

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
INGENIERIA EN COMPUTACION



**“GUIA PARA LA IMPLEMENTACION DE LAS MEJORES
PRACTICAS EN LA DIRECCION DE PROYECTOS APLICADO EN
UN PROYECTO DE DESARROLLO DE SOFTWARE PARA EL
INSTITUTO TECNOLOGICO CENTROAMERICANO**

Trabajo de graduación para optar al título de:
Ingeniero en Ciencias de la Computación

Presentado por:
Guillermo Arturo Cerrato Gámez
Salvador Fonseca Rivas

Asesor:
Ing. Carlos López Linares, MSc, MBA

Soyapango, Marzo del 2008

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
INGENIERIA EN COMPUTACION



AUTORIDADES

RECTOR:

ING. FEDERICO MIGUEL HUGUET

VICERECTOR:

PBRO. VICTOR BERMUDEZ

SECRETARIO GENERAL:

LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERIA:

ING. GODOFREDO ERNESTO GIRON

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
INGENIERIA EN COMPUTACION



**“GUIA PARA LA IMPLEMENTACION DE LAS MEJORES
PRACTICAS EN LA DIRECCION DE PROYECTOS APLICADO EN
UN PROYECTO DE DESARROLLO DE SOFTWARE PARA EL
INSTITUTO TECNOLOGICO CENTROAMERICANO”**

Ing. Carlos H. López Linares **Ing. Melvin A. Carias Rodezno**

Asesor

Lector

AGRADECIMIENTOS.

Arturo Cerrato.

Como primera instancia doy gracias a Dios por la oportunidad de finalizar una etapa más en mi vida.

A mis padres por tener confianza en mis metas.

A mi familia por el apoyo incondicional durante toda la fase de estudio.

A mis amigos que siempre me impulsaron en la finalización y me ayudaron de manera infinita.

A mi amiga de viaje que siempre dispuso de su tiempo de manera infinita.

A mi compañero de tesis por haber compartido este proceso.

Y por último a todas aquellas personas que confiaron y de alguna manera colaboraron para la finalización de mi tesis.

Salvador Fonseca

En primer lugar quiero agradecer a Dios por cada una de las bendiciones que ha derramado en mi vida y que me han permitido culminar esta meta, también quiero agradecer a mi familia quienes me apoyaron incondicionalmente durante todo el proceso y a Marcela quien me animó a continuar hasta el final. Jamás podré pagar todo su amor, sinceramente gracias.

Índice

1	INTRODUCCIÓN.....	I
2	CAPITULO I.....	2
2.1	DEFINICION DEL PROYECTO	2
2.1.1	Antecedentes	2
2.1.2	Definición	3
2.1.3	Justificación	5
2.1.4	Objetivos.....	6
2.1.4.1	Objetivo General	6
2.1.4.2	Objetivos Específicos	6
2.1.5	Alcances.....	7
2.1.6	Limitaciones.....	7
2.1.7	Delimitaciones	8
2.1.8	Proyección social	8
3	CAPITULO II	9
3.1	MARCO TEORICO.....	9
3.1.1	Áreas de conocimiento de la Dirección de Proyectos	9
3.1.1.1	Gestión de la integración de proyectos	9
3.1.1.2	Gestión del Alcance del Proyecto	9
3.1.1.3	Gestión del Tiempo del Proyecto.....	10
3.1.1.4	Gestión de los Costos del Proyecto.....	10
3.1.1.5	Gestión de la Calidad del Proyecto.....	10
3.1.1.6	Gestión de los Recursos Humanos de los Proyectos.....	11
3.1.1.7	Gestión de las Comunicaciones del Proyecto	11
3.1.1.8	Gestión de los riesgos del proyecto	11
3.1.1.9	Gestión de las adquisiciones del proyecto	12
3.1.2	Áreas de experiencia	12
3.1.2.1	Fundamentos de la Dirección de Proyectos	12
3.1.2.2	Conocimientos, normas y regulaciones del área de aplicación	13
3.1.2.3	Comprensión del entorno del proyecto	13
3.1.2.4	Conocimientos y habilidades de dirección general	14
3.1.2.5	Habilidades interpersonales	14
3.1.3	Ciclo de vida del proyecto y organización	15
3.1.3.1	Vida del proyecto	15

3.1.3.2	Características del ciclo de vida del proyecto	15
3.1.3.3	Ciclos de vida de un proyecto.....	15
3.1.3.4	Características de las fases del proyecto	16
3.1.3.5	Relación ciclo de vida del proyecto y ciclo de vida del producto	16
3.1.4	Interesados en el proyecto.....	16
3.1.5	Estructura de la organización	17
3.1.6	Sistema de gestión de proyectos.....	18
3.1.7	Mejores prácticas	18
3.1.8	Análisis de riesgo	19
3.1.9	Gestión de los Recursos Humanos del Proyecto	20
3.1.10	Gestión de calidad del proyecto	21
4	CAPITULO III.....	23
4.1	GUIA PARA LA IMPLEMENTACION DE LAS MEJORES PRÁCTICAS.....	23
4.1.1	Introducción	23
4.1.2	Objetivos.....	24
4.1.3	Ventajas.....	24
4.1.4	Proceso de Inicialización del Proyecto	25
4.1.4.1	Acta de constitución del proyecto	26
4.1.4.2	Emprender un estudio de viabilidad.....	32
4.1.4.3	Establecer los términos de referencia.....	39
4.1.4.4	Designar equipo para el proyecto	45
4.1.4.5	Instalar la oficina para el proyecto.....	49
4.1.4.6	Realizar fase de revisión.....	51
4.1.5	Proceso de Planificación del Proyecto	53
4.1.5.1	Crear un plan de proyecto.....	54
4.1.5.2	Crear un plan de Recursos	59
4.1.5.3	Crear un plan financiero	62
4.1.5.4	Crear un plan de calidad	65
4.1.5.5	Crear un plan de riesgos	68
4.1.5.6	Crear un plan de aceptación.....	74
4.1.5.7	Crear un plan de Comunicación.....	77
4.1.5.8	Crear un plan de Adquisiciones	79
4.1.5.9	Contratar los proveedores	83
4.1.5.10	Continuar con la fase de revisión.....	84

4.1.6	Proceso de Ejecución del Proyecto	87
4.1.6.1	Crear producto(s) final.....	88
4.1.6.2	Monitorear y controlar.....	88
4.1.6.3	Administración del tiempo	89
4.1.6.4	Administración de costos.....	95
4.1.6.5	Administración de la calidad	100
4.1.6.6	Administración de los cambios.....	104
4.1.6.7	Administración de riesgos.	107
4.1.6.8	Gestión de aceptación.....	114
4.1.6.9	Gerencia de comunicación.....	116
4.1.6.10	Gestión de la Integración del Proyecto.	121
4.1.6.11	Gestión del alcance del proyecto	123
4.1.7	Proceso de Cierre del Proyecto	124
4.1.7.1	Realizar el cierre del proyecto	125
4.1.7.2	Revisar la completitud del proyecto	128
4.2	GLOSARIO.....	139
5	CAPITULO IV	141
5.1	APLICACIÓN DEL SISTEMA DE EVALUACIÓN GENERAL DE DESEMPEÑO	141
5.1.1	Acta de constitución.....	141
5.1.2	ACTA DE APROBACIÓN DE MÓDULO	144
5.1.3	Visión del proyecto	145
5.1.4	Cuestionario de entrevista con el cliente	145
5.1.5	Enunciado del Alcance Del Proyecto	148
5.1.5.1	Objetivos que debe cumplir el proyecto	148
5.1.5.2	Características y límites del proyecto	150
5.1.5.3	Productos y servicios relacionados	151
5.1.5.4	Métodos de aceptación y control	155
5.1.6	Estructura de Desglose del Trabajo EDT	156
5.1.7	Secuencia de Actividades.....	161
5.1.8	Cronograma de Actividades	168
5.1.8.1	Cronograma de Línea Base.....	169
5.1.9	Control de Calidad del Proyecto	173
5.1.10	Recursos Humanos del Proyecto	174
5.1.11	Planificación de las Comunicaciones	176
5.1.12	Riesgos del Proyecto	179
5.1.13	ACTA DE APROBACIÓN DEL PROYECTO	184
5.1.14	Estándares de Programación y Diseño	185

5.1.14.1	Objetivos	186
5.1.14.2	Alcance	186
5.1.14.3	Organización de Código y Estilo	187
5.1.14.4	Comentarios.....	189
5.1.14.5	Nomenclatura	191
5.1.14.6	Organización de archivos	194
5.1.14.7	Diseño de Reportes.....	196
5.1.14.8	Diseño de Base de Datos	197
5.1.14.9	Documentación.....	198
5.1.14.10	Manejo de Errores y Excepciones.....	198
5.1.14.11	Conclusión.....	200
5.1.15	Manual de Políticas	200
5.1.15.1	Plan de Reuniones	200
5.1.15.2	Estándar de programación	201
5.1.15.3	Copias de respaldo.....	201
5.1.15.4	Prevención y recuperación de desastres.....	201
5.1.16	Requerimientos del sistema.....	203
5.1.16.1	Perspectiva del cliente/Usuario.....	203
5.1.16.2	Diagrama de Clases	242
5.1.16.3	Diagramas de secuencia.....	243
5.1.16.4	Gráficos de estado	269
5.1.16.5	Glosario general de términos UML	274
6	GLOSARIO GENERAL.....	284
7	CONCLUSION.....	289
8	FUENTES DE CONSULTA.....	290
8.1	SITIOS DE INTERNET VISITADOS.....	290
8.2	BIBLIOGRAFÍA	290

1 INTRODUCCIÓN

En el presente trabajo sentaremos las bases para el desarrollo de un proyecto de software aplicando las técnicas de dirección de proyectos de la PMBOK¹ publicada por el PMI², y utilizando Microsoft Project Server como una herramienta de apoyo para la administración del mismo.

La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer sus requisitos

La dirección de proyectos incluye:

- Identificar los requisitos del proyecto.
- Establecer objetivos claros y posibles de realizar.
- Equilibrar las demandas concurrentes de calidad, alcance, tiempo y costes.
- Adaptar las especificaciones, los planes y el enfoque de las diversas inquietudes y expectativas de los diferentes interesados.

La calidad de los proyectos se ve afectada por el equilibrio entre tres factores importantes: el alcance, el tiempo y el costo. La relación entre estos tres factores es tanta que si cambia uno, se ve afectado por lo menos uno de los otros dos factores.

Como en toda ciencia, los conocimientos de la dirección de proyectos residen en los practicantes que aplican la ciencia y en los académicos que desarrollan el conocimiento. Los fundamentos de la dirección de proyectos, resume el conjunto de conocimientos teóricos y prácticos. Y debido a que continuamente emergen nuevas prácticas ya que la práctica y la teoría se retroalimentan entre si, los fundamentos de la dirección de proyectos están continuamente evolucionando.

¹ Project Management Body of Knowledge – (Fundamentos de la Dirección de proyectos)

² Project Management Institute – (Instituto de Dirección de proyectos)

2 CAPITULO I

2.1 DEFINICION DEL PROYECTO

2.1.1 Antecedentes

La dirección de proyectos es un concepto que surgió mucho tiempo atrás. Si pensamos en todo lo que la humanidad ha tenido que construir a través de la historia, encontramos que tenemos miles de años de experiencia en proyectos para aprender. Experiencia que puede remontarse desde los constructores de pirámides en Egipto o la arquitectura de los acueductos romanos, hasta el moderno desarrollo de software. En cada área los respectivos administradores de proyectos han jugado papeles similares al aplicar la tecnología a los problemas relevantes de cada época. Sin embargo aun en nuestros días, cuando muchos administradores de proyectos intentan mejorar la manera como sus proyectos son administrados, no prestan atención a las lecciones del pasado. Desgraciadamente nuestra referencia histórica en materia de Dirección de proyectos es más cercana al presente de lo que debería ser.

La administración de proyectos actual ha demostrado que la mayoría de proyectos guardan grandes similitudes entre sí. Todos tienen requerimientos, diseños y restricciones. También dependen de las comunicaciones, de la toma de decisiones y de una combinación de pensamiento creativo y lógico. Los proyectos usualmente implican programaciones, presupuestos, e interesados. Sin embargo la actividad principal de los proyectos sigue siendo encaminar el trabajo de diferentes personas a un solo propósito.

2.1.2 Definición

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Un proyecto de desarrollo de software es el conjunto de actividades que permiten asegurar que el software se lleva a cabo a tiempo y de acuerdo con la planificación, así como de acuerdo con los requerimientos del software.

La administración de proyectos de software significa hacer el mejor uso de los recursos existentes para alcanzar la puesta en producción del producto en desarrollo. Los recursos pueden dividirse en laborales (personas), materiales duros (estaciones de trabajo, proyectores y escritorios) y materiales blandos (espacio en disco, tiempo de procesamiento y ancho de banda). El administrador debe asignar tareas a los recursos laborales, los cuales invertirán esfuerzo propio y consumirán recursos materiales asignados, para cumplir con sus responsabilidades.

La factibilidad de realización de un proyecto puede verse comprometida por varios factores, entre ellos el número de personas y el entorno de desarrollo. Si el equipo aumenta en número, entonces la comunicación se volverá cada vez más crítica, podría aumentar el número de licencias de software. Un entorno de desarrollo inapropiado podría restringir tanto el número de personas como la posibilidad de trabajar en forma distribuida.

Los proyectos son temporales porque tienen un inicio y un fin, no así los resultados creados por el proyecto, ya que la mayoría de proyectos tienen impactos sociales y económicos, que perduran mucho más tiempo que los propios proyectos.

Una característica de los proyectos es que crean productos entregables únicos. Los productos entregables pueden ser productos, servicios o resultados. La singularidad

es una característica importante de los productos entregables. Por ejemplo, si se construye un complejo de viviendas, cada vivienda es individual y única, es decir que tienen diferente propietario, diseño y ubicación. La presencia de elementos repetitivos no cambia la condición fundamental de único del trabajo de un proyecto.

La elaboración gradual es otra característica de los proyectos de desarrollo de software y se refiere al desarrollo paso a paso de cada una de las etapas del proyecto.

Las organizaciones realizan una serie de trabajos para alcanzar sus objetivos. Los trabajos se clasifican en proyectos y operaciones. La diferencia entre los proyectos y operaciones radica en que las operaciones son continuas y repetitivas mientras que los proyectos son temporales y únicos.

Los proyectos son una forma de organizar actividades que no pueden ser tratadas dentro de los límites operativos normales de la organización.

Generalmente, los proyectos son resultado de una o más de las siguientes consideraciones estratégicas:

- Una demanda del Mercado.
- Una necesidad de la organización.
- Una solicitud de un cliente.
- Un avance tecnológico.
- Un requisito legal.

2.1.3 Justificación

En la actualidad planificar y dirigir un proyecto de desarrollo de software implica no solo conocer los adelantos tecnológicos más modernos sino desarrollar habilidades para planificar, administrar recursos, acoplarse a un presupuesto y cumplir con los tiempos exigidos, entregando un producto de la mejor calidad.

Lograr cumplir las metas sin fracasar en el intento es un verdadero reto especialmente si no se cuenta con un método que nos ayude a ver más claramente el camino. Con el paso del tiempo los responsables en la dirección de proyectos se han dado cuenta que los principios en la dirección de proyectos aprendidos por la humanidad a lo largo de miles de años de experiencia, son aplicables a los proyectos de desarrollo de software, es entonces cuando surge una nueva rama de la dirección de proyectos que se especializa en el desarrollo de software.

En un mundo donde tomar decisiones se vuelve cada día más difícil, y donde cada problema tiene más de una forma de resolverse, la dirección de proyectos de software nos ayuda a encontrar las mejores alternativas proporcionándonos estándares de mejores prácticas basadas en la experiencia de otros.

Esto resalta la importancia del estudio de la dirección de proyectos, por lo que esta guía será una herramienta útil para todo aquel que desee profundizar en la mejor manera de dirigir un proyecto de desarrollo de software utilizando herramientas como:

1. Mejores prácticas.
2. Análisis de riesgo.
3. Gestión de recursos humanos..
4. Gestión de calidad.

2.1.4 Objetivos

2.1.4.1 Objetivo General

Proporcionar una guía para la aplicación de las mejores prácticas de dirección de proyectos a un proyecto de desarrollo de software. Utilizando como modelo el desarrollo de un proyecto de software llamado “Mecanización del proceso de evaluación general del desempeño” para el Instituto Tecnológico Centroamericano.

2.1.4.2 Objetivos Específicos

- 2.1.4.2.1 Proveer una descripción general de los Fundamentos de la Dirección de Proyectos normados por el PMI³.
- 2.1.4.2.2 Aplicar los conceptos de dirección de proyectos a un proyecto de desarrollo de software.
- 2.1.4.2.3 Desarrollar un proyecto de software llamado “Mecanización del proceso de evaluación general del desempeño” utilizando Java 5.0 y Microsoft SQL Server como plataforma de desarrollo, para el Instituto Tecnológico Centroamericano.
- 2.1.4.2.4 Capacitar al personal del Instituto Tecnológico Centroamericano en el uso y mantenimiento del programa “Mecanización del proceso de evaluación general del desempeño”.
- 2.1.4.2.5 Proporcionar la documentación necesaria para que el Instituto Tecnológico Centroamericano, pueda administrar y modificar el programa “Mecanización del proceso de evaluación general del desempeño”.
- 2.1.4.2.6 Elaborar un documento que pueda utilizarse como guía para la aplicación de las mejores prácticas de dirección de proyectos a un proyecto de desarrollo de software.

³ Project Management Institute.

2.1.4.2.7 Utilizar Microsoft Project Server como parte de la metodología para administrar: tareas, tiempos, asignaciones, etc. durante el desarrollo del proyecto de software.

2.1.5 Alcances

Planificación y administración del desarrollo e implementación de una aplicación de software para el ITCA llamada “Mecanización del proceso de evaluación general del desempeño” a través de la cual se dará seguimiento al plan de trabajo anual de cada empleado de la institución.

Desarrollo de la aplicación mencionada en el punto anterior de acuerdo con los requerimientos del ITCA, utilizando JSP y Microsoft SQL Server 2005 como plataforma de desarrollo.

Capacitar al personal del Instituto Tecnológico Centroamericano en el uso y mantenimiento de la aplicación.

Aplicación de conceptos de dirección de proyectos de la PMBOK.

Elaboración de una guía para aplicar las mejores prácticas de Dirección de proyectos a un proyecto de desarrollo de software.

2.1.6 Limitaciones

Solo se abordaran normas de proyectos individuales y procesos de dirección de proyectos generalmente reconocidos como buenas prácticas.

El estudio de las normas de dirección de proyectos no abordaran la totalidad de los detalles de cada tema.

El desarrollo de software se limitara a los requerimientos proporcionados por el ITCA.

2.1.7 Delimitaciones

Los conceptos de dirección de proyectos serán basados en la PMBOK.

Como software de administración de proyecto utilizaremos Microsoft Project Server 2007.

Las actualizaciones del sistema “Mecanización del proceso de evaluación general del desempeño”, posteriores a la entrega final de este, serán responsabilidad del ITCA.

2.1.8 Proyección social

La guía para la implementación de las mejores prácticas en la dirección de proyectos de software se desarrollará en beneficio de todo el personal administrativo, docente y alumnado de la Universidad Don Bosco. Así como cualquier profesional local o extranjero, con el fin que sea una herramienta útil para el desarrollo de las habilidades de dirección de proyectos de software de cada individuo.

El proyecto de software al que se aplicarán las mejores prácticas de dirección de proyectos se desarrollara en beneficio del Instituto Tecnológico Centroamericano.

3 CAPITULO II

3.1 MARCO TEORICO

3.1.1 Áreas de conocimiento de la Dirección de Proyectos

El estudio actual de la dirección de proyectos de desarrollo de software se divide en 44 procesos aplicables a cualquier proyecto. A su vez estos 44 procesos se clasifican en 9 áreas de conocimiento.

3.1.1.1 Gestión de la integración de proyectos

Procesos y actividades que forman parte de los diversos elementos de la dirección de proyectos de desarrollo de software, que se identifican, definen, combinan, unen y coordinan dentro de los Grupos de Procesos de la Dirección de Proyectos.

Se compone de:

1. Desarrollo del acta de constitución del proyecto.
2. Desarrollo del enunciado del alcance del proyecto preliminar.
3. Desarrollo del plan de gestión del proyecto.
4. Dirigir y gestionar la ejecución del proyecto.
5. Supervisar y controlar el trabajo del proyecto
6. Control integrado de cambios.
7. Cierre del proyecto.

3.1.1.2 Gestión del Alcance del Proyecto

Procesos necesarios para asegurarse de que el proyecto incluya todo el trabajo requerido, y sólo el trabajo requerido, para completar el proyecto satisfactoriamente.

Se compone de:

1. Planificación del alcance.

2. Definición del alcance.
3. Verificación del alcance.
4. Control del alcance

3.1.1.3 Gestión del Tiempo del Proyecto

Procesos relativos a la puntualidad en la conclusión del proyecto.

Se compone de:

1. Definición de las actividades.
2. Establecimiento de la secuencia de actividades.
3. Estimación de recursos de las actividades.
4. Estimación de la duración de las actividades.
5. Desarrollo del cronograma.
6. Control del cronograma.

3.1.1.4 Gestión de los Costos del Proyecto

Procesos involucrados en la planificación, estimación, presupuesto y control de costos de forma que el proyecto se complete dentro del presupuesto aprobado. Se compone de:

1. Estimación de costos.
2. Preparación del presupuesto de costos.
3. Control de costos.

3.1.1.5 Gestión de la Calidad del Proyecto

Procesos necesarios para asegurarse que el proyecto cumpla con los objetivos por los cuales ha sido emprendido.

Se compone de:

1. Planificación de calidad.
2. Realizar aseguramiento de la calidad.

3. Realizar control de calidad.

3.1.1.6 Gestión de los Recursos Humanos de los Proyectos

Procesos que organizan y dirigen el equipo del proyecto.

Se compone de:

1. Planificación de los recursos humanos.
2. Adquisición del equipo del proyecto.
3. Desarrollo del equipo del proyecto.
4. Gestión del equipo del proyecto.

3.1.1.7 Gestión de las Comunicaciones del Proyecto

Procesos relacionados con la generación, recogida, distribución almacenamiento y destino final de la información del proyecto en tiempo y forma. Se compone de:

1. Planificación de las comunicaciones.
2. Distribución de la información.
3. Informar el rendimiento.
4. Gestionar a los interesados.

3.1.1.8 Gestión de los riesgos del proyecto

Procesos relacionados con el desarrollo de la gestión de riesgos de un proyecto.

Se compone de:

1. Planificación de la gestión de riesgos.
2. Identificación de riesgos.
3. Análisis cualitativo de riesgos.
4. Análisis cuantitativo de riesgos.
5. Planificación de la respuesta a los riesgos.
6. Seguimiento y control de riesgos.

3.1.1.9 Gestión de las adquisiciones del proyecto

Procesos para comprar o adquirir productos, servicios o resultados, así como para contratar procesos de dirección.

Se compone de:

1. Planificación de compras y adquisiciones.
2. Planificación de la contratación.
3. Solicitar respuestas de vendedores.
4. Selección de vendedores.
5. Administración del contrato.
6. Cierre del contrato.

3.1.2 Áreas de experiencia

Una dirección de proyecto efectiva requiere que el equipo de dirección del proyecto comprenda y use los conocimientos y habilidades correspondientes a, por lo menos, cinco áreas de experiencia:

3.1.2.1 Fundamentos de la Dirección de Proyectos

Los Fundamentos de la Dirección de Proyectos describen el conocimiento propio del campo de la dirección de proyectos, son:

1. Definición del ciclo de vida del proyecto
2. Grupos de Procesos de Dirección de Proyectos
3. Áreas de conocimiento

3.1.2.2 Conocimientos, normas y regulaciones del área de aplicación

Las áreas de aplicación son categorías de proyectos que tienen elementos significativos comunes pero que no son necesarios ni están presentes en todos los proyectos. Por lo general se definen en términos de:

1. Departamentos funcionales y disciplinas de respaldo, por ejemplo:
 - a. Oficinas legales
 - b. Producción
 - c. Manejo de inventario
 - d. Comercialización
 - e. Logística
 - f. Manejo de personal
2. Elementos técnicos, como el desarrollo o la ingeniería de software y, en algunos casos, un tipo específico de ingeniería como, por ejemplo, la ingeniería de aguas y sanitaria, o la ingeniería de construcción.
3. Especializaciones de gestión, como la contratación por el gobierno, el desarrollo de comunidades y el desarrollo de nuevos productos
4. Grupos de industria, por ejemplo:
 - a. Automotriz
 - b. Químico
 - c. Agrícola
 - d. Servicios financieros

3.1.2.3 Comprensión del entorno del proyecto

Casi todos los proyectos se planifican e implementan en un contexto social, económico y ambiental y tienen impactos positivos y negativos deseados y/o no deseados. Por lo que el equipo del proyecto debe considerar el contexto:

1. Cultural
2. Social

3. Internacional
4. Político
5. Físico

3.1.2.4 Conocimientos y habilidades de dirección general

La dirección general comprende la planificación, organización, selección de personal, ejecución y control de las operaciones de una empresa en funcionamiento. Incluye disciplinas de respaldo como:

1. Gestión financiera y contabilidad
2. Compras y adquisiciones
3. Ventas y comercialización
4. Fabricación y distribución

3.1.2.5 Habilidades interpersonales

La gestión de las relaciones interpersonales incluye:

1. Comunicación efectiva
2. Influencia en la organización
3. Liderazgo
4. Motivación
5. Negociación y gestión de conflictos
6. Resolución de problemas

3.1.3 Ciclo de vida del proyecto y organización

3.1.3.1 Vida del proyecto

Para facilitar la gestión de dirección de proyectos, los directores de estos o la organización pueden dividir los proyectos en *fases*, a lo que se conocen como ciclo de vida del proyecto. Muchas organizaciones identifican un conjunto de ciclos de vida específico para usarlo en todos sus proyectos.

3.1.3.2 Características del ciclo de vida del proyecto

El ciclo de vida del proyecto define las fases que ayudaran al director del proyecto a determinar si:

1. Se conecta el inicio de un proyecto con su fin.
2. Existirá una viabilidad.
3. Se efectuara una ejecución rápida.
4. Se definirá un ciclo de vida más apropiado o preferido.

3.1.3.3 Ciclos de vida de un proyecto

No existe una única manera, que sea la mejor, para definir el ciclo de vida ideal de un proyecto, pero generalmente se define así:

1. **Que** trabajo se debe realizar en cada fase.
2. **Cuando** se deben generar los productos entregables en cada fase.
3. **Quien** está involucrado en cada fase.
4. **Como** controlar y aprobar cada fase.
5. **Transferencias** de información o de componentes técnicos.
6. **Nivel** de costo y de personal.
7. **Certeza** de terminar con éxito el proyecto.

3.1.3.4 Características de las fases del proyecto

1. En cualquier proyecto específico, las fases se pueden subdividir en función de:
 - a. Tamaño
 - b. Complejidad
 - c. Nivel de riesgo
 - d. Restricción del flujo de caja
2. Se puede cerrar una fase sin la decisión de iniciar alguna otra fase.
3. Cada fase se inicia formalmente para producir una salida.
4. Revisión al final de cada fase son también conocidas como:
 - a. Salidas de fase
 - b. Entradas a la fase
 - c. Puntos de cancelación.

3.1.3.5 Relación ciclo de vida del proyecto y ciclo de vida del producto

Las fuerzas impulsoras que crean los estímulos para un proyecto se conocen habitualmente como problemas, oportunidades o requisitos de negocio.

Por lo que el ciclo de vida del proyecto atraviesa una serie de fases para crear el producto. Las organizaciones consideran el ciclo de vida del proyecto como parte de vida del producto.

3.1.4 Interesados en el proyecto

Los interesados tienen niveles de responsabilidad y autoridad variables al participar en un proyecto., que pueden cambiar a lo largo del curso del ciclo de vida del proyecto.

Entre los interesados clave de proyectos se encuentran:

1. Director de proyectos.
2. Cliente/usuario

3. Organización ejecutante
5. Miembros del equipo de proyectos.
6. Equipo de dirección del proyecto.
7. Patrocinador
8. Influyentes
9. Oficina de Gestión de proyectos. (PMO)

Los directores del proyecto deben gestionar las expectativas de los interesados ya que, a menudo, los interesados tienen objetivos muy diferentes o contradictorios.

Las organizaciones basadas en proyectos son aquellas cuyas operaciones se componen principalmente de proyectos. Por ejemplo:

1. Las organizaciones que obtienen sus ingresos principalmente de la ejecución de los proyectos para otros.
2. Las organizaciones que han adoptado la dirección por proyectos.

La cultura de las organizaciones frecuentemente tiene una influencia directa sobre el proyecto.

3.1.5 Estructura de la organización

Esta estructura contiene una organización funcional, como una jerarquía donde cada empleado tiene un superior claramente establecido.

Cada organización es orientada a proyectos: Suelen tener unidades denominadas departamentos, pero estos grupos dependen directamente del director de proyectos.

Existen varios tipos de organizaciones, que son:

- Organización matricial débil:
- Organización matricial equilibrada
- Organización matricial fuerte
- Organización combinada:

3.1.6 Sistema de gestión de proyectos

Este sistema es un conjunto de herramientas, técnicas, metodologías, recursos y procedimientos utilizados para gestionar un proyecto. Ayuda al director del proyecto a gestionar de forma eficaz un proyecto hasta su conclusión.

3.1.7 Mejores prácticas

Las mejores prácticas forman un concepto que se origina en la administración y que afirma que hay una técnica, método, proceso, actividad, incentivo o recompensa que es más efectivo al entregar un resultado en particular que cualquier otra técnica, método o proceso. La idea es que con el proceso adecuado, revisar y probar, un proyecto puede ser culminado con menos problemas e imprevistos.

El concepto de mejores prácticas no es nuevo. Frederick Taylor (1919) dijo: “entre los varios métodos e implementaciones usadas en cada elemento de cada negocio siempre existe un método y una implementación que es mejor y más rápida que cualquiera de las otras”. Este punto de vista fue llamado: “Las mejores prácticas”.

En la actualidad, las mejores prácticas es un concepto muy utilizado. Despierta la necesidad de mejorar los procesos a medida que pasa el tiempo y las cosas se evolucionan, también son consideradas por algunos como una moda en los negocios utilizada para describir los procesos de desarrollo y seguimiento en cuanto a la forma de hacer las cosas que múltiples organizaciones pueden usar para la administración, políticas y especialmente los sistemas de software.

3.1.8 Análisis de riesgo

La Gestión de los riesgos del Proyecto incluye los procesos relacionados con la planificación de la gestión de riesgos, la identificación y el análisis de riesgos, las respuestas a los riesgos, y el seguimiento y control de riesgos de un proyecto; la mayoría de estos procesos se actualizan durante el proyecto. Los objetivos de la gestión de los Riesgos del Proyecto son aumentar la probabilidad del impacto de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos adversos para el proyecto. Los procesos de Gestión de Riesgos del Proyecto incluyen lo siguiente:

- **Planificación de la Gestión de Riesgos**
Decidir cómo enfocar, planificar y ejecutar las actividades de gestión de riesgos para un proyecto.
- **Identificación de Riesgos**
Determinar qué riesgos pueden afectar al proyecto y documentar sus características.
- **Análisis Cualitativo de Riesgos**
Priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando su probabilidad de ocurrencia y su impacto.
- **Análisis Cuantitativo de Riesgos**
Analizar numéricamente el efecto de los riesgos identificados en los objetivos generales del proyecto.
- **Planificación de la Respuesta a los Riesgos**
Desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.
- **Seguimiento y Control de Riesgos**
Realizar el seguimiento de los riesgos identificados, supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos y evaluar su efectividad a lo largo del ciclo de vida del proyecto.

3.1.9 Gestión de los Recursos Humanos del Proyecto

La gestión de los Recursos humanos del Proyecto incluye los procesos que organizan y dirigen el equipo del proyecto. El equipo del proyecto está compuesto por las personas a quienes se les han asignado roles y responsabilidades para concluir con el proyecto. Si bien es común hablar de asignación de roles y responsabilidades, los miembros del equipo deberían participar en gran parte de la planificación y toma de decisiones del proyecto. La participación temprana de los miembros del equipo aporta experiencia durante el proceso de planificación y fortalece el compromiso con el proyecto. El tipo y la cantidad de miembros del equipo del proyecto a menudo pueden cambiar, a medida que avanza el proyecto. Los miembros del equipo del proyecto pueden denominarse personal del proyecto.

El equipo de dirección del proyecto es un subgrupo del equipo del proyecto y es responsable de las actividades de dirección de proyectos, tales como la planificación, el control y el cierre. Este grupo puede denominarse equipo central, equipo ejecutivo o equipo de liderazgo. Para proyectos más pequeños, las responsabilidades de la dirección de proyectos pueden ser compartidas por todo el equipo o administradas únicamente por el director del proyecto. El patrocinador del proyecto trabaja con el equipo de dirección del proyecto, ayudando generalmente con cuestiones tales como la financiación del proyecto, aclarando preguntas sobre el alcance y ejerciendo influencia sobre otros a fin de beneficiar al proyecto.

Los procesos de Gestión de los Recursos Humanos del Proyecto incluyen lo siguiente:

- **Planificación de los Recursos Humanos**
Identificar y documentar los roles del proyecto, las responsabilidades y las relaciones de informe, así como crear el plan de gestión de personal.
- **Adquirir el Equipo del Proyecto**
Obtener los recursos humanos necesarios para concluir el proyecto

- **Desarrollar el Equipo del Proyecto**
Mejorar las competencias y la interacción de los miembros del equipo para lograr un mejor rendimiento del proyecto
- **Gestionar el Equipo del Proyecto**
Hacer un seguimiento del rendimiento de los miembros del equipo, proporcionar retroalimentación, resolver polémicas y coordinar cambios a fin de mejorar el rendimiento del proyecto.

Estos procesos interaccionan entre sí y también con los procesos de las demás áreas de conocimiento. Cada proceso puede implicar el esfuerzo de una o más personas o grupos de personas, dependiendo de las necesidades del proyecto. Cada proceso tiene lugar por lo menos una vez en cada proyecto y se realiza en una o más fases del proyecto, si el proyecto se encuentra dividido en fases.

3.1.10 Gestión de calidad del proyecto

Los procesos de Gestión de Calidad del Proyecto incluyen todas las actividades de la organización ejecutante que determinan las políticas, los objetivos y las responsabilidades relativos a la calidad de modo que el proyecto satisfaga las necesidades por las cuales se emprendió. Implementa el sistema de gestión de calidad a través de la política, los procedimientos y los procesos de planificación de calidad, aseguramiento de calidad y control de calidad, con actividades de mejora continua de los procesos que se realizan durante todo el proyecto, según corresponda.

Los procesos de Gestión de la Calidad del Proyecto incluyen lo siguiente:

Planificación de Calidad

Identificar qué normas de calidad son relevantes para el proyecto y determinar cómo satisfacerlas.

Realizar Aseguramiento de Calidad

Aplicar las actividades planificadas y sistemáticas relativas a la calidad, para asegurar que el proyecto utilice todos los procesos necesarios para cumplir con los requisitos.

Realizar Control de Calidad

Supervisar los resultados específicos del proyecto, para determinar si cumplen con las normas de calidad relevantes e identificar modos de eliminar las causas de un rendimiento insatisfactorio.

4 CAPITULO III

4.1 GUIA PARA LA IMPLEMENTACION DE LAS MEJORES PRÁCTICAS

4.1.1 Introducción

La presente Guía será la referencia orientada para cualquier profesional o personal interesada en trabajar con proyectos. En este manual recoge el conocimiento teórico, la práctica profesional y la información que toda persona necesita conocer para gestionar sus proyectos con éxito.

Se contara como base la guía PMBOK publicada por la PMI, que define a la gerencia de proyecto como: el uso del conocimiento, de las habilidades, y de las técnicas para proyectar actividades para resolver necesidades y expectativas para realizar el alcance de un proyecto.

A lo que agregamos la utilización de un sistema de herramientas y de técnicas para manejar e implementar las mejores practica a un proyecto. Además hay que tomar en cuenta los estándares y los códigos aplicables para el proyecto, la capacidad técnica, la capacidad de la gerencia general, las habilidades del recurso humano, y los alrededores del proyecto.

Según la guía PMBOK, un programa es un grupo de proyectos manejados de una manera coordinada de obtener las ventajas no capaces de ser obtenido manejándolas por separado.

Los costos de ciclo vital se pueden incurrir en con frecuencia sobre la vida útil del proyecto.

Estos costos incluyen responsabilidad del trabajo y del proyecto de la reparación de la garantía y muchas otras. El proyecto se debe hacer dentro de los límites del ambiente físico en el cual ocurre y bajo condiciones presentes. Un proyecto es único porque hay algo que fija cada proyecto aparte de otros. Las metas y los objetivos se indican al principio del proyecto. Estas metas y objetivos se elaboran a ser más detallados mientras que progresa el proyecto. Los proyectos van siempre a ser esfuerzos temporales, porque se ponen intencionalmente juntos con el fin de lograr algo. Una vez que este `` se haya logrado algo ", los recursos que fueron puestos juntos para este propósito se pueden asignar a otros proyectos.

4.1.2 Objetivos.

1. La **"Guía para la implementación de las mejores prácticas en la Dirección de Proyectos aplicado en un proyecto de desarrollo de software para el Instituto Tecnológico Centroamericano "** será la base que se necesita para realizar sus proyectos dentro de un periodo de tiempo, sin superar el presupuesto y a satisfacción del cliente.
2. Que su contenido pueda ser aplicable a cualquier tipo de proyecto.

4.1.3 Ventajas.

Las ventajas de la utilización en esta guía serán enfocadas en reunir muchas de las cosas que son necesarias para poder satisfacer las expectativas del cliente.

Una de las grandes ventajas de la gerencia de proyecto es su capacidad de formar equipos multidisciplinarios con la gente adecuada en el tiempo correcto.

La ventaja obvia de esto es que las habilidades escasas se pueden traer a un proyecto cuando están necesitadas. La gerencia de proyecto trae muchas de estas ventajas juntas en una metodología que pueda ser aplicada con éxito. En la gerencia

de proyecto hablamos con frecuencia de los apremios triples del alcance del proyecto, del presupuesto de proyecto, y del tiempo del proyecto.

Esto significa que los proyectos deben resolver las expectativas de los clientes y ser desarrollados dentro del presupuesto que fue asignado a ellas y en el tiempo asignado a ellas.

4.1.4 Proceso de Inicialización del Proyecto

La primera fase en el ciclo de vida del proyecto es la inicialización. Esta fase implica crear un nuevo proyecto definiendo el problema o la oportunidad de negocio, solución que se entregará mediante los productos y el alcance dentro de los cuales el proyecto será implementado.

La fase de inicialización del proyecto es en gran medida la más crítica, pues permite mantenerlo enfocado en los objetivos que persigue la entidad que gestiona el trabajo, definiendo el alcance del proyecto, las validaciones del mismo y autorizando al director para su ejecución. Si no se inicia correctamente, se puede aumentar el riesgo de no concluirlo o deja de existir la necesidad del negocio en él.

El grupo de proceso de inicialización incluye la realización de ciertas actividades, como se observa en la figura 1.1.

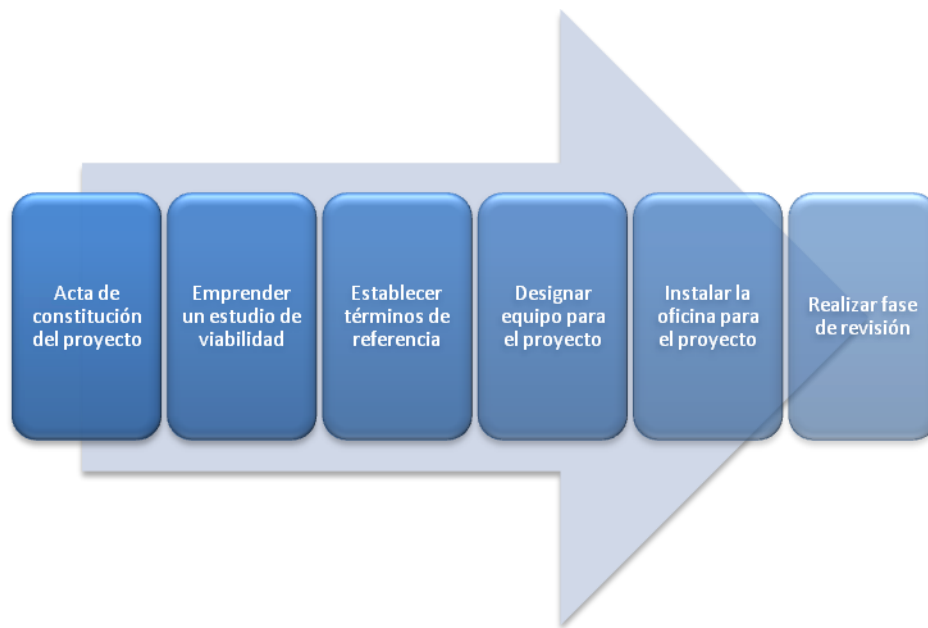


Figura 1.1 Actividades para la iniciación de un proyecto.

4.1.4.1 Acta de constitución del proyecto

Es el proceso relacionado con la autorización del proyecto, tomando como base la documentación de las necesidades del negocio u otros resultados que se pretende obtener para satisfacer esos requisitos. El acta relaciona el trabajo solicitado por la empresa y la autorización del mismo dejando por sentados los requerimientos previamente establecidos.

El acta es un documento que incluye:

- Una descripción detallada del problema o de la situación a resolver.
- Una lista de las opciones disponibles para entregar, la cual será la solución al problema.
- Una lista de los costos y de las ventajas que tienen cada una de las opciones planteadas.
- Una recomendación para la solución a aprobar.

El acta del proyecto ayuda a determinar si los costos, las ventajas, los riesgos y las ediciones están alineados con los documentos de requerimientos originales. Al finalizar el proyecto, se hará una revisión para determinar si este presentó los criterios necesarios; por lo tanto, el éxito del mismo se mide contra su capacidad

resolver dichos criterios. Para la creación de un caso de negocio hay varias fases que se deben completar, las cuales se mencionan en la figura 1.2.

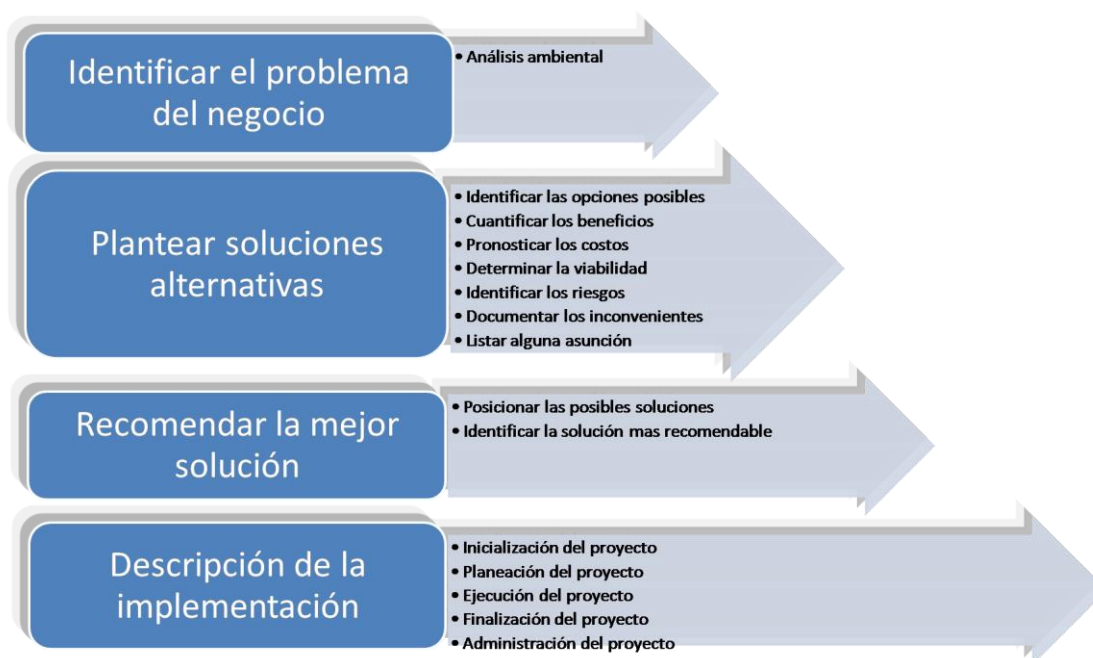


Figura 1.2 Fases para la creación de un caso de negocio

Identificar el problema del negocio

- Análisis Ambiental

Consiste en identificar los cambios o fallos que existen en la empresa en cuestión; se pueden tomar en cuenta cambios que se hayan dado en la visión de la empresa, estrategias u objetivos, procesos o tecnología que no esté operando eficientemente; una nueva amenaza de competencia para la empresa o simplemente requerimientos que la misma estime necesarios, una mejora u optimización de ellos.

Una vez identificados los fallos, se describe el análisis del problema planteado, las razones o factores que dieron origen al proyecto, el impacto que está teniendo el problema en la empresa, así como los intervalos de tiempos en los que puede ser resuelto.

Plantear soluciones alternativas

- Identificar las opciones posibles

Se identifican las que puedan resolver la problemática planteada, con una pequeña descripción de cada una.

- Cuantificar los beneficios

Para cuantificar los beneficios financieros o no financieros se deben plantear aquellos puntos clave que con la implementación de cada posible solución percibirán ganancias o mejoras para la empresa.

Un posible formato para detallar cada uno de los beneficios se encuentra en la tabla 1.1 donde se muestra un ejemplo.

Categoría del Beneficio	Descripción	Valor
Financiero	Reducción de costos	\$xx.xx
Operacional	Reducción del tiempo que se tarda un operario en realizar x cantidad de productos	x horas

Tabla 1.1 Declaración de beneficios con cada alternativa

- Pronosticar los costos

En este momento se identifican los gastos en los que incurrirá la empresa para implementar cada alternativa propuesta, hay que identificar si el gasto proviene del capital de la empresa o del área operacional, de manera que se pueda realizar un análisis de cada uno de los gastos. Para un mejor detalle se puede utilizar la tabla 1.2 como propuesta.

Categoría del Gasto	Descripción	Valor	Tipo de gasto
Personal	Salarios del equipo de trabajo	\$xx.xx	Operativo
	Contratos outsourcing necesarios	\$xx.xx	Capital
Infraestructura	Equipo y materiales	\$xxxx	Operativo

Tabla 1.2 Declaración de costos con cada alternativa

- Determinar la viabilidad

El estudio de la viabilidad demostrará la probabilidad que tiene cada alternativa para satisfacer las necesidades planteadas en los puntos anteriores. Un formato propuesto para dicho estudio se encuentra en la tabla 1.3.

Componente de la solución	Clasificación 1-10	Método utilizado para determinar la viabilidad
Nueva Tecnología	9	Creación de una nueva herramienta para su desarrollo
Nuevos Procesos	3	Con la reingeniería de procesos similares en otras empresas han fracasado

Tabla 1.3 Tabla resumen con el estudio de viabilidad

- Identificar los riesgos

Para los efectos del trabajo se define como cualquier evento que pueda poner en riesgo o afectar el proyecto. Se agrupa cada uno de los riesgos aparentes con la evaluación de cada solución, de encontrarse en una categoría mala se identifican razones y se busca reducirlas. Para un mejor enfoque, observar tabla 1.4

Descripción	Categorización	Impacto	Acciones a realizar para que su viabilidad
Red inhabilitada	Medio	Muy Alto	Tener un plan de contingencia ante cualquier eventualidad
Tecnología y Solución no son compatibles con sistemas comprados	Bajo	Medio	Mantener una estricta administración del proceso de compra durante el proyecto

Tabla 1.4 Identificación de riesgos

- Documentar los inconvenientes

Se reúnen los inconvenientes más altos para cada solución, Estos se definen como eventos que ocurren de manera inadvertida y que afectan la ejecución del proyecto. Para una mejor comprensión, el formato se encuentra en la Tabla 1.5.

Descripción	Prioridad	Acción a tomar
Entrega retrasada	Medio	Contratar fuente externa
Actualización de software	Bajo	Diseñar soluciones basadas en el software que hay en el momento

Tabla 1.5 Documentación de los inconvenientes

- Listar cualquier asunción

Listar cualquier asunción que se haya tenido al momento de elegir una opción, por ejemplo si el equipo solicitado sobra al finalizar el proyecto.

Recomendar la mejor solución

- Posicionar las posibles soluciones

Se debe identificar cada uno de los criterios para comparar las distintas alternativas o soluciones propuestas y que dichos criterios permitan establecer una nota. Un mecanismo puede ser asignar un valor del 1 al 10, pues esta calificación influirá posteriormente en la toma de decisiones para elegir la mejor solución. El formato sugerido es el que se encuentra en la tabla 1.6.

Criterio de comparación	Solución 1 ranking	Solución n ranking
Beneficios	1-10 ranking	1-10 ranking
Incremento de ingresos		
Innovación		
Costos		
Empleados		
maquinaria		

Tabla 1.6 Ranking de las soluciones

- Identificar la solución más recomendable

La solución con mayor puntaje es la más recomendable para optar, ésta debe explicar las necesidades del negocio, puesto que será una de las bases más fuertes para la solución.

Descripción de la implementación

- Inicialización del proyecto

Listar los pasos adecuados para definir un proyecto, creando el plan de proyecto y las instalaciones con los recursos necesarios para llevarlos a cabo.

- Planificación del proyecto

Listar las actividades requeridas para construir la solución propuesta.

- Cierre del proyecto

Listar las actividades involucradas para dar por finalizada la solución, equipo necesario, y planificar la revisión de la post implementación del proyecto.

- Administración del proyecto

Se listan aspectos como: tiempo, costo, calidad, cambios, riesgos, inconvenientes, comunicación y aceptación del proyecto.

4.1.4.2 Emprender un estudio de viabilidad.

El estudio de viabilidad presenta una vía para determinar el problema que se genera en el negocio, identificando varias opciones o soluciones para resolverlo. El estudio de viabilidad provee de lo siguiente:

- Una descripción completa del problema o situación
- Una lista de requerimientos que proporcionan la solución al problema
- Una lista de opciones disponibles para llevar a cabo la solución propuesta
- La asignación de la viabilidad de cada opción
- Lista de los riesgos e inconvenientes asociados a cada opción

- La mejor opción para la implementación

Usualmente el estudio se realiza en paralelo con el acta del proyecto, agregando mayor rigor a la solución u opción presentada; por esta razón los pasos entre el caso del negocio y este estudio las fases son similares.

Para emprender un estudio de viabilidad se deben llevar a cabo 4 grandes fases, las cuales se describen en la figura 1.3

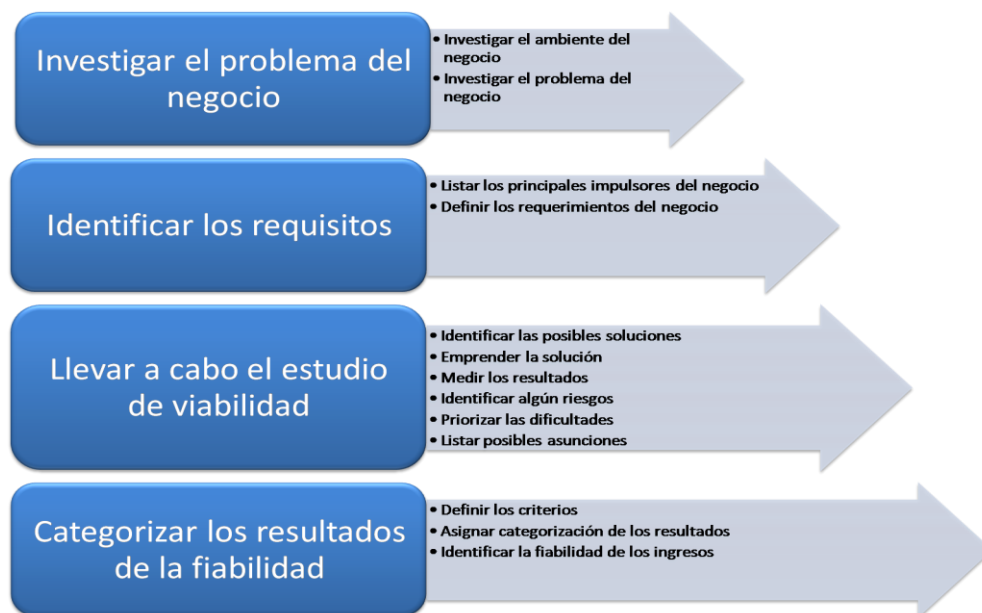


Figura 1.3. Fases para el estudio de viabilidad

Investigar el problema del negocio

- Investigar el ambiente del negocio

Antes de entrar de lleno al estudio de viabilidad se debe tener una apreciación completa del problema a ser abordado. Para lograr esto primero hay que analizar el ambiente en el que se generó el problema.

Análisis Externo

Identificar los agentes externos del problema, los cuales influyen de manera indirecta en él, por ejemplo: tecnología, marcas o cualquier cambio que se dé en el entorno.

Visión del Negocio

Identificar la visión del negocio, estrategia y objetivos.

Áreas del negocio

Identificar cada una de las áreas más relevantes del negocio, ya sea en un organigrama, reporte o un comunicado.

Localidades del Negocio

Identificar geográficamente la ubicación del negocio y de cada una de las unidades; de ser necesario, hacer un diagrama de cada uno de los lugares del negocio involucrados en el proyecto.

Información del Negocio

Se identifica la información más relevante del proyecto, ya sea financiera, humana o de recursos; se listan lugares donde se puede encontrar dicha información ya sea en bases de datos, archivos o sistemas. De ser necesario, crear un diagrama de flujo de datos para analizar cómo se está utilizando la información en la empresa para cada uno de los procesos más relevantes de ésta.

Tecnologías del negocio

Lista cada una de las tecnologías más relevantes del proyecto y describe la función que desempeñan en el proyecto, también contiene el diagrama arquitectónico de la tecnología, y las herramientas para cada una de ellas.

Procesos del negocio

Lista cada uno de los procesos más relevantes que se llevan a cabo en la empresa, y el proyecto a desarrollar proveerá un detalle de la descripción de cada uno.

- Investigar el problema del negocio

Se debe investigar aquellos problemas del negocio, sus razones e impacto, así como identificar las oportunidades de crecimiento del mismo y el tiempo que le tomará alcanzar dicha oportunidad.

Identificar los requisitos

- Listar los principales impulsores del proyecto
- Definir los principales requerimientos

Para definir los requerimientos se sugiere llenar la tabla 1.7, la cual permite definir el problema u oportunidad y su respectivo requerimiento

Problema/Oportunidad	Requerimiento
Descripción...	Descripción...
Descripción....	

Tabla 1.7 Oportunidades y Requerimientos

Llevar a cabo el estudio de viabilidad

Teniendo identificado el problema del negocio y los requerimientos, se debe hacer el estudio de viabilidad. En este punto se describe como realizarlo, identificando la mejor solución y determinando las mejores alternativas para solucionar los requerimientos. Se describen los riesgos, inconvenientes y asunciones asociadas a cada alternativa.

- Identificar las posibles soluciones

Se crea una lista de las soluciones que podrían llegar a resolver el problema planteado; para cada solución se debe listar cada uno de los componentes que ésta, presenta así como una pequeña descripción de ellos. Se debe considerar de la misma manera, cómo cada elemento de

la solución resuelve los requerimientos propuestos ante la problemática planteada.

- Emprender la solución

Para proponer las posibles soluciones se puede utilizar un rango de métodos:

Prototipos: Es un subconjunto de la solución completa, que se construye para resolver la solución. Prototipos son típicos desarrollos para probar las altas áreas de riesgo para resolver la solución.

Encuestas al personal. Son una gran técnica para identificar la viabilidad de las soluciones que implican un cambio a los procesos del negocio. Por ejemplo, pueden ser utilizados para identificar la posible adopción de un nuevo proceso de la gerencia en las cuales se involucre al personal.

Encuestas sobre mercado. Las encuestas sobre mercado se pueden emprender para determinar la viabilidad de introducir productos nuevos o servicios al mercado determinando el nivel de aceptación del mismo.

- Medir los resultados

Se deben medir los resultados de cada planteamiento de los requerimientos actuales y compararlos con respecto a lo esperado, listando aquellas consideraciones no tangibles, también las cosas que se necesitaría tener para que se puedan obtener los resultados deseados, toda la medición se debe realizar planteando un ranking entre ellas.

- Identificar algún riesgo

Listar los riesgos aparentemente identificados con la implementación de cada solución. Los riesgos se definen como cualquier evento que cause efectos negativos para llevar a cabo una determinada actividad. Los riesgos pueden ser de

estratégicos, de desarrollo, financieros, operacionales, técnicos, competitividad o de desarrollo al cliente. Se sugiere documentar los riesgos según la tabla 1.9.

Descripción del riesgo	Prioridad	Impacto	Acciones requeridas para mitigar el riesgo
Descripción...	Bajo	Muy alto	Descripción...
Descripción....	Medio	Alto	Descripción...

Tabla 1.8 Solución de riesgos

- Priorizar las dificultades

Se identifica cualquier dificultad asociada con la adopción de cualquier solución. Las dificultades son definidas como cualquier evento que ocasionalmente pueda afectar a la posible y que como consecuencia de ello no se logre lo planeado. Para documentar las dificultades se sugiere realizarse con la tabla 1.9.

Descripción de la dificultad	Prioridad	Acciones requeridas para resolver las dificultades
Descripción...	Bajo	Descripción...
Descripción....	Medio	Descripción...

Tabla 1.9 Descripción de dificultad

- Listar posibles asunciones

Listar cualquier asunción tomada en cuenta con la adopción de cada solución. Por ejemplo:

- El precio de los materiales a utilizar no se incrementará mientras dure el proyecto

- Outsourcing estará disponible para dar soporte al proyecto en caso sea necesario

Categorizar los resultados de la fiabilidad.

- Definir los criterios

Especificar los criterios usados para establecer la categorización o ranking de las soluciones, y describir el puntaje y el mecanismo usado para obtener los resultados

- Asignar categorización de los resultados

Asignar puntajes a cada opción identificando cada uno de los criterios establecidos en el punto anterior. Para realizarlo se sugiere utilizar la tabla 1.10

Criterio	Opción 1 puntaje	Peso 1	Opción 2 puntaje	Peso 2
Criterio 1...				
Criterio 2....				

Tabla 1.10

El puntaje debe ir de 1 a 10 y el peso debe ir de 0.5 a 1.5. El total es calculado multiplicando el puntaje por el peso.

- Identificar la fiabilidad de los ingresos

Basado en el puntaje total, identificar opción con mayor viabilidad para ser recomendado para la implementación.

Teniendo la mejor opción a recomendar, documentar la opción listando los materiales y recursos necesarios para su aprobación. El siguiente paso en el ciclo del proyecto es para definir los términos de referencia.

4.1.4.3 Establecer los términos de referencia.

Teniendo ya creada la carta del negocio y el estudio de viabilidad, y considerando la mejor solución a proponer, se está listo para definir la visión que se le dará al proyecto, para el usuario. Para definir la finalidad del proyecto se debe realizar un documento de referencia (TOR).

El TOR muestra el propósito del proyecto, la forma como se estructurará y como será implementado, lo describe de la siguiente manera:

- Visión, objetivos, finalidad y entregables.
- Involucrados, roles y responsabilidades
- Recursos tanto financieros como planes de calidad.

El TOR es conocido como un reporte de definición del proyecto. Este documento se presenta a los jefes del proyecto, o aquellas personas involucradas en la aprobación del mismo.



Figura 1.4. Fases a seguir para establecer los términos de referencia

Identificar la visión del proyecto

Para establecer el TOR, es necesario definir el propósito del proyecto, esto incluye la visión y los objetivos claves a ser cumplidos, la visión del trabajo para ser tomado en cuenta y los productos a ser entregados.

- *Visión*

Describe la visión del proyecto. La visión debe ser corta, concisa y realizable.

- *Objetivos*

Listar los objetivos del proyecto que describan con más detalle lo que el proyecto desea realizar. Todos los objetivos listados deben ser específicos, medibles, alcanzables, realistas y en un tiempo establecido.

- *Alcance*

Define el alcance del proyecto incluyendo los productos del negocio, servicios, procesos, departamentos, información y tecnología a afectar. Es muy relevante identificar cualquier área del negocio no afectada por el proyecto, para clarificar las fronteras del mismo.

- *Productos o entregables*

Se deben especificar claramente los elementos a entregar. Para especificar los entregables se sugiere utilizar la Tabla 1.11.

Productos	Componentes	Descripción
Documento 1...	...	Descripción...
Sistema....	Telecomunicaciones	Descripción...

Tabla 1.11. Definición de los entregables del proyecto

Describir la organización del proyecto

- Clientes

Listar los clientes que en algún momento estarán involucrados con los productos a entregar por el proyecto. Los clientes pueden ser individuales, grupales o externos a la compañía. Para documentar a los clientes se sugiere utilizar la tabla 1.12.

Clientes	Representante
Grupo...	Nombre...
....	...

Tabla 1.12. Representación de los usuarios

- Actores

Se deben listar todos los actores o involucrados del proyecto. Un actor puede ser una persona o entidad externa que tiene cierta relación o involucramiento en él. Por ejemplo, la entidad que provee el dinero para llevar a cabo el proyecto es un actor. Para documentar los actores se sugiere utilizar la tabla 1.13.

Actores	Interés de los actores
CEO	Alineamiento con la visión de la compañía
....	...

Tabla 1.13 Documentación de los actores

- Roles

Identificar los roles requeridos para emprender el proyecto. Los roles incluyen un patrocinador del proyecto, jefe de este y el equipo del mismo. Para cada rol se lista el nombre de la persona, la organización, el status asignado y la fecha en que la

persona será asignada al proyecto, si no ha sido asignada aún. Para llenar la información del proyecto se sugiere utilizar la tabla 1.14.

Actores	Tipo	Organización	Status/asignación	Fecha de asignación
CEO	Persona	Organización XY	Asignado/no asignado	dd/mm/yyyy

Tabla 1.14 Responsables del proyecto

- Responsabilidades

Se deben describir las responsabilidades para cada rol, proporcionando de manera sumaria una lista de responsabilidades en esta etapa.

- Estructura

Muestra un reporte estructurado con cada rol y la organización del proyecto. Se debe crear un especie de organigrama de los involucrados en el proyecto y sus roles.

Crear un plan de proyecto general

- Plan de proyecto

Describe los planes que se tienen en el proyecto para inicializar, planificar, ejecutar y cerrar el proyecto, detallando para cada una de éstas las actividades y el tiempo que tomará; se sugiere realizarlo usando la tabla 1.15.

Fase	Actividad	Fecha	Descripción
Iniciación	Descripción	dd/mm/yyyy	Descripción de lo que se hará en el plan para cada actividad....
Planificación			Descripción....

Tabla 1.15 Plan del proyecto

Definidos los eventos en cada etapa y su plan dentro del proyecto, se definen las dependencias externas y la criticidad en él. Las dependencias externas son actividades que impactan o se ven impactadas por alguna actividad del proyecto. Para ello se puede documentar tomando como base la tabla 1.17

Actividad del proyecto	Impacta en...	Es impactado por...	Criticidad	Fecha
Implementación	Movimiento del equipo	Suspensión de energía programado	alto	dd/mm/yyyy
Actividad interna	Actividad externa	Actividad externa	Alto/medio/bajo	Descripción....

Tabla 1.17 Dependencias externas

- Plan de recursos

Describe el nivel de los recursos que son requeridos por el proyecto listado por roles, la fecha de inicio, la fecha de fin y el porcentaje de tiempo que va a ser ocupado en el proyecto, puede utilizarse como ejemplo la tabla 1.18.

Rol	Fecha de inicio	Fecha fin	% de tiempo
Rol del proyecto	dd/mm/yyyy	dd/mm/yyyy	porcentaje
...	...	...	...

Tabla 1.18. Plan de recursos

- Plan de finanzas

Se detallan todos los costos que se generan para el proyecto. Se sugiere utilizar la tabla 1.19

Categoría	Costo	Valor
Personal	Salarios	\$x
	Outsourcing	\$x
	Capacitaciones	\$x
...	...	...

Tabla 1.19 Plan financiero

- Plan de calidad

Se listan todos los procesos dentro del proyecto que deben ser analizados para medir su calidad. Puede ser utilizada la tabla 1.20 como ejemplo.

Proceso	Descripción
Manejo del tiempo	Descripción...
...	...

Tabla 1.20 Documentación de procesos

Listar todas las consideraciones del proyecto

- Riesgos del proyecto

Se deben listar todos los riesgos que sean aparentemente reconocidos en el proyecto, detallando la descripción del mismo, probabilidad, impacto, y el plan de acción a tomar.

- Dificultades de recursos

Se deben listar todas las dificultades encontradas hasta este punto, documentando la dificultad con una pequeña descripción, la prioridad y la acción a tomar para resolverla.

- Asunciones del proyecto

Listar las mayores asunciones que se identificaron al definir el proyecto.

- Restricciones del proyecto

Listar las restricciones identificadas al definir el proyecto, por ejemplo que la empresa a quien se le realice no cuente con una infraestructura adecuada, entre otros.

4.1.4.4 Designar equipo para el proyecto

Teniendo claro los términos de referencia, se tiene claro del tipo de roles requeridos en el proyecto y el número de personas a necesitar para cada uno. El siguiente paso es asignar al personal a dichos roles, para empezar con el proyecto. La primera persona a asignar es el administrador o coordinador, responsable de elegir a los mejores candidatos con las mejores habilidades para integrar al equipo. El

administrador elegirá a los líderes y el equipo en sí, dependiendo de la magnitud del proyecto.



Descripción del trabajo

Para asignar de mejor manera al personal y sus roles se debe crear una descripción de los puestos, donde se determine el papel a desempeñar, responsabilidades, experiencia, habilidades y la calificación necesaria para llevar a cabo de la mejor manera.

- Roles

Describe el propósito de cada uno de los roles, por ejemplo el propósito del administrador del proyecto.

- Responsabilidades

Lista cada una de las responsabilidades de cada rol. Para cada fase del ciclo de vida del proyecto se deben detallarlas.

- Organización

Representa la estructura de la organización del proyecto, es decir, un organigrama de los integrantes con sus roles del proyecto.

- Relaciones

Describe las relaciones entre el administrador del proyecto y cada uno de los involucrados en él, por ejemplo: el patrocinador, grupo de implementación, líder del proyecto o supervisores, y los miembros del equipo en general.

- Habilidades

Identificar las habilidades y los conocimientos necesarios para pertenecer a los roles.

- Experiencia

Identificar el nivel de experiencia requerido para cada rol. La experiencia puede ser definida bajo los siguientes términos:

- a) Tipo de proyectos en los que el candidato ha estado involucrado
- b) Empresas o Industrias en las que ha llevado a cabo los proyectos
- c) Tamaño o nivel de complejidad de los proyectos a su cargo
- d) Número de años de experiencia en proyectos
- e) Si el proyecto o los proyectos han sido a nivel nacional o internacional.

- Calificaciones

Los grados académicos alcanzados son requeridos muchas veces dependiendo del rol que se esté evaluando. La calificación por niveles incluye:

- a) Bachillerato
- b) Universitario
- c) Con especialización

- Personalidad

Describe las características personales que se requieren para cada rol, se pueden mencionar:

- a) Amigable, de mente abierta y acucioso por naturaleza
- b) Estructurado, metódico y capaz de seguir procedimientos
- c) Inspirador, motivador y animado
- d) Fuerte a la hora de tomar decisiones

- Criterios de rendimiento

Listar los indicadores de rendimiento que proveen un mecanismo para monitorear el buen desempeño de las personas en los distintos roles.

- Ambiente de trabajo

Describe el ambiente de trabajo en el que se desarrollará el proyecto, para potenciar el trabajo en general de los candidatos y las condiciones de trabajo asociadas a cada rol. Por ejemplo:

- a) Físico Exterior: trabajo fuera de la empresa.
- b) Físico de desarrollo: Trabajo de escritorio

- Salario

Especificar el salario esperado por rangos y la composición del salario en paquetes, incluyendo comisiones y beneficios dependiendo del rol asociado.

- Condiciones especiales

Listar todas aquellas condiciones especiales asociadas a cada rol, por ejemplo:

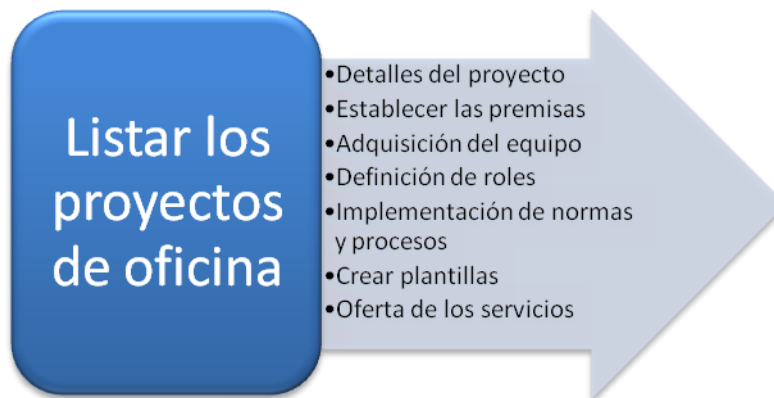
- a) Identificar cualquier licencia en especial, certificados o credenciales requeridos
- b) Disponibilidad de tiempo y de viajar, tanto fuera de la ciudad como del país

- c) Disponibilidad para trabajar horas extras, si es requerido

4.1.4.5 Instalar la oficina para el proyecto

La oficina del proyecto es el lugar donde estará ubicado el equipo que lo llevará a cabo. Esta oficina contiene la infraestructura de comunicación entre el equipo y la tecnología de la información para soportar el proyecto.

En el caso de que el equipo del proyecto esté disperso por muchos lugares en el país, porque las características del mismo así lo requieran, se crea una oficina “virtual”. Con la nueva tecnología las oficinas virtuales permiten la comunicación mediante email, acceso a internet, redes remotas de acceso (VPN), celulares, laptops u otros dispositivos que permitan la comunicación.



- Detalles del Proyecto.

En este punto se debe listar toda la información de proyecto, tal como: Nombre del proyecto, nombre del responsable del proyecto y nombre del responsable de la oficina del proyecto.

- Establecer las premisas

Se deben dejar claras las premisas o condiciones bajo las cuales se pactará la oficina, mediante la documentación de: Las condiciones físicas que el proyecto exige; un contrato formal para las locaciones, consideraciones que se tomaron para elegir el local; si el local no cumple ciertos requisitos a cabalidad, estos se deben

especificar, tareas extras a realizar una vez entregado el local (cableado, aire acondicionado, etc).

- Adquisición del equipo

Se debe establecer el equipo necesario para llevar a cabo el proyecto, ejemplo: número de computadoras, software, proyectores, fax, impresoras, scanner, copiadoras, etc., también se debe hacer una jornada de los mantenimientos del equipo y dejar en un lugar visible; así mismo se deben documentar las medidas de contingencia en caso de que un equipo falle.

- Definición de roles

Se debe documentar el equipo que dependiendo del rol establecido, va a necesitar (computadora, fax, teléfono, Internet, etc.) Una vez establecido, asignar lo necesario, de manera que se haga un inventario de lo que se tiene vs. lo que se necesita.

- Implementación de normas y procesos

Se deben establecer normas o estándares de calidad, planeación, aceptación, reportes de documentación u otros relevantes, por ejemplo ISO.

Una vez documentadas las normas o estándares a implementar, se debe hacerlo también con los procedimientos a seguir: control del tiempo, de los costos, de calidad, de cambios, riesgos, dificultades, aceptación, etc.

- Crear plantillas

Se deben crear las plantillas a utilizar para documentar cada etapa del proyecto (iniciación, planificación, ejecución y cierre).

- Ofertas de los servicios

En esta etapa se documentan aquellas ofertas de los servicios adquiridos, colocando las restricciones de mantenimiento que estas presenten o algún dato importante que se deba tomar en cuenta sobre ello.

4.1.4.6 Realizar fase de revisión

En este momento se debe realizar un alto, pues es el fin de la inicialización del proyecto; es un punto de chequeo para verificar que todo lo planteado hasta aquí conlleve al cumplimiento de los objetivos. Esta fase debe llevarse a cabo para pasar a la de planificación del proyecto, porque es en este momento donde se decide si se aprueba o se cancela. La fase de revisión debe completarse y documentarse como tarea del encargado del proyecto y ser aprobada por el patrocinador del mismo.

Para lograrlo se sugiere utilizar el formato planteado en la siguiente plantilla:

ACTA DE ACEPTACIÓN DE LA FASE DE INICIALIZACIÓN DEL PROYECTO

DETALLES DEL PROYECTO

Nombre del proyecto:	Reporte preparado por:
Encargado del proyecto:	Fecha de preparación:
Patrocinador del proyecto:	Período del reporte:
Descripción del proyecto:	

ESTATUS GLOBAL

Posición Actual:

Calendarización:

Gastos del proyecto:

Entregables:

Riesgos:

Dificultades:

Cambios:

DETALLES DE LA REVISION

Categoría	Pregunta	Respuesta	Variación
Calendario	Fue la fase cumplida en el tiempo y fecha establecida	S/N	
Gastos	Fue la fase cumplida con el presupuesto planteado	S/N	
Entregables	Caso del negocio: ¿fue el caso del negocio aprobado?	S/N	
	Estudio de viabilidad: ¿Fue el estudio de viabilidad aprobado?	S/N	
	Términos de Referencia: ¿Fueron los términos de referencia aprobados?	S/N	
	Definición de Roles: ¿Fue reclutado todo el equipo siguiendo los roles establecidos?	S/N	
Riesgos	¿Existe algún riesgo pendiente de validar?	S/N	
Dificultades	¿Existe alguna dificultad pendiente de validar?	S/N	
Cambio	¿Existe algún cambio pendiente de validar?	S/N	

DETALLES DE APROBACION

Documentación de soporte: (Se hace referencia a documentos que sustenten los detalles mencionados en el documento)

Patrocinador del proyecto:

Firma: _____

Fecha: __/__/__

El proyecto ha sido aprobado para proceder a su fase de planificación.

4.1.5 Proceso de Planificación del Proyecto

Ahora que el proyecto ha sido propiamente definido y el equipo del mismo establecido, se debe planificar a detalle. Las fases de planificación motivan a crear un espacio para documentarlas, lo cual ayuda como guía para el equipo que desarrollará el proyecto.

El grupo de procesos de planificación incluye la realización de ciertas actividades, como se mira en la figura 2.1.

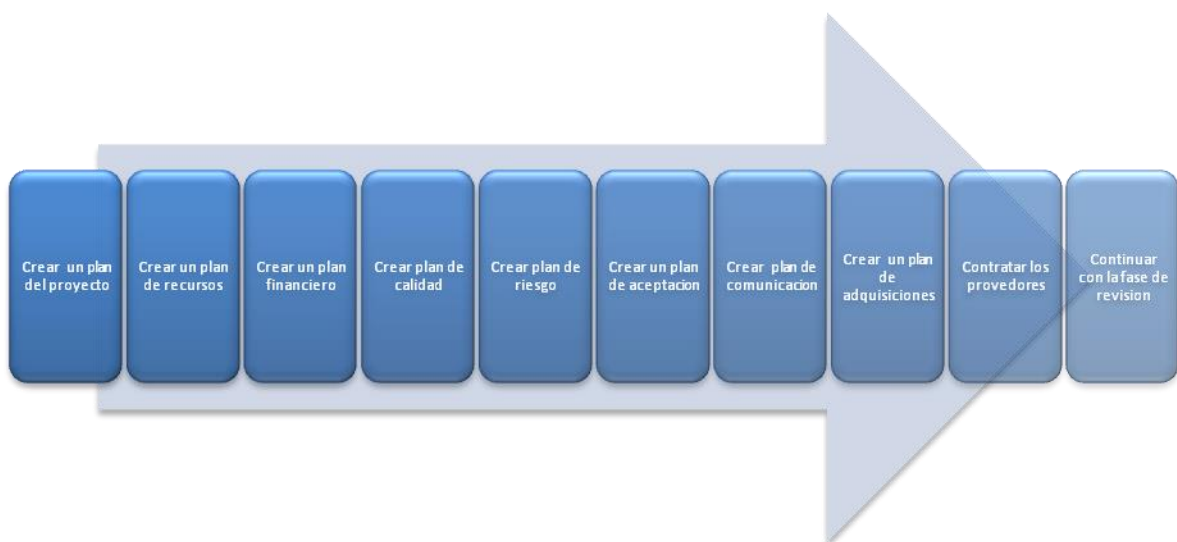


Figura 2.1. Grupo de procesos de planificación

Cuando se está hablando de un proyecto pequeño estas actividades se consolidan en un solo documento pero en el caso de un proyecto grande se crea un documento, por cada etapa para que éste o estos sean aprobados por el encargado y se asegure que cada actividad responde a una secuencia.

El documento más importante a crear durante la fase de planificación es el plan del proyecto. A este documento se hará referencia continuamente en la fase de ejecución, con el fin de que el proyecto esté siguiendo los lineamientos inicialmente planteados. Una buena definición del plan del proyecto ayudará al administrador de esté a tener una visión clara de las actividades y los alcances que los usuarios quieren del mismo.

4.1.5.1 Crear un plan de proyecto

Un plan de proyecto es un documento central donde se maneja el proyecto formalmente. Este plan lista las actividades, tareas y recursos necesarios para completarlo y realizar los beneficios para la empresa planteados en la fase anterior. Un plan de proyecto típico incluye:

- Una descripción de las fases que comprenden el proyecto
- Una calendarización de las actividades, tareas, riesgos, duración, dependencias, recursos y tiempos
- Lista de asunciones y restricciones identificadas durante la planificación de los procesos



Figura 2.2 Figura de procesos de plan de proyecto

Definir las bases de la planificación

- Alcance

El primer escalón para crear un plan de proyecto es reconfirmar el alcance del proyecto, definiendo los términos de referencia. El alcance define lo que se realizará en el proyecto, es decir todas aquellas actividades a realizarse y las tareas que esto requiere para que se complete.

- Fases

Listar y describir las fases en el proyecto. Una fase es un conjunto de actividades necesarias para desarrollar y obtener una porción de todo el proyecto.

- Éxitos

Enumerar o clasificar los puntos de éxito del proyecto.

- Actividades

Listar y describir las actividades en el proyecto. Cualquier actividad es un conjunto de tareas que son requeridas para completar cada porción del proyecto. Una sugerencia para documentar las actividades es la tabla 2.1.

Fase	Actividad	Descripción	Secuencia
Planificación	Efectuar un plan de desarrollo	Produce un documento que describe cualitativamente los procesos y las revisiones de las actividades	Después del plan del proyecto, pero antes de la formulación de los contratos.
...	...	...	...

Tabla 2.1 Tabla para la documentación de actividades

- Tareas

Listar todas las tareas requeridas para llevar a cabo las actividades del proyecto. Una tarea es una asignación de trabajo para ser completada en el proyecto. Para documentar las tareas se sugiere la tabla 2.2

Fase	Actividad	Tarea	Secuencia
Planificación	Efectuar un plan de desarrollo	1. Identificar tareas cualitativas	1ra
		2. Identificar las técnicas	2 da
		3. Identificar las técnicas de control	3ra
		4. Documentar el plan de calidad	4ta
...	...	...	...

Tabla 2.2 Tabla para la documentación de tareas

- Esfuerzo

Para cada tarea se debe especificar el esfuerzo requerido para completarla. Se recomienda realizarlo mediante la tabla 2.3

Tarea	Esfuerzo
1. Identificar tareas cualitativas	1 día
2. Identificar las técnicas	
3. Identificar las técnicas de control	2 días
4. Documentar el plan de calidad	1 día
...	5 días
...	...

Tabla 2.3 Tabla para la documentación de esfuerzos

- Recursos

Para cada tarea listada, se deben identificar los recursos humanos necesarios para completar la tarea.

Crear un calendario del proyecto

- Calendario

Crear detalladamente un calendario del proyecto, listando cada una de las fases, actividades y tareas requerida para completarlo. Un programa sugerido para desarrollar este punto es “Microsoft Project” que es un complemento del paquete de office de Microsoft.

- Dependencias

Dependencias son relaciones lógicas entre fases, actividades o tareas que influyen en el camino del proyecto para ser desarrollado. Dependencias pueden ser internas del proyecto (entre cada actividad del mismo) o externas al proyecto (entre las actividades del proyecto y las actividades de la empresa). Hay 3 tipos de dependencias:

De fin a inicio: La actividad, para ser finalizada depende, del inicio de otra actividad.

De fin a fin: La actividad, para ser finalizada depende, de la finalización de otra actividad.

De inicio a inicio: La actividad, para ser iniciada depende, del inicio de otra actividad.

De Inicio a fin: La actividad, para ser iniciada depende, del fin de otra actividad.

Para documentar las dependencias se puede ejemplificar en la tabla 2.4

Actividad	Depende de...	Tipo de dependencia
Instalar la oficina del proyecto	Contratar todo el equipo	De fin a fin
...	...	...

Tabla 2.4 Tabla para la documentación de dependencias

En este ejemplo, la actividad “instalar la oficina del proyecto” será finalizada hasta que la actividad “contratar todo el equipo” sea concluida.

- Asunciones

Identificar y planificar asunciones a realizarse durante el proceso de planificación. Por ejemplo: “el alcance del proyecto no admite cambios”.

- Restricciones

Identificar y planificar las restricciones. Por ejemplo: El equipo del proyecto realizará las tareas y entregará el producto sin trabajar horas extras.

4.1.5.2 Crear un plan de Recursos

Un plan de recursos describe los recursos físicos que se requieren para completar el proyecto. Este incluye una lista de los diferentes tipos, para cada tarea, equipo y materiales. En una calendarización se identifica cuando cada recurso será utilizado.

Para crear un plan de recursos se deben: listar en general todos los tipos de recursos a utilizar, identificar el número y el propósito de cada tipo de ellos, cuando van a ser utilizados y asignar los recursos a las actividades del proyecto.

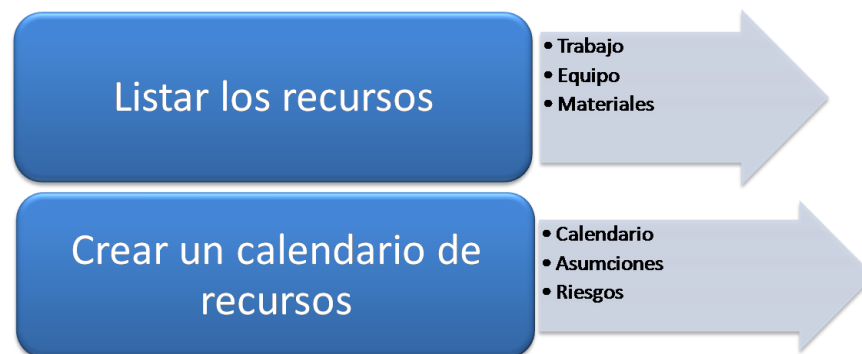


Figura 2.3 Figura de procesos de plan de recursos

Listar los recursos

Se deben listar los recursos a ser utilizados en el proyecto. Un “recurso” está definido como un trabajo, equipo y materiales usados para completar cada actividad en el proyecto.

- Trabajo

Agrupar los roles, responsabilidades y el conjunto de habilidades requeridos para completar el proyecto. Esta etapa se puede documentar mediante la tabla 2.5

Rol	N°	Responsabilidades	Habilidades	Fecha de Inicio	Fecha de Finalización
Administrador del proyecto	1	Proporcionar la solución aprobada para conocer todos los requerimientos del cliente	Administra ción del tiempo de costos de calidad y del personal	dd/mm/yy	dd/mm/yy
...	...	...	...	...	...

Tabla 2.5 Tabla para la documentación de trabajo

El campo N° en la tabla 2.5 representa la cantidad de personas requeridas de dicho rol. Los campos Fecha de inicio y Fecha de finalización identifican cuánto tiempo va a estar vigente el rol. En el caso del administrador del proyecto la fecha de inicio será durante la fase de inicialización y la fecha de finalización será cuando se haya completado la fase de cierre del proyecto.

- Equipo

Ahora que ya se ha identificado el trabajo, es necesario listar todo el equipo a necesitar. Incluye computadoras, muebles, maquinaria, vehículos o cualquier otro equipo necesario para completar el proyecto. Para documentar el equipo se sugiere utilizar la tabla 2.6.

Item	N°	Propósito	Especificación	Fecha de Inicio	Fecha de Finalización
Laptop	1	Proporcionarle al administrador del proyecto una computadora para planificar, monitorear y controlar el proyecto desde cualquier lugar.	Alto procesamiento y pantalla adecuada	dd/mm/yyyy	dd/mm/yyyy
...	...	...	...	...	...

Tabla 2.6 Tabla para la documentación de equipo

El campo N°. Significa el número de equipo requerido, y la fecha de inicio y fin identifican el tiempo que van a ser utilizadas.

- **Materiales**

Ahora se tiene que listar todos los materiales genéricos requeridos para el proyecto, incluyendo materiales de oficina, de computadora, agua, etc., Cada elemento de los materiales debe ser definido mediante la lista y el uso que tendrá.

Crear un calendario de recursos

- Calendario

Ahora que todos los recursos han sido listados, se deben identificar cuando este recurso va a ser utilizado en el proyecto. Se puede hacer esto creando un calendario de recursos, éste permite al administrador del proyecto identificar la cantidad de recursos por día, semana o mes.

- Asunciones

Listar cualquier asunción tomada durante la creación del plan de recursos.

- Riesgos

Listar cualquier riesgo identificado durante la creación del plan de recursos.

4.1.5.3 Crear un plan financiero

Un plan financiero agrupa información financiera descrita en el caso del negocio, estudio de viabilidad, términos de referencia y el documento del plan del proyecto.

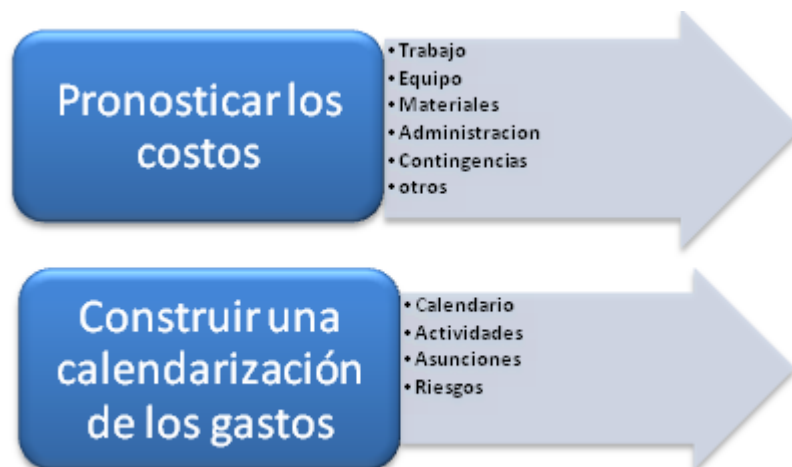


Figura 2.4 Figura de procesos de plan financiero

Pronosticar los costos

En este punto se detallan los puntos financieros en cuanto a los costos requeridos para completar el proyecto

- Trabajo

Se debe detallar por rol el costo unitario que representa, tal como se muestra en la tabla 2.7

Rol	Costo unitario
Administrador del proyecto	\$ Costo por unidad
Coordinador del proyecto	\$ Costo por unidad
...	...

Tabla 2.7 Tabla para la documentación de costo unitario

- Equipo

Listar cada equipo requerido por el proyecto y su costo por unidad, como se detalla en el ejemplo de la tabla 2.8

Equipo	Costo unitario
Computadoras	\$ Costo por unidad
Muebles	\$ Costo por unidad
Vehículos	\$ Costo por unidad
...	...

Tabla 2.8 Tabla para la documentación de equipo

- Materiales

Listar los materiales requeridos en el proyecto con sus costos, tal como se ejemplifica en la tabla 2.9

Material	Costo unitario
Cds	\$ Costo por unidad
Materiales de construcción	\$ Costo por unidad
Páginas	\$ Costo por unidad
...	...

Tabla 2.9 Tabla para la documentación de materiales

- Administración

Listar cualquier costo administrativo en el que se incurra o incurrirá en el proyecto y sus costos asociados

- Contingencias

Identificar cualquier costo de contingencia tomado en cuenta. “Costo de contingencia” se refiere a los impredecibles durante la fase de planificación del proyecto, pero son razonablemente incluidos, aunque se tenga el riesgo de que el proyecto exceda el presupuesto

- Otros

Listar cualquier otro costo en el que se haya incurrido en el proyecto.

Construir una calendarización de gastos

- Calendario

Se necesita decidir cuándo los gastos podrían ocurrir. Se debe hacer una calendarización mensual para identificar cuándo ocurrirá dicho gasto.

- Actividades

Después de conocer los gastos a considerar, es importante saber el costo de sobrellevar dichas actividades (tal como el total de los costos de llevar a cabo el producto a entregar). Se debe cuantificar el total por mes de los costos de cada

actividad listada en el plan del proyecto, pero para construir dicho reporte mensual es necesario para un mejor control llevar un consolidado semanal.

- Asunciones

Identificar cada asunción realizada durante el proceso de la planificación financiera.

- Riesgos

Listar los riesgos que se corren durante el ejercicio de la planificación financiera. Por ejemplo que el equipo requerido no esté disponible cuando se necesite utilizar podría hacer que se incurra en la adquisición de equipo adicional para suplir la necesidad.

4.1.5.4 Crear un plan de calidad

Un plan de calidad puede ser tan extenso como lo requiera el producto a entregar. Calidad es usualmente considerado bajo dos diferentes perspectivas: La calidad de cada producto a entregar para el usuario, y la calidad en la dirección de los procesos involucrados para llevar a cabo el producto final. Por esta razón, un plan de calidad no solo define el nivel de aprobación de calidad de cada entregable, sino ideas de dirección de procesos que influenciaran al producto final para que no se necesite modificar nada al final evitando los riesgos de fracasar.

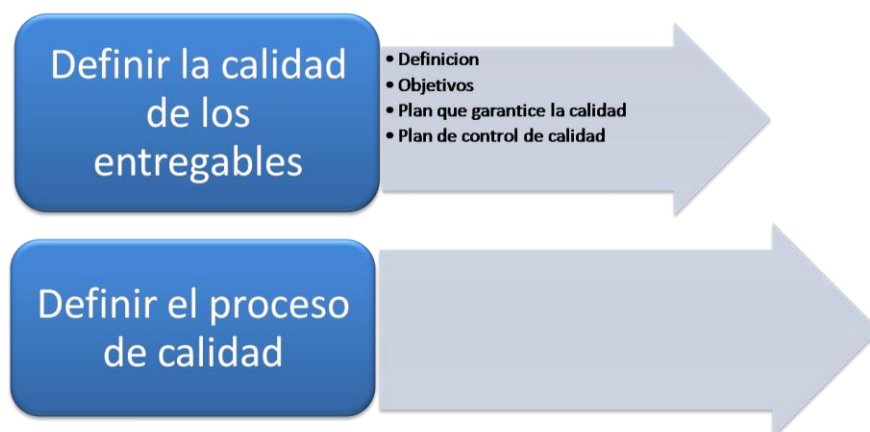


Figura 2.5 Figura para la creación de plan de calidad

Definir la calidad de los entregables o productos

El primer paso es considerar cómo se asegura y controla la calidad de los productos a entregar en el proyecto. Para hacer lo anterior se deben definir los términos de calidad y asignar los diferentes elementos que permitirán monitorear y controlar que esto se lleve a cabo.

- Definición

Para asegurar que se tiene un concepto consistente del término “calidad” en el proyecto, se debe definir formalmente el término así:

“La calidad es el grado en que el producto final se ajusta a las necesidades de los clientes”

- Objetivos

Para el proyecto se deben identificar los objetivos de calidad que se persiguen y una vez encontrados, se debe asegurar que el producto final se apegue a los requerimientos de los usuarios. Un ejemplo de objetivos de calidad se muestra en la tabla 2.10

Requerimientos del proyecto	Producto final del proyecto	Criterio de calidad	Standard de calidad
Poder manejar los estados financieros y todos aquellos pagos y procesos recibidos en el área de las finanzas	Implementación de Oracle Financials (GL), Cuentas de pagos (AP)y cuentas de cobros mediante módulos de un sistema(AR)	Funcionalidad del sistema:	Funcionalidad del sistema:
		GL probado e instalado	GL Operacional, AP operacional sin errores, AR operacional sin errores.
		AP Probado e instalado	
		AR probado e instalado	
		Rendimiento del sistema:	Rendimiento del sistema:
		Respuesta de sistema	99.9% tiempo del sistema
		Disponibilidad del sistema	¿Menor q 5 segundos en tiempo de respuesta?
		Migración de la información	100% de calidad de la información

Tabla 2.10 Ejemplo de objetivos de calidad

- Plan que garantice la calidad

Para proporcionar al cliente la garantía de la calidad de los productos a entregar, técnicas de calidad (QA) se deben definir. Los QA son pasos preventivos que permiten eliminar la variación en la calidad del producto final.

Por ejemplo, para garantizar la calidad algunas tareas sugeridas, son: Revisar información histórica para entender otros proyectos relacionados y las dificultades encontradas en los mismos; definir los requerimientos documentando comprensivamente los que el usuario ha realizado; definir estándares los cuales serán seguidos en el desarrollo del proyecto; reclutar personal calificado o con conocimiento amplio de lo que se desarrollará en el proyecto; efectuar revisión de la

calidad periódicamente; implementar cambios de control, lo cual permitirá cambiar solamente aquellos aspectos que estén erróneos dentro del proyecto.

- Plan de control de calidad

Un plan de control (QC) es una técnica a ser implementada. Los QC definen un paso curativo para eliminar cualquier variante en la calidad del producto final producida desde la calidad de cada una de las actividades realizadas. Los tipos de control son: por revisión, es decir, se revisa el trabajo de cada miembro del equipo, revisión de producto final, documentación de las revisiones, fase de revisiones que son fases formales donde se revisa todo lo realizado en un determinado periodo de tiempo.

Definir el proceso de calidad

Se debe definir el proceso de calidad de las tareas de administración que conllevarán a entregar el producto final. Para cada uno de los procesos siguientes, se deben describir los pasos involucrados para entender el proceso y las responsabilidades que el recurso tiene para llevarlo a cabo: Administración del proceso del tiempo, del proceso de los costos, de los procesos cambiantes, de los riesgos, dificultades, aceptación y adquisiciones.

4.1.5.5 Crear un plan de riesgos

Ahora que se han definido las actividades, recursos y finanzas del proyecto es tiempo de definir a detalle los riesgos del mismo. Un plan de riesgos lista todos los riesgos pronosticables en el proyecto, y las acciones a realizar para prevenir que ocurran y reducir el impacto que estos generen.



Figura 2.6 Figura para la creación de plan de riesgos

Identificar los riesgos

Se deben identificar todos aquellos riesgos que puedan afectar al proyecto y que atenten con evitar el logro de los objetivos. Se deben categorizar los riesgos listados, y para cada categoría asignar los riesgos potenciales.

- Definición

Para asegurar que se tiene un concepto consistente del término “riesgo” en el proyecto, se debe definir formalmente el término así:

“Un riesgo es un evento que pueda afectar negativamente a la capacidad del proyecto para alcanzar los Objetivos definidos.”

- Categorías

Identificar las categorías de los riesgos del proyecto. Una categoría es un aspecto particular del proyecto que podría experimentar algún riesgo durante el curso del proyecto. Las categorías típicas, son: Requerimientos, beneficios, calendarización, presupuesto, producto final, alcance, dificultades, proveedores, aceptación, comunicación, recursos.

- Riesgos

Identificar los riesgos potenciales para cada categoría, un ejemplo puede ser visto en la tabla 2.11

Categoría de riesgo	Descripción del riesgo	ID Riesgo
Requerimientos	Los requerimientos no han sido claramente definidos	1.1
	Los requerimientos especificados no son los que el usuario necesita	1.2
	Los requerimientos especificados no son medibles	1.3
Beneficios	Los beneficios para la empresa no han sido definidos	2.1
	Los beneficios para la empresa no son cuantificables	2.2
	La solución final no logra los beneficios requeridos	2.3
...	...	...

Tabla 2.11 Tabla para identificar los riesgos

Cuantificar los riesgos

- Probabilidad

Crear un sistema de puntuación utilizado para medir la probabilidad de que cada riesgo ocurra, esto se puede realizar siguiendo el formato de la tabla 2.12

Título	Puntuación	Descripción
Muy bajo	20	Altamente probable que no ocurra basado en la información periódica, las circunstancias y demás variables a considerar
Bajo	40	Probable que no ocurra, pero debe ser monitoreado para verificar las circunstancias que pueden repercutir que durante el proyecto ocurra
Medio	60	Puede ocurrir y es claro que el riesgo es eventualmente probable
Alto	80	Muy probable que ocurra, basado en las circunstancias del proyecto
Muy Alto	100	Altamente probable que ocurra, se debe poner mucha atención en estos, puesto que pueden afectar al proyecto grandemente,
...	...	...

Tabla 2.12 Tabla para cuantificar los riesgos

- Impacto

De igual manera que el anterior se debe crear un sistema de puntuación para el impacto que el riesgo tendrá si llegase a ocurrir.

Título	Puntuación	Descripción
Muy bajo	20	Insignificante impacto en el proyecto. Es posible que la medición del impacto sea mínima
Bajo	40	Menor impacto en el proyecto.
Medio	60	Impacto medible en el proyecto
Alto	80	Impacto significativo en el proyecto
Muy Alto	100	Mayor impacto del proyecto.
...	...	...

Tabla 2.13 Tabla para identificar el impacto

- **Prioridad**

La prioridad es una puntuación y se calcula como un promedio entre la probabilidad y el impacto de la siguiente forma:

$$\text{Prioridad} = [\text{probabilidad} + \text{impacto}] / 2$$

El ranking propuesto para la prioridad se puede ver en la tabla 2.14

ID de riesgo	Probabilidad	Impacto	Prioridad	Ranking
1.1	20	80	50	Medio
1.2	80	60	70	Alto
1.3	40	20	30	Bajo
...	...	...	...	...

Tabla 2.14 Tabla para priorizar los riesgos

Este ranking es basado en el cálculo de la prioridad. Se propone en la tabla 2.15 para determinar los valores asignados.

Puntaje de prioridad	Ranking de prioridad	Color
0-20	Muy baja	Blanco
21-40	Baja	Verde
41-60	Medio	Amarillo
61-80	Alto	Anaranjado
81-100	Muy Alto	Rojo

Tabla 2.15 Tabla con valores asignados en priorización

Crear una calendarización de los riesgos

- Calendario

Para cada riesgo identificado, crear acciones preventivas requeridas para reducir la probabilidad de que ocurra, acciones de contingencia para minimizar el impacto que éste tenga en el proyecto si ocurre. A cada acción, se debe asignar un responsable para llevarla a cabo y la fecha en la que se completará. Se sugiere la tabla 2.16 para llevar la información, esta se debe completar con cada riesgo identificado; a los que sean de Alta prioridad se deben asignar más acciones comprensivas, de ser posible.

Ranking	ID Riesgo	Acciones Preventivas	Encargado	Fecha de acción	Acciones de contingencia	Encargado	Fecha de acción
Muy Alto	2.2	Identificar de manera clara los beneficios de la empresa	Gestor del proyecto	dd/mm/yy	Medir los actuales beneficios para empresa con el logro del proyecto	Administrador del proyecto	dd/mm/yy
...	...			...			

Tabla 2.16 Tabla con la documentación de los riesgos

4.1.5.6 Crear un plan de aceptación

La “aceptación” está definida como el momento en el que el usuario está de acuerdo con el producto final del proyecto, bajo los criterios previamente definidos. Estos criterios califican la calidad y los costos del producto final.

El plan de aceptación es referido durante la ejecución del proyecto y se usa para confirmar que cada producto final generado está completado y listo para la aceptación de los usuarios. Este documento se hace referencia en el reporte de cierre y en la fase de retroalimentación de post-implementación.



Figura 2.7 Figura para la creación de plan de aceptación

Identificar los criterios de aceptación

- Definición

Para asegurar que se tiene un concepto consistente del término “aceptación” en el proyecto, se debe definir formalmente así:

“Aceptación está definido como el acuerdo ganado entre los usuarios finales y los ejecutores del proyecto, acordando que el producto final estará basado en los criterios previamente establecidos por el usuario.”

- Éxitos

Listar los éxitos del proyecto y los productos finales que el usuario final aceptará.

Éxitos		Producto Final	
Nombre	Descripción	Nombre	Descripción
Sistema de Finanzas actualizado	Implementación del paquete de software en el hardware	Paquete de software instalado	Implementación del Finanzas General (FG), pago de cuentas (PC)
...	...	...	...

Tabla 2.17 Tabla con los éxitos del proyecto y productos finales

- Criterios

Identificar los criterios y estándares para lograr la aceptación final de los usuarios para cada producto final.

Producto Final	Criterios	Estándares
Instalación del paquete de software	Funcionalidad del sistema	Funcionalidad del sistema
	• P1 probado e instalado • P2 probado e instalado	• P1 operacional sin errores • P2 operacional sin errores
	Rendimiento del sistema	Rendimiento del sistema
	• Tiempo de sistema • Tiempo de respuesta • Transferencia de información	• 99.9% de tiempo de sistema • Menor a 1 tiempo de Respuesta • 100% de la información preciso
	...	...
...	...	...

Tabla 2.18 Tabla con productos finales, criterios y estándares

Construir una calendarización de la aceptación

- Calendario

Crear una calendarización de las revisiones necesarias para generar los productos finales, bajo los criterios y estándares definidos. Un ejemplo se muestra en la tabla 2.19

Éxitos completados	Producto final	Fecha	Método de revisión de aceptación	Revisión	Fecha
Sistema financiero actualizado	Paquete de software instalado	dd/mm/yy	Inspección física	Prueba de sistema	dd/mm/yy
			Pruebas de software	Administración del proyecto	
			Revisión de la información		
...	...			...	

Tabla 2.19 Tabla que construye calendarización de aceptación

- Asunciones

Listar cualquier asunción tomada en cuenta como plan de aceptación. Por ejemplo: No habrá cambios en los requerimientos durante el proyecto, la aceptación de los criterios no cambiarán durante el mismo, los encargados estarán disponibles para conducir las revisiones requeridas.

- Riesgos

Listar cualquier riesgo identificado durante la fase de aceptación. Por ejemplo: que los criterios de aceptación no estén directamente alineados con los requerimientos de los usuarios.

4.1.5.7 Crear un plan de Comunicación

El plan de comunicación describe la información a ser difundida a todos los involucrados en el proyecto para mantenerlos informados de su progreso. Un plan de comunicación claro es vital para el desarrollo del proyecto, pues asegura que todos sus recursos y los involucrados estén trabajando bajo los mismos objetivos, y cualquier obstáculo se supera en conjunto.

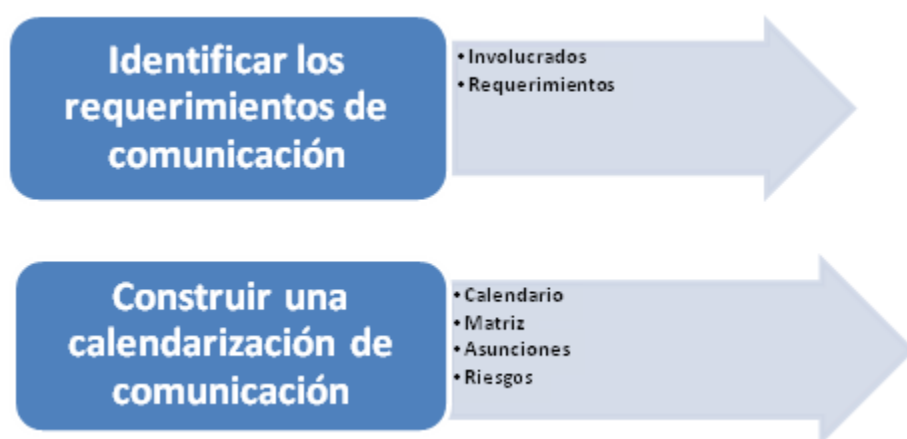


Figura 2.8 Figura para la creación de plan de comunicación

Identificar los requerimientos de comunicación

- Involucrados

La primera fase para definir un plan de comunicación es listar todos los involucrados. Un involucrado en la comunicación es una persona o entidad dentro o fuera del proyecto que requiera información de él. Por defecto, el patrocinador del proyecto estará interesado en conocer su progreso, mientras que los entes externos estén concentrados en que el área legislativa o regulatoria se cumpla (auditoría).

- Requerimientos

Para todos los involucrados identificados, se requiere describir la información que estos necesitan para mantenerlos apropiadamente informados del progreso del proyecto.

Construir una calendarización de la comunicación

- Calendario

Es la acción de definir momentos de comunicación, incluyendo el propósito, método y frecuencia. Como sugerencia se presenta la tabla 2.20.

ID	Evento	Descripción	Propósito	Método	Frecuencia	Fecha
1.1	Reuniones con el equipo del proyecto	Reunión para involucrar al equipo del proyecto y discutir el trabajo en progreso, recientemente completado, y próximos pasos	Mantener al equipo informado del status del proyecto y superar las dificultades, riesgos o cambios alcanzados	Verbal	Semanal	xx/mm/yy
...	...	...	...	...	...	...

Tabla 2.20 Tabla que construye la calendarización de comunicación

- Matriz

Completar una tabla con base a la figura 2.9 para ayudar a identificar las personas participantes en cada evento de comunicación y sus roles. Usar identificador ID en la tabla anterior para hacer referencia al evento.

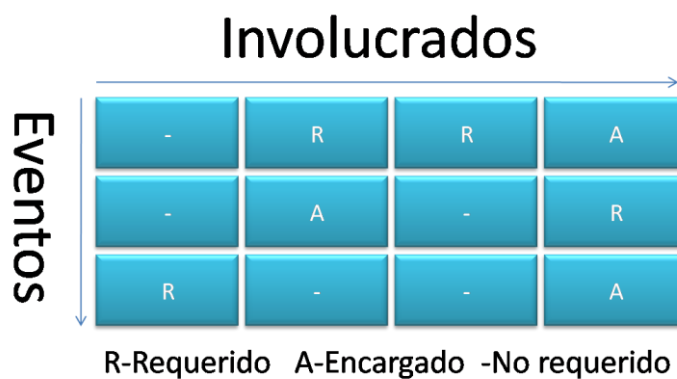


Figura 2.9 Figura para la creación de plan de comunicación

La matriz muestra en las columnas cada uno de los involucrados identificados en el paso 1 y las filas, los eventos detallados en el punto anterior. El cruce de fila y columna significa el rol que desempeñará esa persona en el evento (Requerido, Encargado, No requerido).

- Asunciones

Listar cualquier asunción hecha durante el proceso del plan de comunicación. Por ejemplo, que la herramienta de comunicación será provista cuando sea requerida.

- Riesgos

Listar cualquier riesgo identificado durante la fase de la creación del plan de comunicación. Por ejemplo: Los requerimientos de comunicación pueden cambiar durante la ejecución del proyecto.

4.1.5.8 Crear un plan de Adquisiciones

El plan de adquisiciones identifica los productos a ser adquiridos por proveedores externos y los periodos y métodos de la adquisición. Para efectos de connotación de esta guía el término “productos” se refiere a bienes (computadoras, materiales y equipos o cualquier herramienta a utilizar) y servicios (servicios técnicos, consultorías, administradores, etc). Este plan será desarrollado únicamente para proyectos que requieran de la adquisición de proveedores externos.

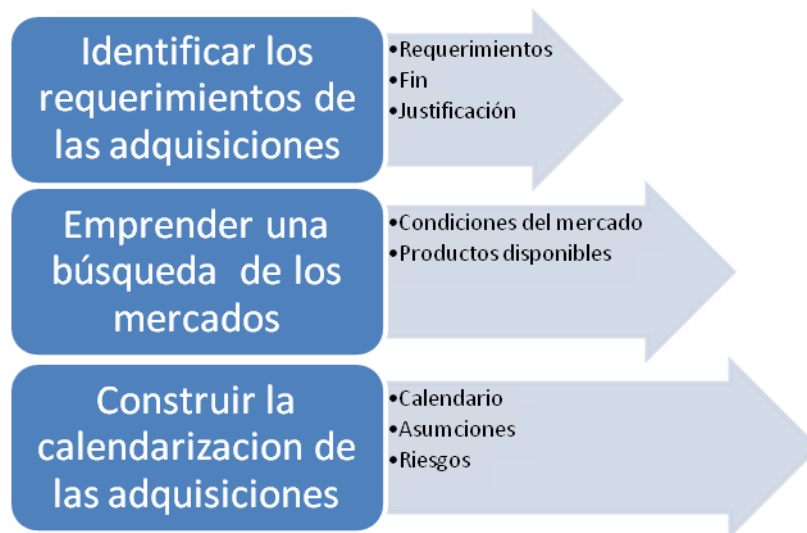


Figura 2.10 Figura para la creación de plan de adquisiciones

Identificar los requerimientos de las adquisiciones

- Requerimientos

Listar y describir los productos a ser adquiridos de un área externa al proyecto. Asignarle a cada producto un ID, y definir la cantidad requerida y el costo presupuestado para cada uno.

Producto	Descripción	ID	Cantidad	Presupuesto
Computadoras (hardware)	Las especificaciones de las computadoras son :	CH1	X	\$
	Monitor 19"			
	Procesador Intel			
	Teclado y mouse			
	Windows XP			
	MS Office			
...	...	...	...	...

Tabla 2.21 Tabla con detalle de productos adquiridos

- Propósito

Se debe detallar el propósito de cada elemento definido en el punto anterior, identificándolo con el ID y el Producto.

- Justificación

Proveer una justificación razonable del porqué dichos productos son adquiridos mediante un proveedor externo. Tal como en el punto anterior, se identifica con el ID y el producto.

Emprender un estudio de mercado

Emprender un estudio de mercado es crítico para identificar los productos adecuados y que estos estén disponibles, en el lugar indicado y el momento requerido.

- Condiciones de mercado

Las condiciones de mercado tratan de establecer las métricas para optar por las actividades de vender y comprar productos y en algunos casos servicios que mejor convengan. Se analizan distintos tipos de vendedores que ofrecen diversos productos o servicios, con el fin de adquirir dichos bienes o servicios.

Al analizar el mercado se puede tener un panorama más claro y contestar preguntas como: ¿Cuales son las nuevas ondas tecnológicas que se ofrecen?, ¿será dicha tecnología una buena oferta que solvente los problemas?, ¿cuántos proveedores en este segmento del mercado se tienen para elegir?, ¿quien domina el mercado y por qué?

- Productos disponibles

Establecer un panorama de los proveedores disponibles y los productos que tienen en el mercado, para que los involucrados del proyecto conozcan las soluciones potenciales encontradas bajo los requerimientos anteriormente planteados. Consultar la tabla 2.22 sugerida para documentar dicho caso.

Producto	Proveedor	Descripción	Presupuesto
CH1	ABC LTD	Intel PC, incluyendo: Monitor 19'' Procesador Intel Teclado, mouse, quemador de DVD	\$/Otra
...	...	...	...

Tabla 2.22 Tabla con detalle de proveedores y productos

Construir la calendarización de las adquisiciones

- Calendario

Calendarizar las actividades requeridas para seleccionar el mejor proveedor, deliberando los productos requeridos.

- Asunciones

Listar cualquier asunción hecha durante el proceso de la planificación de las adquisiciones. Por ejemplo, no habrá cambios en el precio en el mercado del producto.

- Riesgos

Listar cualquier riesgo identificado durante el proceso de planificación de las adquisiciones. Por ejemplo, que los métodos utilizados para elegir el mejor proveedor no fueron los adecuados.

4.1.5.9 Contratar los proveedores

Después de haber creado una serie de planes detallados para el proyecto, su administrador estará listo para contratar los proveedores externos. Si bien el plan de adquisiciones identifica los productos que se adquieren de proveedores externos, no describe en el detalle el método a utilizar para la selección del mejor proveedor con el cual realizar un contrato. Esta sección se refiere a la selección y el nombramiento de uno o más proveedores para el proyecto.

El método habitual de la selección de un proveedor (o proveedores) es proceder a la gestión formal de licitación. Este es un método por el cual los posibles proveedores son identificados, evaluados y seleccionados para el suministro de productos (bienes o servicios) para el proyecto. Este proceso implica la creación de un conjunto de documentos de licitación que esbocen las necesidades que los encargados del proyecto tienen para la adquisición de información y requisitos necesarios de los proveedores sobre la cual se pueda tomar una decisión de a quien elegir. El proceso de licitación se lleva a cabo para garantizar que la selección de los proveedores se haga de forma justa y honesta.

Aunque varios de los diferentes tipos de procesos de gestión de ofertas existen, se sugiere un tipo de mecanismo para la gestión de ofertas: el llamado proceso de RFI / RFP.

Este proceso incluye:

- La emisión de una declaración de trabajo (SOW) que describe las necesidades del proyecto;
- La emisión de una solicitud de información (RFI) que se pide a los proveedores con información general de ellos;
- La emisión de una solicitud de propuestas (RFP), que pide propuestas detalladas por parte del proveedor;

- Recepción y evaluación de las propuestas de los proveedores;
- Elección del proveedor, basados en el mérito;
- La negociación de un contrato con el proveedor elegido.

El proceso se inicia inmediatamente después de que el plan de adquisición ha sido aprobado. Normalmente es realizado por el administrador de proyectos o encargado de las adquisiciones de proveedores externos. Sin embargo, para los grandes proyectos se crea un comité que se encargue de cada oferta y que su tarea sea supervisar el proceso de licitación.

El Proceso se termina sólo cuando los proveedores han sido seleccionados y el Proveedor ha firmado el contrato formal.

4.1.5.10 Continuar con la fase de revisión

Al finalizar con la fase de planificación, se debe realizar la de revisión. Es ésta se hace un alto para revisar y asegurarse que los logros del proyecto vayan enfocados a los objetivos planeados. Una vez completada la fase de revisión debe ser aprobada para proceder con la siguiente del proyecto (Fase de Ejecución).

Esta fase dará el status del proyecto en general, calendarización basado en el plan del mismo, sus gastos de acuerdo al plan financiero, recursos con bases en el plan de recursos, productos finales bajo los planes de calidad, riesgos y dificultades.

Esta fase debe ser completada por el administrador del proyecto y aprobada por el patrocinador. Para obtener la aprobación, el administrador del proyecto usualmente presenta un status del progreso del proyecto para ser considerado (mostrando todos los documentos generados durante dicha fase de planificación).

Para documentar esta fase se presenta el acta de aceptación de la fase de planificación del proyecto en la tabla 2.23.

ACTA DE ACEPTACIÓN DE LA FASE DE PLANIFICACIÓN DEL PROYECTO

DETALLES DEL PROYECTO

Nombre del proyecto:	Reporte preparado por:

Encargado del proyecto:	Fecha de preparación:

Patrocinador del proyecto:	Período del reporte:

Descripción del proyecto:

ESTATUS GLOBAL

Status Global:

Calendarización:

Gastos del proyecto:

Entregables:

Riesgos:

Dificultades:

Cambios:

DETALLES DE LA REVISION

Categoría	Pregunta	Respuesta	Variación
Calendario	Fue la fase cumplida en el tiempo y fecha establecida	S/N	
Gastos	Fue la fase cumplida con el presupuesto planteado	S/N	
Entregables	Plan del proyecto: ¿fue el plan del proyecto aprobado?	S/N	
	Plan de recursos: ¿Fue el plan de recursos aprobado?	S/N	
	Plan Financiero: ¿Fue el plan financiero aprobado?	S/N	
	Plan de calidad: ¿Fue el plan de calidad aprobado?	S/N	
	Plan de riesgos: ¿Fue el plan de riesgos aprobado?	S/N	
	Plan de aceptación: ¿Fue el plan de aceptación aprobado?	S/N	
	Plan de comunicación: ¿Fue el plan de comunicación aprobado?	S/N	
	Plan de adquisiciones: ¿Fue el plan de adquisiciones aprobado?	S/N	
	Contratar proveedores: ¿Fue la contratación de los proveedores aprobada?	S/N	
Riesgos	Existe algún riesgo pendiente de validar	S/N	
Dificultades	Existe alguna dificultad pendiente de validar	S/N	

Cambio	Existe algún cambio pendiente de validar	S/N

DETALLES DE APROBACION

Documentación de soporte: (Se hace referencia a documentos que sustenten los detalles mencionados en el documento)

Patrocinador del proyecto:

Firma: _____

Fecha: __/__/__

El proyecto ha sido aprobado para proceder a la fase de Ejecución.

Tabla 2.23 Tabla con formato para la aprobación de la revisión

4.1.6 Proceso de Ejecución del Proyecto

La fase de ejecución es la más larga del proyecto. Es en esta fase que se desarrollan todos los productos a presentar, para que él o los clientes lo aprueben. Para asegurar que los requerimientos del usuario se están cumpliendo, el administrador del proyecto monitorea y controla la producción de cada producto mediante algunos procedimientos.

Después de que los productos finales han sido desarrollados y aceptados se procede a la fase de revisión, lo cual determinará si se está listo o no para cerrar el proyecto.

El grupo de procesos de ejecución incluye la realización de ciertas actividades, como se mira en la figura 3.1.

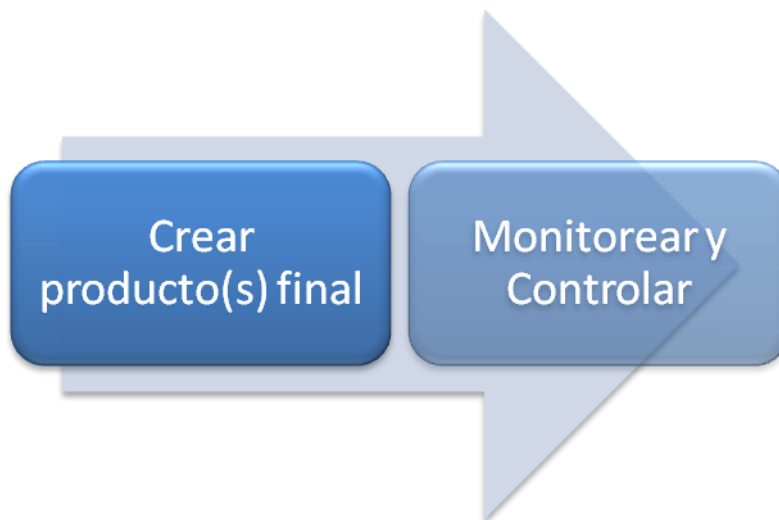


Figura 3.1 Actividades para la ejecución de un proyecto.

4.1.6.1 Crear producto(s) final

Esta etapa consiste en traducir en acciones las formulaciones realizadas en la etapa anterior, a fin de producir algo que satisfaga las necesidades planteadas por el usuario.

La actividad que requiere de más tiempo en el proyecto es la creación de los productos finales. Las etapas para el desarrollo del proyecto dependen exclusivamente de cada proyecto, ya que cada uno es distinto a otro y requieren tareas específicas.

Cada proyecto debe ser desarrollado en etapas sucesivas, hasta terminarlo, o mediante aceptaciones (no se avanza a una siguiente tarea o etapa hasta que no sea aprobada por el usuario, la tarea en cuestión).

4.1.6.2 Monitorear y controlar

El monitoreo facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo (personal, equipos, suministros, etc.) al proyecto. Si éste incluye subcontratación a otras organizaciones, el control del trabajo subcontratado se facilita en la medida en que esas partes encajen bien en la estructura de las fases. El control

de calidad también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).

Realizar el proceso de control del progreso del proyecto, actualizar los planes de proyecto y vigilar de cerca el equipo, son responsabilidades administrativas clave en esta etapa, así como ocuparse de cambios propuestos en el trabajo o en los objetivos del proyecto es, quizá, la tarea más desafiante de todas.

A medida que se implementa el proyecto, se debe controlar continuamente el progreso. Se examina lo que se hizo hasta ese momento, se repasa una vez más el plan o los planes y se observa si hay discrepancias importantes entre ambas cosas. Es común que existan variaciones, lo importante es determinar si las variaciones que se producen son aceptables o no, es decir, si están dentro de lo previsto.

Al igual que el control, el monitoreo o evaluación cumple una importante función de retroalimentación, es un examen objetivo y periódico para determinar el estado de un proyecto en relación con sus objetivos específicos, tienen lugar durante el proyecto y también al final. Al realizar una evaluación en medio del proyecto se pueden utilizar los resultados para modificar el rumbo del trabajo. La evaluación al final del proyecto sirve como ejercicio para verificar lo aprendido.

Las diferencias entre Control y Evaluación se presentan a continuación:

- El control implica una continua verificación de la marcha del proyecto, mientras que la evaluación solo significa la realización de exámenes periódicos.
- El control se concentra en los detalles de lo que está ocurriendo en el proyecto, mientras que la evaluación se ocupa del panorama general.

4.1.6.3 Administración del tiempo

Esta etapa incluye los procesos necesarios para lograr la conclusión del proyecto a tiempo. Se deben cubrir los siguientes aspectos:

1. Evaluación de la definición de las actividades

- Identificar las actividades específicas del cronograma
 - Planificar las actividades a monitorear y evaluar
 - Documentar el trabajo que se planifica realizar
2. Evaluación de la secuencia de las actividades
 - Identifica, documenta y controla la ejecución, las dependencias entre las actividades del cronograma
 - Deben estar ordenadas en forma lógica
 3. Evaluación de la estimación de recursos de las actividades
 - Determina cuales son los recursos definidos y qué cantidad de cada recurso es necesario para realizar cada actividad del cronograma
 4. Estimación de la duración de las actitudes

Es el proceso de velar por el cumplimiento de la estimación de la duración de las actividades del cronograma como:

- Alcance del trabajo de la actividad del cronograma
 - Tipos de recursos necesarios
 - Cantidades de recursos estimadas
 - Calendarios de recursos con su disponibilidad
5. Evaluación del Desarrollo del cronograma

Analiza:

- Las secuencias de las actividades
 - Duración de las actividades, fechas de inicio y finalización
 - Requisitos de los recursos
 - Restricciones del cronograma para crear el cronograma del proyecto
6. Control del cronograma

Implica:

- Determinar el estado actual del cronograma del proyecto
- Influir sobre los factores que crean cambios en el cronograma

- Determinar que el cronograma del proyecto ha cambiado
- Gestionar los cambios reales a medida que suceden

Ejecución de la Administración del tiempo

- ***Proceso de tiempo gerencial***

Este proceso es un método donde se administra el tiempo que se utiliza en las actividades obtenidas por el equipo durante el proyecto. La actualización del tiempo en marcha de cada proceso tiene muchos propósitos, entre los cuales se puede mencionar:

1. Calcula el tiempo total utilizado en cada actividad como un total de costos debido a cada actividad del proyecto.
2. Establece la disponibilidad del nivel de control de recursos por cada actividad utilizada por el gerente del proyecto.
3. Identifica el porcentaje de cada actividad completada como una sola cantidad de trabajo requerido.

El tiempo gerencial transcurre desde que se planifica en las hojas de tiempo de actividades hasta su finalización. Las hojas de tiempo de actividades son los documentos donde se almacena la ubicación del tiempo de todas las actividades propuestas en el plan de trabajo. Estas son completadas semanalmente, por todos los miembros del proyecto, y algunas veces por los distribuidores. Si las hojas de tiempo de actividades no son almacenadas, se tendrá una cantidad de tiempo por actividad no óptima y será imposible el manejo de ajuste del proyecto; así mismo, el costo y la calidad.

Después de ser iniciada la planeación del proyecto y completamente documentado, las hojas de tiempo de actividades podrán ser completadas en cualquier etapa del proyecto, únicamente si es requerido por el gerente del proyecto. Sin embargo, si es necesario el almacenamiento será a través de todo el proyecto para que se asegure que todos los costos estén incluidos en él.

- *Documentación de una hoja de tiempo de actividades.*

Este proceso se refiere a la captura de la información con el tiempo utilizado en referencia a cada actividad del proyecto. Este tiempo será almacenado y estará en compensación de todas las actividades de la fase de ejecución del proyecto. Desde este momento el tiempo para cada actividad será considerado y almacenado en una hoja de actividades. Esta tiene muchos formatos, pero para mantener una calidad de información, al capturar el tiempo en las actividades, es requerido mantener el almacenaje de todas las hojas por el grupo de personas que estén a cargo; en caso contrario, se debe esperar al finalizar los reportes antes de recolectar la información.

- *Aprobación de la hoja de tiempo de actividades.*

Para la inscripción de estas hojas tienen que ser hechas por cada miembro del proyecto y son presentadas al gerente del proyecto, este tiene que ser aprobado periódicamente o semanalmente se acuerde. A continuación existen tres puntos que pueden servir al realizar esta aprobación.

- 1- Confirmar que cada actividad fue validada y marcada en el plan del proyecto.
 - 2- Confirmar que todos los miembros del proyecto constituyeron un recurso para completar las actividades.
 - 3- Decidir si la ganancia del tiempo invertido es razonable.
- Registro de la hoja de tiempo de actividades.

El detalle de todas las hojas de tiempo de actividades se almacenan en un “registro de hojas de tiempo de actividades”. Estos se podrán utilizar por ejemplo:

1. El plan de proyecto debe ser actualizado con un resumen de tiempo de todas las actividades
2. El costo de cada miembro del proyecto es calculado y monitoreado a través del mismo

3. La identificación de tiempo extra en el Proyecto.

Estas partes están disponibles para el administrador de proyectos como una ayuda, y permite:

1. Producir una vista general diaria del proyecto.
2. Identificar las actividades que no han podido ser completadas por falta de tiempo
3. Identificar algunas excepciones, como por ejemplo, donde las actividades deben ser completadas utilizando más tiempo que se les ha sido asignado.

El administrador del proyecto debe disponer de una copia actualizada del plan de proyecto, así como el crecimiento de las áreas. El debe tomar las acciones requeridas basadas en:

1. Cantidad de recursos disponible para cada actividad
2. Disponibilidad de fondos para poder completar la actividad.
3. Requerimiento de asistencia externa para poder completar las actividades.
4. Disponibilidad de patrocinadores.

Asignación de puestos en la administración del tiempo

Estará constituida por asignaciones de puestos y responsabilidades, donde será distribuido el tiempo que será administrado durante todo el proyecto. Entre los cuales se describen:

- Miembros del proyecto

Son responsables de:

- Que cada empresa asignará las actividades según su mejor disponibilidad.
- Se requiere un nivel detallado de las hojas de tiempo de actividades.
- Proveer al director de proyecto más información con respecto al tiempo en que la empresa requiere para cada actividad, si es necesario.

- Gerente del proyecto.

El gerente de proyecto es el que tiene la responsabilidad general del tiempo en el proceso, esto incluye:

1. Delegar las actividades a los miembros del proyecto y proveerlos de recursos para que la empresa realice efectivamente sus actividades.
2. Garantizar que todo el personal esté informado del proceso gerencial, ellos deben entender cuándo deben ingresar las hojas de tiempo de actividades, así como a quiénes y cómo deben ser detalladas.
3. Repasar y aprobar de las hojas de tiempo de actividad del proyecto.
4. Crear planes de acciones orientadas desde un plan de desviaciones.
5. Resolver todas las incógnitas junto a los miembros del proyecto y el crecimiento de relación de tiempo en el proyecto.

- Administrador de proyectos.

El administrador del proyecto trabaja el día a día con los reportes de tiempo de actividad, debido a que:

1. Provee a todo el personal las hojas de tiempo de actividades básicas para que sean completadas.
2. Garantiza que todas las hojas de tiempo de actividades sean completadas a su tiempo y con todo su nivel de detalles.
3. Verifica que todas las hojas de tiempo de actividades estén revisadas y aprobadas por el gerente del proyecto.
4. Obtener información de los miembros del proyecto, si es requerida por el gerente de proyecto.
5. Almacena todos los detalles de las hojas de tiempo de actividades en los registros.

6. Concluye un registro general de información generado en la planeación e identificación del proyecto.

Documentos de administración del tiempo.

- Formato de hoja de tiempo de actividades.

Este formato es un documento que es completado por el personal y se almacena formalmente el tiempo invertido en las actividades o asignaciones del proyecto.

- Registro de tiempo de actividades.

Este registro es un documento central donde se almacena toda la información de las hojas de tiempo de actividades de un proyecto.

4.1.6.4 Administración de costos

En la gestión de costos del proyecto se deben considerar todos los procesos de planificación, estimación, preparación del presupuesto y control del presupuesto aprobado. En lo que se consideran tres puntos importantes:

1. Evaluación de la estimación de costo

- En este punto se desarrollará una aproximación de los costos de los recursos necesarios para completar las actividades del proyecto.
- El estimado debe considerar las posibles causas de variación de las estimaciones de costos, incluyendo los costos.
- Esta estimación incluye la identificación y consideración de diversas alternativas de costos
- Hay que considerar que la exactitud de la estimación de un proyecto aumenta a medida que avanza el proyecto a lo largo del ciclo de la vida del proyecto.

2. Revisión del presupuesto de costos

Se analiza:

- La suma de los costos estimados de actividades individuales, a fin de establecer una línea base de costos
- Como medir el rendimiento del proyecto.

3. Control de costos

Muestra una descripción de:

- La influencia sobre los factores que producen cambios en la línea base de costos.
- Asegurar que los cambios solicitados sean acordados
- Gestiona los cambios reales cuando y a medida se produzcan
- Evita que se incluyan cambios incorrectos, adecuados o no, aprobados en el costo o en el uso de los recursos.

En los procesos de estimación se ocupan principalmente de los costos de los recursos necesarios, para cual se debe contemplar los siguientes procesos para poder completar las actividades del cronograma. Las que se pueden mencionar

- *Proceso de costo gerencial*

Este es un método donde el costo se divide entre los gastos incurridos del proyecto que son formalmente identificados y aprobados. Por ejemplo:

1. Costos laborales.
2. Costos de los equipos.
3. Costos de materiales.
4. Costos de administración.

Su propósito es almacenar la de información actual de costos entre la acumulación de gastos durante el ciclo de vida del proyecto. La administración de costos de una empresa es a través de las formas de aprobación y finalización de los gastos. En su

forma de gastos es un documento que es completado por los miembros del equipo que requieren solventar los gastos que ya se han incurrido en el proyecto. Cada gasto debe especificar:

- Las actividades y tareas serán enumeradas en el plan del proyecto.
- La fecha cuando ocurrió los gastos.
- El tipo de gasto.
- Una descripción detallada de los gastos.
- Cantidad de gastos efectuados.
- Numero de factura relativa a los gastos, cuando aplique.

Para almacenar los procesos y procedimientos en la administración de costos se realizaran documentos de gastos, asignación de gastos y un registro de gastos durante la ejecución del proyecto.

Documentos de gastos

El primer paso es la captura de información relacionada con los gastos incurridos en el proyecto, así como las tareas y actividades del proyecto en la empresa. Sin embargo, es importante identificar las actividades y tareas relacionadas con cada gasto incurrido como un total de costos del proyecto de la empresa.

Los registros son regularmente completados por:

1. Personal del proyecto que han incurrido en los gastos.
2. Administradores del proyecto.
3. Contratistas asignados para los servicios previstos en el proyecto.

Asignación de gastos

Definido los documentos de los gastos en el proyecto, se envía al gerente del proyecto para su aprobación. El gerente del proyecto también debe considerar:

- Las actividades donde se ha incurrido un gasto válido, y serán listados en el plan de proyecto.
- Los gastos deben estar originalmente presupuestados, como se define en el plan financiero.
- Se deben tomar en cuenta los gastos sin presupuesto, son entendibles.

El gerente del proyecto puede tener la autoridad de autorizar el presupuesto que no esté considerado. Los gastos que no estén en el presupuesto y que estén fuera del límite deberán ser aprobados por el patrocinador.

El gerente del proyecto debe estar sabedor de:

- Los gastos aprobados y la orientación de los pagos por la administración del proyecto.
- Solicitud de información de la persona a quien se le asignara los gastos.
- Disminución de los gastos.

La aprobación formal de los gastos será ejecutada por el gerente del proyecto y los pagos deben ser programados.

Registro de gastos

Después de que cada pago ha sido programado, el registro de gastos debe actualizarse con calidad de información de pagos y aprobaciones que se han efectuado. También estos registros deben ser actualizados después de los gastos que han sido aprobados, estos se actualizan a través del proceso de asegurar que el gerente del proyecto este informado del progreso de los gastos en todas las fases del ciclo de aprobación.

El registro de gastos almacena todos los detalles de todos los gastos efectuados, permitiendo así:

1. Que el plan de proyecto se actualice mediante el almacenamiento de los gastos por cada actividad.
2. Que el costo de cada miembro del proyecto se calcule y monitoree durante todo el proyecto.
3. Que el director de proyecto identifique el costo actual contra el presupuestado durante el proyecto.

En una base común, el administrador de proyecto actualiza el plan de éste con el total de costos por cada actividad y se detallan en el registro de gastos. El administrador de proyecto suministra una copia actualizada del plan de proyectos al gerente de proyecto, así mismo identifica los costos hasta la fecha.

Asignación de puestos en la administración de costos.

Estará constituida por asignaciones de puestos y responsabilidades, donde se relacionara con la gerencia de costos entre los gastos durante el proyecto. Entre los cuales se describen:

- ***Miembros del proyecto.***

Su responsabilidad es:

1. Están bajo el dominio de delegar cada actividad con la mejor disponibilidad de ésta.
2. Completar los gastos generales con todos los detalles incluidos.
3. Incorporar las formas de gastos a la aprobación del gerente del proyecto.
4. Distribuir al gerente del proyecto más información respecto a los gastos, si es necesario.

- **Administrador de proyecto.**

El administrador del proyecto está encargado del proceso de la administración de costos, lo que incluye:

1. Proveer a todos los miembros los formularios de gastos para que sean completados.
2. Asegurar que los miembros del proyecto cumplan y completen los formularios a tiempo.
3. Dirigir todos los formularios de gastos al gerente del proyecto para su aprobación.
4. Asegurar que todo registro de gasto incluya información de calidad y actualizada
5. Disponer de información de gastos incorporados a la planeación e identificación del proyecto.
6. Mantener acuerdos de pagos para cada gasto que esté aprobado.

- **Gerente del proyecto.**

El gerente del proyecto es responsable de todos los procesos donde se deben administrar los costos, incluyendo:

1. Asegurar que todos los miembros estén informados de todos los procesos de costos; ellos deben entender, cuando las formas de gastos se deben inscribir, a quién será dirigido y cuál será el nivel de detalle;
2. Ejecutar el repaso y aprobación de todos los formularios de gastos;
3. Resolver todos los inconvenientes de gastos o todo lo relacionado con ello con los miembros del proyecto durante el proyecto.
4. Desarrollar el monitoreo de aseguramiento de la calidad.

4.1.6.5 Administración de la calidad

Este proceso nos refiere a que incluye todas las actividades de la organización ejecutante que determinan los objetivos y las responsabilidades relativas a la calidad

de modo que el proyecto satisfaga las necesidades por las cuales se emprendió. Este comprende:

1. Planificación de calidad.
 - Identifica normas de calidad que son relevantes para el proyecto y para determinar cómo satisfacerlas
2. Aseguramiento de calidad.
 - Aplicación de las actividades planificadas y sistemáticas para asegurar que el proyecto utilice todos los procesos necesarios.
3. Realizar control de calidad
 - En esta etapa supervisa los resultados específicos del proyecto, para determinar si cumplen las normas de calidad relevantes e identificar modos de eliminar las causas de un rendimiento insatisfactorio.

- *Proceso de calidad gerencial*

Este es un método en que cada producto de calidad y su proceso serán verificados y controlados durante el proyecto. Relaciona a la empresa en una variedad de evaluaciones y procesos de calidad del proyecto. Para ser más específico, en este proceso se debe incluir:

1. Los objetivos que serán considerados en el plan de calidad.
2. Los tipos de técnicas de medición que serán desarrollados o establecidos en la empresa.
3. La implementación de aseguramiento y control de técnicas de calidad.
4. La toma de acción en los niveles de mejora y proceso de calidad.
5. Reportar el nivel de calidad alcanzado.

Este proceso de aseguramiento de calidad en la empresa durara solo en la fase de ejecución del proyecto. Sin un adecuado proceso de aseguramiento de calidad los objetivos de tiempo, costo y calidad no podrán ser alcanzados. Este proceso se considera finalizado siempre y cuando todos los productos y la administración de estos estén completados.

Medición de la calidad alcanzada.

Cuando ya estamos claros de cuáles son los objetivos de calidad a alcanzar, es tiempo de ejecutar las técnicas de aseguramiento y control de calidad para asegurar el nivel de calidad de cada producto realizado.

- Realizar una administración de calidad.

El uso de métodos de aseguramiento de la calidad es afianzar el nivel de calidad de productos y sus procesos. Las técnicas son utilizadas a menudo como un nivel de resumen del proyecto, suministrado por un recurso externo del proyecto.

Algunos ejemplos de estas técnicas, son:

1. Informar al equipo de proyectos, cuales son los objetivos definidos.
2. Utilizar los procedimientos de cambios de control que minimizan el número potencial en gestiones de calidad.
3. Revisar la calidad de la empresa para mantener la confianza de que el proyecto va en buen camino.

Control de calidad de la empresa.

Generalmente la empresa detalla los niveles del proyecto mediante un recurso interno. Ejemplo de algunos métodos de control de calidad:

1. Revisiones partidarias.
2. Revisiones de procesos.
3. Revisiones de fases.
4. Revisiones de entregas.
5. Revisiones de documentos.

Asignaciones de puestos en la gestión de calidad

Las asignaciones y responsabilidades se mostrarán en los niveles de aseguramiento y control de calidad durante el proyecto. Entre los cuales se describen:

- Gerente.

El gerente se asegura que el proyecto produzca un conjunto de productos que incluye un nivel de calidad en mutuo acuerdo con el cliente. El gerente es responsable de:

1. Asegurar el entendimiento de los objetivos de calidad definidos para cada producto.
2. Implementar los métodos de aseguramiento de la calidad para asegurar que el producto del proyecto sea de calidad.
3. Implementar las técnicas de aseguramiento de la calidad de los productos que estén ya elaborados en el proyecto.
4. Almacenar el nivel de calidad obtenido;
5. Identificar las desviaciones de calidad y las nuevas acciones de implementación;
6. Reportar el nivel de calidad al gerente del proyecto.

Documentos en la gestión de calidad.

- Formato de revisión de calidad.

Este formato es de la empresa que provee al cliente la exactitud de la calidad de los productos y el manejo de los procesos que son aceptables. Son usados dos tipos genéricos de revisión para los procesos de entrega y revisión en los proyectos.

- Formato de producto de entrega.

Este formato es utilizado para alcanzar el nivel de calidad de los productos que han sido creados para el cliente. Para lograr mejores resultados, la empresa entregará revisiones durante la creación de cada producto, así como en la fase de construcción de entrega.

- Formato del proceso de revisión.

Este es usado para evaluar el nivel de calidad de los procesos gerenciales de la empresa durante la trayectoria del proyecto. Estos procesos de revisión determinan el nivel de conformidad actual del proceso de la empresa en las cuales han requerido un plan de calidad.

4.1.6.6 Administración de los cambios

El proceso de administración de cambios permite que los encargados del proyecto puedan llevar documentado cualquier cambio que se haya dado durante la ejecución del proyecto, de manera que se pueda constatar si existiese algún inconveniente.

Proceso de gestión de cambio.

Este proceso es un método en el cual los cambios van dirigidos al alcance del proyecto, productos, escalas de tiempo o recursos deben ser identificados, evaluados y aprobados antes de su implementación. Este proceso se complementa con una variedad de procedimientos de control para asegurar que sí se ha implementado.

Este proceso dura la fase de ejecución del proyecto, formalmente definido y planificado. En teoría, cualquier cambio en ese lapso necesitará ser manejado como parte del proceso. Sin un cambio formal en este, la disponibilidad de dirigir un proyecto será efectiva al realizar su alcance como se ha propuesto. El proceso de cambio se dará por finalizado cuando la fase de ejecución este completa.

Envío de requerimiento de cambio.

Para iniciar un proceso de cambio, se debe permitir que cada miembro del equipo del proyecto envíe un requerimiento. La persona encargada de su alcance es llamada “solicitante de cambios”. Este debe documentar los requisitos de cambios del proyecto, completando el formulario de requerimiento, así como los gastos, descripción de los cambios, el impacto y las aprobaciones requeridas.

Aprobación del requerimiento de cambios.

Es una revisión formal del formato de requerimiento de cambio que está a cargo de la aprobación del grupo. La decisión de ellos será basada en el nivel de riesgo y el impacto de los resultados del proyecto.

Implementación del requerimiento de cambios.

Aprobados los cambios, luego son implementados y se debe relacionar así:

1. Identificar de la fecha de implementación de los cambios.
2. Implementar el cambio.
3. Revisión y seguimiento.
4. Almacenamiento de todas las actividades en el registro de cambio.
 - Cambio de asignaciones.

Las asignaciones y responsabilidades serán involucradas en la gerencia de cambios.

- Solicitante de cambios.

Este inicialmente reconoce la necesidad de cambios en el proyecto y formalmente comunica estos requisitos al gerente de cambios. El solicitante será responsable:

1. –De identificar la necesidad para poder realizar el cambio en el proyecto.

2. -De documentar la necesidad de cambios completados con el formato de solicitud de cambios
3. -De enviar el formato al gerente para su revisión.

- Administrador de cambio.

Este administrador recibe, monitorea y controla el progreso de todos los cambios durante el proyecto. Su responsabilidad es de:

1. Recibir todos los formularios y almacenarlos en el registro de cambio.
 2. Categorizar y priorizar todas las solicitudes de cambios.
 3. Revisión de todos los formularios para determinar cualquier información si es requerida.
 4. Determinar cualquier factibilidad de estudio de cambio si es requerido.
 5. Reportar y comunicar todas las decisiones hechas para la aprobación del grupo.
- Aprobación del cambio de grupo.

Esta es la autoridad principal de todos los formularios de cambios enviados por el gerente. La aprobación grupal es responsable de:

1. -Revisión de todos los formularios enviados por el gerente.
 2. -Consideración de toda la documentación de soporte.
 3. -Aprobación o desacuerdo de cada formulario basado en sus propios meritos.
 4. -Resolver los conflictos de cambios, donde dos o más cambios existan.
 5. -Identificación de los cambios aprobados en la fase de implementación.
- Implementación del grupo de cambio.

En este grupo se calendariza y se implementan todos los cambios. Esta implementación es responsable de:

1. Calendarizar todos los cambios por periodos, distribuidos por el grupo de cambios aprobados.
 2. Evaluación de todos los cambios, antes de la implementación.
 3. Implementar todos los cambios durante el proyecto.
 4. Revisión de todos los alcances de cada cambio, luego se implementan.
 5. Solicitar al gerente de cambio un cierre en cada registro de cambio.
- Registro de cambio.

Es un archivo donde están todas las solicitudes de cambios ya registradas y revisadas hasta su implementación.

4.1.6.7 Administración de riesgos.

Proceso de administración de riesgos.

Es un método en el cual los riesgos del proyecto son formalmente identificados, cuantificados y manejados durante la ejecución del proyecto. El proceso se complementa con un número de acciones que reduce las actividades parecidas y el impacto de riesgo de cada una. El proceso de riesgo es usado para asegurar que cada riesgo este formalmente identificado, cuantificado, monitoreado y transferido en el proyecto.

También el proceso de riesgo en la empresa se da durante la fase de ejecución del proyecto, los riesgos deben ser identificados en cualquier parte del ciclo de vida del proyecto. En teoría, cualquier riesgo identificado durante la vida del proyecto deberá ser formalmente manejado en el proceso de administración de riesgos. Sin un adecuado proceso de administración de riesgos, no se podrían identificar riesgos de mayor impacto en el proyecto para poder alcanzar sus objetivos. El proceso de administración de riesgos se es finalizado solo cuando la fase de ejecución del proyecto esté terminada.

- Aumento de riesgo.

Al iniciar el proceso de administración de riesgos, se debe permitir que todos los miembros del equipo del proyecto participen en el proyecto de riesgo. Esta persona identifica un riesgo de aspecto común (ya sean en los alcances, productos, horarios de actividades o recursos) y se completa con la descripción natural con que se ha identificado ese riesgo.

- Registro de riesgos.

El administrador o gerente del proyecto revisa todos los riesgos próximos y determina cual o no es aplicable como tal en el proyecto. Esta decisión será basada siempre por los impactos de riesgos, tales como:

1. -Productos especificados en el registro de calidad
2. -Objetivos de calidad especificados en el plan de calidad.
3. -Periodos especificados en la fase de planeación.
4. -Objetivos de Recursos especificados en la fase de recursos.
5. -Objetivos financieros especificados en el plan de finanzas

Si el riesgo es considerado como tal por el gerente del proyecto, se debe documentar este riesgo como registro y asignarle un número de registro. El gerente del proyecto deberá asignar el nivel de impacto y la similitud de riesgo basado en la gravedad de riesgos pasados.

- Asignación de acciones de riesgos.

La revisión de grupo del proyecto se completara con cada riesgo se haya mencionado en el registro de riesgos y se decida el impacto y su similitud, por lo tanto:

1. -Se asignaran acciones de riesgos para minimizar su impacto.
2. -Se puede incrementar la solicitud de cambios, solo si es requerido para minimizar el riesgo.
3. Se puede cerrar el riesgo en su propio registro, solo si no es satisfactorio las acciones tomadas o no tiene un impacto duradero en el proyecto.

- Implementación de acciones de riesgos.

Para minimizar la toma de acciones por los riesgos en el proyecto se deberá tomar en cuenta lo siguiente:

1. -Elaborar una agenda con cada acción a implementar.
2. -Implementar cada acción programada.
3. -Revisar cada acción implementada con éxito.

Asignaciones de la administración de riesgos.

Se refiere a las asignaciones y responsabilidades que integran la gestión del riesgo durante el proyecto.

- Iniciador de riesgos.

Esta persona identifica los riesgos del proyecto, complementa los formularios y envía esta forma al gerente del proyecto para su revisión.

- Gerente de proyecto.

El gerente recibe cada formulario de riesgo y monitorea el progreso de todos los riesgos durante el proyecto. El es responsable de:

1. Recibir todos los formularios de riesgos y identificar si es un riesgo afectara el proyecto.
2. Almacenar todos los riesgos en un registro y monitorear posteriormente el nivel de riesgo.
3. Presentar todos los riesgos al grupo de revisión del proyecto.
4. Comunicar de todos las decisiones hechas por el grupo de revisión del proyecto.
5. Monitorear el progreso de las acciones asignadas para minimizar los riesgos.

Grupo de revisión del proyecto.

Este grupo es el encargado de evaluar la similitud de los riesgos y su impacto, luego se le asignara las acciones apropiadas para minimizar estos. Este será responsable también de:

1. Revisar regularmente el registro de riesgos.
2. Identificar las solicitudes de cambios requeridos para minimizar el aumento de riesgos.
3. Asignar acciones para minimizar los riesgos.
4. Finalizar algunos riesgos que no se consideran de gran impacto en el proyecto.

Equipo del proyecto.

Este equipo delega todas las acciones para minimizar los riesgos, estos serán supervisados por el grupo de revisión del proyecto.

Documentación de riesgos.

Existen dos tipos de documentos.

1. Formularios de riesgos: son completados por los miembros del proyecto al iniciar un nuevo proyecto.
2. Registro de riesgos: es un archivo en el cual todos los riesgos se almacenan y se orientan al cierre del proyecto.

Administración de proveedores.

Este proceso es un método en el cual los productos son adquiridos por fuentes externas del proyecto. Esto incluye el control de ordenar, recibir, revisar y aprobar la adquisición de los productos que se estén administrando. El proceso de proveedores es usado para asegurar que todos los productos adquiridos sean en el tiempo correcto, su nivel de calidad definida y su presupuesto de costo sea identificado. Este

proceso también precisa de procedimientos que aseguran la relación entre los administradores y sus colaboradores.

Se inicia este proceso después de haber aprobado un plan y firmado un contrato con los proveedores. Este proceso es usualmente manejado por un simple recurso durante el proyecto y monitoreado por el gerente del proyecto.

Si no existe un proceso formal, será dificultoso confirmar la entrega de los productos a los proveedores. Así como el manejo de este, aumentando aun más el nivel de riesgo en el proyecto. Se finalizara este proceso siempre y cuando la fase de ejecución este completada.

- Orden de compra.

En este proceso se inicia con la compra del producto al proveedor mediante una orden de compra. Por lo que se requiere tener en cuenta cuatro puntos importantes:

1. Identificar el producto: se debe identificar los productos que serán comprados y un plan de proveedores. Deberá existir periodos de entrega y verificar el nivel de calidad de estos.
2. Documento de orden de compra: el gerente de proveedores debe completar esta orden, en la cual debe especificar los productos requeridos, sus componentes, sus periodos de entrega y su centro de costo donde será cargado la compra. Existirá una lista de varios productos para un solo proveedor. Esto es importante que la orden de compra esta bien completada antes de la fecha de entrega para que el distribuidor este notificado cualquier cambio que ocurra.
3. Aprobación de la orden de compra: Esta orden se envía al gerente del proyecto para su aprobación. El gerente debe considerar su propio criterio para la aprobación de esta compra.
4. Periodos de entrega: El distribuidor debe confirmar los recibos de orden de compra y su tiempo de estimado de llegada de los productos. Estos tiempos

estimados de llegada se describen como los periodos de entrega de un producto en un lugar asignado en el proyecto.

- Aceptación de Orden de compra.

Esto se relaciona con el cumplimiento de la orden de compra, ya que después de ser procesada por el proveedor y que su producto debe ser entregado como se esperaba.

En la administración de proveedores se debe revisar, identificar y resolver cualquier inconveniente con los productos a fin que ellos se puedan aceptar y así suministrar los pagos autorizados.

- Aceptación del producto.

Una vez que todos los inconvenientes de los productos han sido resueltos, el gerente del proyecto aceptara en el nombre del equipo del proyecto el producto y se le notificara a su proveedor.

- Aprobación de pago.

El proveedor debe de enviar al equipo de proyecto una factura en el cual se refleje el pago autorizado por el gerente del proyecto.

- Administración del contrato del proveedor.

Mientras los procedimientos se llevan a cabo por cada orden de compra en el proyecto. Existe un proceso paralelo de compra, con el que refleja un rendimiento general de todos los proveedores que determinan si es necesario continuar con los compromisos ya obtenidos en el contrato.

Rendimiento de los proveedores.

Este rendimiento se medirá según:

1. Identificación de los productos distribuidos por los proveedores.
 2. Confirmación de cada una de las condiciones mediante el contrato del proveedor.
- Identificación de inconvenientes.

Algunos inconvenientes son identificados durante la revisión del proyecto y el impacto se podrá cuantificar. Estos se documentaran y formalmente aumentaran las soluciones proporcionadas por el gerente del proyecto.

- Resolver los inconvenientes.

El gerente del proyecto supervisara cada inconveniente con la ayuda del gerente de proveedores para poder negociar una mejor solución.

- Asignaciones de proveedores.

En esta etapa se asignan las tareas y responsabilidades con el personal vinculado al proyecto de proveedores.

- Gerente de los proveedores.

Es responsable de asegurar que todo el proceso con los proveedores se realice en forma efectiva, pero se lograra mediante:

- La identificación de los productos como un recurso externo.
- Documentar cada orden de compra y su futura aprobación por el gerente del proyecto.
- Comunicación de periodos de entregas.
- Aceptación del producto y aprobación de la solicitud de pago.
- Identificar y resolver cualquier inconveniente de entrega de un producto.

Documentación de los proveedores.

Existen dos tipos documentos:

1. Formato de orden de compra,
2. Registro de proveedores.

4.1.6.8 Gestión de aceptación.

Se definirá como un método donde los productos ya producidos son revisados y aceptados por el cliente. Este proceso implica una variedad de técnicas para poder aceptar los productos bajo un criterio de aceptación única. El proceso de aceptación se enfocara hacia la finalización de la fase de ejecución del proyecto.

Sin un adecuado proceso de aceptación, el cliente no podrá aceptar los productos que se han producido en el proyecto. El proceso de aceptación se dará por finalizado hasta que la fase de ejecución este completada.

Antes de su petición formal de aceptación de productos por el cliente. Estos deben completar el nivel de calidad como una ganancia a la aceptación del cliente. Esto se refiere a:

- Requerir que todas las tareas de la empresa estén finalizada para poder entregar sus productos.
- Documentar los componentes entregables.
- Informar al gerente del proyecto que los productos están listos para la aceptación del cliente.

Evaluación de aceptación.

El gerente del proyecto organiza una evaluación para el cliente para determinar si o no se especifica un criterio en el plan de aceptación. Esto se refiere a:

1. Comparación de resultados con los criterios originales de aceptación.
 2. Determinar si o no será un criterio a seguir.
 3. Inicializar con el trabajo necesario para una mejor entrega del producto, si es necesario.
 4. Completar un formato de aceptación para la aprobación del producto.
- Aceptación del producto.

El producto finalmente ha sido aceptado por el cliente, lo que conlleva a:

1. La obtención del formulario de aceptación por el cliente.
2. Repasar el formulario de aceptación para asegurar de tener un mismo criterio.
3. Y por último, transferir el producto a un ambiente adecuado para su aceptación.

En referencia a las tareas y asignaciones se vinculara con la aceptación de los productos por los clientes durante el proyecto.

- Gerente de proyecto.

El gerente del proyecto estará encargado de supervisar en el proceso de aceptación de todos los productos en el proyecto. Sera responsable de:

1. Recibir una notificación del equipo del proyecto de que esta completa y lista la aceptación del cliente.
2. Organizar las entregas de los productos a los clientes, programar y documentar los resultados.
3. Completar el formulario de aceptación por el cliente.
4. Administrar las tareas que se requiera en el proyecto para que los productos se desarrollen en un ambiente de ventas.

- Cliente.

El tiene la autoridad final de aceptación. El cliente es responsable de:

1. Revisión de la aprobación (donde sea posible).
2. Autorización del formato de aprobación el cual estará en espera de la confirmación final de que el producto fue especificado en el plan de aceptación.
3. El producto se desarrollara en un ambiente de negocio.

Documentos de la gestión de aceptación.

Esto ayudara a la empresa en la parte de gerencia de aceptación, como un ejemplo del formato de aceptación y de registro de aceptación que ellos suministran.

- Formato de aceptación.

Este formato será completado mediante una petición formal de aceptación de un producto del proyecto elaborado por el cliente.

- Registro de aceptación.

Este se refiere a un archivo en el cual la solicitud es registrada y tiene su seguimiento para su aprobación.

4.1.6.9 Gerencia de comunicación.

Proceso de la gerencia de comunicación.

Se refiere a un método en el cual los mensajes son identificados, creados, revisados y comunicados durante todo el proyecto. El éxito de cualquier proyecto ocurrirá en la calidad de comunicación en un tiempo crítico. Así asegurara a los involucrados que tengan una información correcta, en el tiempo correcto, en el cual se podrán tomar decisiones correctas.

Los métodos de comunicación en la empresa siempre deben se debe considerar:

1. Proporcionar el contenido del mensaje, su público, el tiempo y formato.
2. Crear el mensaje que se enviara.
3. Revisar un mensaje previo para su distribución.
4. Comunicar el mensaje a sus destinatarios.

Estos cuatro procesos deben ser aplicados en cualquier tipo de comunicación formal en el proyecto, incluyendo la distribución de:

1. Reportes regulares de su nivel en el proyecto.
2. Resultados de la fase de revisión.
3. Calidad de los reportes revisados.
4. Actas y otros documentos de información general.
5. Duración de las reuniones del equipo del proyecto.

Los procesos de comunicación de la empresa deben ser documentados, la duración de la comunicación debe durar en todas las fases del proyecto. Este proceso se debe aplicar en comunicaciones formales de la empresa durante el ciclo de vida del proyecto. Sin una buena gerencia de comunicación empresarial dificultara a los involucrados recibir información correcta en el tiempo correcto.

- Identificación del mensaje.

Esta administración iniciara con el contenido del mensaje, su público, el tiempo y el formulario de aprobación y distribución de un proyecto.

- Identificación del contenido del mensaje.

Primero, se necesita una comunicación identificada. En cualquier proyecto se puede identificar una necesidad de comunicación. Cualquier gerente deberá estar de acuerdo cual su necesidad real y quiénes son los interesados para ser distribuidos el mensaje.

El contenido del mensaje deberá ser documentado. El contenido debe incluir:

1. Un nivel de actividad o de información financiera.
 2. Información de sus productos.
 3. Disponibilidad de proveedores.
 4. Riesgos, asignaciones y cambios identificados.
- Identificación del público.

Este público será identificado, algunos ejemplos serán:

1. El equipo de administración de proyecto.
 2. El grupo de revisión.
 3. El equipo del proyecto.
 4. Todos los involucrados del proyecto.
- Identificar su duración.

En este momento, el tipo de contenido será distribuido y su público será específicamente claro. El próximo paso será identificar el periodo de frecuencia del mensaje. En algunos casos será necesario tener una comunicación más continua, como reportar semanal o mensualmente un informe, en cualquiera de los casos debe ser satisfactoria la entrega.

- Creación de un mensaje.

Como primer filtro del contenido del mensaje será el equipo de comunicaciones. El gerente revisará el contenido y enviará los cambios a su equipo a su aprobación y envío.

- Revisión del mensaje.

Dependiendo del tamaño del proyecto y tipo de mensaje que se distribuirá el mensaje, el gerente del proyecto debe aprobar el mensaje para su distribución. Se puede mencionar algunos tipos de mensajes en que el gerente del proyecto generalmente aprueba, lo que incluirá:

1. El nivel del proyecto.
2. Sus riesgos y consecuencias.
3. Solicitud de cambios.
4. Nuevos puntos a tratar.

Ciertos tipos de mensajes requieren una aprobación del grupo antes de su distribución, tales como:

1. Cambios en el alcance del proyecto, objetivos y cronogramas.
2. Riesgos o consecuencias identificadas.
3. Demoras o tiempos innecesarios del proyecto.
4. Cierre prematuro del proyecto.

- Mensaje de comunicación.

Este mensaje es aprobado y distribuido al público involucrado, usando los métodos de comunicación aprobados. Después de distribuidos, el registro de comunicación será actualizado con los últimos eventos.

Asignación de puestos en la gestión de comunicación.

Las asignaciones y responsabilidades serán relacionadas en el proyecto de comunicaciones de la empresa, a los que se pueden mencionar:

- Equipo de comunicación.

Este edifica cada mensaje de comunicación y realiza alternativas requeridas por el gerente de comunicación. Ellos son responsables de:

1. Documentación formal de cada mensaje.
2. Realizar cambios a cada mensaje si son requeridos.
3. Publicación del mensaje final.
4. Distribución del mensaje publicado a un público específico.

- Administrador de comunicación.

Este rol permite comunicar a todo el equipo mediante la revisión de todos los mensajes de comunicación previamente aprobados, los cuales incluye:

1. Asisten al administrador del proyecto con la identificación del contenido, público, periodos y formularios de todos los requerimientos para la comunicación del proyecto.
2. Revisión de los elementos de la comunicación y recomendaciones para futuros cambios.
3. Envío del borrador final al administrador del proyecto para su aprobación.

- Administrador del proyecto.

El administrador de proyecto tiene una sola responsabilidad de asegurar la calidad de envío y el periodo de tiempo de los mensajes durante el proyecto. Esto incluye:

1. Un plan de comunicación de documentación en el inicio del proyecto.
2. Identificar el contenido, la audiencia, los periodos y los formularios de comunicación.
3. Aprobar u obtener la aprobación de los mensajes de comunicación.
4. Retroalimentación de los mensajes enviados.

Documentos de comunicación.

- Reporte de Status del proyecto

El reporte de status del proyecto es usado por los administradores del proyecto para comunicar formalmente el status del proyecto. Provee una vista del progreso del proyecto, y asegurar que los involucrados del proyecto estén informados, dificultades y riesgos inherentes al proyecto también deben ser agregados en este proyecto. El administrador del proyecto debería reportar semanalmente al encargado del proyecto o gerente del proyecto y mensualmente al comité de la empresa.

- Registro de la comunicación

Es la bitácora donde todos los mensajes de comunicación son capturados para el proyecto.

4.1.6.10 Gestión de la Integración del Proyecto.

La importancia de la integración será concentrar recursos y esfuerzos en el proyecto. En esta integración se deben tener en cuenta las actividades siguientes:

1. Analizar y comprender el alcance del proyecto.
2. Documentar los requisitos del mismo.
3. Medir y supervisar sus procesos del proyecto.
4. Analizar sus riesgos.

La integración del proyecto se relaciona, principalmente, con la integración efectiva de los procesos entre los grupos de dirección que son necesarios para lograr los objetivos dentro de los procedimientos definidos de una organización.

En todo proceso de integración en la elaboración de un proyecto se deben incluir los siguientes pasos:

1. Desarrollar el acta de Constitución del proyecto.
 - Autoriza formalmente un proyecto o fase de este
 - Se relaciona principalmente con la documentación de necesidades de negocio, la justificación del proyecto, la comprensión efectiva de los requisitos del cliente.

2. Desarrollar el enunciado del Alcance del proyecto Preliminar
Aborda y documenta las características y los límites del proyecto

3. Desarrollar el plan de gestión del proyecto

Documentar las acciones necesarias para

- Definir
- Preparar
- Integrar
- Coordinar el área de aplicación y complejidad del proyecto

El plan de gestión del proyecto define los diferentes aspectos.

- Como se ejecuta
- Como se supervisa y controla
- Y como se cierra el proyecto

4. Supervisar y controlar el trabajo del proyecto

La supervisión y control son requeridos para iniciar, planificar, ejecutar y cerrar un proyecto, a fin de cumplir con los objetivos de rendimiento definidos en el plan de gestión del proyecto.

5. Control integrado de cambio. Implica:
 - Revisar todas las solicitudes

- Aprobar los cambios
- Controlar los cambios

6. Cerrar proyecto

Cierra la parte del alcance del proyecto y las actividades relacionadas aplicables a una fase determinada

4.1.6.11 Gestión del alcance del proyecto

Esta parte se relaciona principalmente con la definición y el control de lo que está y no está incluido en el proyecto.

Cabe mencionar las siguientes áreas de conocimiento para obtener una descripción general de los procesos de gestión del alcance del proyecto.

1. Planificación del alcance

Esta planificación del proyecto debe reflejar diferentes aspectos, así:

- ¿Cómo la definirá?
- ¿Cómo verificara y controlara el alcance del proyecto?
- ¿Cómo creara y definirá la Estructura de Desglose del Trabajo (EDT)?

2. Definición del alcance

Puntos a tomar en cuenta:

- Se desarrollará el enunciado sobre el alcance del proyecto
- Las necesidades, deseos y expectativas de los interesados se analizan y se convierten en requisitos.

3. Crear Estructura del Desglose del Trabajo(EDT)

Se tendrán en cuenta las siguientes características:

1. Estará orientada al producto a entregar
2. Deberá ser ejecutado por el equipo del proyecto
3. Estará enfocada hacia el logro de los objetivos del proyectos

Otro aspecto a considerar será que la EDT es la que organiza y define el alcance total del proyecto.

4. Verificación del alcance

En esta fase debe establecerse y documentar el nivel y el alcance de lo contemplado mediante los siguientes aspectos:

- Cumplir los requisitos de calidad
- Supervisar el control del alcance
- Controlar los cambios en el alcance del proyecto
- Asegurar que todos los cambios solicitados y las acciones correctiva recomendadas se procesen a través del **Control Integrado de Cambio de Proyectos**

4.1.7 Proceso de Cierre del Proyecto

Al finalizar la fase de ejecución, todos los productos finales habrán sido desarrollados y aceptados por el usuario. Es aquí donde se puede realizar una fase de revisión formal para determinar si el proyecto ha logrado los objetivos planteados en un inicio, así como los beneficios que el mismo trae a la empresa.

Es común que muchos proyectos nunca alcancen esta fase, porque una vez terminados y aceptados los productos finales, simplemente se pasa a un nuevo proyecto; sin embargo, en la presente guía se sugieren un grupo de procesos de cierre, los cuales incluyen la realización de ciertas actividades, como se observa en la figura 4.1.



Figura 4.1. Etapas de cierre de un proyecto

La realización de estas dos actividades, permitirán detectar las fallas ocurridas durante el proyecto y las soluciones planteadas, de manera que es un aprendizaje obtenido para que los futuros sean mejor completados.

Una vez finalizado con las fases planteadas, hay que desarrollar diversas tareas finales que van a depender de la naturaleza del proyecto. Por ejemplo, si se utilizó equipamiento es preciso tenerlo en cuenta y, si es posible, destinarlo a nuevos usos, asignar nuevas tareas a los miembros del equipo de proyecto o el mantenimiento del mismo. Esta es una actividad permanente y de duración infinita que se encuentra fuera del ciclo de vida de los proyectos.

4.1.7.1 Realizar el cierre del proyecto

Para iniciarlo se debe crear un reporte de cierre del proyecto. Este lista todas las actividades e identifica los recursos responsables para dicha actividad.

Documentar el cierre del proyecto

El reporte de cierre del proyecto es un documento que formaliza dicha actividad, provee la confirmación que la aceptación del usuario ha sido alcanzada y las solicitudes cerradas. El reporte de cierre es realizado al inicio por el administrador del proyecto, y enviado al patrocinador del mismo para su firma.

- Validar la completitud del proyecto

Esta parte de la guía permite confirmar que los criterios requeridos para completar el proyecto están tomados en cuenta y que cualquier criterio fuera de ellos se ha incluido en esta etapa, se sugiere utilizar la tabla 4.1 donde se detallan las categorías y los criterios tomados en cuenta y si fueron o no aceptados por el usuario.

Categoría	Criterio	Aceptación del usuario
Objetivos	Mencionar todos los criterios medibles en los objetivos	S/N
Entregables	Todos los productos finales pactados	S/N
...	...	...

Tala 4.1. Criterios aceptados por usuario

- Identificar las actividades de cierre

Posteriormente identificar las actividades que requiere el cierre del proyecto, esto incluye el manejo de los productos finales y documentación para el usuario, terminar con los contratos de los proveedores, distribuir los recursos al lugar inicial, y la comunicación a todos los involucrados del cierre del proyecto.

La documentación depende del tipo de proyecto, pero se debe poner toda la información necesaria para que sea entendible por los usuarios si en un futuro se desea conocer o utilizar por cualquier persona no involucrada en su desarrollo del proyecto.

Se sugiere llevar una bitácora de la vigencia de los contratos con los proveedores, de manera que tenga el control de los contratos activos y los que su duración caducó con el cierre del proyecto. La tabla 4.2 muestra un ejemplo para la documentación.

Proveedor	N°. Contrato	Actividad de terminación	Fecha	Encargado
Nombre	Referencia de contrato N°.	Actividad con la que se termino el contrato	dd/mm/yyyy	Nombre de la persona encargada de la actividad
...	...			...

Tabla 4.2 Bitácora de vigencia de contratos a proveedores

Para la comunicación del cierre del proyecto con los involucrados, se sugiere completar la información en la tabla 4.3, como un mecanismo de planificación para la difusión a todos los involucrados del cierre del proyecto.

Destinatario	Mensaje	Método	Fecha	Encargado
Nombre(s) de los destinatarios del mensaje	El proyecto "X" ha sido finalizado satisfactoriamente y actualmente está en la fase de cierre.	E-mail a cada involucrado.	dd/mm/yyyy en la que el comunicado debe ser enviado	Nombre de la persona encargada de la actividad de comunicación
...	...			...

Tabla 4.3 Planificación de comunicación de cierre del proyecto

Es momento entonces de juntar toda la información redactada o completada y efectuar el reporte, pasando a la siguiente etapa que son las acciones de cierre identificadas en el éste.

Completar las acciones de cierre del proyecto

Llevar a cabo todas las actividades reportadas en el documento de cierre, las cuales involucran manejo de los productos finales y documentación para el usuario, terminar con los contratos de los proveedores, distribuir los recursos al lugar inicial, y la comunicación a todos los involucrados en la conclusión del proyecto.

4.1.7.2 Revisar la completitud del proyecto

El punto final en el ciclo de vida del proyecto es la revisión de su completa realización. Una revisión de la post-implementación es necesaria para identificar el aprendizaje logrado con él.

Llevar a cabo una revisión de “post-implementación”

La revisión de “post-implementación” (RPI) es un análisis de todos los sucesos que se han tenido en el proyecto. El RPI conduce a concluir los logros obtenidos desde el plan original hasta su cierre, y analizar el cumplimiento de los planes desarrollados durante la planificación del proyecto.

El propósito del RPI, es identificar las lecciones que nos deja el proyecto y desarrollar posibles recomendaciones para futuros proyectos.

- Beneficios

Identificar alguna desviación entre los beneficios pronosticados por el proyecto y los beneficios obtenidos.

- Objetivos

Identificar el porcentaje de cumplimiento de los objetivos, y de no haberse alcanzado, las razones de ello y sus posibles recomendaciones para su mejora en el futuro.

- Alcance

Identificar las posibles desviaciones del alcance original, definido en los términos de referencia, las razones que las justifiquen y las recomendaciones.

- Producto final

Listar los productos finales originales, sus criterios de calidad definidos en el plan y compararlos con el nivel de calidad alcanzado usando la tabla 4.4

Productos Finales	Criterio de calidad	Estándares de calidad	% logrado
Nombre(s) de los entregables	Criterios de calidad definidos en el plan de calidad	Estándares de calidad definidos en el plan de calidad	0-100%
...	...		

Tabla 4.4 Productos finales resultantes

- Calendarización

Comparar el calendario de actividades obtenido al finalizar la ejecución del proyecto con el realizado inicialmente en el plan de proyecto, y analizar los factores que influyeron en su desfase o su adelanto.

- Gastos

Identificar cualquier desviación entre los gastos pronosticados en la planificación del proyecto y los que se tuvieron, tal como fueron documentados en el plan financiero.

- Recursos

Comparar la cantidad de recursos pronosticados a ser utilizados para el proyecto y cuanto fue lo que realmente se uso (detallado en el plan de recursos).

Efectuar una revisión de “planificación”

La revisión de “Administración” (RA) es un análisis de todos los planes que se han tenido en el proyecto. El RA conduce a evaluar la administración obtenida durante el proyecto, la cual se realizó en la fase de iniciación y planificación del mismo

- Administración del tiempo

Identificar la buena planificación del tiempo de duración del proyecto y sus actividades a realizar. Se sugiere utilizar la tabla 4.5 donde se responde S/N a cada pregunta; sin embargo, si se tiene otra pregunta, puede ser incluida.

Criterios	S/N
¿Fue una buena distribución del tiempo para las actividades en el plan de proyecto?	S/N
¿Estaban incluidas en la calendarización todas las actividades, tareas y dependencias?	S/N
¿Fueron los recursos claramente estimados durante el plan del proyecto?	S/N
¿Utilizó el equipo de trabajo el tiempo requerido para actualizar sus tareas y tiempo de duración?	S/N
¿Se alcanzó alguna dificultad de tiempo?	S/N
¿Se requirió tiempo adicional para alguna actividad sobre lo planeado?	S/N
¿Existió alguna actividad en la que no se haya planificado tiempo para su desarrollo?	
...	...

Tabla 4.5 Administración del tiempo

- Administración de los costos

Identificar la buena planificación de los costos del proyecto. Se sugiere utilizar la tabla 4.6 donde se responde S/N a cada pregunta; sin embargo, si se tiene otra interrogante, puede ser incluida.

Criterios	S/N
¿Fueron todos los tipos de gastos identificados en el plan financiero?	S/N
¿Fueron todos los gastos adecuadamente cuantificados en el plan financiero?	S/N
¿El proyecto incurrió en todos los gastos planificados?	S/N
¿Fue completado el formulario de gastos para todos los gastos experimentados?	S/N
¿Fue el procedimiento de aprobación seguido al pie de la letra?	S/N
¿Existieron gastos fuera del presupuesto?	S/N
¿Estuvo de acuerdo el usuario con los gastos incurridos durante el proyecto?	S/N
...	...

Tabla 4.6 Administración del costos

- Administración de la calidad

Identificar la buena calidad de los productos finales. Se sugiere utilizar la tabla 4.7 donde se responde S/N a cada pregunta, sin embargo si se tiene otra interrogante, puede ser incluida.

Criterios	S/N
¿Estuvieron todos los productos finales incluidos en el plan de calidad?	S/N
¿Fueron bien identificados los criterios y estándares de calidad?	S/N
¿Fue el control de calidad aplicado mediante técnicas definidas?	S/N
¿Fueron los roles de calidad formalmente definidos?	S/N
¿El equipo de trabajo operó de acuerdo a los roles definidos?	S/N
¿Fueron todos los productos finales registrados bajo la calidad definida?	S/N
¿Se llenaron todos los formularios de calidad correspondientes?	S/N
¿Se tuvo alguna desviación de calidad en los productos finales?	S/N
...	...

Tabla 4.7 Administración de la calidad

- Administración de los cambios

Se propone el uso de la tabla 4.8 para la evaluación de los cambios; sin embargo, si se tiene alguna otra pregunta adicional, se agrega a la lista planteada.

Criterios	S/N
¿Fueron todos los cambios manejados bajo el proceso formal de los cambios?	S/N
¿Fueron los formularios de los cambios llenados con los cambios?	S/N
¿Fue el control de calidad aplicado mediante técnicas definidas?	S/N
¿Fueron los roles de calidad formalmente definidos?	S/N
¿El equipo de trabajo operó de acuerdo a los roles definidos?	S/N
¿Fueron todos los productos finales registrados bajo la calidad definida?	S/N
¿Se llenaron todos los formularios de calidad correspondientes?	S/N
¿Se tuvo alguna desviación de calidad en los productos finales?	S/N
...	...

Tabla 4.8 Administración de los cambios

- Administración de riesgos

Se propone el uso de la tabla 4.9, para la evaluación de riesgos; sin embargo, si se tiene alguna otra pregunta adicional, se agrega a la lista planteada.

Criterios	S/N
¿Fueron los riesgos del proyecto oportunamente identificados?	S/N
¿Fueron los riesgos administrados mediante un proceso formal?	S/N
¿Fueron los riesgos documentados formalmente?	S/N
¿Fue la probabilidad y el impacto de los riesgos asignado apropiadamente?	S/N
¿Fue el plan de mitigación claramente identificado para cada riesgo?	S/N
¿Fueron las acciones de mitigación realizadas adecuadamente?	S/N
¿Fueron los roles adecuadamente definidos?	S/N
¿El equipo trabajo adecuadamente con cada rol?	S/N

¿Existió algún riesgo que impacto eventualmente el proyecto?	
...	S/N
...	...

Tabla 4.9 Administración de riesgos

- Administración de dificultades

Se propone el uso de la tabla 4.10, para la administración de dificultades; sin embargo, si se tiene alguna otra pregunta adicional, se agrega a la lista planteada.

Criterios	S/N
¿Fueron las dificultades del proyecto oportunamente identificados?	S/N
¿Fueron las dificultades administradas mediante un proceso formal?	S/N
¿Fueron las dificultades documentados formalmente?	S/N
¿Fueron las recomendaciones claramente identificadas para cada dificultad?	S/N
¿Fueron las dificultades superadas adecuadamente?	S/N
¿Fueron los roles adecuadamente definidos?	S/N
¿El equipo trabajo adecuadamente con cada rol?	S/N
¿Existió alguna dificultad que impidió el desarrollo eventualmente del proyecto?	S/N
...	...

Tabla 4.10 Administración de dificultades

- Administración de la comunicación

Se deben discutir los mecanismos utilizados para la comunicación durante el proyecto, se sugiere utilizar el mismo mecanismo de las preguntas pre-formuladas

como en los puntos anteriores, sin embargo la forma de evaluar la comunicación depende del tipo de proyecto, así como las preguntas a realizar.

Identificar los logros del proyecto

Una vez analizados los planes desarrollados, se deben listar los logros que se alcanzaron durante el proyecto, y describir los efectos positivos que cada logro tuvo para los usuarios.

Identificar los fallos del proyecto

Detectados los fallos, deben listarse junto a los efectos que cada uno de ellos tuvo para la empresa.

Documentar las lecciones obtenidas

Listar las lecciones que el proyecto dejó a los integrantes del equipo, se deben poner en común y en conjunto en un solo documento, y una posible recomendación para futuros proyectos.

4.2 Glosario

- **Fases.**

Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos temporales correspondientes a la asignación de recursos (humanos, financieros o materiales).

Cuanto más grande y complejo sea un proyecto, mayor detalle se necesitará en la definición de las fases para que el contenido de cada una siga siendo manejable. De esta forma, cada fase de un proyecto puede considerarse un “*micro-proyecto*” en sí mismo, compuesto por un conjunto de micro-fases.



Otro motivo para descomponer una fase en subfases menores puede ser el interés de separar partes temporales del proyecto que se subcontraten a otras organizaciones, requiriendo distintos procesos de gestión.

Cada fase viene definida por un conjunto de elementos observables externamente, como son las actividades con las que se relaciona, los datos de entrada (resultados de la fase anterior, documentos o productos requeridos para la fase, experiencias de proyectos anteriores), los datos de salida (resultados a utilizar por la fase posterior, experiencia acumulada, pruebas o resultados efectuados) y la estructura interna de la fase.



Figura 5.1. Esquema general de operación de una fase

- **Entregables o productos ("deliverables").**

Son los productos intermedios que generan las fases. Pueden ser materiales (componentes, equipos) o inmateriales (documentos, software). Los entregables permiten evaluar la marcha del proyecto mediante comprobaciones de su adecuación o no a los requisitos funcionales y de condiciones de realización previamente establecidos. Cada una de estas evaluaciones puede servir, además, para la toma de decisiones a lo largo del desarrollo del proyecto.

- **Proyecto.**

Un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado.

- **Ciclo de vida de un proyecto.**

Un conjunto de fases del proyecto que, generalmente son secuenciales, cuyos nombres y números son determinadas por las necesidades de control de la organización u organizaciones involucradas en el proyecto.

- **Estructura de Desglose del trabajo (EDT).**

Una descomposición jerárquica con orientación hacia el producto entregable relativa al *trabajo* que será *ejecutado* por el *equipo de proyecto* para lograr los *objetivos* del proyecto y crear los productos requeridos.

5 CAPITULO IV

5.1 Aplicación del Sistema de evaluación general de desempeño

5.1.1 Acta de constitución

Sistema de evaluación general de desempeño

INSTITUTO TECNOLÓGICO CENTROAMERICANO

Objetivo

1. Será la base que se necesita para autorizar formalmente un proyecto dentro de un periodo de tiempo, sin superar el presupuesto y cumpliendo los requisitos del Instituto Tecnológico Centroamericano.

Desarrollo

DETALLES DEL PROYECTO

Nombre del proyecto: Guía para la implementación de las mejores prácticas en la dirección de proyectos aplicado en un proyecto de desarrollo de software para el Instituto Tecnológico Centroamericano.

Reporte preparado por: Salvador Fonseca

Encargado del proyecto: Arturo Cerrato

Fecha de preparación: 15 Agosto 2007

Patrocinador del proyecto: Instituto Tecnológico Centroamericano.

Período del reporte: Anual.

Descripción del proyecto: Planificación anual de tareas y proyectos asignados a cada empleado.

ESTATUS GLOBAL

Status Global: Iniciado

Calendarización: inicio 16Agosto2007- finalización 15Marzo2008

Gastos del proyecto: \$4,817.14

Entregables: Sistema de evaluación general de desempeño.

DETALLES DE APROBACION

Esta acta será orientada para poder realizar y cumplir el inicio de las expectativas del Instituto para la administración del proyecto “sistema de evaluación general de desempeño”, el cual la gerencia de tecnología designo al señor Arturo Cerrato como director del proyecto para dar inicio a este.

Lo que interesa en esta primera fase es identificar o buscar soluciones, en el caso particular de estudio se realizo actividades para resolver las necesidades en el área de avance tecnológico como petición del cliente (ITCA), en el acta se debe definir cuál es el problema y sus expectativas para poder realizar su futuro alcance del proyecto. La definición de la problemática del proyecto se debe hacer dentro de los límites del ambiente físico en el cual ocurre y bajo condiciones presentes. Definido su necesidad se pretende desarrollar un sistema informático para su uso interno que deberá ser capaz de administrar la planificación anual de cada empleado, las asignaciones de tareas de desarrollo profesional, actividades no planificadas y sus respectivas evaluaciones.

Los proyectos van siempre a ser esfuerzos temporales. Las metas y los objetivos se indican al principio del proyecto mediante la realización de una fecha inicio y una final para el proyecto. La fecha de inicio que formalmente dará inicio fue el 16 de Agosto del 2007 con una fecha final del 15 de Marzo del 2008. Estas fechas se

pueden validar dependiendo de las nuevas decisiones tomadas durante el desarrollo del proyecto a partir del original.

Además hay que tomar en cuenta los estándares y los códigos aplicables para el proyecto en el instituto, la capacidad técnica, la capacidad de la gerencia general, las habilidades del recurso humano, e información que fueron proporcionados por ellos.

Patrocinador del proyecto:

Firma: _____

Fecha: 16/08/2007

Carlos Humberto López Linares

El proyecto ha sido aprobado para proceder a la fase de planificación del proyecto.

5.1.2 ACTA DE APROBACIÓN DE MÓDULO

Por este medio hacemos constar que las pruebas realizadas al módulo de “Planificación Anual”, han cumplido satisfactoriamente todos los requisitos solicitados en la recolección de requerimientos. Por lo tanto, aceptamos la entrega de este módulo por parte del Sr. Arturo Cerrato Director del proyecto de desarrollo y damos por finalizado el desarrollo de este módulo.

Sr. Arturo Cerrato

Ing. Carlos López

San Salvador, 22 de Septiembre de 2007

5.1.3 Visión del proyecto

Objetivo

Presentar una explicación general y breve de las metas, alcances y beneficios que aportará el nuevo sistema.

Visión

El “Sistema de evaluación general de desempeño” permitirá al ITCA tener un control integral del desempeño de cada uno de sus empleados.

Se dividirá en cuatro módulos que medirán el desempeño de:

1. La planificación anual de tareas y proyectos asignados a cada empleado.
Tareas, proyectos y presupuestos asignados a cada gerente.
2. Actividades asignadas a cada empleado para su desarrollo profesional.
3. Actividades no planificadas realizadas por cada empleado.
4. Evaluación anual de 360°.

5.1.4 Cuestionario de entrevista con el cliente

Objetivo

Entender el problema desde el punto de vista del cliente considerando como interactúan con el sistema, pero sin involucrar aspectos tecnológicos o de diseño.

Identificar las necesidades y refinar los requerimientos que el cliente necesita del sistema.

Cuestionario

1. ¿Quién utilizará el sistema?

Todo el personal administrativo.

2. ¿Dónde será utilizado el sistema?

a. En las oficinas centrales del ITCA.

b. En las oficinas regionales del ITCA. (Interior del País)

c. En el hogar de los empleados.

3. ¿Cómo manejan el plan de trabajo por empleado en este momento?

En este momento el control se lleva en hojas de MS Excel.

4. ¿Cómo manejan las tareas no planificadas?

En la mayoría de los casos no se lleva un control y quienes lo llevan lo hacen de manera independiente.

5. ¿Cómo manejan el plan de desarrollo del empleado en este momento?

Cada jefe lleva en hojas de MS Excel un control del plan de desarrollo de sus subalternos.

6. ¿Cómo manejan la evaluación anual del desempeño del personal 360° en este momento?

Este control se maneja en hojas de papel y es realizado por cada jefe de área.

7. ¿Quién asigna las tareas de cada empleado?

El jefe de área o el gerente de departamento.

8. ¿Quién ingresará la información al sistema?

Cada empleado o su respectivo jefe.

9. ¿Quién puede ver, modificar o aprobar las tareas de los empleados?

Cada empleado puede modificar solamente su información personal, y los jefes y gerentes pueden modificar la información de sus subalternos.

10. ¿Quién inicia el proceso de evaluación anual de un empleado?

Su superior inmediato.

11. ¿Cuándo se inicia el proceso de evaluación anual del empleado?
Un año después que el empleado ha iniciado labores en la empresa.
12. ¿Cuántos cuestionarios se llenan por empleado en la evaluación anual?
Depende del número de evaluadores que el superior designe para la evaluación. (Máximo 5)
13. ¿Quién selecciona los evaluadores de cada empleado?
Su superior inmediato.
14. ¿Cuál es la base para seleccionar a los evaluadores?
Se toman en consideración 4 tipos de evaluadores: subalternos (en caso que existan), superiores, compañeros del mismo nivel jerárquico y compañeros de otros departamentos que se han visto involucrados en el trabajo del empleado en evaluación.
15. ¿Cómo se notifica a los evaluadores que deben llenar el cuestionario?
Actualmente cada jefe o gerente notifica a los evaluadores por correo electrónico.
16. ¿Cómo y dónde llenan el cuestionario los evaluadores?
Son formularios impresos que cada evaluador llena a mano.
17. ¿Existe algún caso especial que pudiera mencionar en este momento?
No.
18. Se requiere algún formato especial para los reportes como hojas de Excel o archivos pdf?
No.

5.1.5 Enunciado del Alcance Del Proyecto

Objetivo

Definir el proyecto y los objetivos que deben cumplirse. Documentar las características y los límites del proyecto, sus productos y servicios relacionados así como los métodos de aceptación y el control del alcance.

Desarrollo

Definición

El “Sistema de evaluación general de desempeño” permitirá al ITCA tener un control integral del desempeño de cada uno de sus empleados.

Se dividirá en cuatro módulos que medirán el desempeño de:

1. La planificación anual de tareas y proyectos asignados a cada empleado.
Tareas, proyectos y presupuestos asignados a cada gerente.
2. Actividades asignadas a cada empleado para su desarrollo profesional.
3. Actividades no planificadas realizadas por cada empleado.
4. Evaluación anual de 360°.

5.1.5.1 Objetivos que debe cumplir el proyecto

5.1.5.1.1 Módulo de Planificación Anual

Registrar la planificación anual de tareas por proyecto para cada empleado de la institución.

Se podrán crear proyectos por año y a cada uno de estos proyectos se le asignarán tareas que tendrán como responsable un único empleado de la institución. A cada tarea se le asignará un porcentaje de avance proyectado para cada mes del año.

5.1.5.1.2 Módulo de Desarrollo Profesional

Se le asignara a cada empleado de la institución las tareas de desarrollo profesional que tendrá como meta para el presente año. Estas tareas de desarrollo profesional deberán ser de carácter educativo y podrán ser o no financiados por la institución. A cada tarea de desarrollo profesional se le asignara un porcentaje mensual de avance.

5.1.5.1.3 Módulo de Actividades no Planificadas

Este módulo registrará todas aquellas actividades que son realizadas por los empleados pero que no fueron planificadas dentro de la Planificación Anual.

5.1.5.1.4 Módulo de Evaluación General

Administra las evaluaciones anuales que la institución realiza a los empleados. Esta es una evaluación de 360° ya que cada empleado es evaluado en cada aspecto de su trabajo desde diferentes perspectivas estas incluyen:

1. Su superior
2. Sus subalternos
3. Sus compañeros de Área
4. Otras personas de la institución vinculadas a su trabajo

5.1.5.1.5 Generalidades

El programa deberá contar con un sistema de seguridad capaz de identificar a cada usuario a través de un nombre y una contraseña. Se manejarán dos perfiles de usuario, uno para jefes y supervisores y otro para el resto de empleados.

La diferencia entre los perfiles será que los supervisores tendrán acceso a la información de sus subalternos mientras que los empleados solo podrán ver la información relacionada a ellos mismos. Adicionalmente los jefes y supervisores podrán crear y administrar usuarios dentro del sistema.

El sistema también deberá manejar información de cada usuario como correo electrónico puesto de trabajo y superior.

5.1.5.2 Características y límites del proyecto

El sistema podrá ser utilizado por cada empleado de la institución desde su puesto de trabajo (esto incluye oficinas centrales y oficinas regionales) y adicionalmente podrán utilizarlo desde sus hogares.

El sistema utilizará tecnología Web como plataforma de desarrollo. El lenguaje de programación deberá ser JAVA y el motor de Base de Datos será MS SQL Server 2005.

Se podrá hacer uso de MS Reporting Services como plataforma para crear los reportes.

El sistema final deberá ser instalado en los servidores de producción del ITCA designados por las autoridades competentes.

5.1.5.3 Productos y servicios relacionados

5.1.5.3.1 Módulo de Planificación Anual

[illegible]

5.1.5.3.2 Módulo de Desarrollo Profesional

No.	OBJETIVOS ESPECÍFICOS Y ACTIVIDADES	MESES												RESULTADOS		EVALUACIÓN DE RESULTADOS del 30 DE ABRIL DE 2007 Acuerdo/Comentarios/Comprensión por completo	
		1	2	3	4	5	6	7	8	9	10	11	12	60%	100%		
2 PLAN DE DESARROLLO DEL EMPLEADO																	
2.1 Primer semestre																	
2.1.1	Forma física	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%
2.1.2	Formación y desarrollo profesional	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%
2.2 Segundo Semestre																	
Completar con el mismo título																	
Según seminario y dictando la norma (T)																	
3 OTRAS ACTIVIDADES																	
3.1	Reporte de actividades del área de Física	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%
3.2	Reporte de actividades del área de Biología	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%	8.00%
4 CONSULTA																	
Indique tres cargos de personas que puedan ser consultados, como parte de la revisión, por el Jefe inmediato.																	
CARGO CONSULTADO																	
NOMBRE CONSULTADO																	
GERENTE DE ESCUELA COMPLECIÓN																	
5 FIRMAS																	
Firma del empleador:																	
Firma del jefe inmediato:																	
Fecha de elaboración:																	
18/05/2007																	
6 APROBACIÓN (Presidente Ejecutivo, Director, Coordinador, Gerente de Área, Jefe)																	
Aprobado por:																	
TMC CARLOS HUMBERTO ORTIZ																	

5.1.5.3.3 Módulo de Actividades no Planificadas

[illegible]

5.1.5.3.4 Módulo de Evaluación Anual

Reportes

[ITCA-FEPADE]

FRHCO-12: REVISION DEL DESEMPEÑO CARGOS INTERRELACIONADOS PERSONAL EJECUTIVO

REVISIÓN: 1

2007 CICLO I-EVALUACION DEL DESEMPEÑO

CENTRO DE COSTO: 110806 - INFORMATICA

Empleado: JACQUELINE YOLANDA ESCALANTE HENRIQUEZ

Sep 7 2007 7:52:51-310PM

VALORACIÓN

AREA	0 NA	1 NM	2 R	3 B	4 MB	5 E
CAPACIDAD						
1 Por los efectos de su trabajo considera usted que es una persona competente en su cargo.	0	0	0	1	2	1
2 Domina su área trabajo.	0	0	0	1	2	1
3 Incorpora nuevos conocimientos en su trabajo.	0	0	0	1	1	2
4 Resuelve conflictos de su área de una manera efectiva y acorde a los lineamientos institucionales.	0	0	0	1	2	1
Total CAPACIDAD	0	0	0	4	7	5
COMUNICACIÓN Y PARTICIPACION						
1 Demuestra capacidad para expresar sus ideas con claridad.	0	0	0	0	2	2
2 Sabe escuchar a los demás.	0	0	1	1	1	1
3 Es una persona participativa (en iniciativas de mejora, reuniones y actividades institucionales) .	0	0	0	1	2	1
4 Es de las personas que brinda apoyo si se le solicita.	0	0	0	0	2	2
Total COMUNICACIÓN Y PARTICIPACION	0	0	1	2	7	6
GESTION DEL PERSONAL						
1 Faculta al personal para tomar decisiones.	0	0	0	1	2	1
2 Fomenta la comunicación efectiva al interior de la institución.	0	0	0	2	2	0
3 Valora a los empleados como personas.	0	0	0	1	2	1
4 Genera un ambiente de confianza y de solidaridad entre sus compañeros	0	0	0	1	2	1
5 Motiva a que realice un trabajo con calidad.	0	0	0	1	2	1
6 Es amigable en el trato con los demás.	0	0	0	1	1	2
7 Ante un problema se puede negociar con él (ella).	0	0	0	1	3	0

Done

My Computer

FRHCO-19: REVISION DEL DESEMPEÑO CARGOS INTERRELACIONADOS
REVISIÓN: 1
2007 CICLO I-EVALUACION DEL DESEMPEÑO
CENTRO DE COSTO: 110806 - INFORMATICA
Empleado: KAREN ARELY(BECADA) REYES ALFARO

Sep 7 2007 7:50:11:077PM

AREA	VALORACIÓN					
	0 NA	1 NM	2 R	3 B	4 MB	5 E
CAPACIDAD						
1 Domina su área trabajo.	0	0	0	3	2	0
2 Incorpora nuevos conocimientos en su trabajo.	0	0	1	2	1	1
3 Es una fuente de nuevas ideas.	0	0	1	2	2	0
4 Resuelve adecuadamente los problemas.	0	0	1	1	3	0
5 Logra el óptimo desempeño de sus funciones.	0	0	1	2	2	0
6 Revisa y mejora la efectividad de su trabajo.	0	0	1	2	2	0
7 Se esmera porque sus clientes internos y externos estén satisfechos con sus servicios.	0	0	1	1	2	1
8 Realiza su trabajo con calidad.	0	0	1	1	2	1
9 Planifica adecuadamente su trabajo.	0	0	1	1	2	1
Total CAPACIDAD	0	0	8	15	18	4
COMUNICACIÓN Y PARTICIPACION						
1 Demuestra capacidad para expresar sus ideas con claridad.	0	0	2	0	3	0
2 Permite el acercamiento con sus compañeros.	0	0	1	1	2	1
3 Sabe escuchar a los demás.	0	0	1	0	4	0
4 Participa en actividades institucionales.	0	0	0	3	2	0
5 Participa activamente en iniciativas de mejora en su área.	0	0	0	3	1	1
6 Participa activamente en las reuniones.	0	0	2	0	2	1
7 Tiene habilidad para comunicarse efectivamente con los demás.	0	0	1	1	3	0
Total COMUNICACIÓN Y PARTICIPACION	0	0	7	8	17	3
RELACIONES INTERPERSONALES						
1 Fomenta un ambiente de trabajo de armonia y compañerismo.	0	0	1	1	3	0
2 Se dirige de manera respetuosa hacia los demás.	0	0	0	2	2	1

5.1.5.4 Métodos de aceptación y control

El sistema se desarrollara en cinco etapas, una por cada módulo:

1. Módulo de Planificación Anual.
2. Módulo de Desarrollo Profesional.
3. Módulo de Actividades no Planificadas.
4. Módulo de Evaluación Anual.
5. Módulo de Seguridad.

Para cada una de estas etapas:

1. Se sostendrán reuniones para la recolección de requerimientos con el personal designado por el ITCA. Al final de la recolección de requerimientos se creara un informe que recopilara toda la información obtenida, este documento será firmado por el Gerente de IT del ITCA como una muestra de aprobación de la información recopilada.

2. Una vez se termine la programación cada módulo el personal designado por el ITCA realizará pruebas a los programas. Si se encontrara problemas de funcionamiento, se procederá a corregir el programa. Si se encuentran inconsistencias entre el programa y el documento que recopila los requerimientos, se procederá a realizar los cambios necesarios. Si se solicitan cambios debido a requerimientos que no estaban contemplados en el documento que recopila los requerimientos, se discutirán con el gerente para determinar si se deben incluir en el sistema o se incluirán en una versión posterior.
3. Finalizadas las pruebas y los cambios solicitados, se proceder a la implementación del módulo en los servidores de producción.
4. El personal designado por el ITCA realizara una revisión final del módulo y de no encontrar insatisfacciones, se procederá a firmar el documento de aceptación.
5. Con la aceptación del último módulo del sistema se dará por finalizado el proyecto.

5.1.6 Estructura de Desglose del Trabajo EDT

Objetivo

Presentar una descomposición jerárquica orientada al producto entregable del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requeridos. La EDT subdivide el trabajo del proyecto en porciones de trabajo más pequeñas y fáciles de manejar, donde cada nivel descendente de la EDT representa una definición cada vez más detallada del trabajo de proyecto.

Desarrollo

☐ Sistema de Evaluación General de Desempeño
☐ Primera Etapa
☐ Iniciación del proyecto
☐ Investigación preliminar
Administración de proyectos
Aplicaciones JAVA, JSP y JSF
UML
Elaboración del Alcance del proyecto
Elaboración de la Estructura de Desglose de Trabajo (EDT)
Elaboración de la Estructura de Desglose de la Organización (EDO)
Elaborar la Planificación de Actividades del proyecto (preliminar)
Elaborar Presupuesto preliminar del proyecto
Elaborar Plan de Administración de Cambios
Elaboración del Acta de constitución del proyecto
Elaborar Estándares del proyecto
Elaborar Manual de políticas y procedimientos del proyecto
Compra MS Project Server
Compra de 2 LapTop Pavillion
Compra de Servidor DELL PowerEdge 2900
Revisión de la documentación de iniciación del proyecto junto con autoridades del ITCA
Correcciones a la Documentación de iniciación del proyecto
Aprobación de la documentación e inicio formal del proyecto
☐ Control y Monitoreo
Reunión de seguimiento

☐ Módulo de Planificación Anual
Proceso de iniciación del módulo
☐ Recolección de Requerimientos
Elaboración de Diagrama de Casos de uso
☐ Elaboración de Casos de uso
☐ Agregar Proyecto
☐ Agregar Tarea
☐ Agregar Presupuesto
Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITC/
Correcciones a los Diagramas de Casos de uso y Actividad
Aprobación de los Diagramas de Casos de Uso y Actividad
☐ Programación de Casos de uso
Agregar Proyecto
Agregar Tarea
Agregar Presupuesto
Pruebas del módulo
Correcciones al módulo
Implementación del módulo en producción
Proceso de finalización del módulo

☐ Módulo de Desarrollo Profesional
Proceso de iniciación del módulo
☐ Recolección de Requerimientos
Elaboración de Diagrama de Casos de uso
☐ Elaboración de Casos de uso
☐ Agregar Tarea
Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA
Correcciones a los Diagramas de Casos de uso y Actividad
Aprobación de los Diagramas de Casos de Uso y Actividad
☐ Programación de Casos de uso
Agregar Tarea
Pruebas del módulo
Correcciones al módulo
Implementación del módulo en producción
Proceso de finalización del módulo

☐ Módulo de Actividades no planificadas
Proceso de iniciación del módulo
☐ Recolección de Requerimientos
Elaboración de Diagramas de Casos de uso
☐ Elaboración de Casos de uso
☐ Agregar Tarea
Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA
Correcciones a los Diagramas de Casos de uso y Actividad
Aprobación de los Diagramas de Casos de Uso y Actividad
☐ Programación de Casos de uso
Agregar Tarea
Pruebas del módulo
Correcciones al módulo
Implementación del módulo en producción
Proceso de finalización del módulo

☐ Módulo de Evaluación Anual
Proceso de iniciación del módulo
☐ Recolección de Requerimientos
Elaboración de Diagramas de Casos de uso
☐ Elaboración de Casos de uso y Diagramas de Actividad
☐ Agregar Cuestionario
☐ Configuración de la evaluación
☐ Iniciar Evaluación
☐ Contestar Cuestionario
Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA
Correcciones a los Diagramas de Casos de uso y Actividad
Aprobación de los Diagramas de Casos de Uso y Actividad
☐ Programación de Casos de uso
Agregar Cuestionario
Configuración de la evaluación
Iniciar Evaluación
Contestar Cuestionario
Pruebas del módulo
Correcciones al módulo
Implementación del módulo en producción
Proceso de finalización del módulo

☐ Módulo de Seguridad
Proceso de iniciación del módulo
☐ Recolección de Requerimientos
Elaboración de Diagramas de Casos de uso
☐ Elaboración de Casos de uso y Diagramas de Actividad
⊕ Agregar Departamento
⊕ Agregar Cargo de Empleado
⊕ Agregar Empleado
⊕ Registrar Ingreso al sistema
⊕ Cambio de Contraseña
Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA
Correcciones a los Diagramas de Casos de uso y Actividad
Aprobación de los Diagramas de Casos de Uso y Actividad
☐ Programación de Casos de uso
Agregar Departamento
Agregar Cargo de Empleado
Agregar Empleado
Registrar Ingreso al sistema
Cambio de Contraseña
Pruebas del módulo
Correcciones al módulo
Implementación del módulo en producción
Proceso de finalización del módulo

☐ Segunda Etapa
Elaboración de reportes
☐ Elaboración de Manuales
Elaboración de Manual de usuario
Elaboración de Manual Técnico
Capacitación de usuarios de la aplicación
Finalización del proyecto

5.1.7 Secuencia de Actividades

Objetivo

Pretende identificar y documentar las relaciones lógicas entre las actividades del cronograma.

Desarrollo

Las actividades del cronograma pueden estar ordenadas de forma lógica con relaciones de precedencia adecuadas, así como también adelantos y retrasos, para respaldar el desarrollo posterior de un cronograma del proyecto realista y factible.

Todas las actividades dentro de este proyecto se vincularan a través de relaciones de tipo: Final a Inicio, esto quiere decir que cuando una actividad dependa de otra, el inicio de la actividad sucesora depende de la finalización de la actividad predecesora.

	 Task Name	Predeces
1	[-] Sistema de Evaluación General de Desempeño	
2	[+] Control y Monitoreo	
34	[-] Primera Defensa	
35	[-] Iniciación del proyecto	
36	[-] Investigación preliminar	
37	Administración de proyectos	
38	Aplicaciones JAVA, JSP y JSF	
39	UML	38
40	Elaboración del Alcance del proyecto	37
41	Elaboración de la Estructura de Desglose de Trabajo (EDT)	40
42	Elaboración de la Estructura de Desglose de la Organización (EDO)	41
43	Elaborar la Planificación de Actividades del proyecto (preliminar)	42
44	Elaborar Presupuesto preliminar del proyecto	43
45	Elaborar Plan de Administración de Cambios	37,39
46	Elaboración del Acta de constitución del proyecto	44
47	Elaborar Estándares del proyecto	37,45
48	Elaborar Manual de políticas y procedimientos del proyecto	47
49	 Compra MS Project Server	
50	 Compra de 2 LapTop Pavillion	
51	 Compra de Servidor DELL PowerEdge 2900	
52	Revisión de la documentación de iniciación del proyecto junto con autoridades del ITCA	46,45
53	Correcciones a la Documentación de iniciación del proyecto	52
54	Aprobación de la documentación e inicio formal del proyecto	53

55	☐ Módulo de Planificación Anual	35
56	Proceso de iniciación del módulo	
57	☐ Recolección de Requerimientos	56
58	Elaboración de Diagrama de Casos de uso	
59	☐ Elaboración de Casos de uso	58
60	☐ Agregar Proyecto	
61	Elaborar Definición de Caso de uso y Flujos de eventos	
62	Diagrama de Actividad	61
63	Elaboración de Diagrama de Clases	62
64	Elaboración de Diagramas de Secuencias	63
65	☐ Agregar Tarea	60
66	Elaborar Definición de Caso de uso y Flujos de eventos	
67	Diagrama de Actividad	66
68	Elaboración de Diagrama de Clases	67
69	Elaboración de Diagramas de Secuencias	68
70	☐ Agregar Presupuesto	65
71	Elaborar Definición de Caso de uso y Flujos de eventos	
72	Diagrama de Actividad	71
73	Elaboración de Diagrama de Clases	72
74	Elaboración de Diagramas de Secuencias	73
75	Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA	59
76	Correcciones a los Diagramas de Casos de uso y Actividad	75
77	Aprobación de los Diagramas de Casos de Uso y Actividad	76
78	☐ Programación de Casos de uso	57
79	Agregar Proyecto	
80	Agregar Tarea	79
81	Agregar Presupuesto	80
82	Pruebas del módulo	78
83	Correcciones al módulo	82
84	Implementación del módulo en producción	83

85	Proceso de finalización del módulo	84
86	☐ Módulo de Desarrollo Profesional	35
87	Proceso de iniciación del módulo	
88	☐ Recolección de Requerimientos	87
89	Elaboración de Diagrama de Casos de uso	
90	☐ Elaboración de Casos de uso	89
91	☐ Agregar Tarea	
92	Elaborar Definición de Caso de uso y Flujos de eventos	
93	Diagrama de Actividad	92
94	Elaboración de Diagrama de Clases	93
95	Elaboración de Diagramas de Secuencias	94
96	Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA	90
97	Correcciones a los Diagramas de Casos de uso y Actividad	96
98	Aprobación de los Diagramas de Casos de Uso y Actividad	97
99	☐ Programación de Casos de uso	88
100	Agregar Tarea	
101	Pruebas del módulo	99
102	Correcciones al módulo	101
103	Implementación del módulo en producción	102
104	Proceso de finalización del módulo	103
105	☐ Módulo de Actividades no planificadas	86
106	Proceso de iniciación del módulo	
107	☐ Recolección de Requerimientos	106
108	Elaboración de Diagramas de Casos de uso	
109	☐ Elaboración de Casos de uso	108
110	☐ Agregar Tarea	
111	Elaborar Definición de Caso de uso y Flujos de eventos	
112	Diagrama de Actividad	111
113	Elaboración de Diagrama de Clases	112

114	Elaboración de Diagramas de Secuencias	113
115	Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA	109
116	Correcciones a los Diagramas de Casos de uso y Actividad	115
117	Aprobación de los Diagramas de Casos de Uso y Actividad	116
118	▢ Programación de Casos de uso	107
119	Agregar Tarea	
120	Pruebas del módulo	118
121	Correcciones al módulo	120
122	Implementación del módulo en producción	121
123	Proceso de finalización del módulo	122
124	▢ Módulo de Evaluación Anual	55
125	Proceso de iniciación del módulo	
126	▢ Recolección de Requerimientos	125
127	Elaboración de Diagramas de Casos de uso	
128	▢ Elaboración de Casos de uso y Diagramas de Actividad	127
129	▢ Agregar Cuestionario	
130	Elaborar Definición de Caso de uso y Flujos de eventos	
131	Diagrama de Actividad	130
132	Elaboración de Diagrama de Clases	131
133	Elaboración de Diagramas de Secuencias	132
134	▢ Configuración de la evaluación	129
135	Elaborar Definición de Caso de uso y Flujos de eventos	
136	Diagrama de Actividad	135
137	Elaboración de Diagrama de Clases	136
138	Elaboración de Diagramas de Secuencias	137
139	▢ Iniciar Evaluación	134
140	Elaborar Definición de Caso de uso y Flujos de eventos	
141	Diagrama de Actividad	140
142	Elaboración de Diagrama de Clases	141

143	Elaboración de Diagramas de Secuencias	142
144	▢ Contestar Cuestionario	139
145	Elaborar Definición de Caso de uso y Flujos de eventos	
146	Diagrama de Actividad	145
147	Elaboración de Diagrama de Clases	146
148	Elaboración de Diagramas de Secuencias	147
149	Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA	128
150	Correcciones a los Diagramas de Casos de uso y Actividad	149
151	Aprobación de los Diagramas de Casos de Uso y Actividad	150
152	▢ Programación de Casos de uso	126
153	Agregar Cuestionario	
154	Configuración de la evaluación	153
155	Iniciar Evaluación	154
156	Contestar Cuestionario	155
157	Pruebas del módulo	152
158	Correcciones al módulo	157
159	Implementación del módulo en producción	158
160	Proceso de finalización del módulo	159
161	▢ Módulo de Seguridad	105
162	Proceso de iniciación del módulo	
163	▢ Recolección de Requerimientos	162
164	Elaboración de Diagramas de Casos de uso	
165	▢ Elaboración de Casos de uso y Diagramas de Actividad	164
166	▢ Agregar Departamento	
167	Elaborar Definición de Caso de uso y Flujos de eventos	
168	Diagrama de Actividad	167
169	Elaboración de Diagrama de Clases	168
170	Elaboración de Diagramas de Secuencias	169
171	▢ Agregar Cargo de Empleado	166
172	Elaborar Definición de Caso de uso y Flujos de eventos	
173	Diagrama de Actividad	172
174	Elaboración de Diagrama de Clases	173
175	Elaboración de Diagramas de Secuencias	174
176	▢ Agregar Empleado	171
177	Elaborar Definición de Caso de uso y Flujos de eventos	
178	Diagrama de Actividad	177
179	Elaboración de Diagrama de Clases	178
180	Elaboración de Diagramas de Secuencias	179
181	▢ Registrar Ingreso al sistema	176
182	Elaborar Definición de Caso de uso y Flujos de eventos	
183	Diagrama de Actividad	182
184	Elaboración de Diagrama de Clases	183
185	Elaboración de Diagramas de Secuencias	184
186	▢ Cambio de Contraseña	181
187	Elaborar Definición de Caso de uso y Flujos de eventos	
188	Diagrama de Actividad	187
189	Elaboración de Diagrama de Clases	188
190	Elaboración de Diagramas de Secuencias	189
191	Revisión de los Diagramas de Casos de uso y Actividad junto con autoridades del ITCA	165
192	Correcciones a los Diagramas de Casos de uso y Actividad	191
193	Aprobación de los Diagramas de Casos de Uso y Actividad	192

194	☐ Programación de Casos de uso	163
195	Agregar Departamento	
196	Agregar Cargo de Empleado	195
197	Agregar Empleado	196
198	Registrar Ingreso al sistema	197
199	Cambio de Contraseña	198
200	Pruebas del módulo	194
201	Correcciones al módulo	200
202	Implementación del módulo en producción	201
203	Proceso de finalización del módulo	202
204	☐ Segunda Defensa	34
205	Elaboración de reportes	
206	☐ Elaboración de Manuales	205
207	Elaboración de Manual de usuario	
208	Elaboración de Manual Técnico	
209	Capacitación de usuarios de la aplicación	206
210	Finalización del proyecto	209

5.1.8 Cronograma de Actividades

Objetivo

Determinar las fechas de inicio y finalización planificadas para las actividades del proyecto.

Desarrollo

El desarrollo del cronograma exige que se revisen y se corrijan las estimaciones de duración y las estimaciones de los recursos para crear un cronograma del proyecto aprobado que pueda servir como línea base con respecto a la cual poder medir el avance. El desarrollo del cronograma continua a lo largo del proyecto, a medida que el trabajo avanza, el plan de gestión del proyecto cambia, y los eventos de riesgo anticipados ocurren o desaparecen al tiempo que se identifican nuevos riesgos.

5.1.8.1 Cronograma de Línea Base

Task Name	Predeces	Duration	Resource Names	% Complete	Cost	Start	Finish
[-] Sistema de Evaluación General de Desempeño		252 days		0%	\$4,807.50	Mon 02/07/07	Sun 09/03/08
[-] Control y Monitoreo		239 days		0%	\$60.00	Sun 08/07/07	Sun 02/03/08
[-] Primera Defensa		225 days		0%	\$4,627.50	Mon 02/07/07	Mon 11/02/08
[-] Iniciación del proyecto		57 days		0%	\$3,975.00	Mon 02/07/07	Mon 27/08/07
[-] Investigación preliminar		51 days		0%	\$202.50	Mon 02/07/07	Tue 21/08/07
Administración de proyectos		30 days	Acerrato	0%	\$75.00	Mon 02/07/07	Tue 31/07/07
Aplicaciones JAVA, JSP y JSF		30 days	SFonseca	0%	\$75.00	Mon 02/07/07	Tue 31/07/07
UML	31	21 days	SFonseca	0%	\$52.50	Wed 01/08/07	Tue 21/08/07
Elaboración del Alcance del proyecto	30	3 days	Acerrato	0%	\$7.50	Wed 01/08/07	Fri 03/08/07
Elaboración de la Estructura de Desglose de Trabajo (EDT)	33	3 days	Acerrato	0%	\$7.50	Sat 04/08/07	Mon 06/08/07
Elaboración de la Estructura de Desglose de la Organización (E	34	3 days	Acerrato	0%	\$7.50	Tue 07/08/07	Thu 09/08/07
Elaborar la Planificación de Actividades del proyecto (preliminar	35	3 days	Acerrato	0%	\$7.50	Fri 10/08/07	Sun 12/08/07
Elaborar Presupuesto preliminar del proyecto	36	2 days	Acerrato	0%	\$5.00	Mon 13/08/07	Tue 14/08/07
Elaboración del Acta de constitución del proyecto	37	3 days	Acerrato	0%	\$7.50	Wed 15/08/07	Fri 17/08/07
Elaborar Estándares del proyecto	32	5 days	SFonseca	0%	\$12.50	Wed 22/08/07	Sun 26/08/07
Elaborar Manual de políticas y procedimientos del proyecto	39	1 day	SFonseca	0%	\$2.50	Mon 27/08/07	Mon 27/08/07
Compra MS Project Server		1 day	ITCA	0%	\$600.00	Mon 16/07/07	Mon 16/07/07
Compra de 2 LapTop Pavillion		1 day	ITCA	0%	\$1,600.00	Mon 23/07/07	Mon 23/07/07
Compra de Servidor DELL PowerEdge 2900		1 day	ITCA	0%	\$1,500.00	Tue 24/07/07	Tue 24/07/07
Revisión de la documentación de iniciación del proyecto junto c	38	1 day	ITCA	0%	\$0.00	Sat 18/08/07	Sat 18/08/07
Correcciones a la Documentación de iniciación del proyecto	44	5 days	Acerrato	0%	\$12.50	Sun 19/08/07	Thu 23/08/07
Aprobación de la documentación e inicio formal del proyecto	45	1 day	Clópez	0%	\$2.50	Fri 24/08/07	Fri 24/08/07

[-] Módulo de Planificación Anual	28	47 days		0%	\$117.50	Tue 28/08/07	Sat 13/10/07
Proceso de iniciación del módulo		1 day	SFonseca	0%	\$2.50	Tue 28/08/07	Tue 28/08/07
[-] Recolección de Requerimientos	48	26 days		0%	\$65.00	Wed 29/08/07	Sun 23/09/07
Elaboración de Diagrama de Casos de uso		1 day	SFonseca	0%	\$2.50	Wed 29/08/07	Wed 29/08/07
[-] Elaboración de Casos de uso	50	21 days		0%	\$52.50	Thu 30/08/07	Wed 19/09/07
[-] Agregar Proyecto		7 days		0%	\$17.50	Thu 30/08/07	Wed 05/09/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Thu 30/08/07	Sat 01/09/07
Diagrama de Actividad	53	1 day	SFonseca	0%	\$2.50	Sun 02/09/07	Sun 02/09/07
Elaboración de Diagrama de Secuencia	54	2 days	SFonseca	0%	\$5.00	Mon 03/09/07	Tue 04/09/07
Elaboración de Diagrama de Estados	55	1 day	SFonseca	0%	\$2.50	Wed 05/09/07	Wed 05/09/07
[-] Agregar Tarea	52	7 days		0%	\$17.50	Thu 06/09/07	Wed 12/09/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Thu 06/09/07	Sat 08/09/07
Diagrama de Actividad	58	1 day	SFonseca	0%	\$2.50	Sun 09/09/07	Sun 09/09/07
Elaboración de Diagrama de Secuencia	59	2 days	SFonseca	0%	\$5.00	Mon 10/09/07	Tue 11/09/07
Elaboración de Diagramas de Estados	60	1 day	SFonseca	0%	\$2.50	Wed 12/09/07	Wed 12/09/07
[-] Agregar Presupuesto	57	7 days		0%	\$17.50	Thu 13/09/07	Wed 19/09/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Thu 13/09/07	Sat 15/09/07
Diagrama de Actividad	63	1 day	SFonseca	0%	\$2.50	Sun 16/09/07	Sun 16/09/07
Elaboración de Diagrama de Secuencia	64	2 days	SFonseca	0%	\$5.00	Mon 17/09/07	Tue 18/09/07
Elaboración de Diagramas de Estados	65	1 day	SFonseca	0%	\$2.50	Wed 19/09/07	Wed 19/09/07
Revisión de los Diagramas de Casos de uso y Actividad jun	51	1 day	SFonseca	0%	\$2.50	Thu 20/09/07	Thu 20/09/07
Correcciones a los Diagramas de Casos de uso y Actividad	67	2 days	SFonseca	0%	\$5.00	Fri 21/09/07	Sat 22/09/07
Aprobación de los Diagramas de Casos de Uso y Actividad	68	1 day	SFonseca	0%	\$2.50	Sun 23/09/07	Sun 23/09/07

Programación de Casos de uso	49	9 days		0%	\$22.50	Mon 24/09/07	Tue 02/10/07
Agregar Proyecto		3 days	SFonseca	0%	\$7.50	Mon 24/09/07	Wed 26/09/07
Agregar Tarea	71	3 days	SFonseca	0%	\$7.50	Thu 27/09/07	Sat 29/09/07
Agregar Presupuesto	72	3 days	SFonseca	0%	\$7.50	Sun 30/09/07	Tue 02/10/07
Pruebas del módulo	70	5 days	SFonseca	0%	\$12.50	Wed 03/10/07	Sun 07/10/07
Correcciones al módulo	74	3 days	SFonseca	0%	\$7.50	Mon 08/10/07	Wed 10/10/07
Implementación del módulo en producción	75	2 days	SFonseca	0%	\$5.00	Thu 11/10/07	Fri 12/10/07
Proceso de finalización del módulo	76	1 day	SFonseca	0%	\$2.50	Sat 13/10/07	Sat 13/10/07
Módulo de Desarrollo Profesional	28	27 days		0%	\$67.50	Sun 16/09/07	Fri 12/10/07
Proceso de iniciación del módulo		1 day	Acerrato	0%	\$2.50	Sun 16/09/07	Sun 16/09/07
Recolección de Requerimientos	79	12 days		0%	\$30.00	Mon 17/09/07	Fri 28/09/07
Elaboración de Diagrama de Casos de uso		1 day	Acerrato	0%	\$2.50	Mon 17/09/07	Mon 17/09/07
Elaboración de Casos de uso	81	7 days		0%	\$17.50	Tue 18/09/07	Mon 24/09/07
Agregar Tarea		7 days		0%	\$17.50	Tue 18/09/07	Mon 24/09/07
Elaborar Definición de Caso de uso y Flujos de evento		3 days	Acerrato	0%	\$7.50	Tue 18/09/07	Thu 20/09/07
Diagrama de Actividad	84	1 day	Acerrato	0%	\$2.50	Fri 21/09/07	Fri 21/09/07
Elaboración de Diagrama de Secuencia	85	2 days	Acerrato	0%	\$5.00	Sat 22/09/07	Sun 23/09/07
Elaboración de Diagramas de Estados	86	1 day	Acerrato	0%	\$2.50	Mon 24/09/07	Mon 24/09/07
Revisión de los Diagramas de Casos de uso y Actividad jun	82	1 day	Acerrato	0%	\$2.50	Tue 25/09/07	Tue 25/09/07
Correcciones a los Diagramas de Casos de uso y Actividad	88	2 days	Acerrato	0%	\$5.00	Wed 26/09/07	Thu 27/09/07
Aprobación de los Diagramas de Casos de Uso y Actividad	89	1 day	Acerrato	0%	\$2.50	Fri 28/09/07	Fri 28/09/07
Programación de Casos de uso	80	3 days		0%	\$7.50	Sat 29/09/07	Mon 01/10/07
Agregar Tarea		3 days	Acerrato	0%	\$7.50	Sat 29/09/07	Mon 01/10/07
Pruebas del módulo	91	5 days	Acerrato	0%	\$12.50	Tue 02/10/07	Sat 06/10/07
Correcciones al módulo	93	3 days	Acerrato	0%	\$7.50	Sun 07/10/07	Tue 09/10/07
Implementación del módulo en producción	94	2 days	Acerrato	0%	\$5.00	Wed 10/10/07	Thu 11/10/07
Proceso de finalización del módulo	95	1 day	Acerrato	0%	\$2.50	Fri 12/10/07	Fri 12/10/07
Módulo de Actividades no planificadas	78	27 days		0%	\$67.50	Sat 13/10/07	Thu 08/11/07
Proceso de iniciación del módulo		1 day	Acerrato	0%	\$2.50	Sat 13/10/07	Sat 13/10/07
Recolección de Requerimientos	98	12 days		0%	\$30.00	Sun 14/10/07	Thu 25/10/07
Elaboración de Diagramas de Casos de uso		1 day	Acerrato	0%	\$2.50	Sun 14/10/07	Sun 14/10/07
Elaboración de Casos de uso	100	7 days		0%	\$17.50	Mon 15/10/07	Sun 21/10/07
Agregar Tarea		7 days		0%	\$17.50	Mon 15/10/07	Sun 21/10/07
Elaborar Definición de Caso de uso y Flujos de evento		3 days	Acerrato	0%	\$7.50	Mon 15/10/07	Wed 17/10/07
Diagrama de Actividad	103	1 day	Acerrato	0%	\$2.50	Thu 18/10/07	Thu 18/10/07
Elaboración de Diagrama de Secuencia	104	2 days	Acerrato	0%	\$5.00	Fri 19/10/07	Sat 20/10/07
Elaboración de Diagramas de Estado	105	1 day	Acerrato	0%	\$2.50	Sun 21/10/07	Sun 21/10/07
Revisión de los Diagramas de Casos de uso y Actividad jun	101	1 day	Acerrato	0%	\$2.50	Mon 22/10/07	Mon 22/10/07
Correcciones a los Diagramas de Casos de uso y Actividad	107	2 days	Acerrato	0%	\$5.00	Tue 23/10/07	Wed 24/10/07
Aprobación de los Diagramas de Casos de Uso y Actividad	108	1 day	Acerrato	0%	\$2.50	Thu 25/10/07	Thu 25/10/07
Programación de Casos de uso	99	3 days		0%	\$7.50	Fri 26/10/07	Sun 28/10/07
Agregar Tarea		3 days	Acerrato	0%	\$7.50	Fri 26/10/07	Sun 28/10/07
Pruebas del módulo	110	5 days	Acerrato	0%	\$12.50	Mon 29/10/07	Fri 02/11/07
Correcciones al módulo	112	3 days	Acerrato	0%	\$7.50	Sat 03/11/07	Mon 05/11/07
Implementación del módulo en producción	113	2 days	Acerrato	0%	\$5.00	Tue 06/11/07	Wed 07/11/07
Proceso de finalización del módulo	114	1 day	Acerrato	0%	\$2.50	Thu 08/11/07	Thu 08/11/07

☐ Módulo de Evaluación Anual	47	65 days		0%	\$162.50	Sun 14/10/07	Mon 17/12/07
Proceso de iniciación del módulo		1 day	SFonseca	0%	\$2.50	Sun 14/10/07	Sun 14/10/07
☐ Recolección de Requerimientos	117	33 days		0%	\$82.50	Mon 15/10/07	Fri 16/11/07
Elaboración de Diagramas de Casos de uso		1 day	SFonseca	0%	\$2.50	Mon 15/10/07	Mon 15/10/07
☐ Elaboración de Casos de uso y Diagramas de Actividad	119	28 days		0%	\$70.00	Tue 16/10/07	Mon 12/11/07
☐ Agregar Cuestionario		7 days		0%	\$17.50	Tue 16/10/07	Mon 22/10/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Tue 16/10/07	Thu 18/10/07
Diagrama de Actividad	122	1 day	SFonseca	0%	\$2.50	Fri 19/10/07	Fri 19/10/07
Elaboración de Diagrama de Secuencia	123	2 days	SFonseca	0%	\$5.00	Sat 20/10/07	Sun 21/10/07
Elaboración de Diagramas de Estado	124	1 day	SFonseca	0%	\$2.50	Mon 22/10/07	Mon 22/10/07
☐ Configuración de la evaluación	121	7 days		0%	\$17.50	Tue 23/10/07	Mon 29/10/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Tue 23/10/07	Thu 25/10/07
Diagrama de Actividad	127	1 day	SFonseca	0%	\$2.50	Fri 26/10/07	Fri 26/10/07
Elaboración de Diagrama de Secuencia	128	2 days	SFonseca	0%	\$5.00	Sat 27/10/07	Sun 28/10/07
Elaboración de Diagramas de Estado	129	1 day	SFonseca	0%	\$2.50	Mon 29/10/07	Mon 29/10/07
☐ Iniciar Evaluación	126	7 days		0%	\$17.50	Tue 30/10/07	Mon 05/11/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Tue 30/10/07	Thu 01/11/07
Diagrama de Actividad	132	1 day	SFonseca	0%	\$2.50	Fri 02/11/07	Fri 02/11/07
Elaboración de Diagrama de Secuencia	133	2 days	SFonseca	0%	\$5.00	Sat 03/11/07	Sun 04/11/07
Elaboración de Diagramas de Estado	134	1 day	SFonseca	0%	\$2.50	Mon 05/11/07	Mon 05/11/07
☐ Contestar Cuestionario	131	7 days		0%	\$17.50	Tue 06/11/07	Mon 12/11/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	SFonseca	0%	\$7.50	Tue 06/11/07	Thu 08/11/07
Diagrama de Actividad	137	1 day	SFonseca	0%	\$2.50	Fri 09/11/07	Fri 09/11/07
Elaboración de Diagrama de Secuencia	138	2 days	SFonseca	0%	\$5.00	Sat 10/11/07	Sun 11/11/07
Elaboración de Diagramas de Estado	139	1 day	SFonseca	0%	\$2.50	Mon 12/11/07	Mon 12/11/07

Revisión de los Diagramas de Casos de uso y Actividad jun	120	1 day	SFonseca	0%	\$2.50	Tue 13/11/07	Tue 13/11/07
Correcciones a los Diagramas de Casos de uso y Actividad	141	2 days	SFonseca	0%	\$5.00	Wed 14/11/07	Thu 15/11/07
Aprobación de los Diagramas de Casos de Uso y Actividad	142	1 day	SFonseca	0%	\$2.50	Fri 16/11/07	Fri 16/11/07
☐ Programación de Casos de uso	118	20 days		0%	\$50.00	Sat 17/11/07	Thu 06/12/07
Agregar Cuestionario		5 days	SFonseca	0%	\$12.50	Sat 17/11/07	Wed 21/11/07
Configuración de la evaluación	145	5 days	SFonseca	0%	\$12.50	Thu 22/11/07	Mon 26/11/07
Iniciar Evaluación	146	5 days	SFonseca	0%	\$12.50	Tue 27/11/07	Sat 01/12/07
Contestar Cuestionario	147	5 days	SFonseca	0%	\$12.50	Sun 02/12/07	Thu 06/12/07
Pruebas del módulo	144	5 days	SFonseca	0%	\$12.50	Fri 07/12/07	Tue 11/12/07
Correcciones al módulo	149	3 days	SFonseca	0%	\$7.50	Wed 12/12/07	Fri 14/12/07
Implementación del módulo en producción	150	2 days	SFonseca	0%	\$5.00	Sat 15/12/07	Sun 16/12/07
Proceso de finalización del módulo	151	1 day	SFonseca	0%	\$2.50	Mon 17/12/07	Mon 17/12/07
☐ Módulo de Seguridad	97	75 days		0%	\$187.50	Fri 09/11/07	Tue 22/01/08
Proceso de iniciación del módulo		1 day	Acerrato	0%	\$2.50	Fri 09/11/07	Fri 09/11/07
☐ Recolección de Requerimientos	154	43 days		0%	\$107.50	Sat 10/11/07	Sat 22/12/07
Elaboración de Diagramas de Casos de uso		1 day	Acerrato	0%	\$2.50	Sat 10/11/07	Sat 10/11/07
☐ Elaboración de Casos de uso y Diagramas de Actividad	156	35 days		0%	\$87.50	Sun 11/11/07	Sat 15/12/07
☐ Agregar Departamento		7 days		0%	\$17.50	Sun 11/11/07	Sat 17/11/07
Elaborar Definición de Caso de uso y Flujos de event		3 days	Acerrato	0%	\$7.50	Sun 11/11/07	Tue 13/11/07
Diagrama de Actividad	159	1 day	Acerrato	0%	\$2.50	Wed 14/11/07	Wed 14/11/07
Elaboración de Diagrama de Secuencia	160	2 days	Acerrato	0%	\$5.00	Thu 15/11/07	Fri 16/11/07
Elaboración de Diagramas de Estado	161	1 day	Acerrato	0%	\$2.50	Sat 17/11/07	Sat 17/11/07

☐ Agregar Cargo de Empleado	158	7 days		0%	\$17.50	Sun 18/11/07	Sat 24/11/07
Elaborar Definición de Caso de uso y Flujos de evento		3 days	Acerrato	0%	\$7.50	Sun 18/11/07	Tue 20/11/07
Diagrama de Actividad	164	1 day	Acerrato	0%	\$2.50	Wed 21/11/07	Wed 21/11/07
Elaboración de Diagrama de Secuencia	165	2 days	Acerrato	0%	\$5.00	Thu 22/11/07	Fri 23/11/07
Elaboración de Diagramas de Estado	166	1 day	Acerrato	0%	\$2.50	Sat 24/11/07	Sat 24/11/07
☐ Agregar Empleado	163	7 days		0%	\$17.50	Sun 25/11/07	Sat 01/12/07
Elaborar Definición de Caso de uso y Flujos de evento		3 days	Acerrato	0%	\$7.50	Sun 25/11/07	Tue 27/11/07
Diagrama de Actividad	169	1 day	Acerrato	0%	\$2.50	Wed 28/11/07	Wed 28/11/07
Elaboración de Diagrama de Secuencia	170	2 days	Acerrato	0%	\$5.00	Thu 29/11/07	Fri 30/11/07
Elaboración de Diagramas de Estado	171	1 day	Acerrato	0%	\$2.50	Sat 01/12/07	Sat 01/12/07
☐ Registrar Ingreso al sistema	168	7 days		0%	\$17.50	Sun 02/12/07	Sat 08/12/07
Elaborar Definición de Caso de uso y Flujos de evento		3 days	Acerrato	0%	\$7.50	Sun 02/12/07	Tue 04/12/07
Diagrama de Actividad	174	1 day	Acerrato	0%	\$2.50	Wed 05/12/07	Wed 05/12/07
Elaboración de Diagrama de Secuencia	175	2 days	Acerrato	0%	\$5.00	Thu 06/12/07	Fri 07/12/07
Elaboración de Diagramas de Estado	176	1 day	Acerrato	0%	\$2.50	Sat 08/12/07	Sat 08/12/07
☐ Cambio de Contraseña	173	7 days		0%	\$17.50	Sun 09/12/07	Sat 15/12/07
Elaborar Definición de Caso de uso y Flujos de evento		3 days	Acerrato	0%	\$7.50	Sun 09/12/07	Tue 11/12/07
Diagrama de Actividad	179	1 day	Acerrato	0%	\$2.50	Wed 12/12/07	Wed 12/12/07
Elaboración de Diagrama de Secuencia	180	2 days	Acerrato	0%	\$5.00	Thu 13/12/07	Fri 14/12/07
Elaboración de Diagramas de Estados	181	1 day	Acerrato	0%	\$2.50	Sat 15/12/07	Sat 15/12/07
Revisión de los Diagramas de Casos de uso y Actividad jun	157	1 day	Acerrato	0%	\$2.50	Sun 16/12/07	Sun 16/12/07
Correcciones a los Diagramas de Casos de uso y Actividad	183	5 days	Acerrato	0%	\$12.50	Mon 17/12/07	Fri 21/12/07
Aprobación de los Diagramas de Casos de Uso y Actividad	184	1 day	Acerrato	0%	\$2.50	Sat 22/12/07	Sat 22/12/07

☐ Programación de Casos de uso	155	21 days		0%	\$52.50	Sun 23/12/07	Sat 12/01/08
Agregar Departamento		5 days	Acerrato	0%	\$12.50	Sun 23/12/07	Thu 27/12/07
Agregar Cargo de Empleado	187	5 days	Acerrato	0%	\$12.50	Fri 28/12/07	Tue 01/01/08
Agregar Empleado	188	5 days	Acerrato	0%	\$12.50	Wed 02/01/08	Sun 06/01/08
Registrar Ingreso al sistema	189	3 days	Acerrato	0%	\$7.50	Mon 07/01/08	Wed 09/01/08
Cambio de Contraseña	190	3 days	Acerrato	0%	\$7.50	Thu 10/01/08	Sat 12/01/08
Pruebas del módulo	186	5 days	Acerrato	0%	\$12.50	Sun 13/01/08	Thu 17/01/08
Correcciones al módulo	192	3 days	Acerrato	0%	\$7.50	Fri 18/01/08	Sun 20/01/08
Implementación del módulo en producción	193	1 day	Acerrato	0%	\$2.50	Mon 21/01/08	Mon 21/01/08
Proceso de finalización del módulo	194	1 day	Acerrato	0%	\$2.50	Tue 22/01/08	Tue 22/01/08
☐ Elaboración de reportes	195	20 days		0%	\$50.00	Wed 23/01/08	Mon 11/02/08
Control de la Planificación Anual		5 days	Acerrato	0%	\$12.50	Wed 23/01/08	Sun 27/01/08
Control del Desarrollo profesional	197	5 days	Acerrato	0%	\$12.50	Mon 28/01/08	Fri 01/02/08
Control de Actividades no planificadas	198	5 days	Acerrato	0%	\$12.50	Sat 02/02/08	Wed 06/02/08
Reporte de la Evaluación Anual	199	5 days	Acerrato	0%	\$12.50	Thu 07/02/08	Mon 11/02/08
☐ Segunda Defensa	27	27 days		0%	\$120.00	Tue 12/02/08	Sun 09/03/08
☐ Elaboración de Manuales		21 days		0%	\$105.00	Tue 12/02/08	Mon 03/03/08
Elaboración de Manual de usuario		21 days	Acerrato	0%	\$52.50	Tue 12/02/08	Mon 03/03/08
Elaboración de Manual Técnico		21 days	SFonseca	0%	\$52.50	Tue 12/02/08	Mon 03/03/08
Capacitación de usuarios de la aplicación	202	5 days	Acerrato	0%	\$12.50	Tue 04/03/08	Sat 08/03/08
Finalización del proyecto	205	1 day	SFonseca	0%	\$2.50	Sun 09/03/08	Sun 09/03/08

5.1.9 Control de Calidad del Proyecto

Objetivo

Definir todas las actividades que se realizaran en el Instituto para determinar sus políticas, objetivos y responsabilidades relativos a un control calidad del proyecto.

Desarrollo

En el desarrollo del proyecto se identificaron objetivos de calidad para poder asegurar que el sistema de evaluación se apegara a los requerimientos de los usuarios. Con lo cual se definió:

Requerimientos del proyecto.

1. Planificación anual de tareas y proyectos asignados a cada empleado.
2. Actividades asignadas a cada empleado para su desarrollo profesional.
3. Actividades no planificadas.
4. Evaluación anual de 360°.

Producto final del proyecto.

Implementación del “Sistema de evaluación general de desempeño” lo que permitirá al ITCA tener un control integral del desempeño de cada uno de sus empleados.

Criterio de calidad.

-Funcionalidad del sistema:

Sistema de evaluación general de desempeño probado e instalado.

-Rendimiento del sistema:

Respuesta de sistema

Disponibilidad del sistema

Migración de la información

Standard de calidad.

-Funcionalidad del sistema

Sistema de evaluación general de desempeño operacional sin errores.

-Rendimiento del sistema

99.9% tiempo del sistema disponible

Menor tiempo de respuesta

100% de calidad de la información

5.1.10 Recursos Humanos del Proyecto**Objetivo**

Definir las asignaciones de los miembros que conforman el equipo del proyecto y sus responsabilidades que deben cumplir.

Desarrollo

Planificación de los recursos humanos.

Desglose del trabajo:

Gerente-administrador del proyecto: Carlos Humberto López.

Miembros del proyecto: Salvador Fonseca/Arturo Cerrato

Ejecutante: empleados del ITCA.

Usuarios: superiores, subalternos, compañeros de área y otras personas vinculadas a su trabajo.

Desarrollar el equipo del proyecto.

Actividades:

- Se sostendrán reuniones con el personal designados por el ITCA para la recolección de requerimientos.
- Con personal asignados por el Instituto se realizaran pruebas de los programas. Si se encontrase discrepancia se procederá a realizar los cambios necesarios.
- El personal del ITCA realizara una revisión final de los módulos, de no encontrar discrepancias se procederá a la aceptación de este finalizado el sistema.
- Tiempo estimado de utilización de recurso humano dura su ejecución.

Recurso	jul-07	ago-07	sep-07	oct-07	nov-07	dic-07	ene-08	feb-08	mar-08	Total Horas
SFonseca	150h	155h	150h	155h	150h	85h	0h	90h	20h	955 hrs
Acerrato	150h	110h	75h	155h	150h	155h	155h	145h	40h	1,135 hrs
ITCA	15h	5h	0h	0h	0h	0h	0h	0h	0h	20 hrs
Clópez	10h	15h	15h	10h	10h	10h	10h	10h	5h	95 hrs

5.1.11 Planificación de las Comunicaciones

Objetivo

Determinar las necesidades de información y comunicaciones de los interesados en el proyecto, por ejemplo, quién necesita qué información, cuándo la necesita, cómo le será suministrada y por quién.

Desarrollo

El desarrollo de cada módulo del sistema estará dividido en subetapas las cuales serán:

1. Recolección de requerimientos
2. Programación
3. Pruebas
4. Correcciones
5. Implementación
6. Finalización

Cada una de estas sub-etapas tendrá productos entregables, entre los cuales se encontraran documentos impresos de aceptación de la culminación de cada etapa.

1. Recolección de requerimientos

Nombre del Documento: Requerimientos del módulo

Objetivo del Documento: Recopilar la información obtenida durante la recolección de requerimientos. Este documento servirá de base para el desarrollo del sistema, por lo que deberá contar con la aprobación del Jefe de desarrollo y del gerente de IT de ITCA, quienes firmarán el documento como muestra de su conformidad, todo requerimiento posterior a este documento, será discutido con el director del proyecto para determinar si se procederá a implementarse el cambio o se dejara para una versión posterior.

Responsable del documento: Programadores asignados al proyecto, quienes levantarán los requerimientos del sistema, a través de entrevistas con el personal del ITCA.

Interesados: Gerente de IT y jefe de Desarrollo del ITCA.

Formato de entrega: Documento impreso.

Fecha de elaboración: Este documento se elaborara al inicio del desarrollo de cada módulo. Su elaboración será posterior al levantamiento de requerimientos y anterior al desarrollo del módulo.

2. Programación

Nombre del Documento: Acta de aprobación del módulo

Objetivo del Documento: Dar por finalizado el desarrollo de un módulo en particular. Este documento será firmado por el jefe de desarrollo y el gerente de IT como muestra de su conformidad con el funcionamiento del módulo en particular.

Responsable del documento: Director del Proyecto.

Interesados: Jefe de desarrollo y Gerente de IT.

Formato de entrega: Documento impreso.

Fecha de elaboración: Este documento se elaborará al finalizar el desarrollo de un módulo.

3. Pruebas

Nombre del Documento: Resultado de pruebas al módulo

Objetivo del Documento: Recopilar las observaciones hechas al módulo durante las pruebas. Puede documentar mal funcionamiento en el sistema, carencia de funcionalidad solicitada en los requerimientos, requerimientos nuevos (sujetos a discusión y aprobación del director del proyecto).

Responsable del documento: Jefe de desarrollo

Interesados: Programadores.

Formato de entrega: Documento impreso.

Fecha de elaboración: Posterior a la realización de pruebas por parte del personal del ITCA.

4. Implementación

Nombre del Documento: Manuales de usuario y programador.

Objetivo del Documento: Documentar el funcionamiento del sistema desde la perspectiva del usuario y del programador,

Responsable del documento: Programadores asignados al proyecto.

Interesados: Jefe de desarrollo.

Formato de entrega: Documentos impresos

Fecha de elaboración: al finalizar el proyecto

5. Finalización

Nombre del Documento: Acta de finalización del proyecto

Objetivo del Documento: Finalizar formalmente el proyecto. Esta acta será firmada por el gerente de IT y el director del proyecto.

Responsable del documento: Director del proyecto.

Interesados: Gerente de IT.

Formato de entrega: Documento impreso.

Fecha de elaboración: Al finalizar el proyecto.

5.1.12 Riesgos del Proyecto

Objetivo

Definir los procesos relacionados con la planificación, identificación y análisis de riesgo, así como a las respuestas a los riesgos durante todo el proceso del proyecto.

Desarrollo

En la planificación de la gestión de riesgos en el desarrollo del sistema se decidió identificar, planificar y ejecutar actividades de aquellos riesgos que pudieran afectar alcanzar los objetivos del proyecto.

Identificación de riesgos.

Se elaboro un sistema de puntuación utilizado para medir la probabilidad de cada riesgo ocurra y su impacto de este. Como se puede ver en la tabla 1.1

Id	Riesgo	Probabilidad	Impacto	Prioridad
1	Insuficiente información por parte del ITCA	2	3	3.5
2	Falta de colaboración de parte del ITCA	2	3	3.5
3	Falta de fondos por parte del ITCA	2	3	3.5
4	Falta de fondos por parte del equipo de tesis	3	5	5.5
5	Enfermedad	2	4	4
6	Poca experiencia desarrollando aplicaciones JAVA	4	5	6.5

Tabla 1.1

Riesgos.

Se definieron algunos riesgos potenciales. Como lo describe en la tabla 1.2

Categoría de riesgo	Descripción del riesgo	ID Riesgo
Requerimientos	Insuficiente información por parte del ITCA	1
	Falta de colaboración de parte del ITCA	2
	Falta de fondos por parte del ITCA	3
	Falta fondos por parte del equipo de tesis	4
	Enfermedad	5
	Poca experiencia desarrollando aplicaciones en JAVA	6

Tabla 1.2

Análisis cualitativo y cuantitativo de riesgos.

Probabilidad.

Se midió la probabilidad de cada riesgo ocurra, como se expone en la tabla 1.3

Titulo	Puntuación	Descripción
Muy bajo	1	Altamente probable que no ocurra basado en la información periódica, las circunstancias y demás variables a considerar
Bajo	2	Probable que no ocurra, pero debe ser monitoreado para verificar las circunstancias que puede repercutir que durante el proyecto ocurra
Medio	3	Puede ocurrir y es claro que el riesgo es eventualmente probable
Alto	4	Muy probable que ocurra, basado en las circunstancias del proyecto
Muy Alto	5	Altamente probable que ocurra, se debe poner mucha atención en estos puesto que pueden afectar al proyecto grandemente,

Tabla 1.3

Impacto.

Se utilizo una puntuación para el impacto que el riesgo tendrá si llegase a ocurrir, como se muestra en la tabla 1.4

Titulo	Puntuación	Descripción
Muy bajo	1	Insignificante impacto en el proyecto. Es posible que la medición del impacto sea mínima
Bajo	2	Menor impacto en el proyecto.
Medio	3	Impacto medible en el proyecto
Alto	4	Impacto significativo en el proyecto
Muy Alto	5	Mayor impacto en el proyecto

Tabla 1.4

Prioridad.

Se utilizo como puntuación un promedio entre la probabilidad y el impacto de la siguiente forma, como se muestra en la tabla 1.5.

$$\text{Prioridad} = [\text{probabilidad} + \text{impacto}] / 2$$

ID	Probabilidad	Impacto	Prioridad	Ranking
4	3	5	5.5	Medio
6	4	5	6.5	Alto
2	2	3	3.5	Bajo

Tabla 1.5

Seguimiento y control de riesgos.

Y por último se creó acciones preventivas requeridas para reducir las probabilidades que ocurra, acciones de contingencia para reducir el impacto que tenga en el proyecto si este ocurre.

5.1.13 ACTA DE APROBACIÓN DEL PROYECTO

Por este medio hacemos constar que las todas pruebas realizadas al “Sistema de Evaluación General de Desempeño”, han cumplido satisfactoriamente todos los requisitos solicitados en la recolección de requerimientos. Por lo tanto, aceptamos la entrega de este sistema por parte del Sr. Arturo Cerrato Director del proyecto de desarrollo y damos por finalizado el desarrollo de este sistema.

Sr. Arturo Cerrato

Ing. Carlos López

San Salvador, 23 de Febrero de 2008

5.1.14 Estándares de Programación y Diseño

Introducción

Un aspecto muy importante en el desarrollo de software es el Estándar de Programación. Un Estándar de Programación define un conjunto de reglas de notación y nomenclatura en el estilo de codificación, dicho Estándar ayuda a mantener un código ordenado y, a su vez, permite un fácil mantenimiento.

No existe una terminología o estilo mejor que otro. La valoración de dichas terminologías no se basa en que al programador le guste, sino primordialmente en el uso adecuado de una terminología específica. Cabe recalcar que un buen programador debe estar en la capacidad de adaptarse a cualquier Estándar de Programación que establezca el equipo de desarrollo al que pertenece.

Un Estándar de Desarrollo debe incluir:

- **Factor Nemotécnico**

El Factor Nemotécnico consiste en que el programador pueda recordar fácilmente el nombre de un objeto.

- **Factor Sugestivo**

El Factor Sugestivo permite a otros programadores poder entender fácilmente el código fuente.

- **Consistencia**

La Consistencia radica en usar las mismas convenciones de nomenclatura en todo el programa, para lograr que el texto del código sea legible.

5.1.14.1 Objetivos

- Definir un conjunto de reglas de notación y nomenclatura para la creación del código fuente, que permitan el desarrollo de un código legible y de fácil mantenimiento.
- Establecer lineamientos para el diseño de la interfaz gráfica de usuario.
- Estandarizar el diseño de Base de Datos.

5.1.14.2 Alcance

Establecimiento de reglas de nomenclatura y reglas que garanticen el orden y la legibilidad del código escrito. Así como el establecimiento de lineamientos para el diseño de interfaces gráficas y diseño de la base de datos.

El presente documento cubre las siguientes áreas:

- **Reglas de legibilidad del código**
Como indentar el código por cada bloque o estructura que se emplea.
- **Reglas de documentación**
Como establecer comentarios y ayuda del sistema.
- **Reglas de nomenclatura**
Cómo nombrar constantes, variables, objetos, procedimientos, funciones, métodos, tablas, campos y archivos.
- **Reglas de Organización de Archivos**
Cómo organizar el árbol de directorios.
- **Reglas de diseño de reportes**
Formato general de reportes.

- **Reglas de diseño de interfaz gráfica**

Lineamientos de diseño de interfaz gráfica de usuario.

- **Reglas de diseño de Base de Datos**

Lineamientos para nombrar tablas, llaves y campos.

5.1.14.3 Organización de Código y Estilo

Debido a que un programador pasa la mayor parte del tiempo escribiendo, leyendo y revisando el código fuente, debemos poner especial énfasis en la legibilidad de éste. Podemos hacer éste trabajo mucho más fácil siguiendo las convenciones de nomenclatura de los estándares y a la vez haciendo nuestro código legible y bien documentado.

5.1.14.3.1 Legibilidad del código

Para mayor claridad en la lectura, se utilizará un espacio en blanco entre operaciones, parámetros, asignaciones y condicionales.

Ejemplo:

```
//USO INCORRECTO
public int intTotalFrutas(int intDocenasManzanas,int intDocenasPeras){
    intTotalFrutas=(intDocenasManzanas+intDocenasPeras)*12;
    return intTotalFrutas;
}

//USO CORRECTO
public int intTotalFrutas(int intDocenasManzanas, int intDocenasPeras){
    intTotalFrutas = (intDocenasManzanas + intDocenasPeras) * 12;
    return intTotalFrutas;
}
```

5.1.14.3.1.1 Longitud de línea

Con la finalidad de una mejor comprensión del código y una lectura más ágil se espera que las líneas de código no superen los 80 caracteres. Si una línea supera este tamaño, se deberá cortar en base a los siguientes criterios:

- Cortar antes de una coma.
- Cortar antes de un operador.
- Alinear la nueva línea con el principio de la línea anterior.

5.1.14.3.1.2 Identado en el código fuente

Para evitar la disparidad en la cantidad de espacios de indentación, se utilizarán tabuladores.

Los bloques de código estarán indentados en una posición con respecto a la sentencia que lo utilice, de acuerdo al siguiente ejemplo:

```
While(condición){  
    if (condición){  
        Bloque de código  
    }  
}
```

5.1.14.3.1.3 Llaves de inicio y cierre de bloque

- La llave de apertura '{', irá en el mismo renglón de la sentencia que requiere el bloque de código.
- La llave de cierre irá en una línea independiente.
- La utilización de llaves es recomendada aún cuando el bloque de código consista solo de una línea.

5.1.14.3.2 Declaraciones

5.1.14.3.2.1 Declaración de variables

Con respecto a la declaración de variables debemos tener en cuenta lo siguiente:

- Se recomienda la declaración de una variable por reglón.
- La declaración de variables debe hacerse al inicio del bloque de código.
- Preferiblemente las variables se inicializarán en la misma línea de declaración.

Ejemplo:

```
Int intIdProducto = 7;
String strConexion;
String strSQL;

strConexion = "jdbc:sqlserver://localhost:1433;" +
              "databaseName=SEP;user=sa;password=123;";
strSQL = "select * from producto where idProducto = " + intProducto;
```

5.1.14.3.2.2 Declaración de clases

En la definición de clases se usarán las siguientes reglas:

- No se usará ningún espacio entre el nombre del método y el paréntesis de abertura de la lista de parámetros.

5.1.14.4 Comentarios

Los programas deberán estar debidamente documentados con comentarios que expliquen de manera clara cada proceso dentro de la codificación. Se sugiere que los comentarios se escriban en letra mayúscula.

Ejemplo:

```
//FORMA INCORRECTA
public double calcularSalario(double dblSB,dblBonif){
double dblDscto;
if (dblSB>700){
dblDscto=dblSB*0.1;
```

```

}
return dblSB+dblBonif-dblDscto;
}

//FORMA CORRECTA
public double calcularSalario(double dblSalarioBase, dblBonificacion){
    //RECIBIMOS COMO PARÁMETROS EL SUELDO BASE Y SUS BONIFICACIONES
    double dblDescuento;
    dblDescuento = 0;

    //SI ESL SUELDO BASE PASA DE 700 SE LE HACE UNA RETENCIÓN DEL 10%
    if (dblSalarioBase > 700){
        dblDescuento = dblSalarioBase * 0.1;
    }

    //FINALMENTE SE RETORNA EL SALARIO CON EL SIGUIENTE CÁLCULO
    return dblSalarioBase + dblBonificacion - dblDescuento;
}

```

5.1.14.4.1 *Encabezados*

El siguiente Estándar ha sido establecido como encabezado para la documentación de clases, procedimientos, programas, métodos y funciones:

```

/*
*EL PRIMER PÁRRAFO RESUME EL PROPÓSITO DEL PROGRAMA
*
*EL SEGUNDO PÁRRAFO CONTIENE DETALLES ESPECÍFICOS DE LA OPERACIÓN
*DEL PROGRAMA, Y CUALQUIER DEPENDENCIA O PROTOCOLO ESPECIAL,
*QUE LOS DESARROLLADORES DEBERÍAN CONOCER AL USAR EL PROGRAMA.
*
*VARIABLES DE ENTRADA
*VARIABLE 1: RESUME LA UTILIZACIÓN DE LAS VARIABLES DE ENTRADA.
*VARIABLE 2: RESUME LA UTILIZACIÓN DE LAS VARIABLES DE ENTRADA.
*
*VALOR DE SALIDA
*DESCRIBE EL TIPO DE VALOR QUE DEVUELVE EL PROGRAMA Y LOS VALORES POSIBLES.
*
*AUTOR: NOMBRE DEL AUTOR
*FECHA DE CREACIÓN: YYYY-MM-DD
*
*REVISIÓN 1
*AUTOR: NOMBRE DEL AUTOR

```

```

*MOTIVO: MOTIVO DE LA MODIFICACIÓN
*FECHA: YYYY-MM-DD
*
*REVISIÓN 2
*AUTOR: NOMBRE DEL AUTOR
*MOTIVO: MOTIVO DE LA MODIFICACIÓN
*FECHA: YYYY-MM-DD
*/

```

5.1.14.4.2 *Descripción de variables*

Cada declaración de variables, se documentara con una breve descripción de la finalidad de dicha variable.

Ejemplo:

```

String strCadenaConexion //CONTIENE LA CADENA DE CONEXIÓN.
String strSQL;           //CONTIENE UNA EXPRESIÓN SQL.

```

5.1.14.4.3 *Casos especiales*

En todos aquellos casos en que sea necesario modificar el código de un programa se agregará el siguiente comentario junto a la modificación realizada:

```

/*
*REVISIÓN (NÚMERO DE REVISIÓN)
*DESCRIPCIÓN DE LA MODIFICACIÓN
*/

```

Nota: En aquellos casos que se considere necesario, se podrá comentar el código obsoleto, y dejarlo como referencia para revisiones futuras.

5.1.14.5 *Nomenclatura*

5.1.14.5.1 *Nombramiento de elementos de programación*

Para asignar nombres de variables, clases, procedimientos, programas, métodos y funciones, se utilizará una combinación de notación Húngara y Notación Pascal CasIng. De la siguiente forma:

- Al nombre se antepondrá un prefijo que indique el ámbito y el tipo de objeto que define, en el caso de variables el tipo de dato que almacena.
- Los nombres comenzarán con letra minúscula.
- Si el nombre está compuesto por múltiples palabras, estas se escribirán juntas, no tendrán ningún carácter de separación y solo la letra inicial de cada palabra será mayúscula.
- A las variables globales se les antepondrá la letra 'g'.
- A los parámetros que se pasen por referencia se les antepondrá el prefijo 'pr'.
- A las constantes se les antepondrá el prefijo 'cns'.
- A los nombres de programas se les podrán añadir al final del prefijo tres letras que distingan únicamente al modulo al que pertenecen.
- La letra 'ñ' será sustituida por las letras 'ni'.
- Las vocales con tilde, deberán ser sustituidas por la vocal sin tilde equivalente.
- Los nombres de constantes se escribirán en notación Versal, (notación Versal es aquella en la que todo se escribe en mayúsculas).

Los nombres deberán ser lo suficientemente descriptivos para facilitar su manejo y se recomienda evitar el uso de abreviaturas.

Debe evitarse el uso de palabras que no brinden información (como: de, del, la, el).

Por ejemplo:

```
//FORMA INCORRECTA
String strNombreDeUsuario;

//FORMA CORRECTA
Strint strNombreUsuario;
```

5.1.14.5.2 *Prefijos*

5.1.14.5.2.1 **Prefijos para nombres de variables**

La siguiente tabla describe los prefijos para los nombres de variables:

Tipo de dato	Prefijo
Integer	int
Byte	
Tiny	
Int	
Short	
Long	
Boolean	bln
String	str
Double	dbl
Datetime	dt
Char	chr
Varchar	vch
Numeric	num
Object	Obj

5.1.14.5.2.2 Prefijos para nombres de controles

La siguiente tabla describe los prefijos para los nombres de controles:

Control	Prefijo
TextBox TextArea	txt
DropDownList ComboBox	cbo
ListBox	lst
CheckBox	chk
Option Radio button	rbt
Label	lbl
Button Submit	btn

5.1.14.6 Organización de archivos

5.1.14.6.1 Archivos *que contengan clases*

Cada clase debe estar en uno solo archivo, y un archivo no puede contener más de una clase.

5.1.14.6.2 Estructura de directorios

Todo grupo de archivos que estén relacionados o guarden alguna semejanza deberán guardarse en un directorio aparte bajo el directorio raíz.

Ejemplo:

- Formularios
 - Mantenimientos
- Reportes
- Clases

5.1.14.6.3 Nombramiento de archivos

Los nombres de archivos tendrán los siguientes prefijos:

Tipo de Archivo	Prefijo
Mantenimientos	frmMto
Procesos	frmPro
Reportes	frmRpt/frmVRp
Utilerías	frmUtl
Consultas	frmCon
Transacciones	frmTrn
Auxiliares	frmAMt
Seguridad	frmSec
Reportes	rpt

5.1.14.8 Diseño de Base de Datos

5.1.14.8.1 Reglas Generales

Los nombres de tablas y campos se deben escribir en notación lowerCamelCase, de la siguiente forma:

- La primera letra del nombre se escribe en minúscula.
- No se permiten espacios en blanco, caracteres de puntuación, caracteres especiales ni símbolos.
- Si el nombre está compuesto por más de una palabra, cada palabra iniciara en mayúscula, se escribirán juntas y no se separaran por espacios en blanco o guión bajo.
- Únicamente se utilizarán caracteres alfabéticos, salvo que por la naturaleza del nombre se necesiten dígitos numéricos
- Las vocales acentuadas se reemplazarán por sus equivalentes no acentuadas.
- La letra 'ñ' se reemplazara por 'ni'.
- El nombre debe ser lo más descriptivo posible, evitando términos ambiguos que se presten a distintas interpretaciones.
- Es preferible no utilizar abreviaciones.

5.1.14.8.2 Nombres de Tablas

- Los nombres deben especificarse en singular.
- En el caso de tablas que se relacionan específicamente con otra tabla, esta relación debe quedar expresada en el nombre, ejemplo: domicilioPersona, categoriaMunicipio.

5.1.14.8.3 Nombres de Campos

- Se recomienda describir cada campo (en el espacio suministrado por los administradores de Bases de Datos) con una breve descripción de su función dentro del programa, valor que almacenará y su significado dentro de la base de datos.

5.1.14.8.4 Campos Llave

- Toda tabla debe poseer siempre por lo menos un campo como llave primaria.
- Toda relación entre las tablas debe implementarse mediante constraints con integridad referencial.
- Los campos que conforman la llave primaria deben colocarse al inicio de la definición de la tabla, es decir deben ser los primeros.
- Si la llave primaria esta constituida solamente por un campo, el nombre debe estar compuesto por “id” + nombre de la tabla en singular.
- Las llaves foráneas conservan el mismo nombre del campo al cual hacen referencia en la tabla relacionada.

5.1.14.8.5 Otros Campos

- Todo lo que represente un nombre o descripción, se colocará inmediatamente después de los campos clave, y se nombrará igual que la tabla a la que pertenece.
- Los campos boléanos deberán nombrarse de acuerdo al estado correspondiente al valor 1/Verdadero/True del contexto en el que se encuentran modelados. Ejemplo: autorizado, vigente.

5.1.14.9 Documentación

5.1.14.9.1 Ayuda del sistema

Todo programa deberá contar con su propio manual de usuario, el cual estará disponible en papel y electrónicamente a través del mismo sistema.

5.1.14.10 Manejo de Errores y Excepciones

Es importante que siempre se verifique la existencia de un estado de error a través del manejo de excepciones, esto significa que si un método lanza una excepción,

ésta debe ser atrapada. Debido a que una excepción es un objeto, se puede almacenar más información que en una simple variable de estatus y nos permite tomar una mejor decisión.

Ejemplo:

```
try{
    //BLOQUE DE CÓDIGO
}
catch (...) {
    //INSTRUCCIONES QUE SE EJECUTAN EN CASO
    //DE QUE UN ERROR SEA ATRAPADO.
}
finally{    //SOLO SI ES NECESARIO
    //ESTE CÓDIGO SE EJECUTA SIEMPRE,
    //YA SEA QUE OCURRA UNA EXCEPCIÓN O NO.
    //SU UTILIDAD NO ES MUCHA.
    //YA QUE ESTAS INSTRUCCIONES PUEDEN COLOCARSE FUERA DEL BLOQUE.
}
```

5.1.14.10.1 Buenas Prácticas

- No se deben usar las excepciones para controlar el flujo normal.
- No se deben usar excepciones para indicar condiciones anormales.
- Si existe información de contexto suficiente, siempre se debe manejarse el error. Caso contrario debe propagarse siguiendo la jerarquía de llamados hacia niveles superiores.
- Se deben manejar grupos de sentencias (no encerrar cada sentencia en un bloque de try/catch).
- Usar excepciones en constructores.
- Un bloque finally no debe contener sentencias de: break, continue, o return.
- Mantener un esquema de excepciones simple.
- Generar un sistema más seguro invirtiendo tiempo en debug.
- Las excepciones en el catch deben ir de lo más específico a lo más general.

- Siempre enviar mensajes de error al usuario, pero transformar la excepción a un lenguaje entendible para el usuario

5.1.14.11 Conclusión

La esencia de los estándares de programación es mantener la consistencia del código siguiendo una determinada convención de nombres y estilo. Esto ayuda a escribir, mantener y reusar el código en una forma más eficaz y eficiente. Entender código elaborado por otra persona fomentando el trabajo en equipo

Se puede ahorrar segundos prescindiendo de los estándares, pero se perderá mucho más tiempo si los programadores persisten en programar sin un orden establecido.

5.1.15 Manual de Políticas

Objetivo de este documento

Recopilar las buenas prácticas a seguir dentro del desarrollo del proyecto.

Desarrollo

5.1.15.1 Plan de Reuniones

Se programarán reuniones quincenales para evaluar el desarrollo del proyecto. Estas reuniones se realizarán junto con el Director del proyecto, los programadores asignados al proyecto, el gerente de IT y el jefe de desarrollo.

También se realizaran reuniones al inicio del desarrollo de cada módulo con el personal asignado por el ITCA con el fin de recolectar todos los requerimientos del sistema.

Se realizará una reunión durante cada entrega formal de un módulo, con el fin que cada una de las partes formalicen su conformidad en un documento escrito.

5.1.15.2 Estándar de programación

Durante el desarrollo del sistema, los programadores se apegarán a un conjunto de normas recopiladas en el documento llamado: “Estándares de Programación”. Esto se hará con el fin de hacer más ordenado el trabajo de los programadores y para que el posterior mantenimiento y actualización del sistema sea lo más sencillo posible.

5.1.15.3 Copias de respaldo

Durante el desarrollo del sistema, se realizarán copias de seguridad en CD, una vez por semana. Los encargados de realizar estas copias de seguridad serán los programadores asignados al proyecto.

5.1.15.4 Prevención y recuperación de desastres

Con el fin de prevenir posibles desastres y/ o recuperarse de ellos, se seguirán las siguientes normas.

5.1.15.4.1 Prevención

5.1.15.4.1.1 Mantenimiento del equipo

Con el fin de mantener en buen estado el equipo, se realizará mensualmente en cada equipo:

1. Limpieza física del equipo.
2. Desfragmentación del Disco duro.
3. Eliminación de archivos temporales.
4. Actualización de antivirus.
5. Instalación de actualizaciones, service Paks y parches, a los programas que lo requieran.

5.1.15.4.1.2 Antivirus

A cada máquina se le instalará: “Norton Antivirus”, como sistema de protección contra el ataque de virus.

Norton Antivirus se configurará para que descargue las actualizaciones de definiciones de antivirus, una vez por semana.

5.1.15.4.1.3 Copias de respaldo

Durante el desarrollo del sistema, se realizarán copias de seguridad en CD, una vez por semana. Los encargados de realizar estas copias de seguridad serán los programadores asignados al proyecto.

5.1.15.4.2 *Recuperación*

5.1.15.4.2.1 Pérdida de información

En el caso que exista pérdida de información, ya sea por daño del equipo, extravío, robo, desastres naturales, negligencia, etc. Se procederá a instalar la copia de respaldo más reciente.

5.1.15.4.2.2 Pérdida o daño de equipo

En el caso que parte o la totalidad del equipo sea extraviado, o dañado, tendrá que procederse a la compra de nuevo equipo, durante el proceso de compra, se contara con apoyo del ITCA quienes suministrarán equipo de cómputo temporal que podrá ser usado dentro de sus instalaciones.

5.1.16Requerimientos del sistema

Objetivo

Describir el sistema desde el punto de vista del cliente, usuario y programador, para obtener un claro entendimiento de este.

5.1.16.1 Perspectiva del cliente/Usuario

Se busca comprender el sistema desde el punto de vista del cliente y usuario considerando como interactúan con el sistema para obtener el beneficio esperado. No se deben tomar en cuenta implicaciones tecnológicas o de diseño.

5.1.16.1.1 Actores

5.1.16.1.1.1 Empleados

5.1.16.1.1.2 Supervisores

Un supervisor es un empleado especializado, esto quiere decir que además de aquellas características propias de un supervisor tiene todas las características de un empleado.

5.1.16.1.2 Módulo de planificación anual

5.1.16.1.2.1 Diagrama general de casos de uso

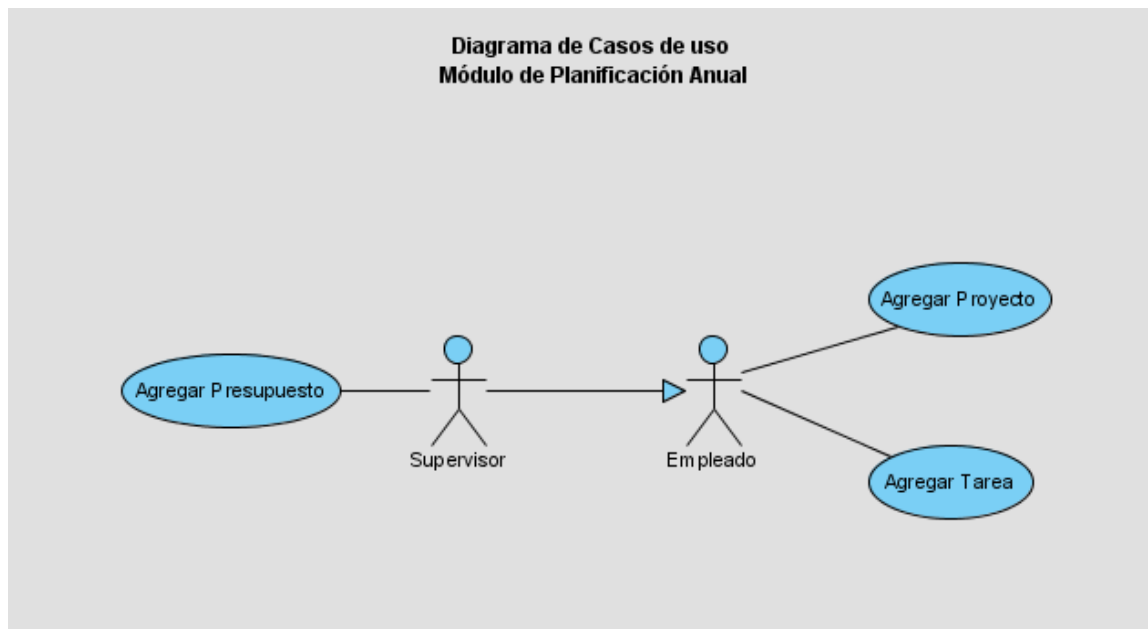


Diagrama general de caso de uso para el Módulo de planificación anual.

5.1.16.1.2.2 Casos de uso

5.1.16.1.2.2.1 Agregar proyecto

Nombre del caso de uso: Agregar proyecto.

Descripción: El empleado utiliza el caso de uso Agregar proyecto para registrar los proyectos que le han sido asignados. Una vez el proyecto ha sido registrado, será visible para el empleado y sus superiores y podrán agregársele las tareas respectivas.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.

Restricciones de la implementación: El empleado puede agregar un proyecto desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado agrega al sistema un proyecto.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado agrega el proyecto nuevo.
3. El nuevo proyecto es mostrado en la lista de tareas por proyectos.

Flujo alternativo de eventos: El supervisor agrega al sistema el proyecto de un empleado.

1. Se muestra una lista de empleados ordenados alfabéticamente.
2. El supervisor selecciona un empleado y ve una lista de los proyectos asignados al empleado.
3. El supervisor agrega el proyecto nuevo.
4. El nuevo proyecto es mostrado en la lista de tareas por proyectos.

Flujo alternativo de eventos: El empleado modifica el nombre de un proyecto.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado modifica el proyecto seleccionado.

4. La información es actualizada y la lista de tareas es renovada.

Flujo alterno de eventos: Tarea duplicada.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado agrega el proyecto nuevo. El proyecto ya existe.
3. El sistema informa al empleado que el proyecto ya existe. La lista de tareas no sufre cambios.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar el proyecto, debido a problemas en el sistema o en las comunicaciones.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado agrega el proyecto nuevo. El sistema es incapaz de agregar el proyecto, debido a problemas en el sistema o en las comunicaciones.
3. El sistema informa del error al empleado, con el mayor nivel de detalle posible. La lista de tareas no sufre cambios.
4. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

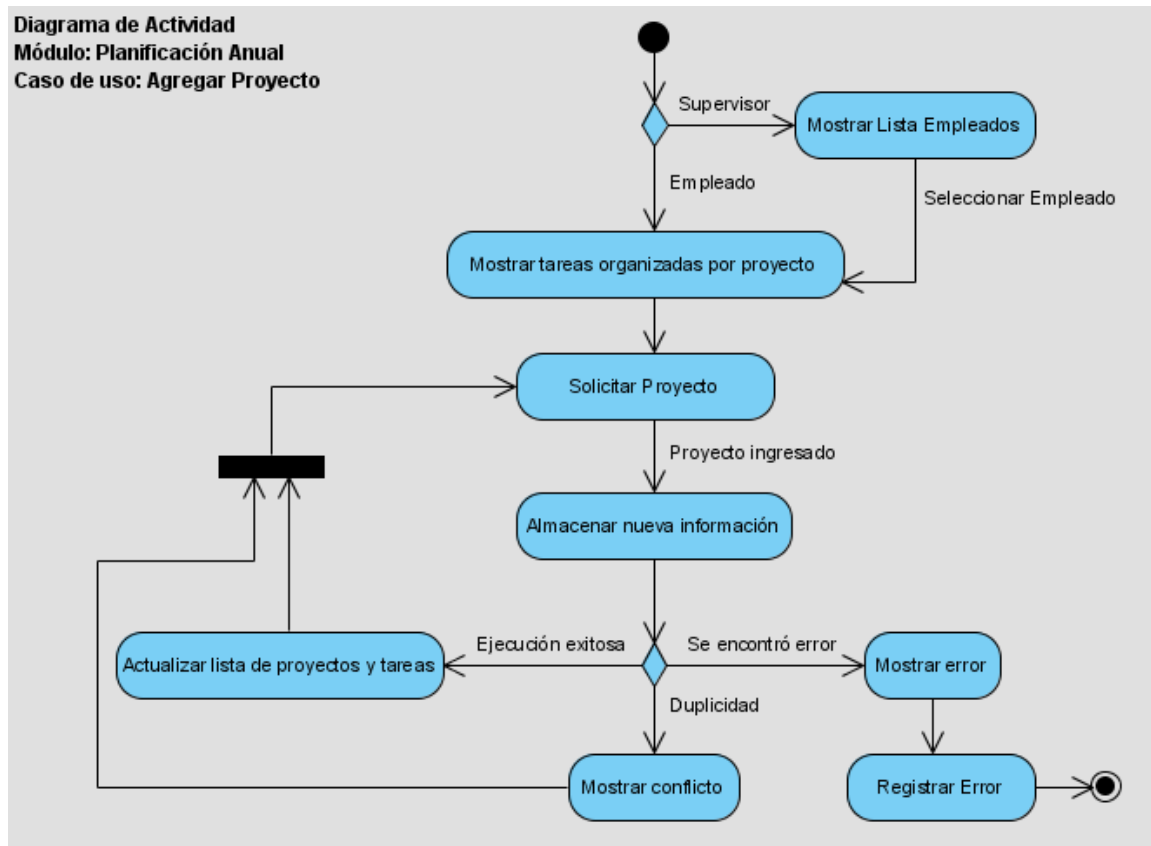


Diagrama de actividad para el caso de uso: Agregar Proyecto.

5.1.16.1.2.2.2 Agregar tarea

Nombre del caso de uso: Agregar tarea.

Descripción: El empleado utiliza el caso de uso Agregar tarea para registrar todas las actividades que le han sido asignadas por proyecto y programadas en su planificación anual. Una vez la actividad ha sido ingresada, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Restricciones de la implementación: El empleado puede agregar una tarea desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado agrega al sistema una actividad.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado agrega una tarea en el proyecto seleccionado para su planificación anual. La nueva tarea es mostrada en la lista de tareas por proyectos.

Flujo alterno de eventos: El supervisor agrega al sistema la tarea de un empleado.

1. Se muestra una lista de empleados ordenados alfabéticamente.
2. El supervisor selecciona un empleado y ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
3. El supervisor selecciona un proyecto.
4. El supervisor agrega una tarea en el proyecto seleccionado para la planificación anual del empleado. La nueva tarea es mostrada en la lista de tareas por proyectos.

Flujo alterno de eventos: El empleado modifica una tarea.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.

2. El empleado selecciona un proyecto.
3. El empleado selecciona una tarea existente.
4. El empleado modifica una tarea en el proyecto.
5. La información es actualizada y la lista de tareas es renovada.

Flujo alterno de eventos: Tarea duplicada.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado agrega una tarea en el proyecto seleccionado para su planificación anual. La tarea ya existe en el proyecto.
4. El sistema informa al empleado que la tarea ya existe en el proyecto. La lista de tareas no sufre cambios.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar la tarea, debido a problemas en el sistema o en las comunicaciones.

1. El empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado agrega una tarea en el proyecto seleccionado para su planificación anual. El sistema es incapaz de agregar la tarea, debido a problemas en el sistema o en las comunicaciones.
4. El sistema informa del error al empleado, con el mayor nivel de detalle posible. La lista de tareas no sufre cambios.
5. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

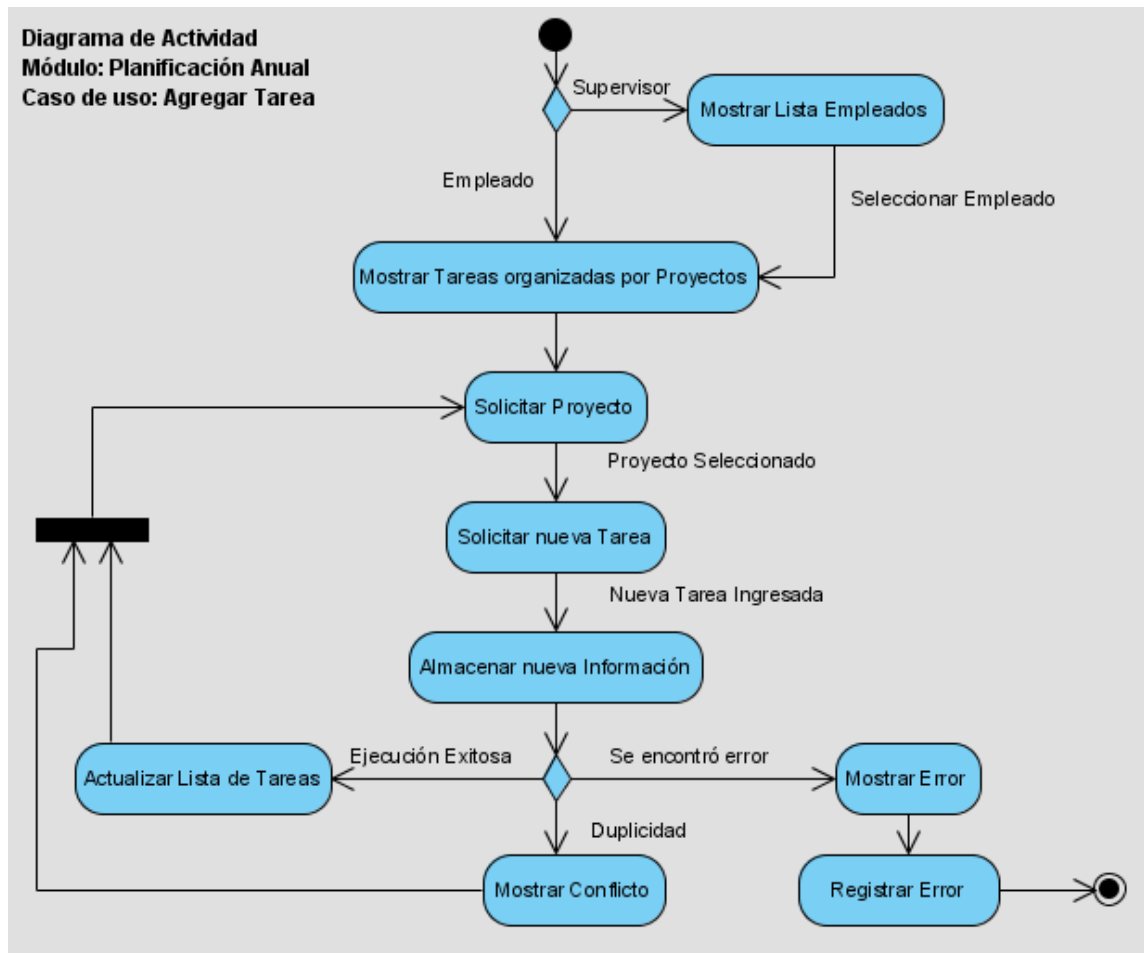


Diagrama de actividad para el caso de uso: Agregar Tarea.

5.1.16.1.2.2.3 Agregar presupuesto

Nombre del caso de uso: Agregar presupuesto.

Descripción: El empleado utiliza el caso de uso Agregar Presupuesto para registrar todos los gastos presupuestados de un proyecto dentro de su planificación anual. Una vez el gasto ha sido ingresado, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Restricciones de la implementación: El empleado puede agregar el presupuesto desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado agrega al sistema un gasto presupuestado en un proyecto.

1. El empleado ve una lista de sus gastos presupuestados asignados por proyecto.
2. El empleado selecciona un proyecto.
3. El empleado agrega un gasto en el presupuesto de un proyecto dentro de su planificación anual. El nuevo gasto es mostrado en la lista del presupuesto.

Flujo alterno de eventos: El empleado modifica un gasto.

1. El empleado ve una lista de sus gastos presupuestados asignados por proyecto.
2. El empleado selecciona un proyecto.
3. El empleado selecciona un gasto del presupuesto.
4. El empleado modifica el gasto presupuestado.
5. La información es actualizada y la lista de gastos del presupuesto es renovada.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar el gasto en el presupuesto, debido a problemas en el sistema o en las comunicaciones.

1. El empleado ve una lista de sus gastos presupuestados asignados por proyecto.
2. El empleado selecciona un proyecto.
3. El empleado agrega un gasto en el presupuesto de un proyecto dentro de su planificación anual El sistema es incapaz de agregar el gasto dentro del presupuesto, debido a problemas en el sistema o en las comunicaciones.
4. El sistema informa del error al empleado, con el mayor nivel de detalle posible.
La lista de gastos del presupuesto no sufre cambios.
5. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

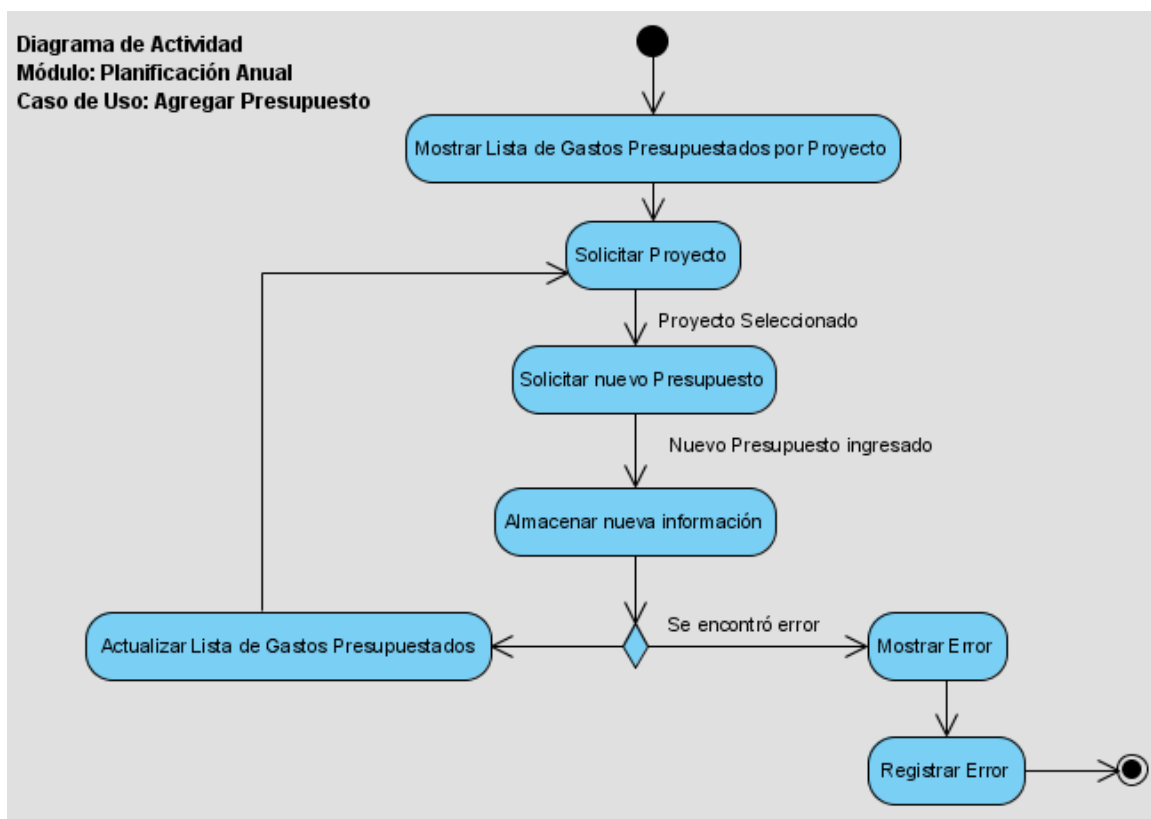


Diagrama de actividad para el caso de uso: Agregar Presupuesto.

5.1.16.1.3 *Módulo de desarrollo profesional*

5.1.16.1.3.1 Diagrama general de casos de uso

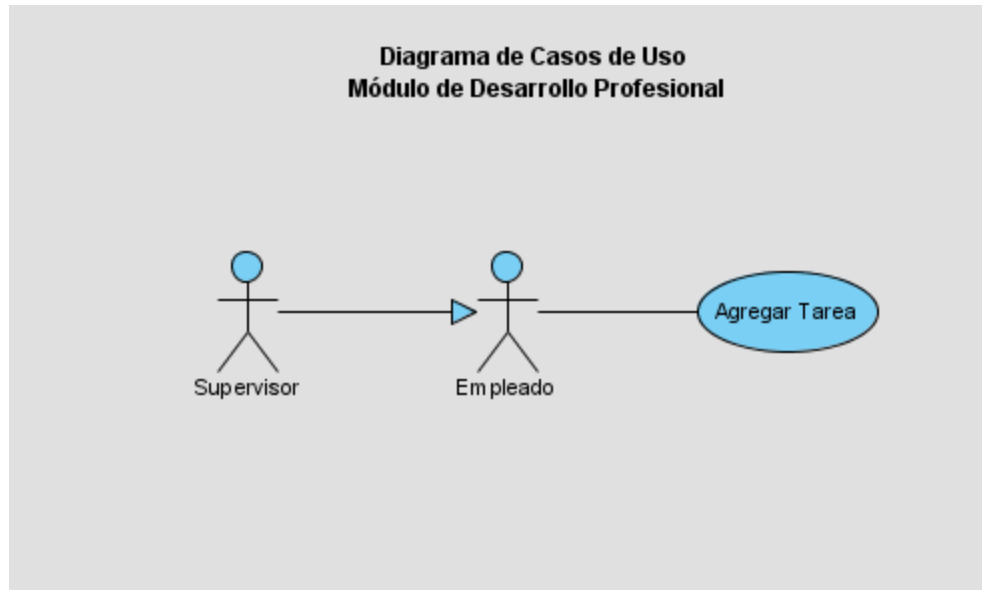


Diagrama general de caso de uso para el Módulo de Desarrollo Profesional.

5.1.16.1.3.2 Casos de uso

5.1.16.1.3.2.1 Agregar tarea

Nombre del caso de uso: Agregar tarea.

Descripción: El empleado utiliza el caso de uso Agregar tarea para registrar todas las actividades de estudio que le han sido asignadas para su desarrollo profesional. Una vez la actividad ha sido ingresada, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.

Restricciones de la implementación: El empleado puede agregar una tarea desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado agrega al sistema una actividad.

1. El empleado ve una lista de sus tareas asignadas.
2. El empleado agrega una tarea en su planificación de desarrollo. La nueva tarea es mostrada en la lista de tareas.

Flujo alterno de eventos: El supervisor agrega al sistema la tarea de un empleado.

1. Se muestra una lista de empleados ordenados alfabéticamente.
2. El supervisor selecciona un empleado y ve una lista de sus tareas asignadas.
3. El supervisor agrega una tarea para la planificación de desarrollo profesional del empleado. La nueva tarea es mostrada en la lista de tareas.

Flujo alterno de eventos: El empleado modifica una tarea.

1. El empleado ve una lista de sus tareas asignadas.
2. El empleado selecciona una tarea existente.
3. El empleado modifica una tarea en su planificación de desarrollo profesional.
4. La información es actualizada y la lista de tareas es renovada.

Flujo alterno de eventos: Tarea duplicada.

1. El empleado ve una lista de sus tareas asignadas.
2. El empleado agrega una tarea en su planificación de desarrollo profesional. La tarea ya existe en su planificación de desarrollo profesional.
3. El sistema informa al empleado que la tarea ya. La lista de tareas no sufre cambios.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar la tarea, debido a problemas en el sistema o en las comunicaciones.

1. El empleado ve una lista de sus tareas asignadas.
2. El empleado agrega una tarea en su planificación de desarrollo profesional. El sistema es incapaz de agregar la tarea, debido a problemas en el sistema o en las comunicaciones.
3. El sistema informa del error al empleado, con el mayor nivel de detalle posible. La lista de tareas no sufre cambios.
4. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

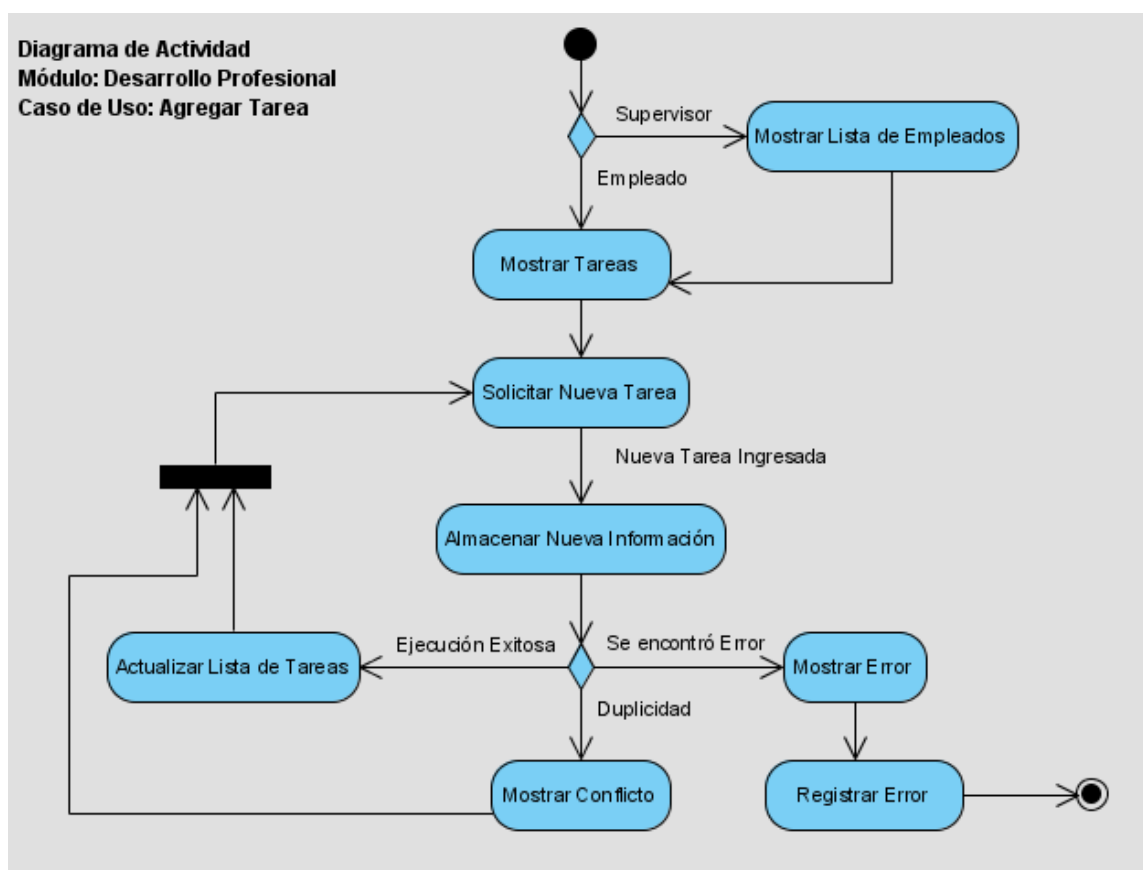


Diagrama de actividad para el caso de uso: Agregar Tarea.

5.1.16.1.4 Módulo de actividades no planificadas

5.1.16.1.4.1 Diagrama general de casos de uso

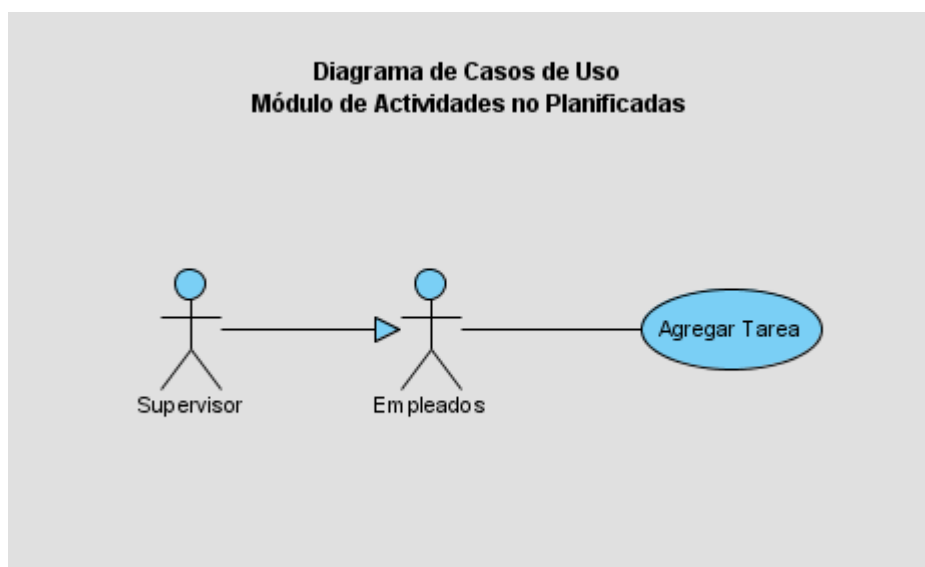


Diagrama general de caso de uso para el Módulo de Actividades no Planificadas.

5.1.16.1.4.2 Casos de uso

5.1.16.1.4.2.1 Agregar tarea

Nombre del caso de uso: Agregar tarea.

Descripción: El empleado utiliza el caso de uso Agregar tarea para registrar todas las actividades que no han sido programadas en su planificación anual. La actividad de asocia a un proyecto y una vez ha sido ingresada, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Restricciones de la implementación: El empleado puede agregar una tarea desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado agrega al sistema una actividad.

1. El empleado ve una lista de sus tareas no planificadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado agrega una tarea en el proyecto seleccionado. La nueva tarea es mostrada en la lista de tareas por proyectos.

Flujo alterno de eventos: El supervisor agrega al sistema la tarea de un empleado.

1. Se muestra una lista de empleados ordenados alfabéticamente.
2. El supervisor selecciona un empleado y ve una lista de sus tareas no planificadas, las tareas están organizadas por proyectos.
3. El supervisor selecciona un proyecto.
4. El supervisor agrega una tarea en el proyecto seleccionado. La nueva tarea es mostrada en la lista de tareas por proyectos.

Flujo alterno de eventos: El empleado modifica una tarea.

1. El empleado ve una lista de sus tareas no planificadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado selecciona una tarea existente.
4. El empleado modifica una tarea en el proyecto.
5. La información es actualizada y la lista de tareas es renovada.

Flujo alterno de eventos: Tarea duplicada.

1. El empleado ve una lista de sus tareas no planificadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado agrega una tarea en el proyecto seleccionado. La tarea ya existe en el proyecto.
4. El sistema informa al empleado que la tarea ya existe en el proyecto. La lista de tareas no sufre cambios.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar la tarea, debido a problemas en el sistema o en las comunicaciones.

1. El empleado ve una lista de sus tareas no planificadas, las tareas están organizadas por proyectos.
2. El empleado selecciona un proyecto.
3. El empleado agrega una tarea en el proyecto seleccionado. El sistema es incapaz de agregar la tarea, debido a problemas en el sistema o en las comunicaciones.
4. El sistema informa del error al empleado, con el mayor nivel de detalle posible. La lista de tareas no sufre cambios.
5. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

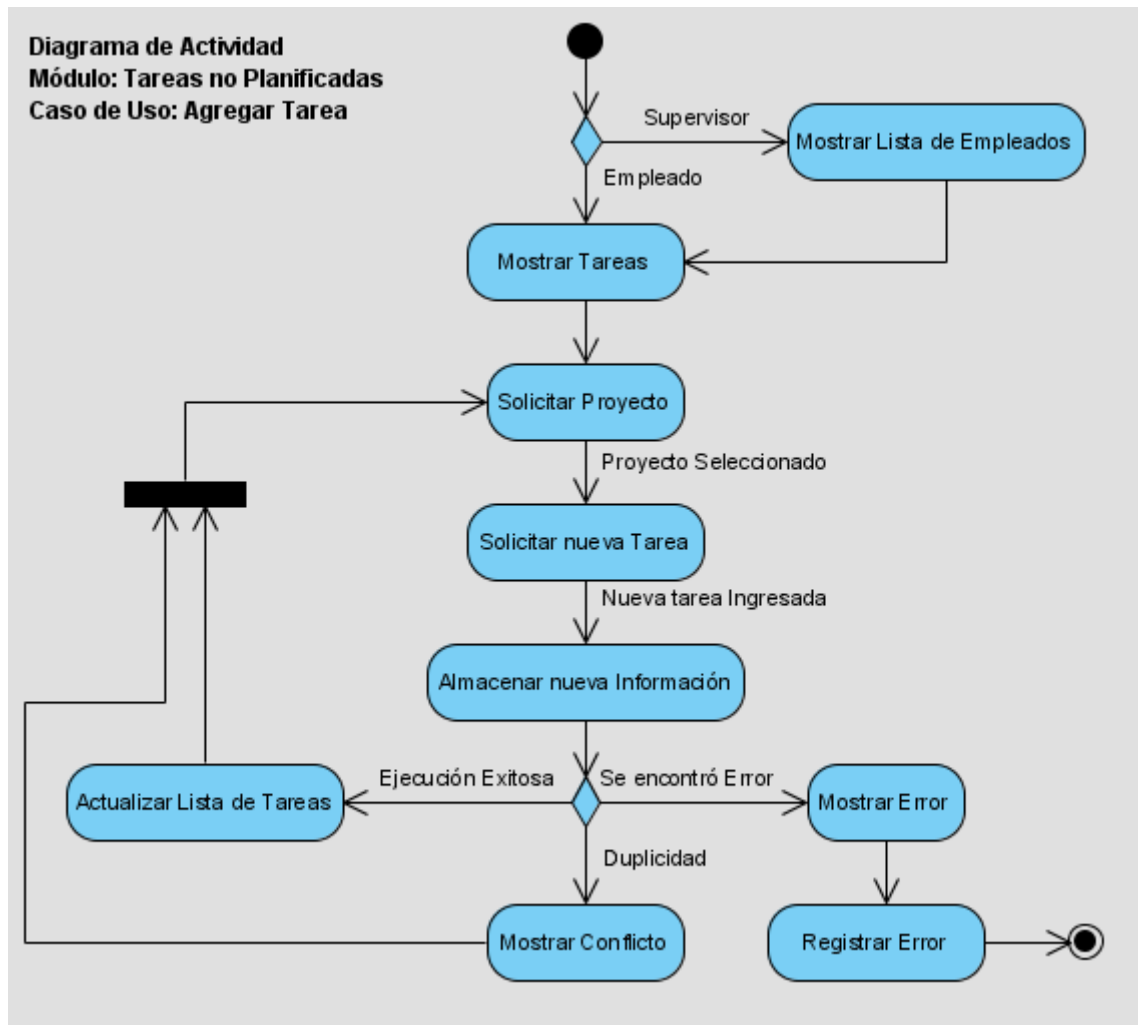


Diagrama de actividad para el caso de uso: Agregar Tarea.

5.1.16.1.5 Módulo de evaluación anual

5.1.16.1.5.1 Diagrama general de casos de uso

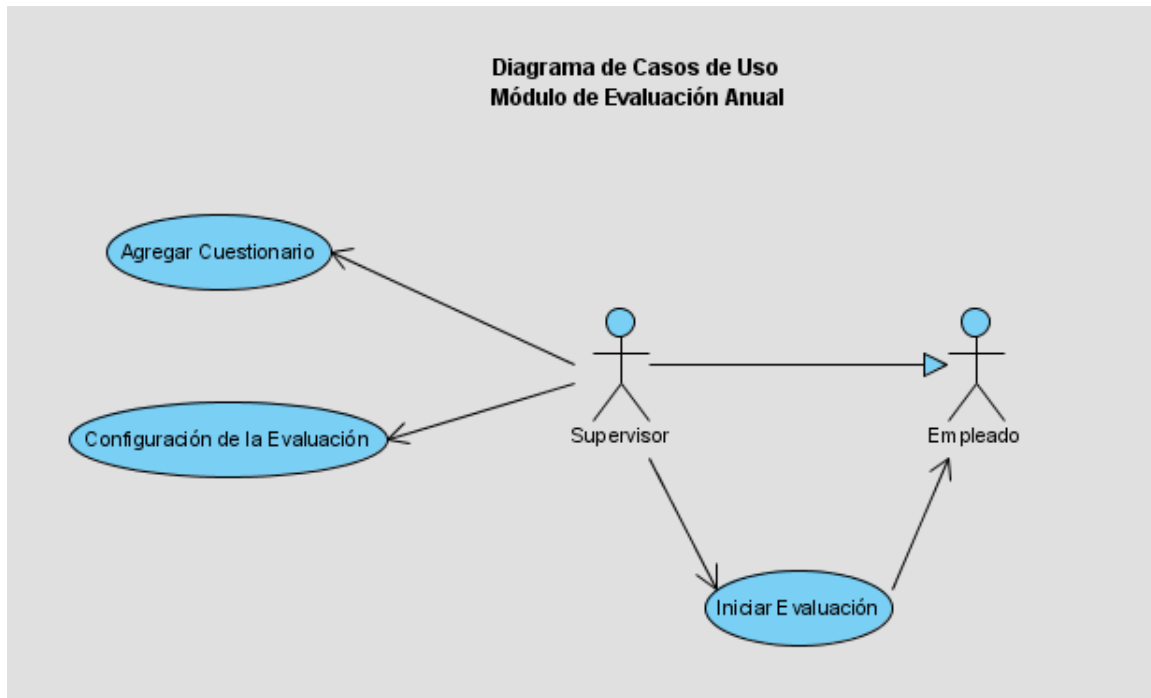


Diagrama general de caso de uso para el Módulo de Evaluación Anual.

5.1.16.1.5.2 Casos de uso

5.1.16.1.5.2.1 Agregar cuestionario

Nombre del caso de uso: Agregar cuestionario.

Descripción: El supervisor utiliza el caso de uso Agregar cuestionario para ingresar al sistema las preguntas de un cuestionario nuevo. Una vez el cuestionario ha sido ingresado, es visible para todos los supervisores.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Restricciones de la implementación: El supervisor puede agregar un cuestionario desde la oficina o desde su hogar.

Flujo normal de eventos: El supervisor agrega al sistema un cuestionario.

1. El supervisor ve una lista de los cuestionarios existentes, los cuestionarios están ordenados alfabéticamente.
2. El supervisor agrega un nuevo cuestionario especificando su nombre. El supervisor ve la lista de preguntas en blanco del cuestionario.
3. El supervisor agrega las preguntas del cuestionario. La lista de preguntas del cuestionario es actualizada.

Flujo alterno de eventos: El supervisor modifica un cuestionario.

1. El supervisor ve una lista de los cuestionarios existentes, los cuestionarios están ordenados alfabéticamente.
2. El supervisor selecciona un cuestionario.
3. El empleado modifica las preguntas del cuestionario.
4. La lista de preguntas del cuestionario es actualizada.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar el cuestionario, debido a problemas en el sistema o en las comunicaciones.

1. El supervisor ve una lista de los cuestionarios existentes, los cuestionarios están ordenados alfabéticamente.

2. El supervisor agrega un nuevo cuestionario especificando su nombre. El supervisor ve la lista de preguntas en blanco del cuestionario.
3. El supervisor agrega las preguntas del cuestionario. El sistema es incapaz de agregar las preguntas, debido a problemas en el sistema o en las comunicaciones.
4. El sistema informa del error al supervisor, con el mayor nivel de detalle posible. La lista de cuestionarios no sufre cambios.
5. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

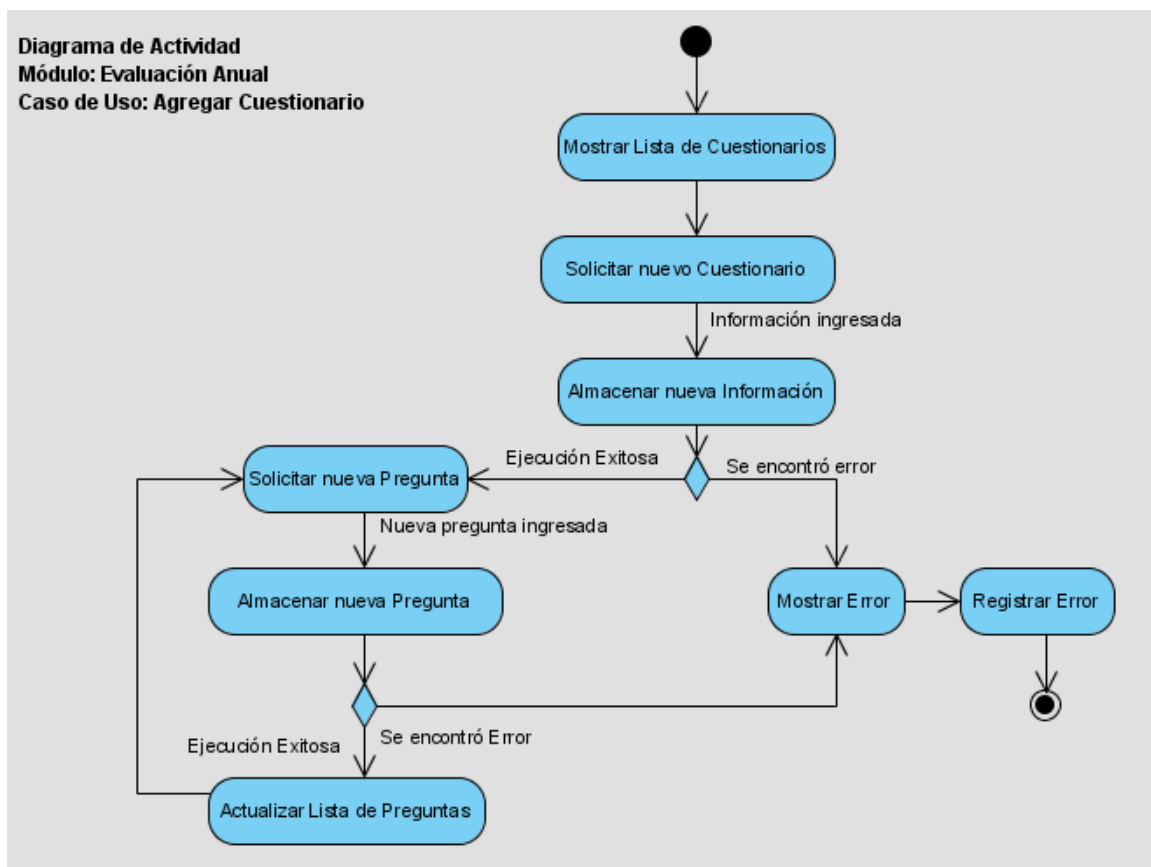


Diagrama de actividad para el caso de uso: Agregar Tarea.

5.1.16.1.5.2.2 Configuración de la evaluación

Nombre del caso de uso: Configuración de la evaluación.

Descripción: El supervisor utiliza el caso de uso Configuración de la evaluación para seleccionar a los empleados que contestarán los cuestionarios de evaluación de un empleado en evaluación. El supervisor solo podrá configurar evaluaciones para sus subalternos. Una vez la evaluación ha sido configurada puede dar inicio la evaluación.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.
3. Los cuestionarios de la evaluación debes estar creados.

Restricciones de la implementación: El supervisor puede configurar la evaluación desde la oficina o desde su hogar.

Flujo normal de eventos: El supervisor configura una nueva evaluación.

1. El supervisor ve una lista de empleados a su cargo, los empleados están organizados por departamento.
2. El supervisor selecciona un empleado.
3. El sistema muestra una lista de cuestionarios.
4. El supervisor selecciona los cuestionarios que deberán ser contestados para la evaluación, selecciona a los empleados que contestarán los cuestionarios y define una fecha de inicio de la evaluación.

Flujo alternativo de eventos: El supervisor modifica la configuración de una evaluación.

1. El supervisor ve una lista de empleados a su cargo, los empleados están organizados por departamento.
2. El supervisor selecciona un empleado.
3. El sistema muestra una lista de evaluaciones ordenadas por fecha de evaluación.
4. El supervisor modifica la configuración de la evaluación.
5. La información es actualizada en el sistema.

Excepciones o flujo de errores de eventos: El sistema es incapaz de configurar una evaluación, debido a problemas en el sistema o en las comunicaciones.

1. El supervisor ve una lista de empleados a su cargo, los empleados están organizados por departamento.
2. El supervisor selecciona un empleado.
3. El sistema muestra una lista de cuestionarios.
4. El supervisor selecciona los cuestionarios que deberán ser contestados para la evaluación, selecciona a los empleados que contestarán los cuestionarios y define una fecha de inicio de la evaluación. El sistema es incapaz de configurar la nueva evaluación, debido a problemas en el sistema o en las comunicaciones.
5. El sistema informa del error al empleado, con el mayor nivel de detalle posible.
6. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

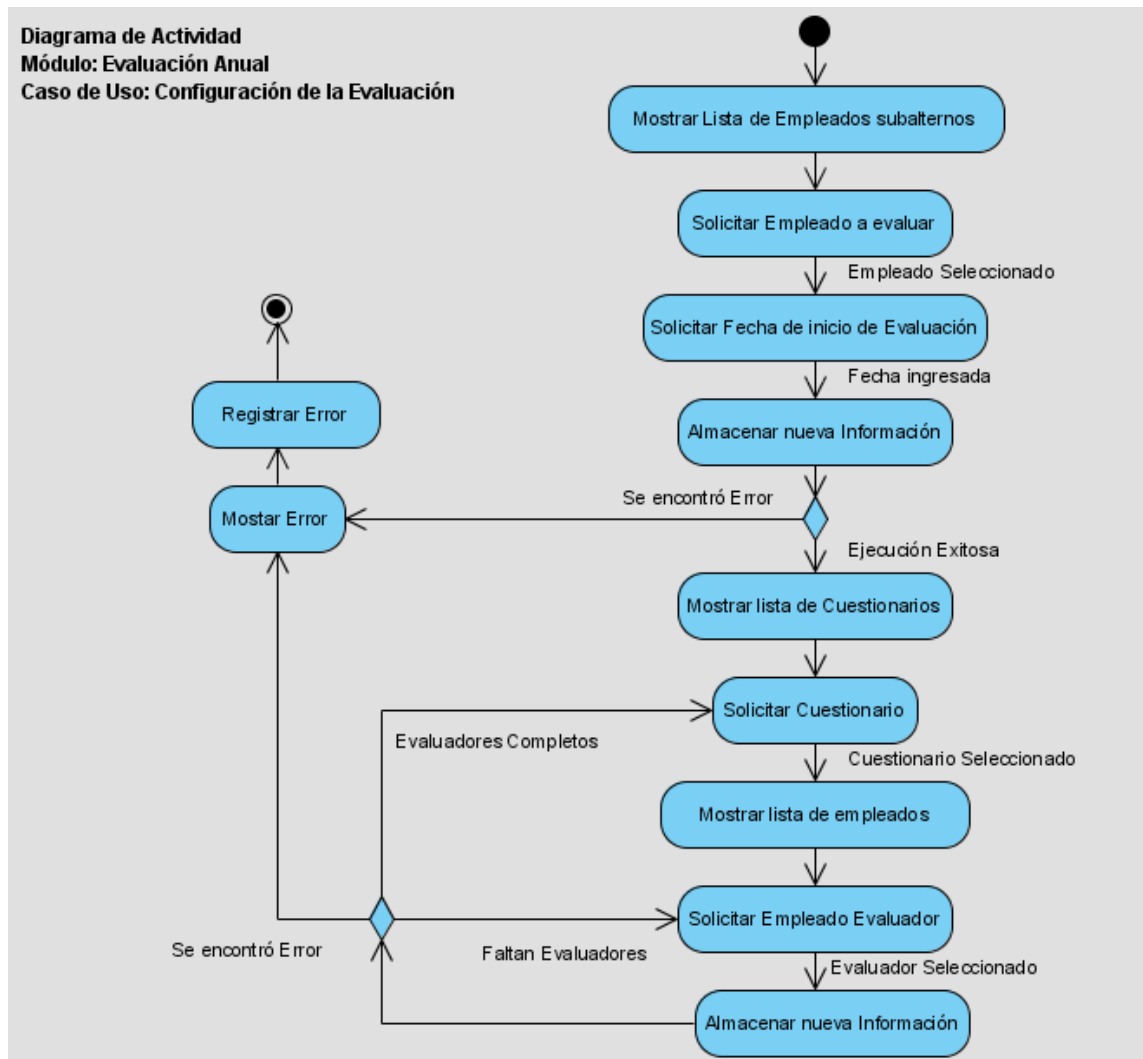


Diagrama de actividad para el caso de uso: Agregar Tarea.

5.1.16.1.5.2.3 Iniciar evaluación

Nombre del caso de uso: Iniciar evaluación.

Descripción: El supervisor utiliza el caso de uso Iniciar evaluación para permitir a los empleados evaluadores iniciar la evaluación del empleado en evaluación, al finalizar este proceso notifica por correo a los evaluadores que la evaluación ha iniciado.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.
3. Los cuestionarios debe estar creados.
4. La configuración de la evaluación debe estar terminada.

Restricciones de la implementación: ninguna.**Flujo normal de eventos:** El supervisor inicia una evaluación.

1. El supervisor ve una lista de empleados a su cargo, los empleados están organizados por departamento.
2. El supervisor selecciona un empleado.
3. El sistema muestra una lista de evaluaciones pendientes ordenadas por fecha de evaluación.
4. El supervisor activa la evaluación. El sistema envía un correo de notificación a los empleados evaluadores. Los empleados evaluadores pueden contestar los cuestionarios de evaluación.

Excepciones o flujo de errores de eventos: El sistema es incapaz de iniciar una evaluación, debido a problemas en el sistema o en las comunicaciones.

5. El supervisor ve una lista de empleados a su cargo, los empleados están organizados por departamento.
6. El supervisor selecciona un empleado.
7. El sistema muestra una lista de evaluaciones pendientes ordenadas por fecha de evaluación.
6. El supervisor activa la evaluación. El sistema es incapaz de iniciar la evaluación debido a problemas en el sistema o en las comunicaciones.
7. El sistema informa del error al supervisor, con el mayor nivel de detalle posible. Ningún correo es enviado.

8. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

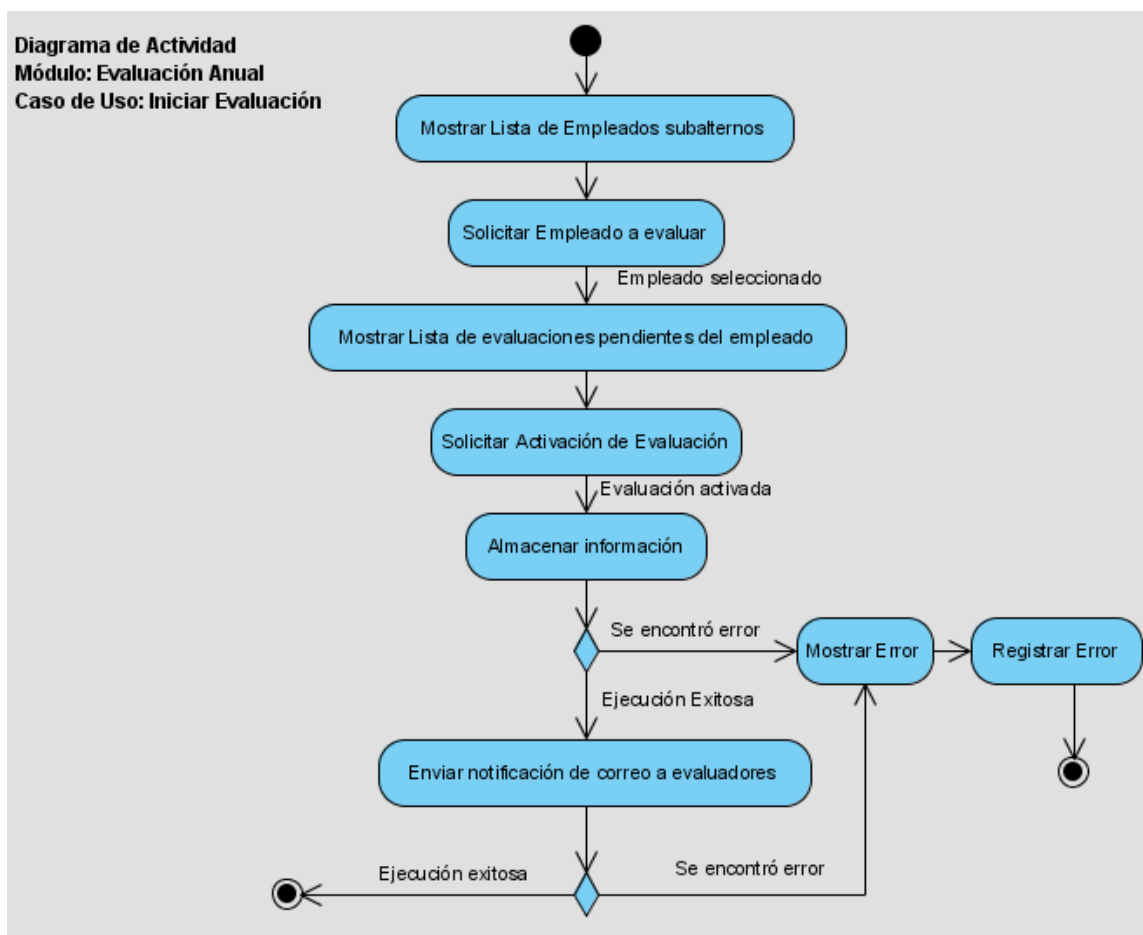


Diagrama de actividad para el caso de uso: Iniciar Evaluación.

5.1.16.1.5.2.4 Contestar cuestionario

Nombre del caso de uso: Contestar cuestionario.

Descripción: El empleado utiliza el caso de uso Contestar cuestionario para evaluar el desempeño anual de otro empleado. Una vez el cuestionario ha sido llenado, es visible para los superiores del empleado en evaluación.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El cuestionario debe estar creado.
4. La configuración de la evaluación debe estar terminada.
5. La evaluación debe estar iniciada.

Restricciones de la implementación: El empleado puede agregar una tarea desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado contesta un cuestionario para la evaluación de otro empleado.

1. El empleado ve una lista de evaluaciones pendientes de contestar.
2. El empleado selecciona una evaluación.
3. El empleado ve la lista de preguntas del cuestionario junto con el nombre del empleado en evaluación.
4. El empleado contesta las preguntas.

Excepciones o flujo de errores de eventos: El sistema es incapaz de guardar el cuestionario contestado, debido a problemas en el sistema o en las comunicaciones.

1. El empleado ve la lista de preguntas del cuestionario junto con el nombre del empleado en evaluación.
2. El empleado contesta las preguntas. El sistema es incapaz de guardar el cuestionario contestado debido a problemas en el sistema o en las comunicaciones.
3. El sistema informa del error al empleado, con el mayor nivel de detalle posible.
4. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

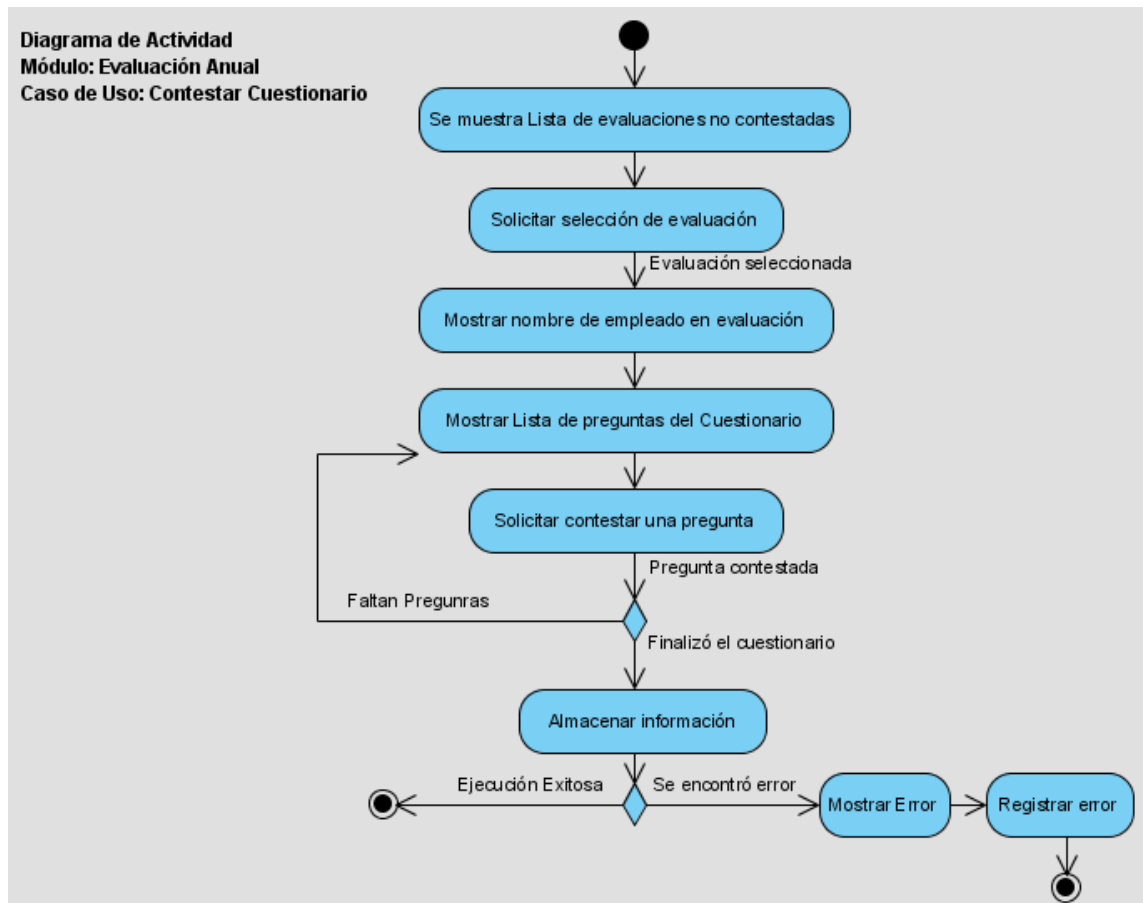


Diagrama de actividad para el caso de uso: Contestar Cuestionario.

5.1.16.1.6 Módulo de seguridad

5.1.16.1.6.1 Diagrama general de casos de uso

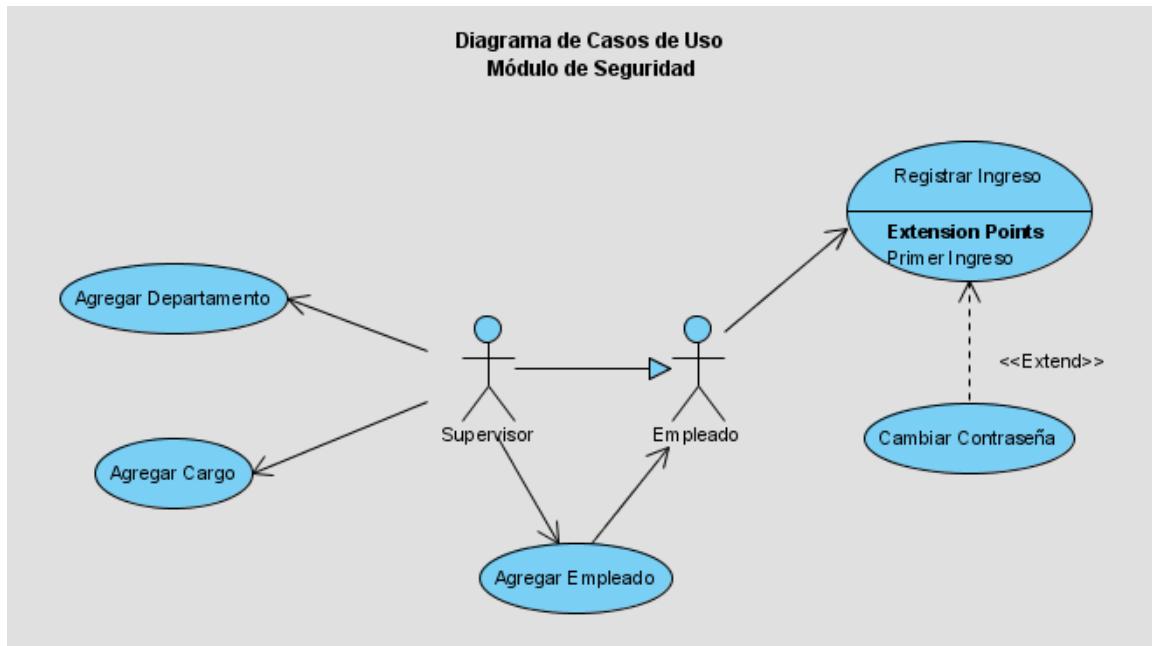


Diagrama general de caso de uso para el Módulo de Seguridad.

5.1.16.1.6.2 Casos de uso

5.1.16.1.6.2.1 Agregar departamento

Nombre del caso de uso: Agregar departamento.

Descripción: El supervisor utiliza el caso de uso Agregar departamento para registrar un departamento o área que están bajo su cargo. Una vez el departamento ha sido ingresado, es visible en todo el sistema.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.

2. El supervisor debe registrar su ingreso al sistema.

Restricciones de la implementación: El supervisor puede agregar un departamento desde la oficina o desde su hogar.

Flujo normal de eventos: El supervisor agrega un departamento.

1. El supervisor ve una lista de los departamentos que están bajo su cargo.
2. El supervisor agrega un departamento. El nuevo departamento es mostrado en la lista de departamentos.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar el proyecto, debido a problemas en el sistema o en las comunicaciones.

1. El supervisor ve una lista de los departamentos que están bajo su cargo.
2. El supervisor agrega un departamento. El sistema es incapaz de agregar el departamento, debido a problemas en el sistema o en las comunicaciones.
3. El sistema informa del error al supervisor, con el mayor nivel de detalle posible. La lista de tareas no sufre cambios.
4. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

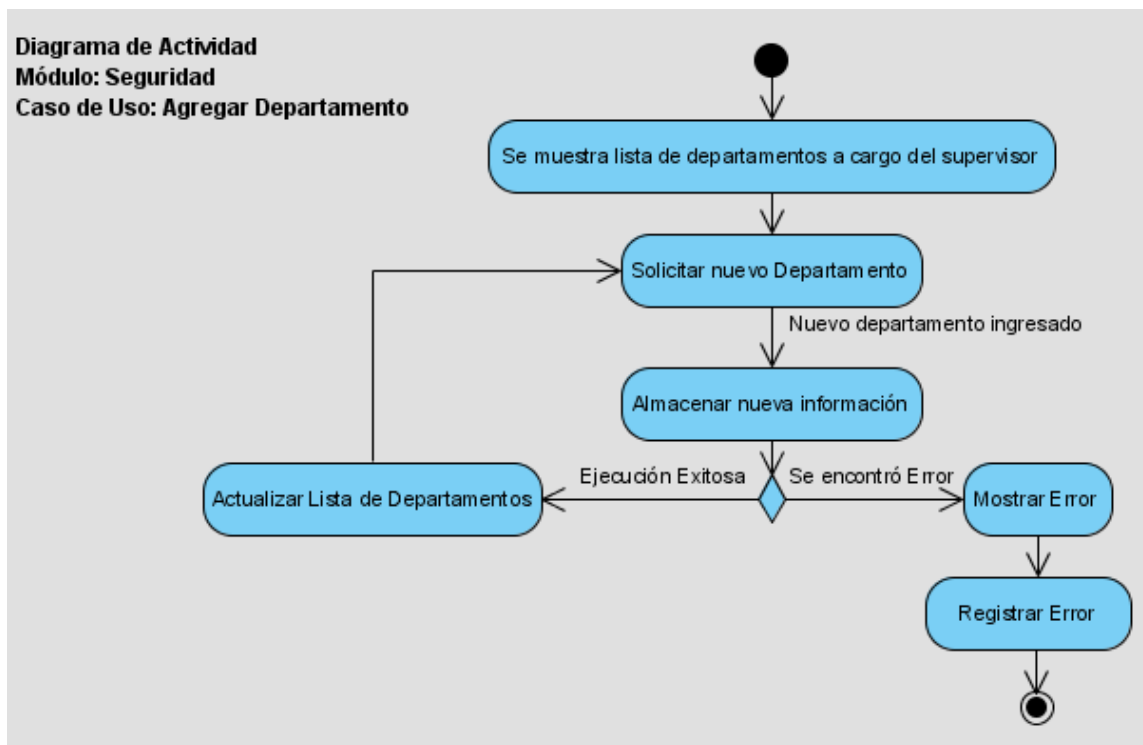


Diagrama de actividad para el caso de uso: Agregar Departamento.

5.1.16.1.6.2.2 Agregar cargo de empleado

Nombre del caso de uso: Agregar cargo de empleado.

Descripción: El supervisor utiliza el caso de uso Agregar cargo de empleado para registrar los cargos de sus empleados subalternos. Una vez el cargo ha sido ingresado, es visible en los departamentos que están a cargo del supervisor.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.

2. El supervisor debe registrar su ingreso al sistema.
3. El departamento debe estar ingresado en el sistema.

Restricciones de la implementación: El supervisor puede agregar un cargo de empleado desde la oficina o desde su hogar.

Flujo normal de eventos: El supervisor agrega un cargo de empleado.

1. El supervisor ve una lista de cargos de empleados agrupados por departamento.
2. El supervisor agrega un cargo de empleado. El nuevo cargo de empleado es mostrado en la lista de departamentos.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar el nuevo cargo de empleado, debido a problemas en el sistema o en las comunicaciones.

1. El supervisor ve una lista de cargos por empleado agrupados por departamento.
2. El supervisor agrega un cargo de empleado El sistema es incapaz de agregar el departamento, debido a problemas en el sistema o en las comunicaciones.
3. El sistema informa del error al supervisor, con el mayor nivel de detalle posible. La lista de tareas no sufre cambios.
4. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

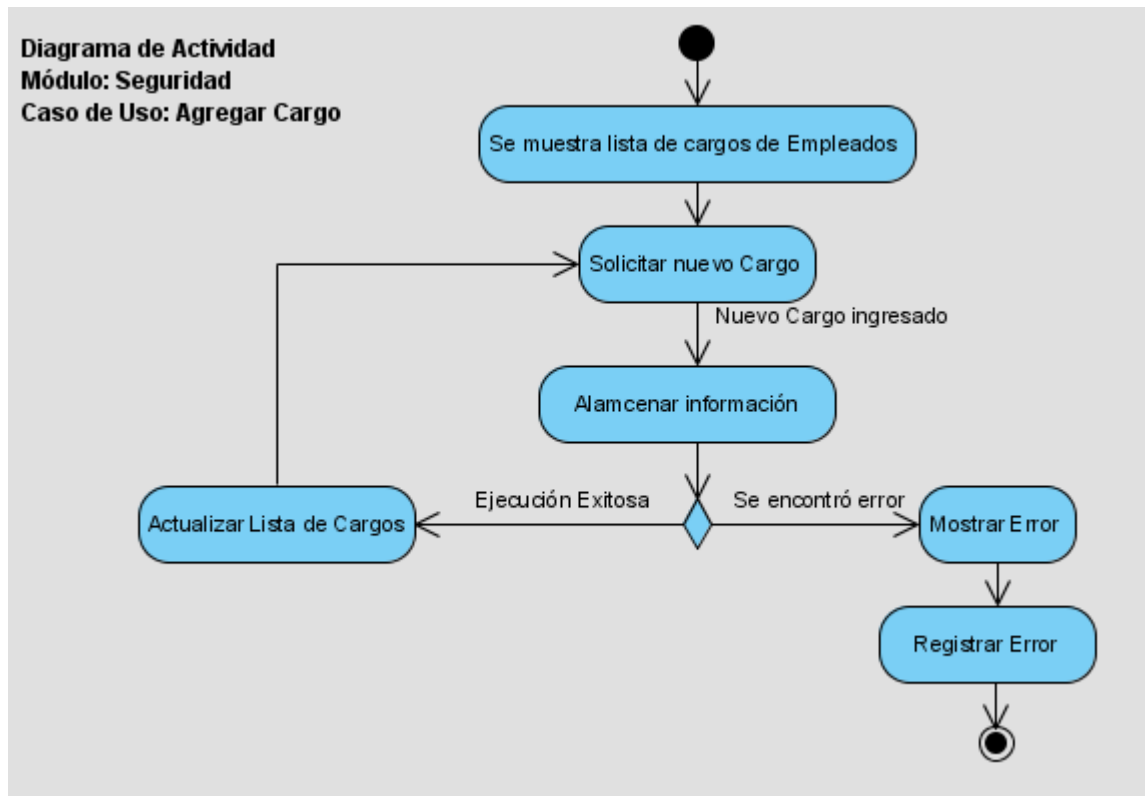


Diagrama de actividad para el caso de uso: Agregar Cargo.

5.1.16.1.6.2.3 Agregar empleado

Nombre del caso de uso: Agregar empleado.

Descripción: El caso de uso agregar empleado, permite que el supervisor agregue un nuevo empleado dentro del área o departamento que es responsable. Una vez ha sido agregado al sistema, el empleado puede usar el sistema.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.

3. El departamento del empleado debe estar creado.
4. El cargo del empleado debe estar creado.

Restricciones de la implementación: El supervisor puede agregar un empleado desde la oficina o desde su hogar.

Flujo normal de eventos: El supervisor agrega un empleado al sistema.

1. El supervisor ve una lista de los empleados ordenados por departamento.
2. El supervisor selecciona el departamento al que pertenece el nuevo empleado.
3. El supervisor selecciona el cargo que tendrá el nuevo empleado.
4. El supervisor define si el nuevo empleado será supervisor.
5. El supervisor agrega a un empleado con su nombre, correo electrónico y contraseña.
6. El sistema envía un correo al nuevo usuario, notificándole su alta en el sistema y su contraseña.
7. El nuevo empleado es mostrado en la lista. El empleado puede registrar sus actividades en el sistema.

Excepciones o flujo de errores de eventos: El sistema es incapaz de agregar al empleado, debido a problemas en el sistema o en las comunicaciones.

1. El supervisor ve una lista de los empleados ordenados por departamento.
2. El supervisor selecciona el departamento al que pertenece el nuevo empleado.
3. El supervisor selecciona el cargo que tendrá el nuevo empleado.
4. El supervisor define si el nuevo empleado será supervisor.
5. El supervisor agrega a un empleado con su nombre, correo electrónico, nombre de usuario dentro del sistema y su contraseña. El sistema es incapaz de agregar al empleado, debido a problemas en el sistema o en las comunicaciones.

6. El sistema informa del error al supervisor, con el mayor nivel de detalle posible. La lista de empleados no sufre cambios.
7. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

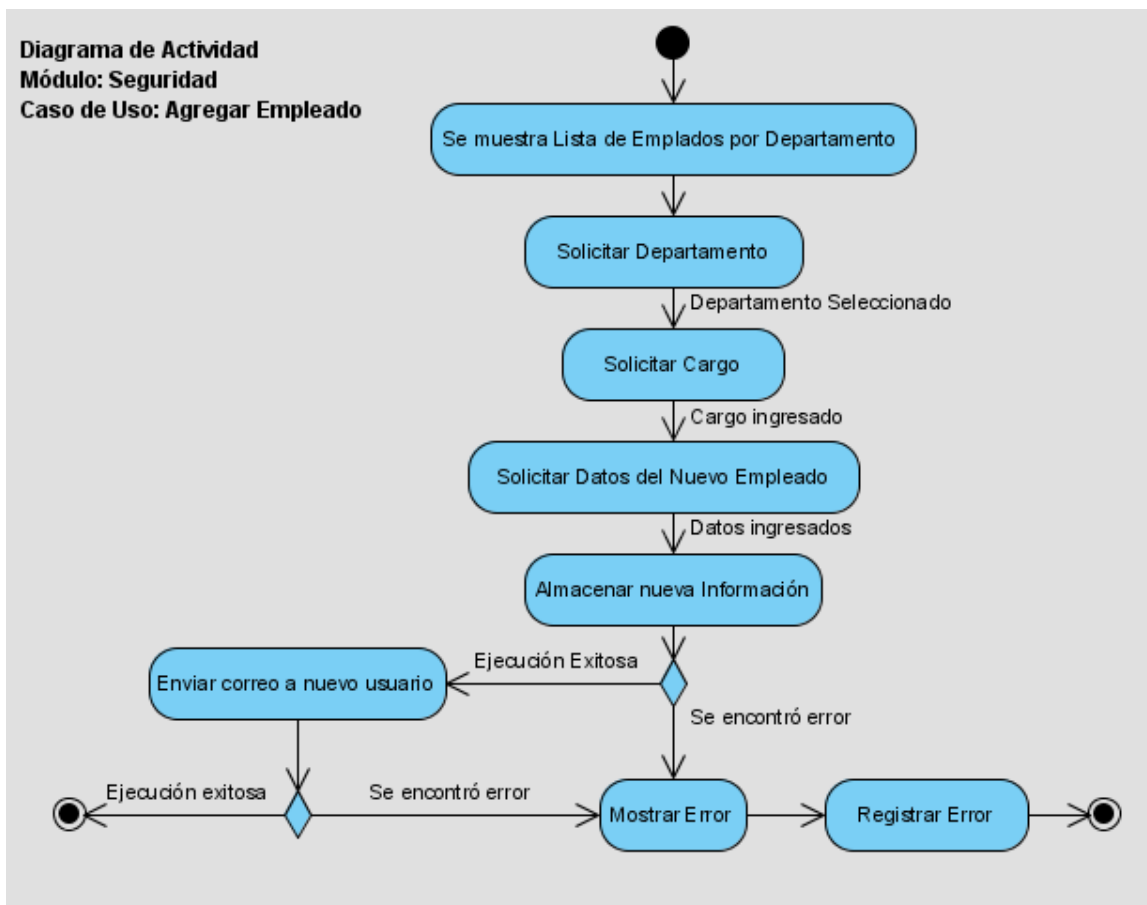


Diagrama de actividad para el caso de uso: Agregar Empleado.

Requerimientos no funcionales: Ninguno.

Notas:

1. ¿Se requiere información adicional al nombre, correo electrónico y contraseña del empleado?

2. ¿El empleado podrá cambiar su contraseña?
3. ¿Se requiere que los empleados estén organizados por departamento?

5.1.16.1.6.2.4 Registrar ingreso al sistema

Nombre del caso de uso: Registrar ingreso al sistema.

Descripción: El caso de uso Registrar ingreso al sistema permite que el empleado acceda a las funciones del sistema.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.

Restricciones de la implementación: El empleado puede registrar su ingreso al sistema desde la oficina o desde su hogar.

Flujo normal de eventos: El nombre de usuario y contraseña del empleado son validos.

1. El empleado proporciona su nombre de usuario y contraseña.
2. El empleado es identificado como un supervisor o como un empleado. Esto no es determinado por una elección del usuario durante el proceso de registro, es determinado por el nombre de usuario.

Flujo alterno de eventos: Primer ingreso.

1. El empleado proporciona su nombre de usuario y contraseña.

2. El empleado es identificado como un supervisor o como un empleado. Esto no es determinado por una elección del usuario durante el proceso de registro, es determinado por el nombre de usuario.
3. El sistema pregunta al empleado si desea cambiar su contraseña.
4. Se ejecuta el caso de uso cambiar contraseña en este punto de extensión.

Flujo alterno de eventos: Información de autenticación no válida.

1. El empleado proporciona su nombre de usuario y contraseña.
2. Se notifica al empleado que ha suministrado información no válida.
3. El sistema permite que el empleado proporcione nuevamente la información de autenticación indefinidamente.

Diagrama de actividad:

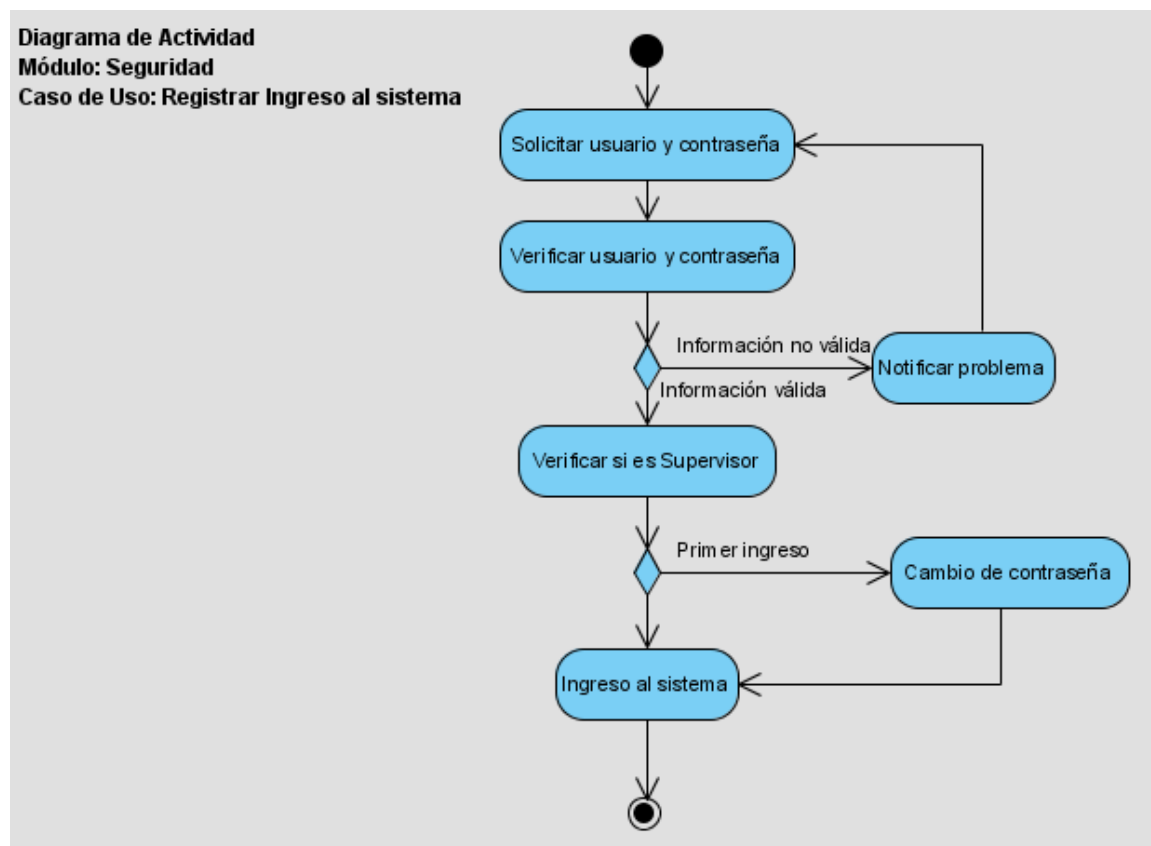


Diagrama de actividad para el caso de uso: Registrar Ingreso.

5.1.16.1.6.2.5 Cambio de contraseña

Nombre del caso de uso: Cambio de contraseña.

Descripción: El caso de uso Cambio de contraseña permite que el empleado cambie su contraseña del sistema.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.

Restricciones de la implementación: El empleado puede registrar su ingreso al sistema desde la oficina o desde su hogar.

Flujo normal de eventos: El empleado cambia su contraseña.

1. El empleado proporciona su contraseña actual y la contraseña nueva dos veces.
2. El empleado es notificado que la contraseña ha sido cambiada.

Flujo alternativo de eventos: Contraseña actual no válida.

1. El empleado proporciona su contraseña actual y la contraseña nueva dos veces.
2. El empleado es notificado que la contraseña actual no es válida.
3. El sistema permite que el empleado proporcione nuevamente la contraseña indefinidamente.

Flujo alternativo de eventos: La nueva contraseña no coincide con la verificación.

1. El empleado proporciona su contraseña actual y la contraseña nueva dos veces.
4. El empleado es notificado que la nueva contraseña no coincide con la verificación.
2. El sistema permite que el empleado proporcione nuevamente la contraseña indefinidamente.

Excepciones o flujo de errores de eventos: El sistema es incapaz de cambiar la contraseña, debido a problemas en el sistema o en las comunicaciones.

1. El empleado proporciona su contraseña actual y la contraseña nueva dos veces.
2. El sistema es incapaz de cambiar la contraseña, debido a problemas en el sistema o en las comunicaciones.
3. El sistema informa del error al empleado, con el mayor nivel de detalle posible.
4. Si es posible el error es registrado en una bitácora de errores.

Diagrama de actividad:

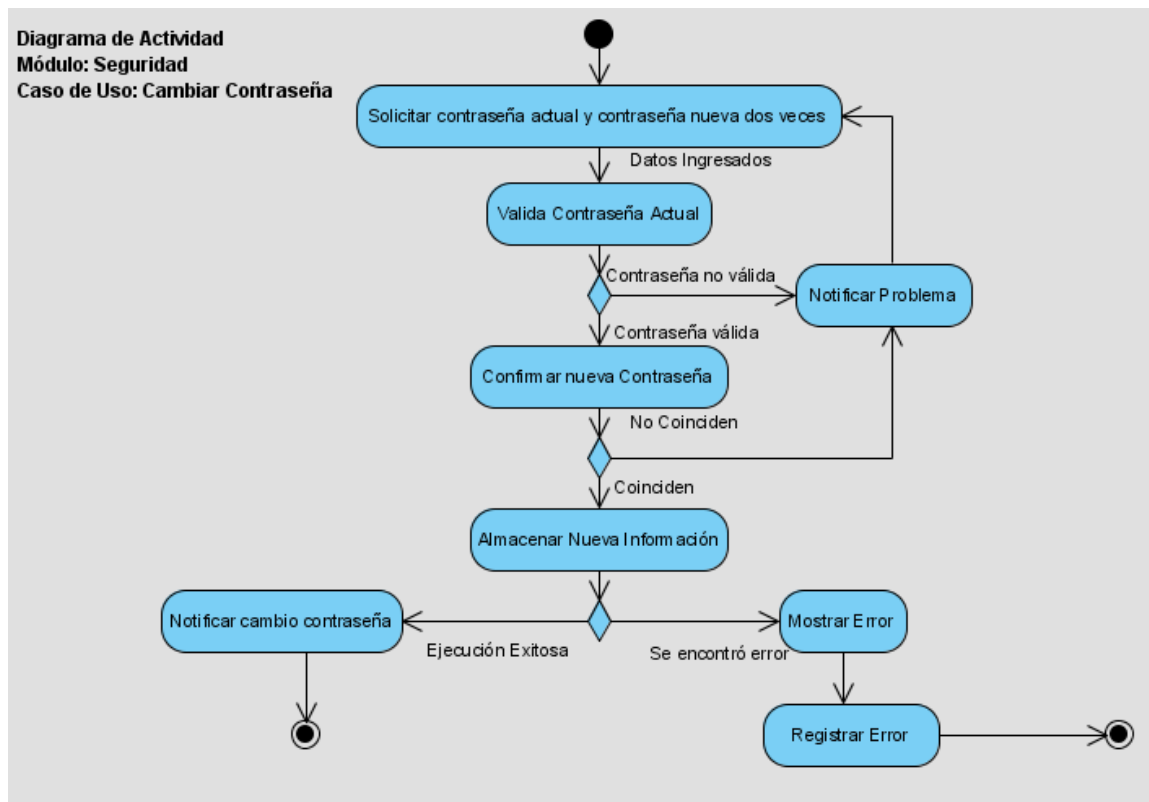
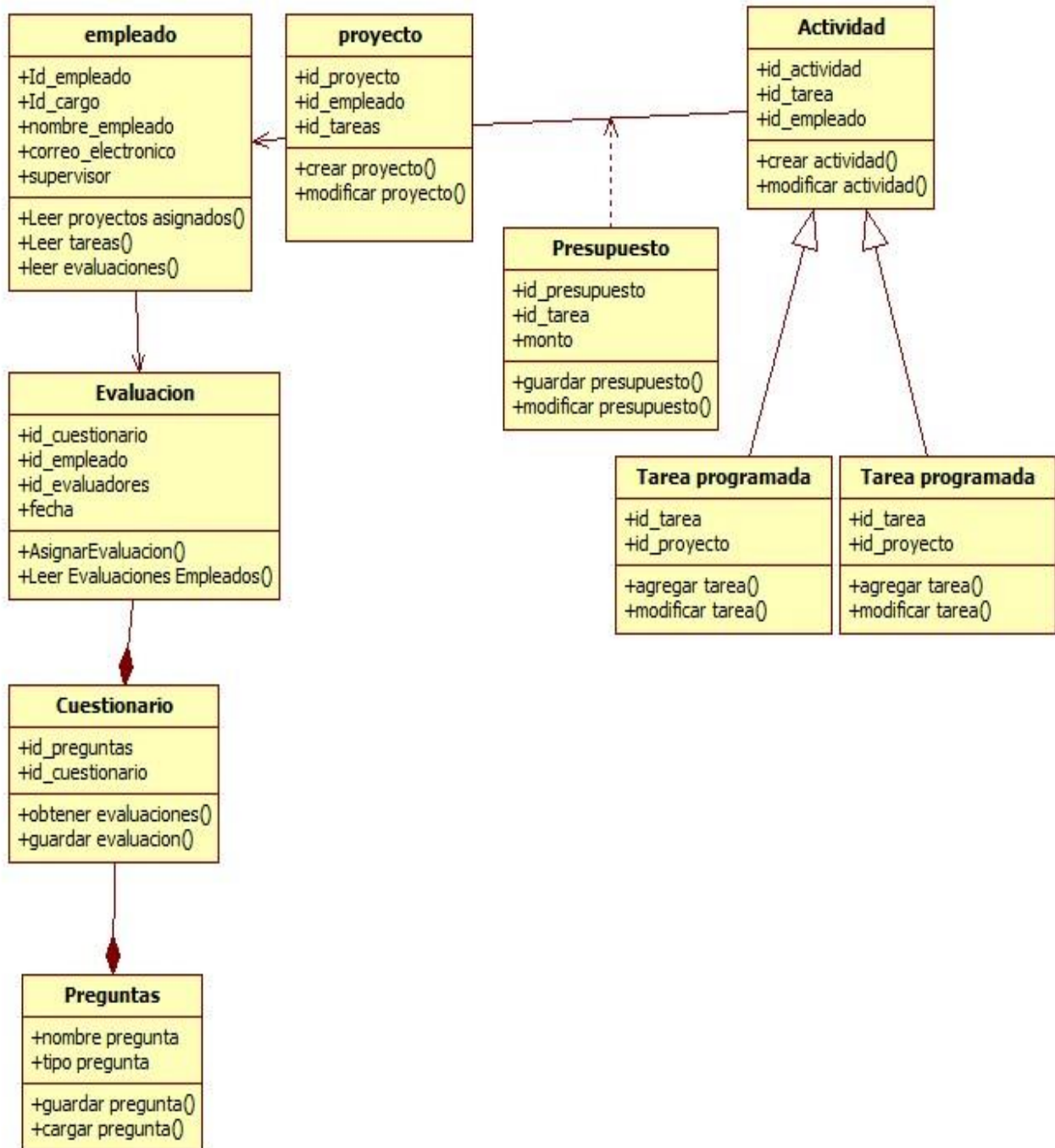


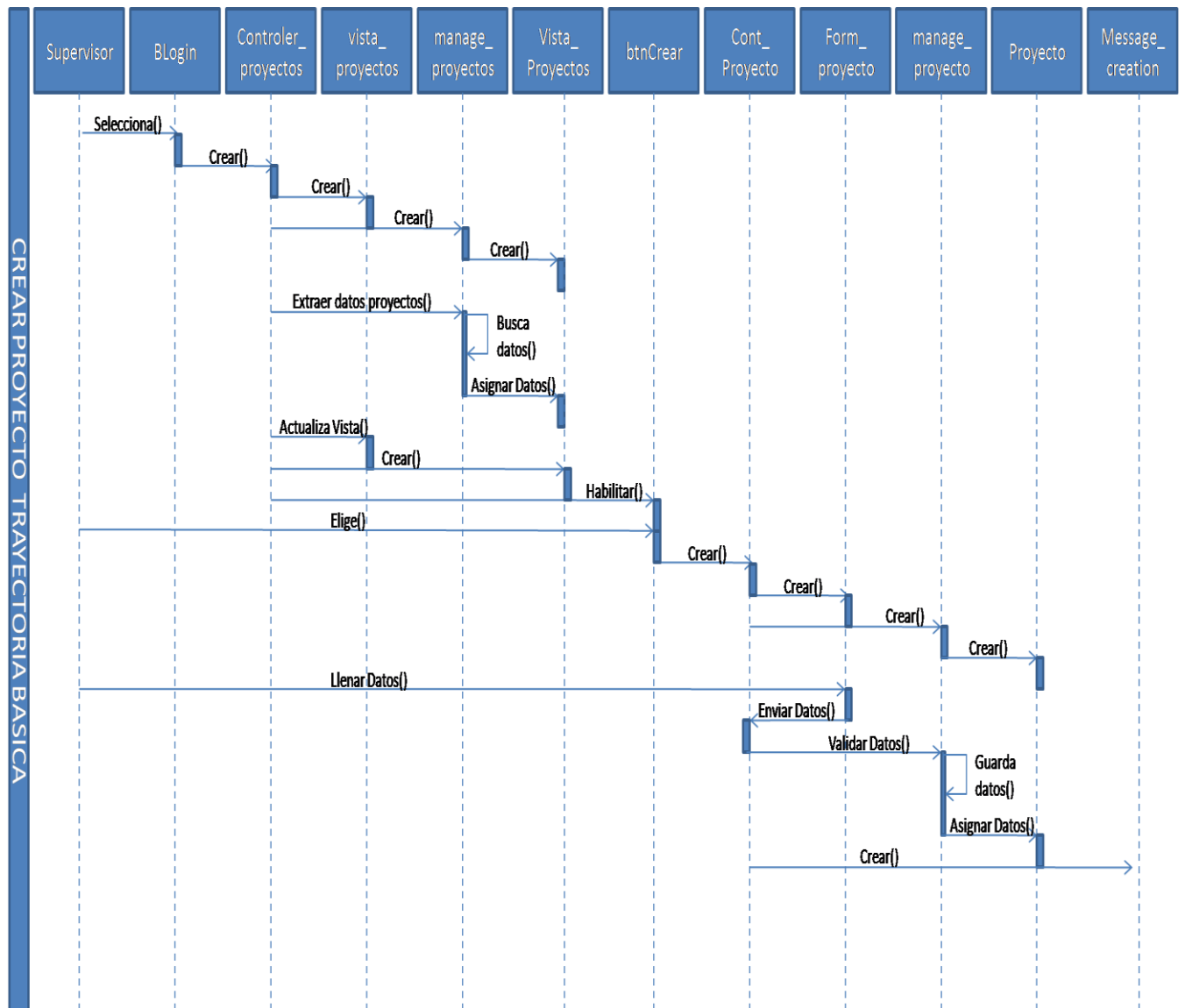
Diagrama de actividad para el caso de uso: Cambiar Contraseña.

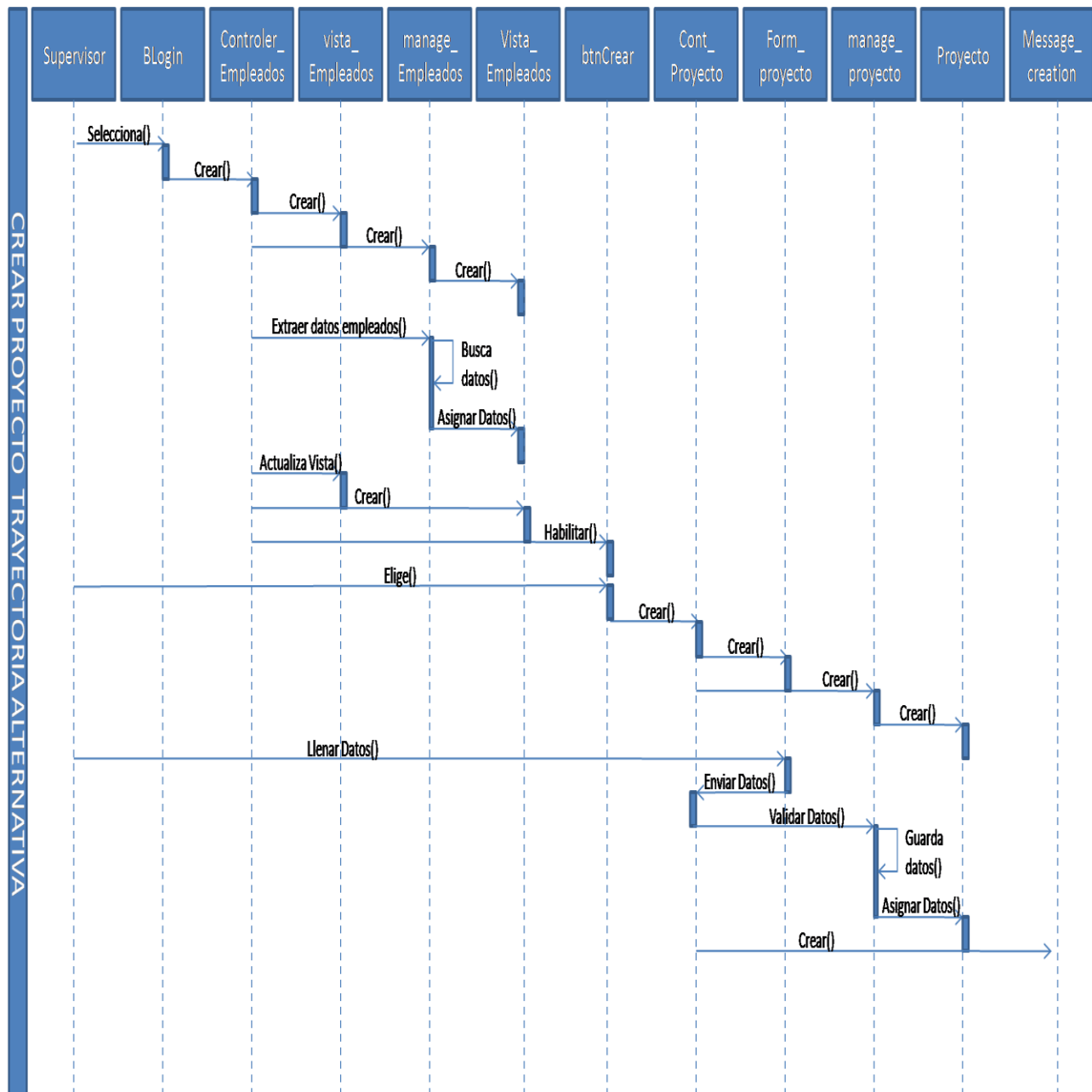
5.1.16.2 Diagrama de Clases

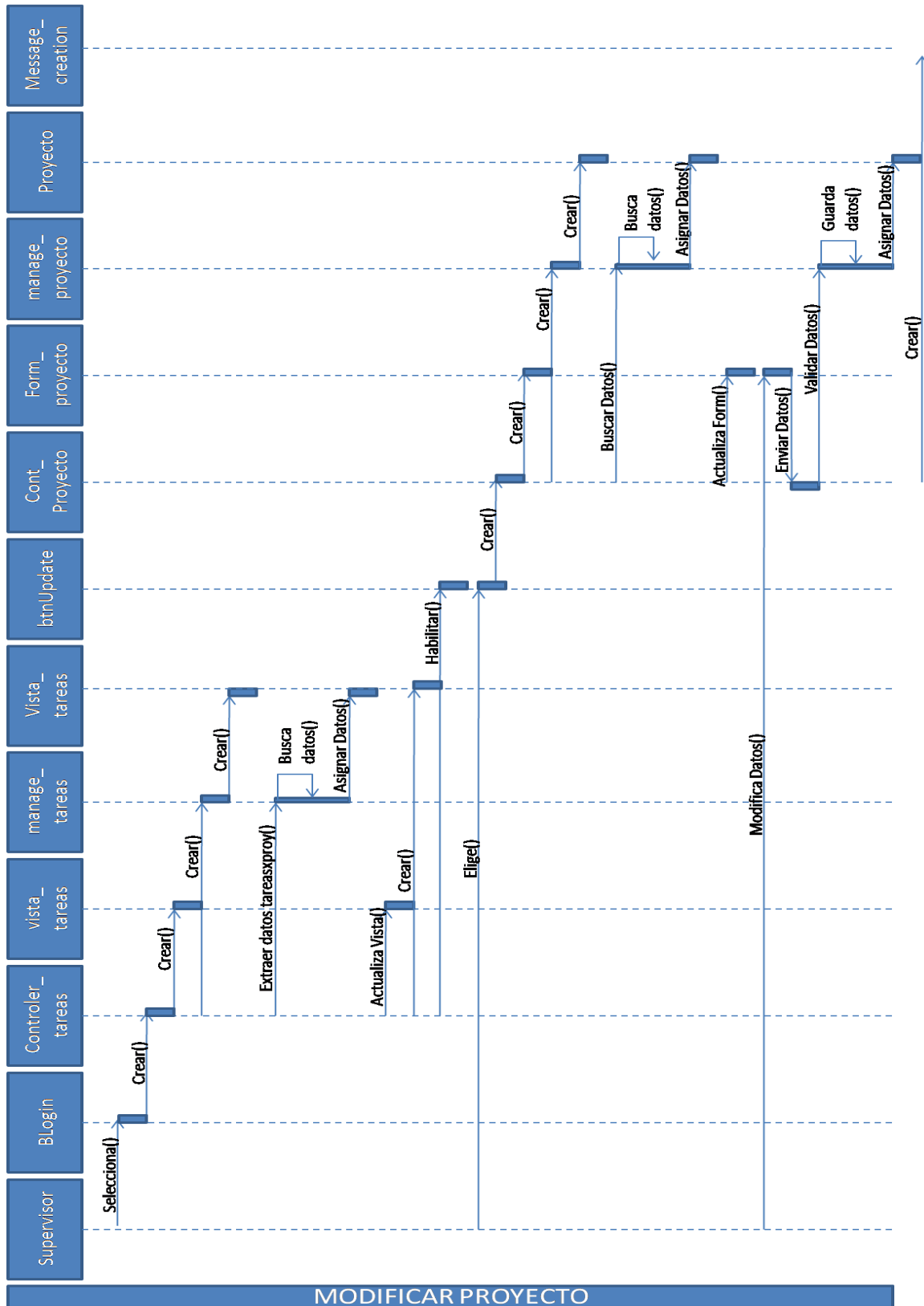


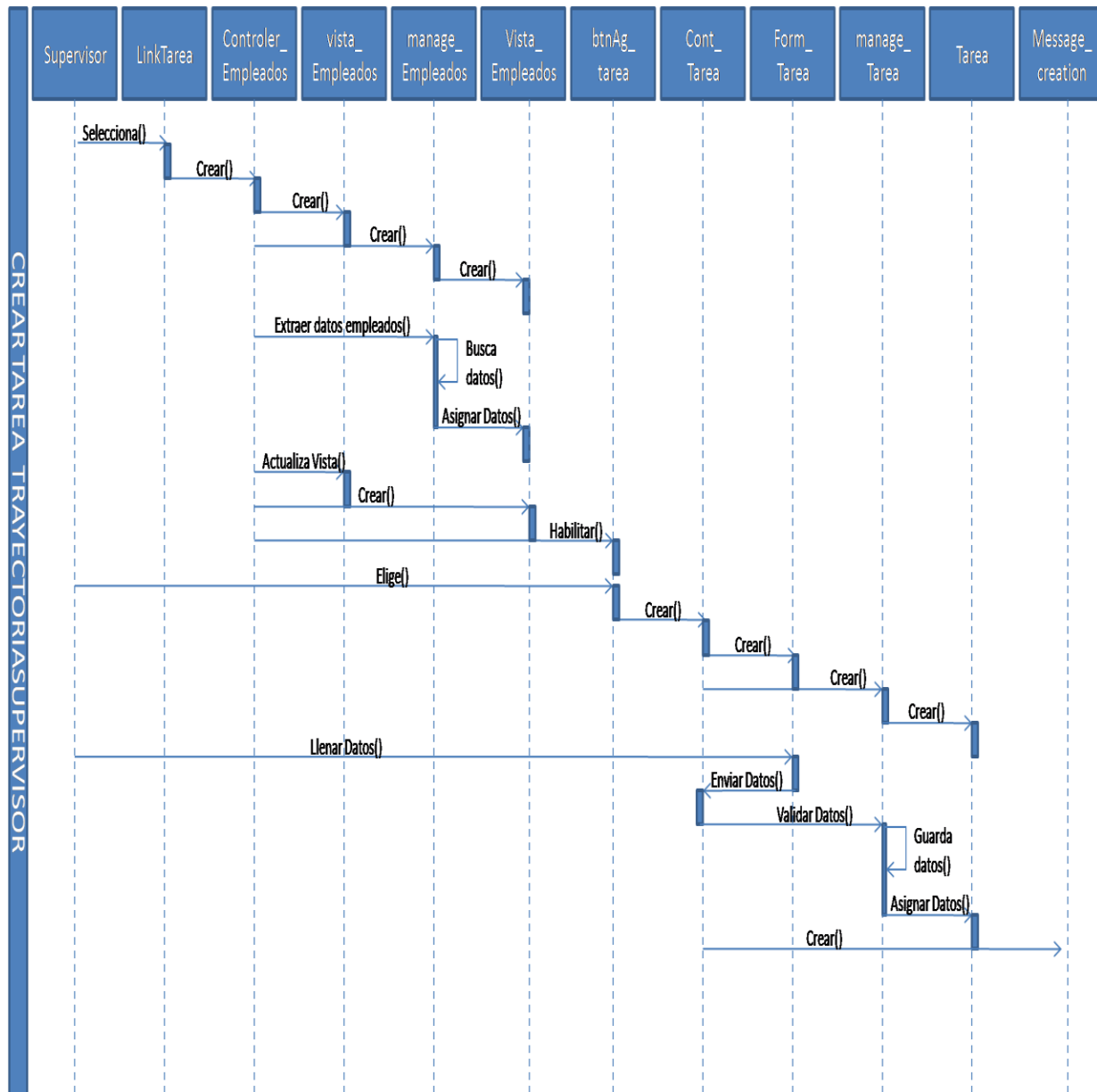
5.1.16.3 Diagramas de secuencia

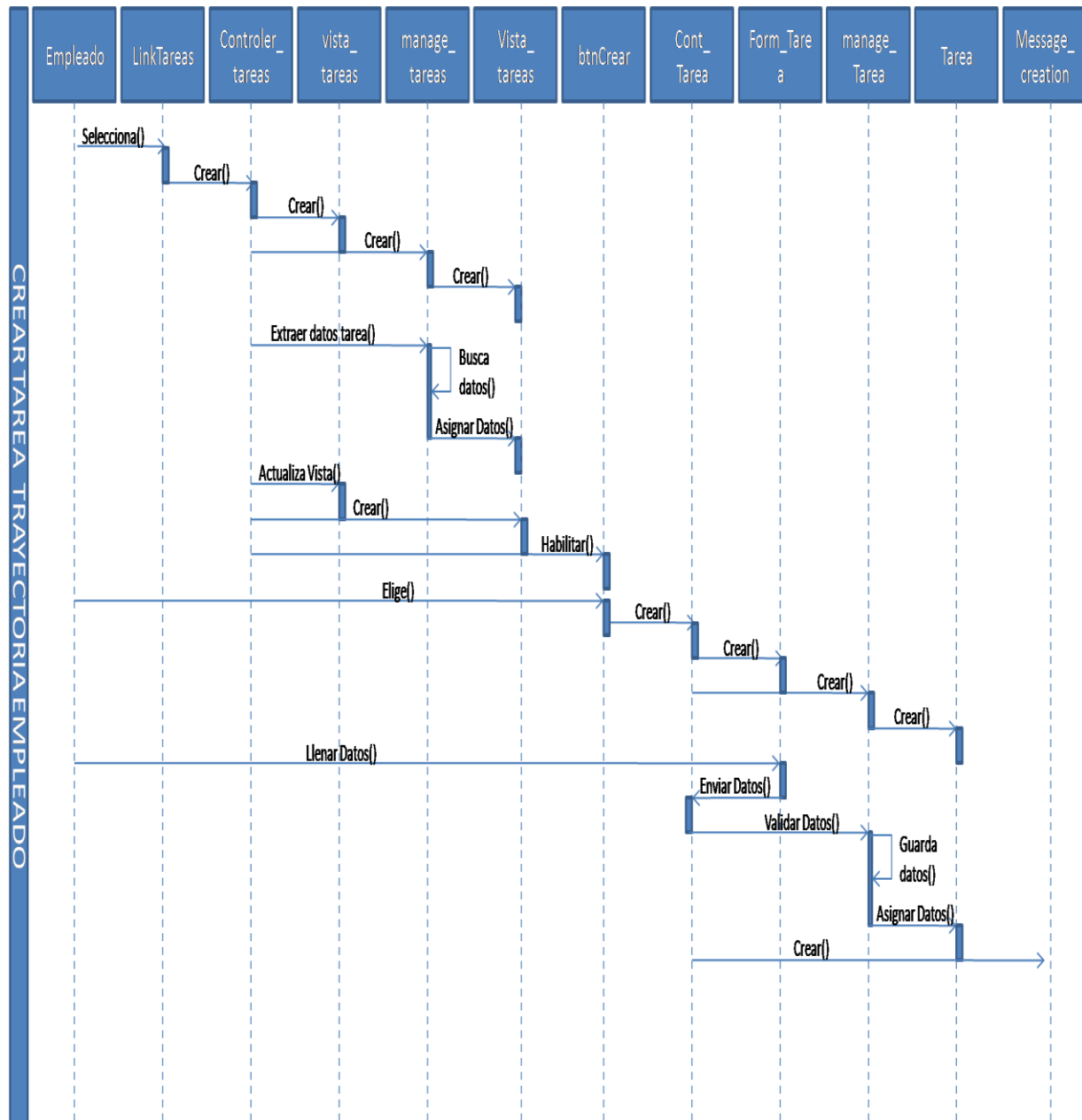
MODULO DE PLANIFICACION ANUAL

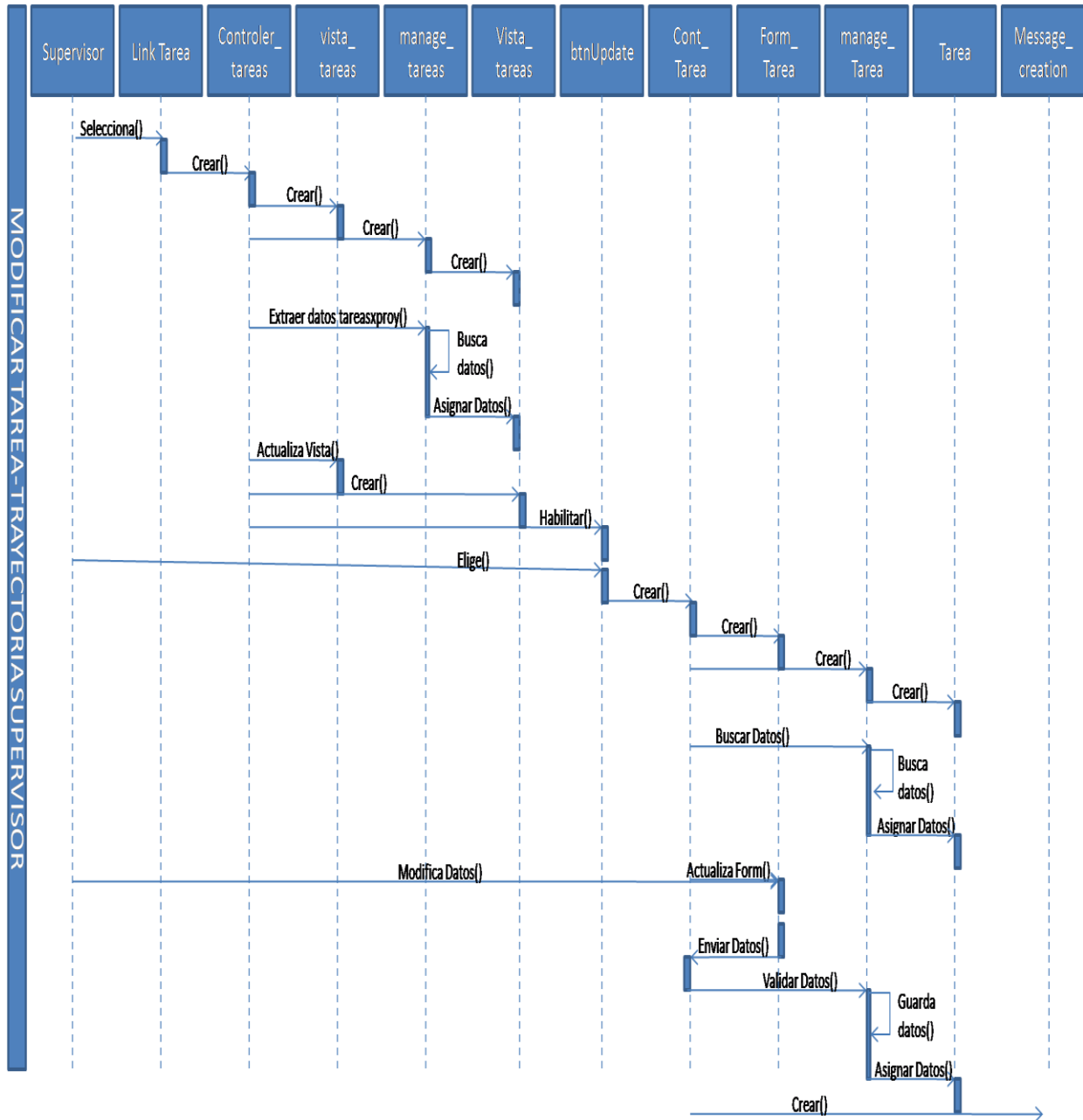


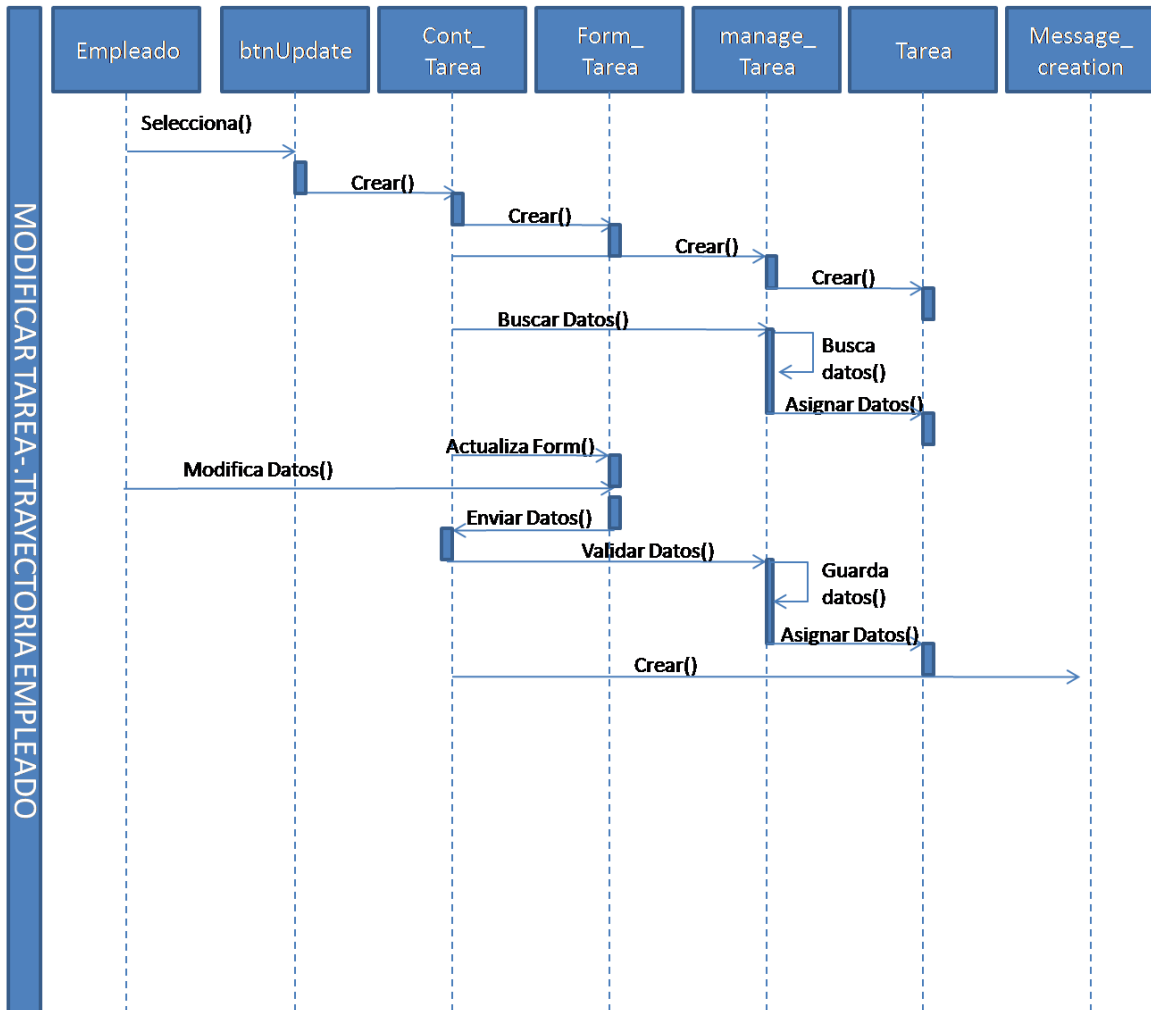


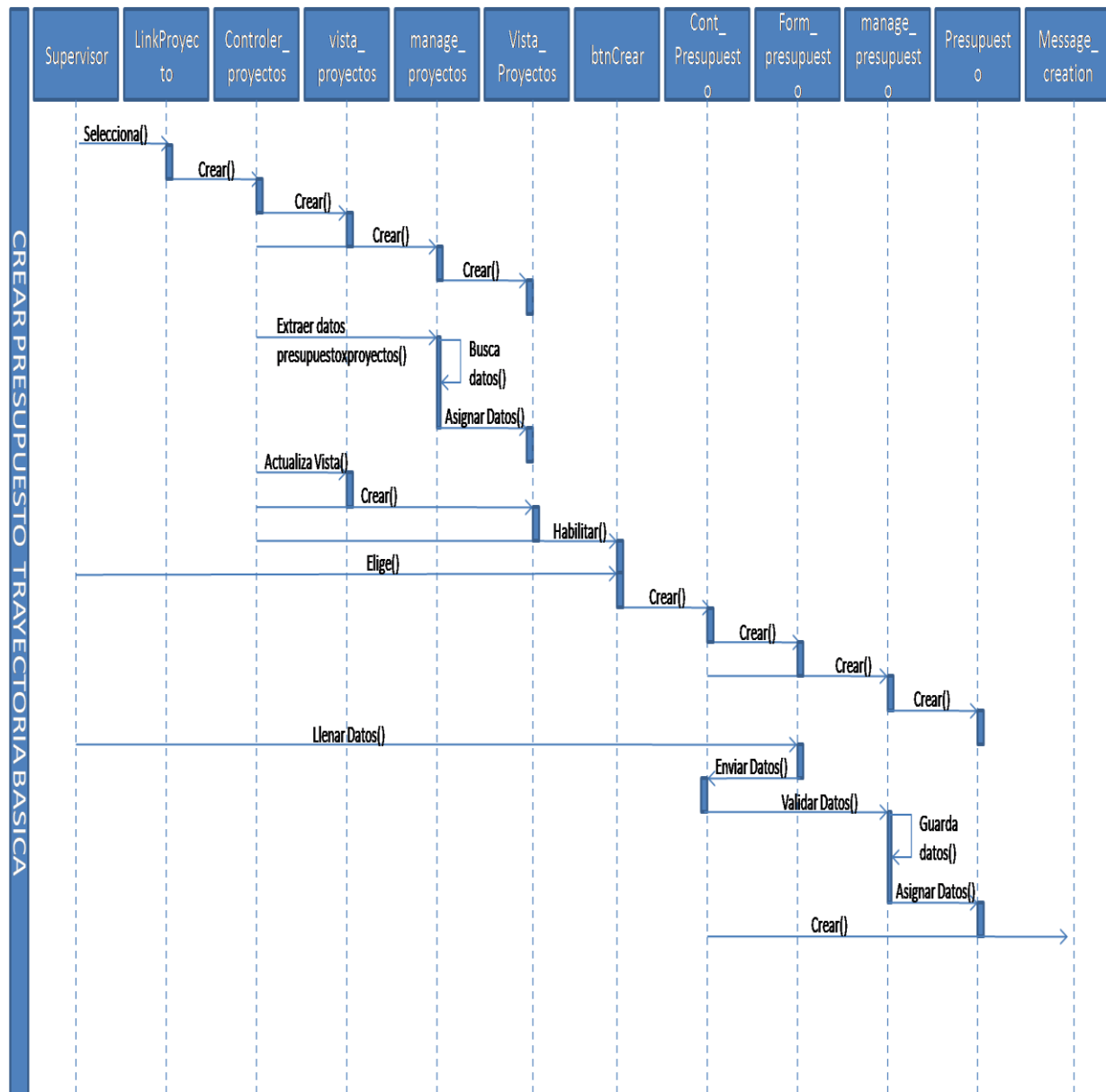


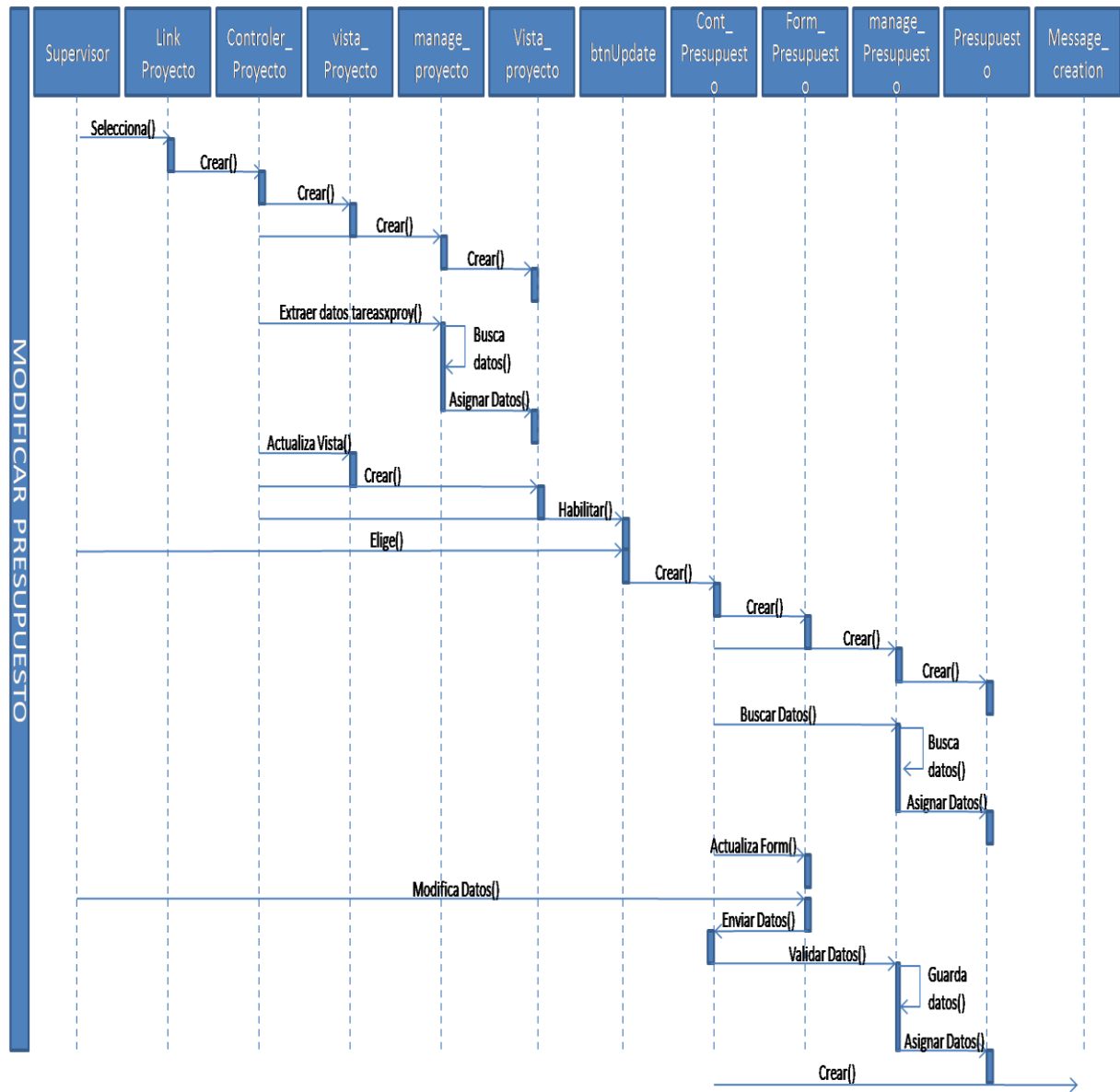


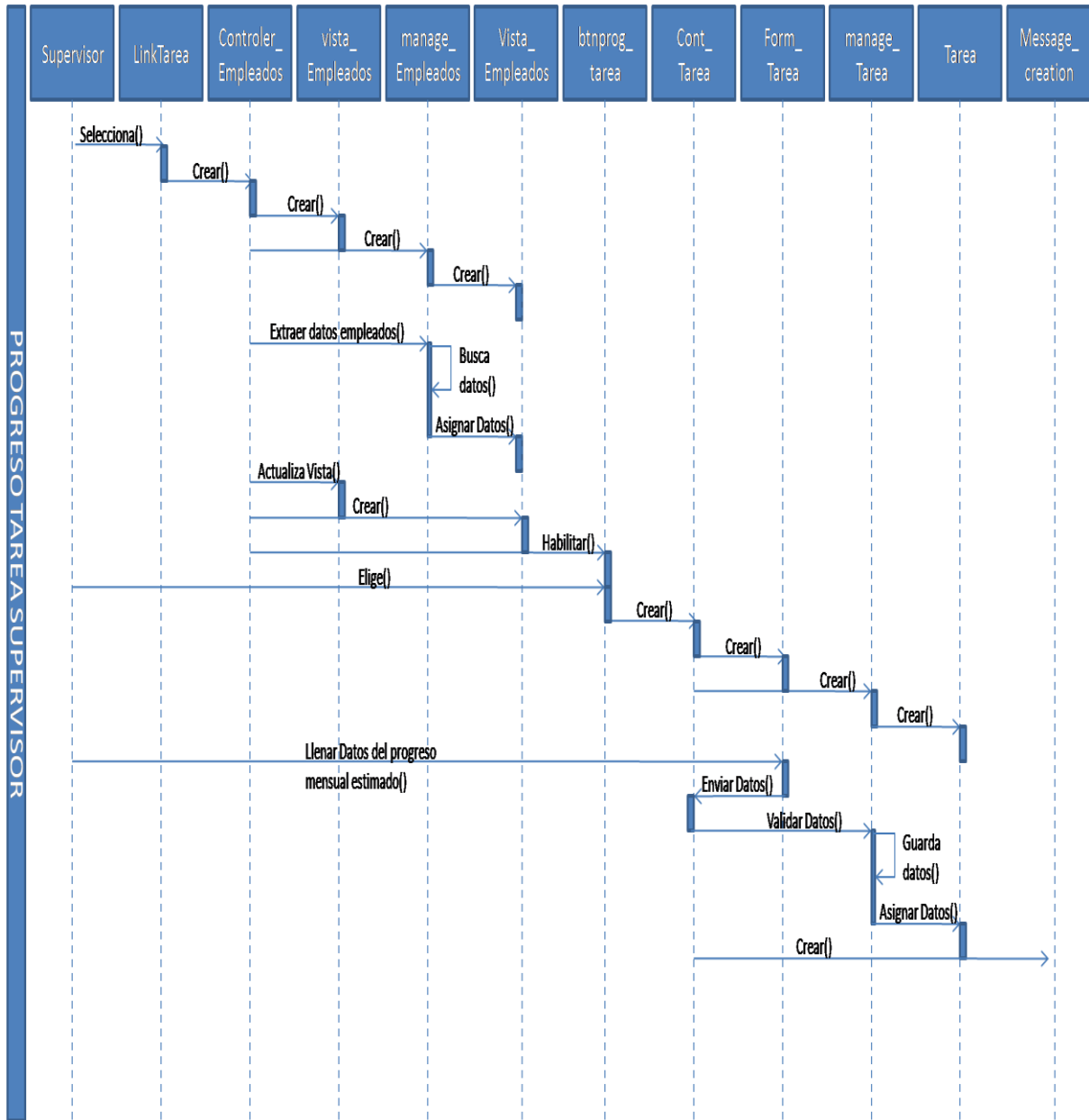


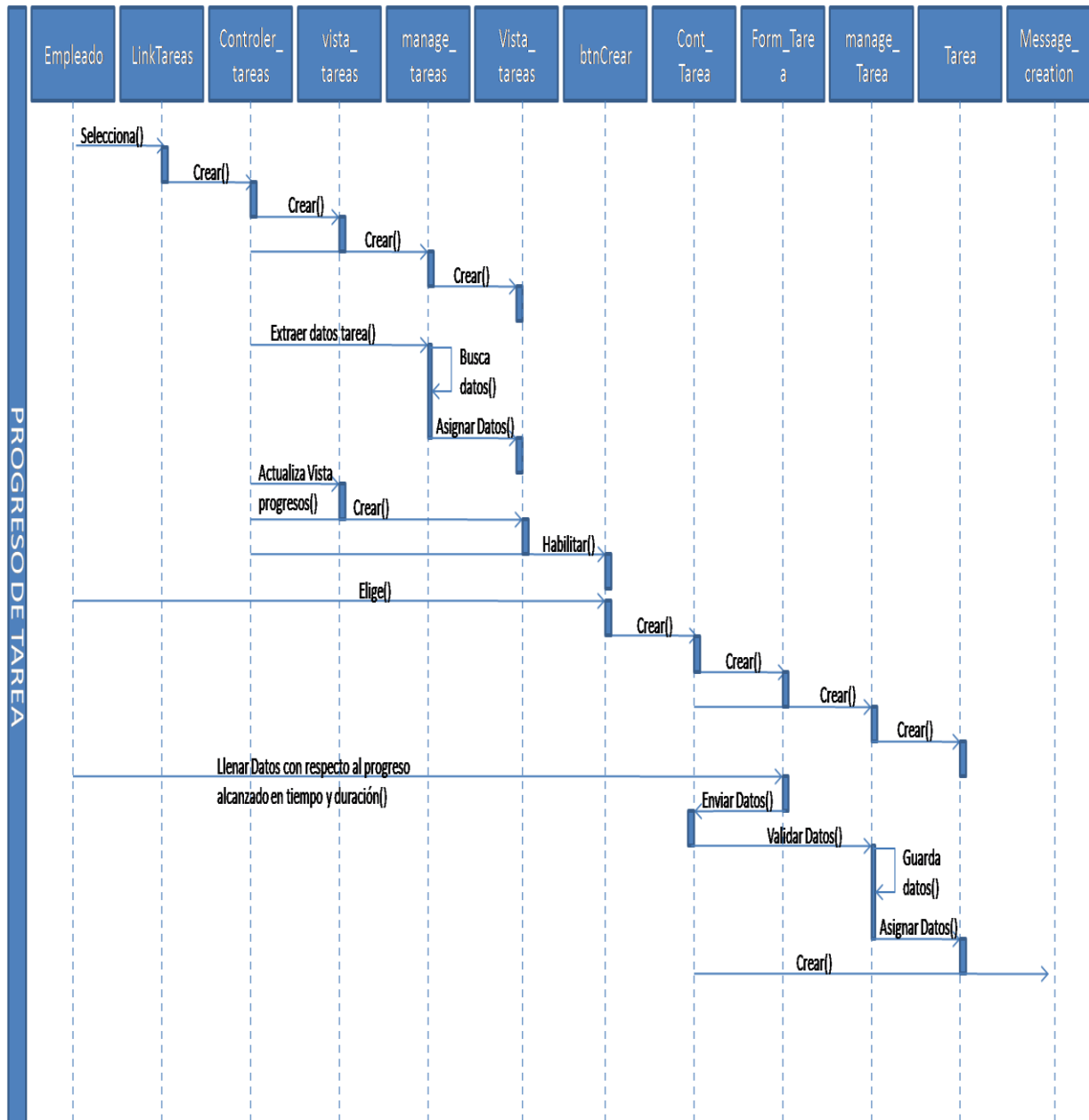


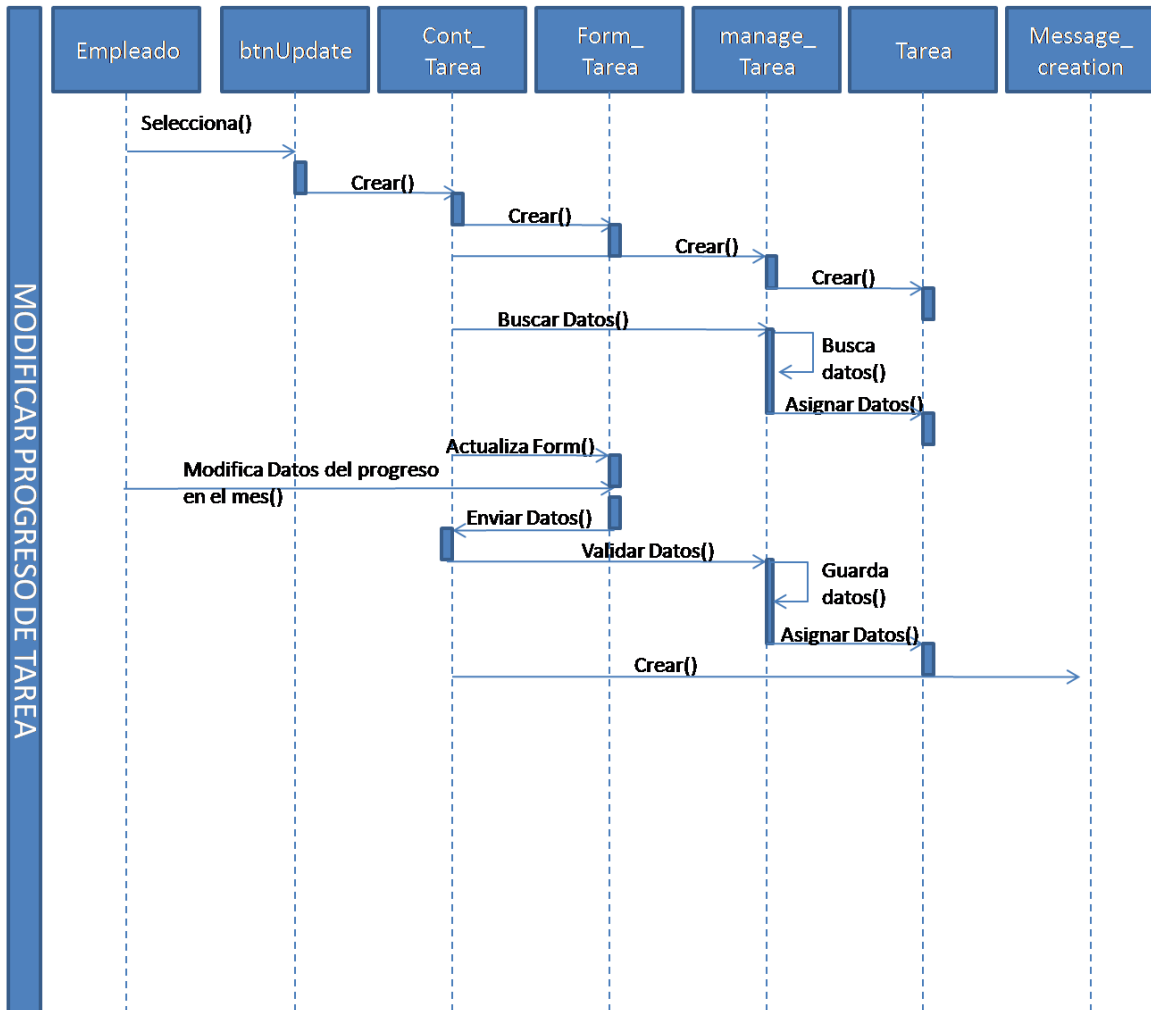




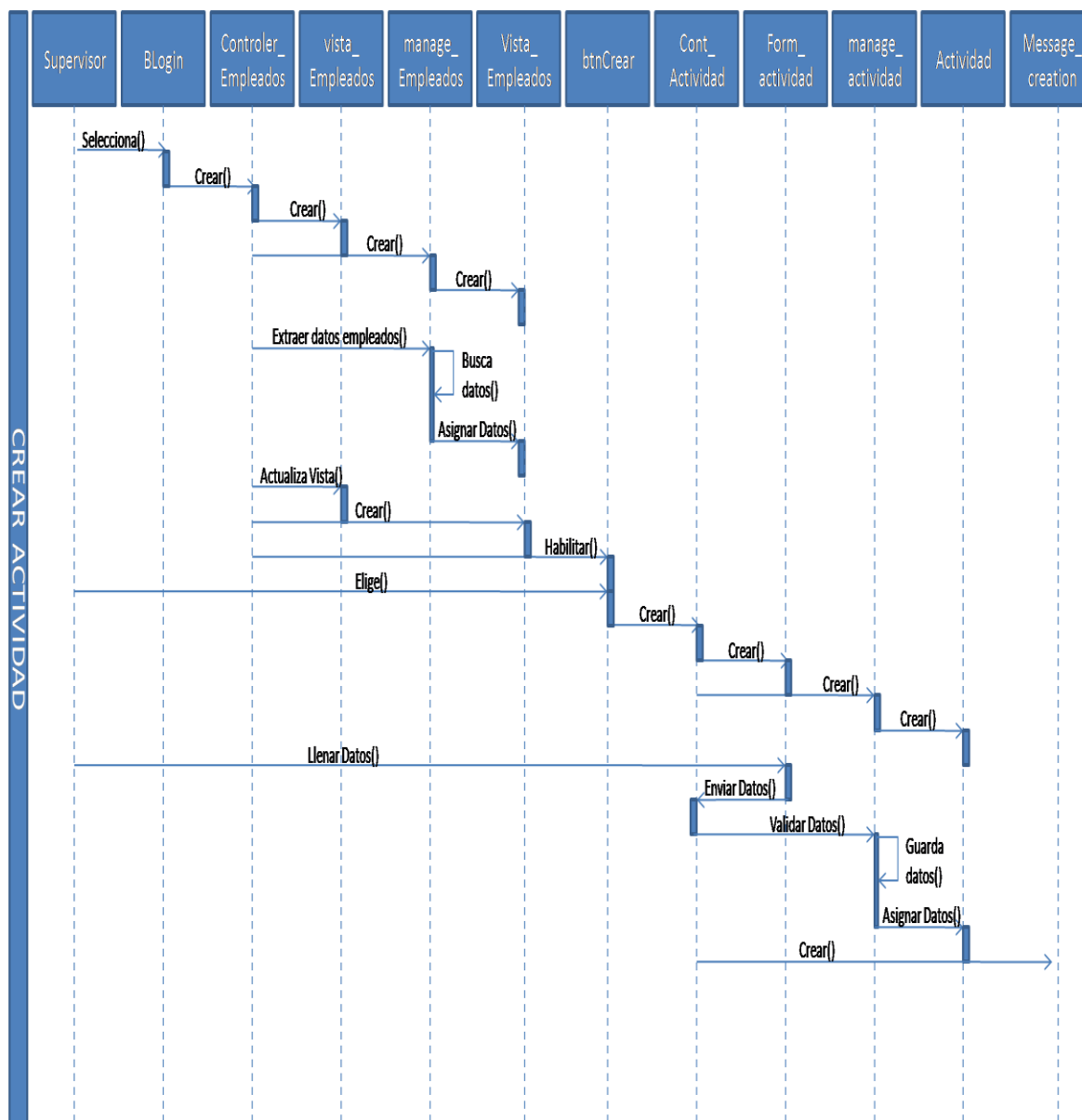


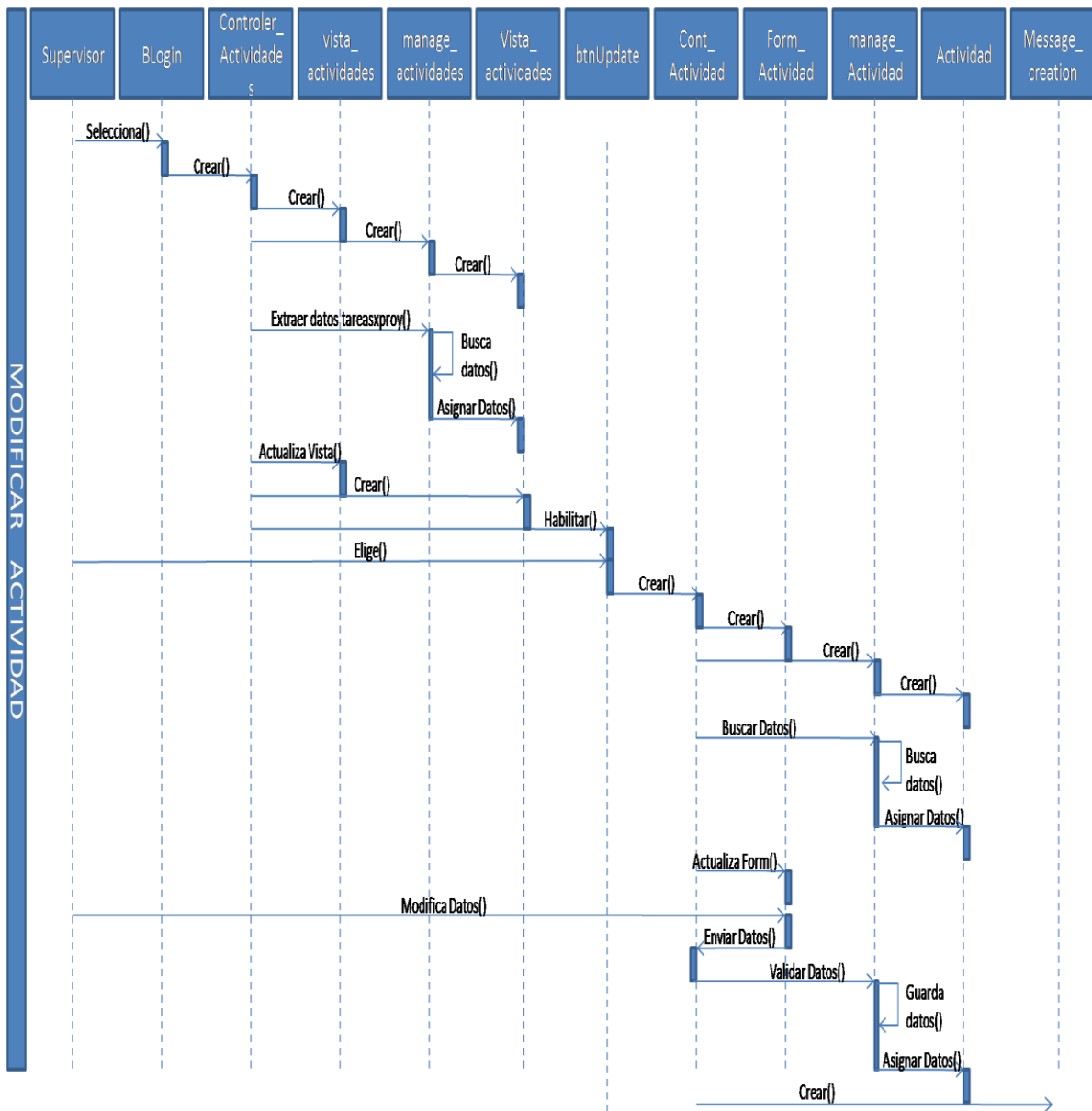




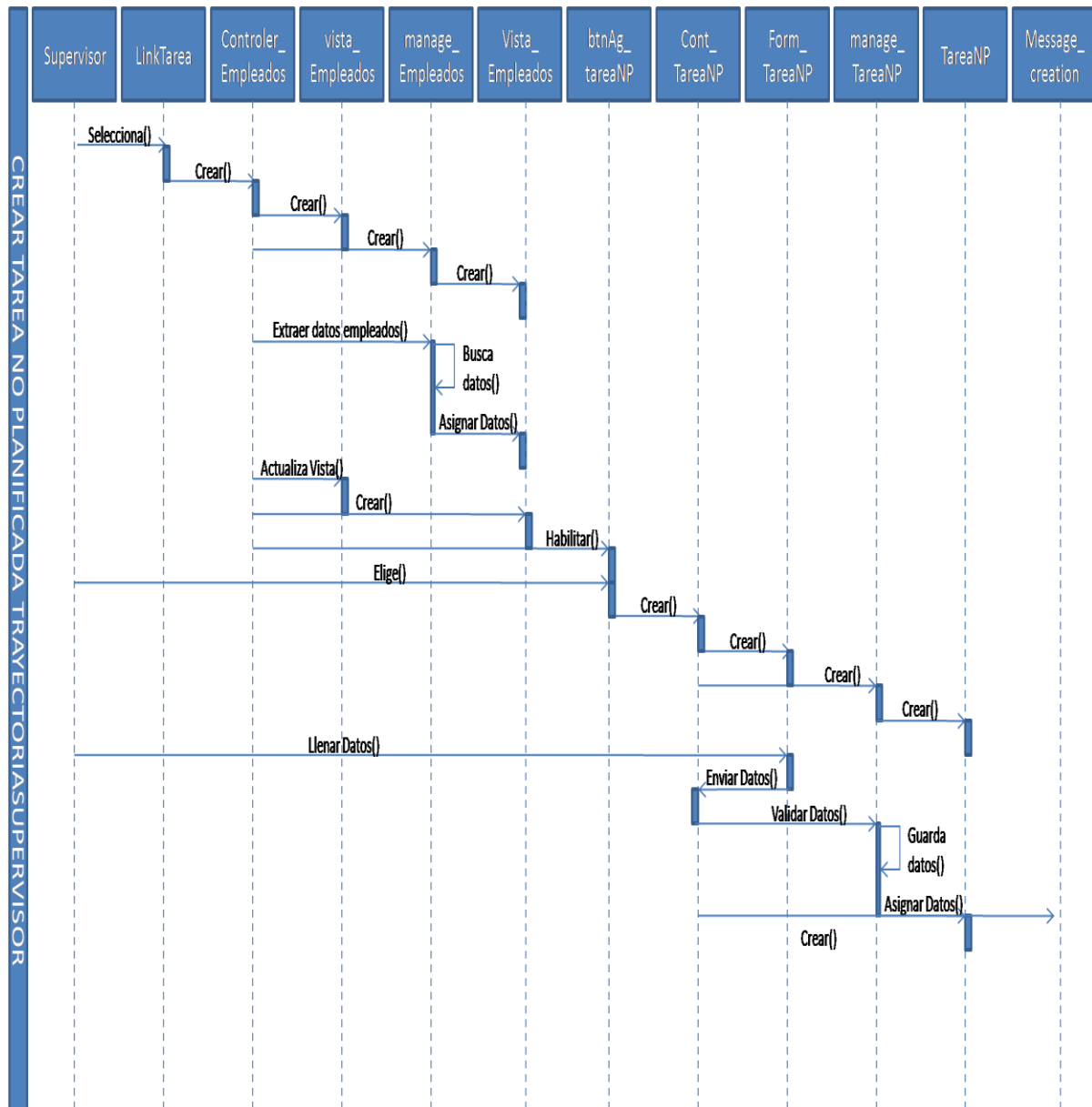


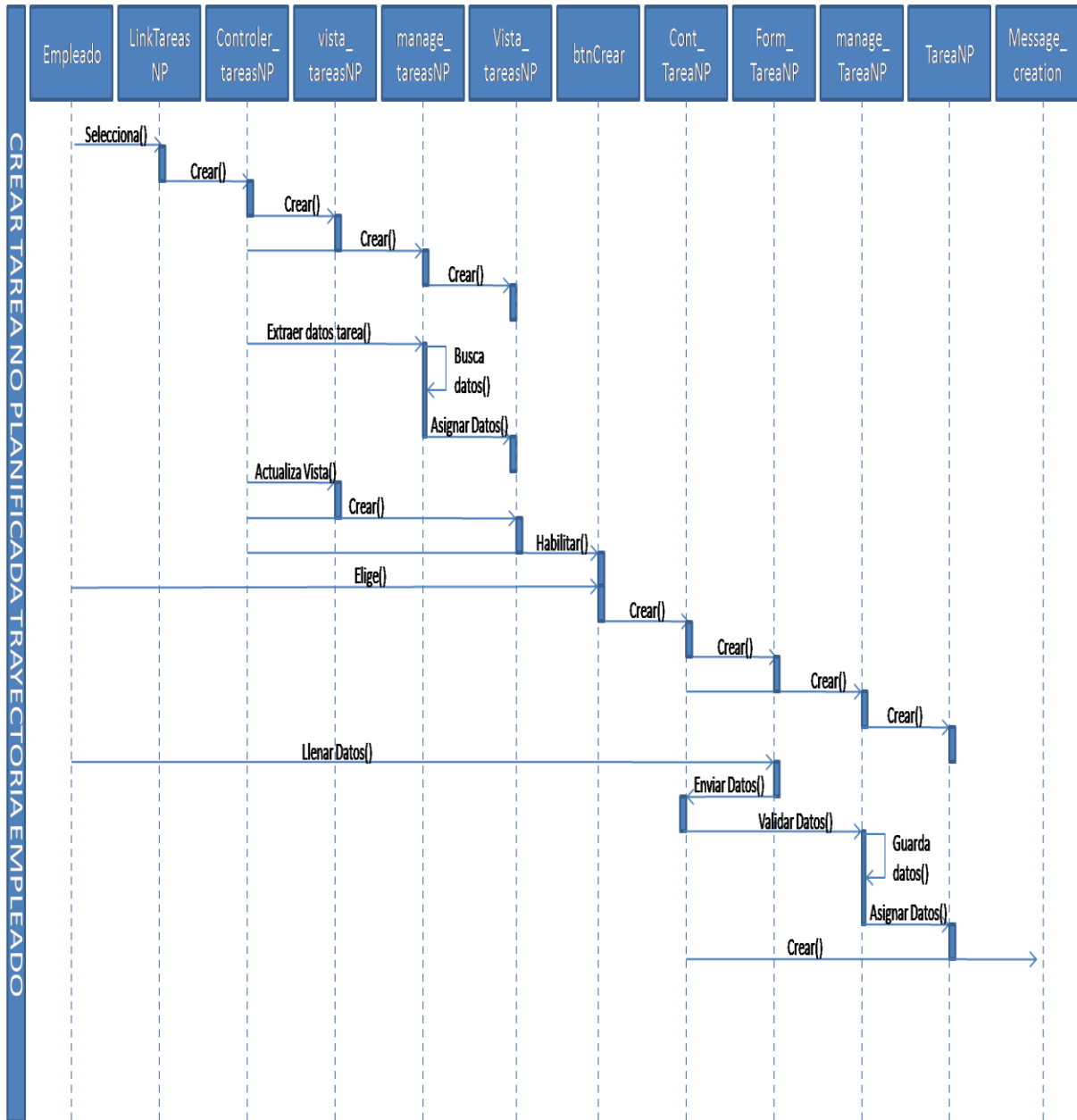
METODO DE DESARROLLO PROFESIONAL

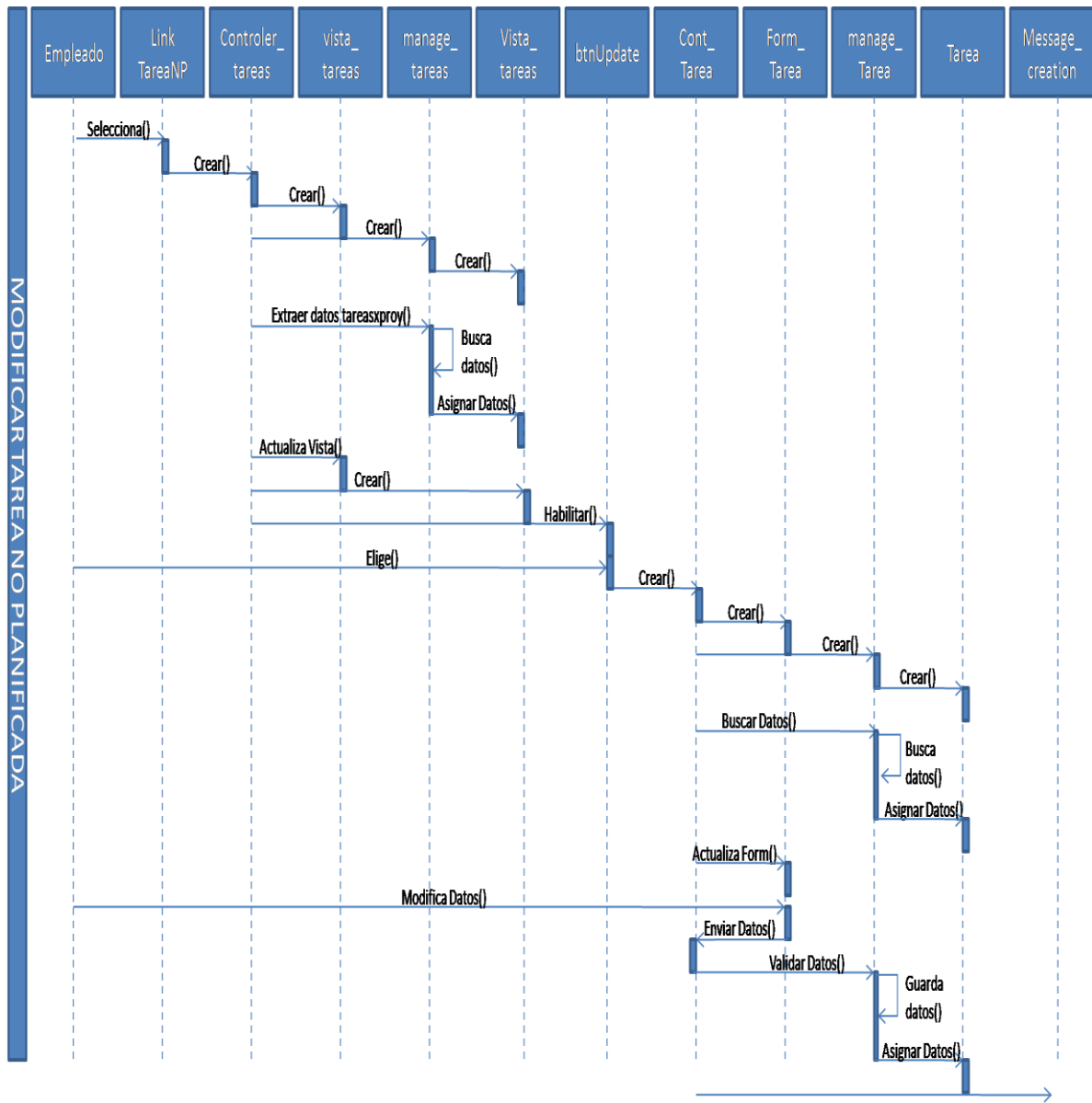




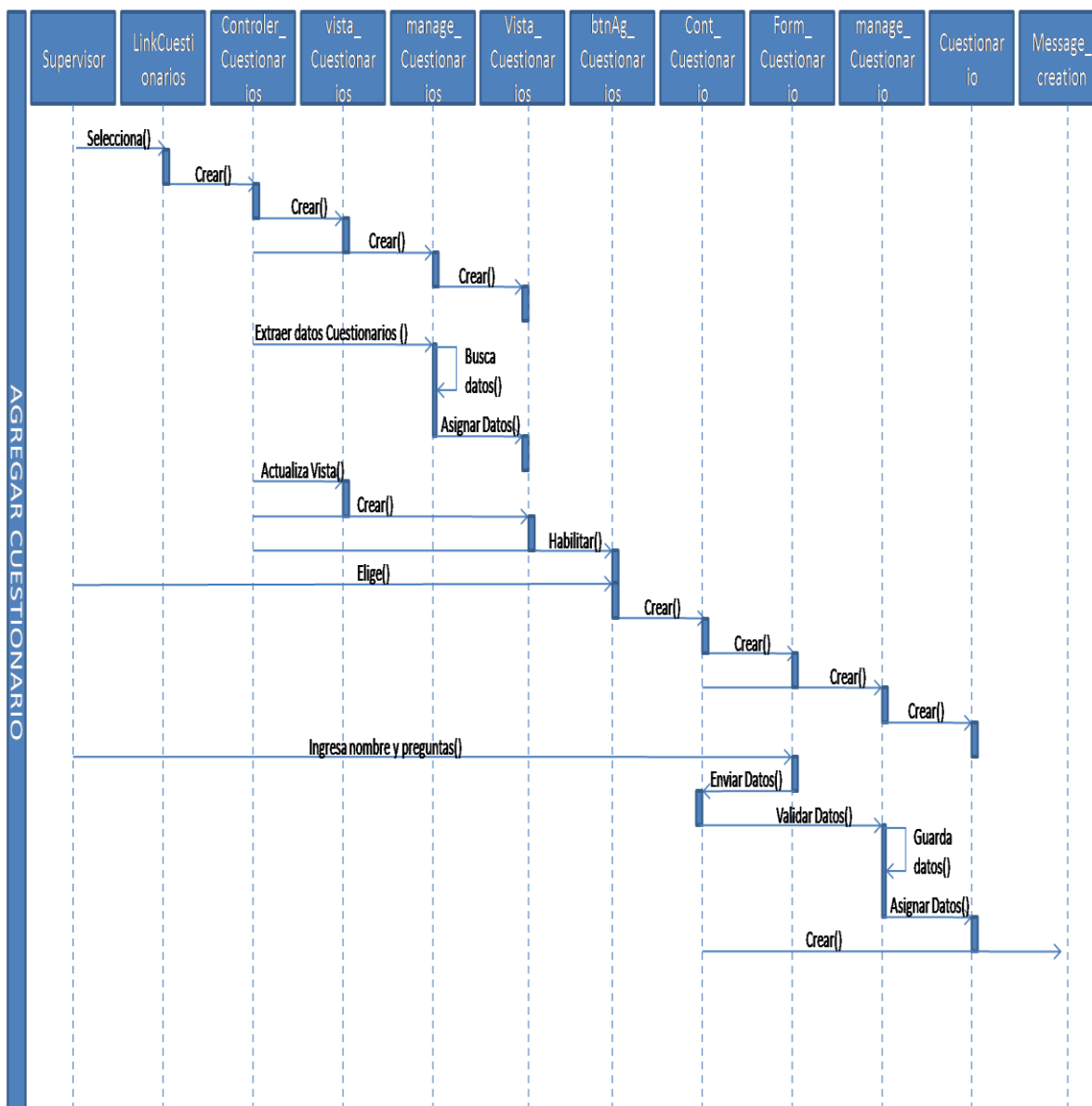
METODO DE ACTIVIDADES NO PLANIFICADAS

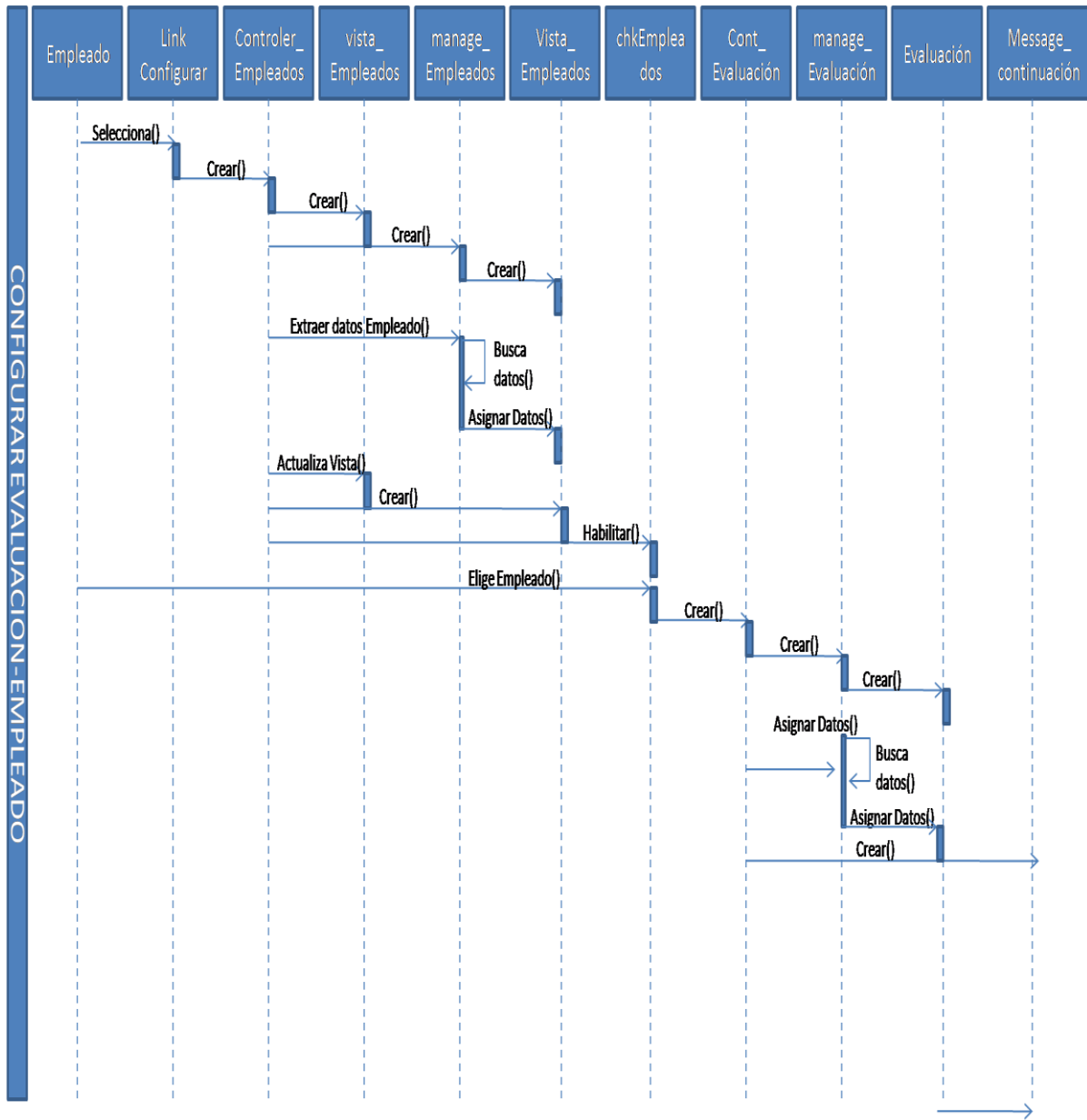


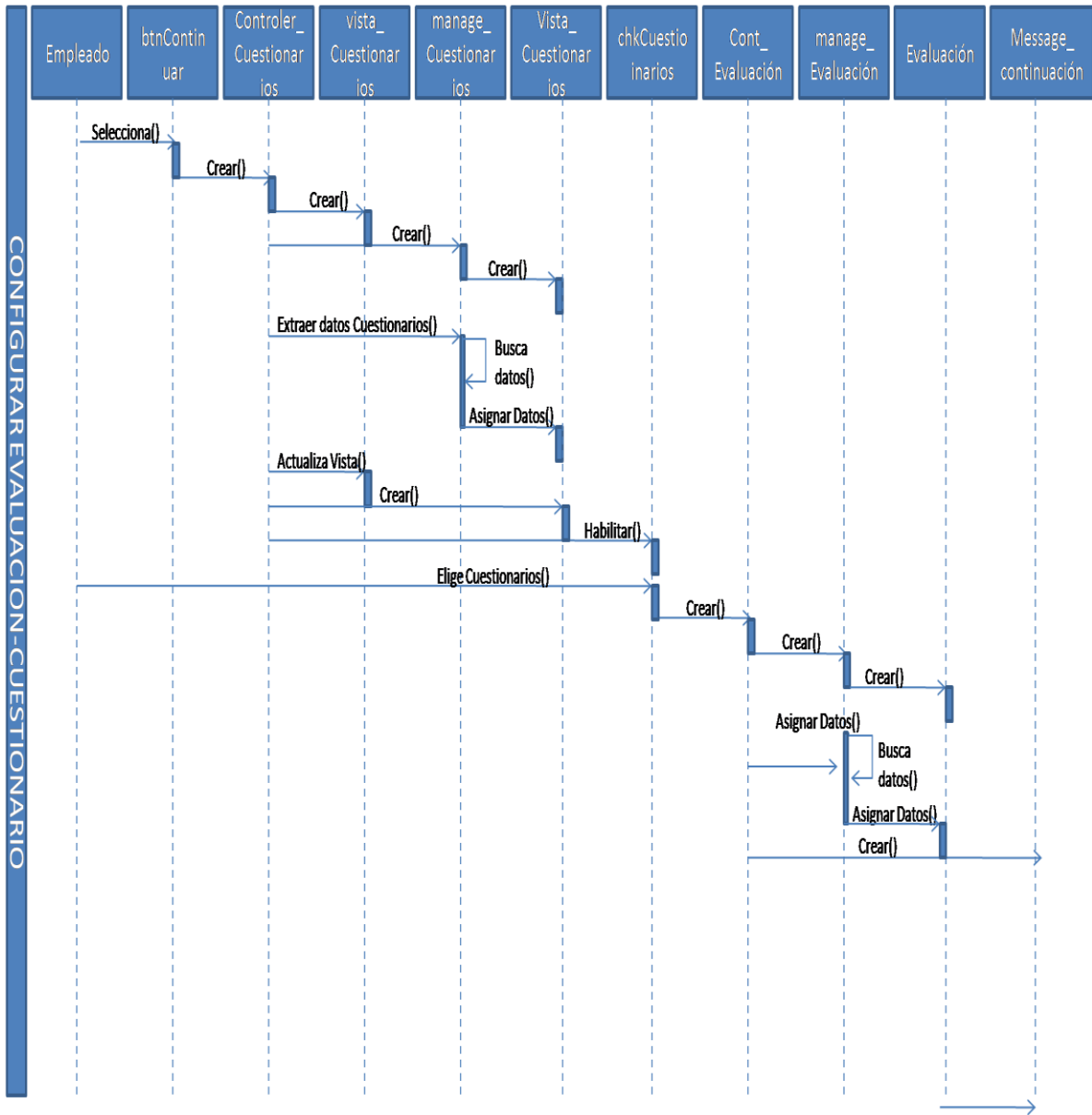


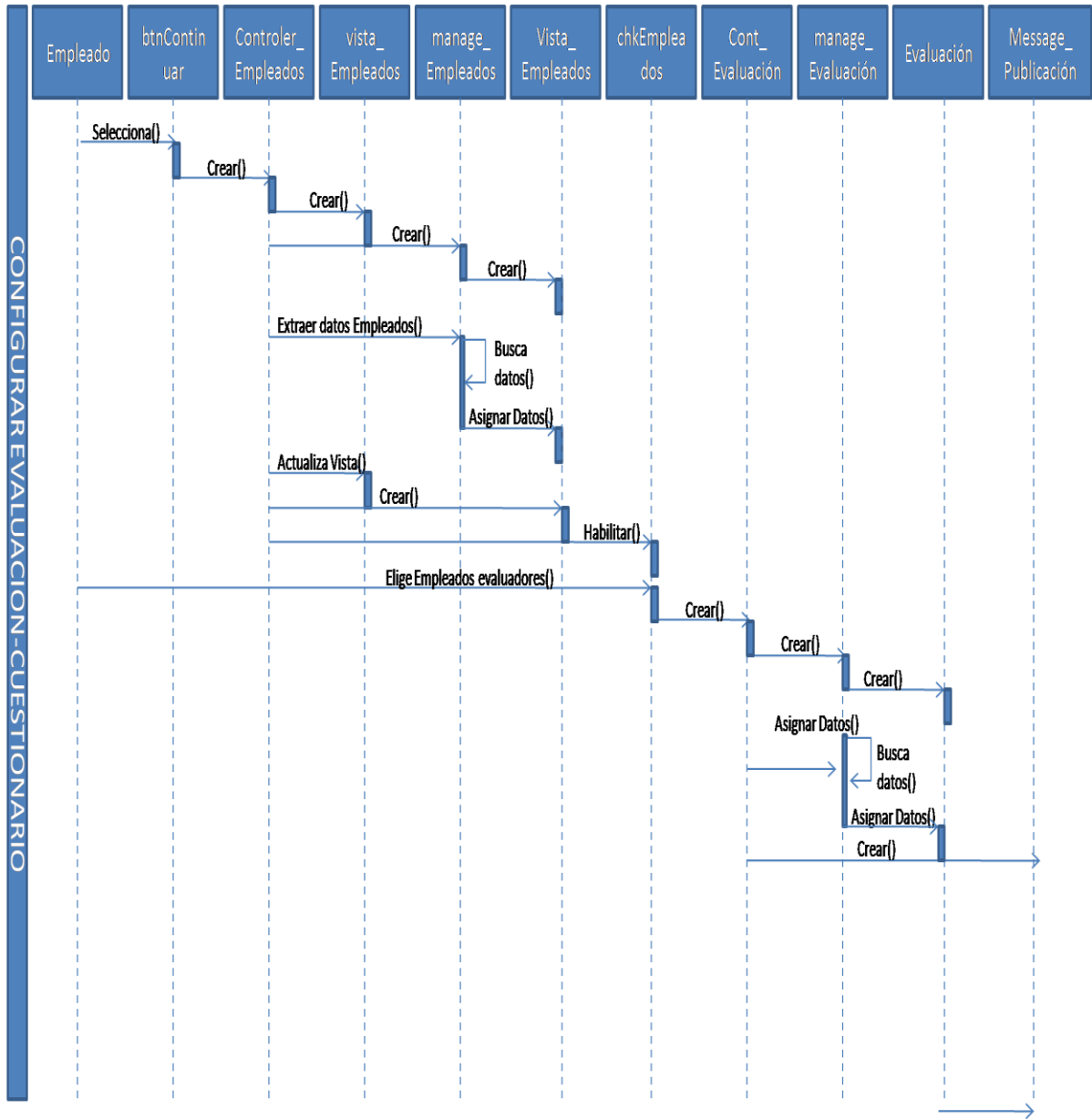


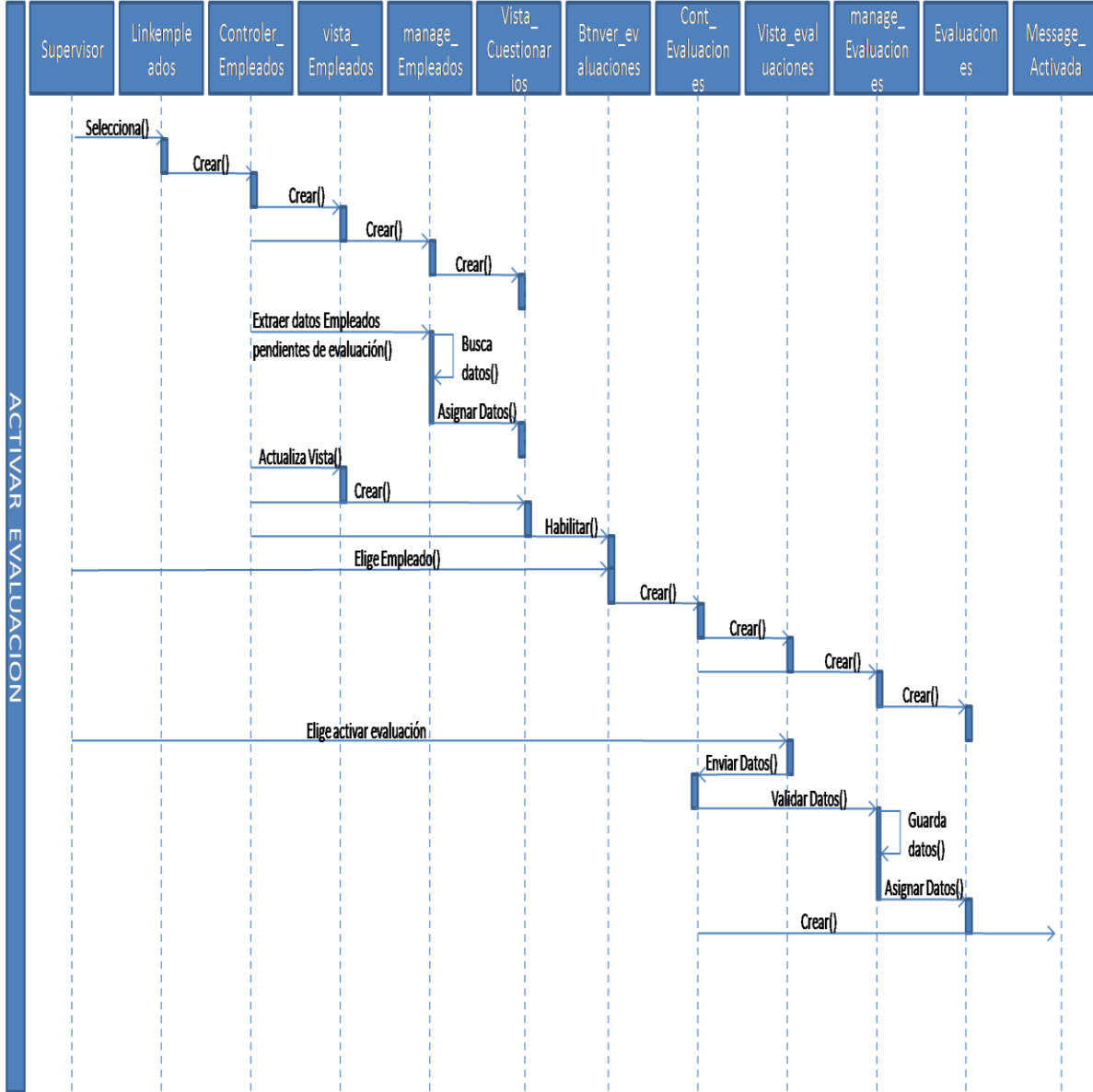
MODULO EVALUACION ANUAL

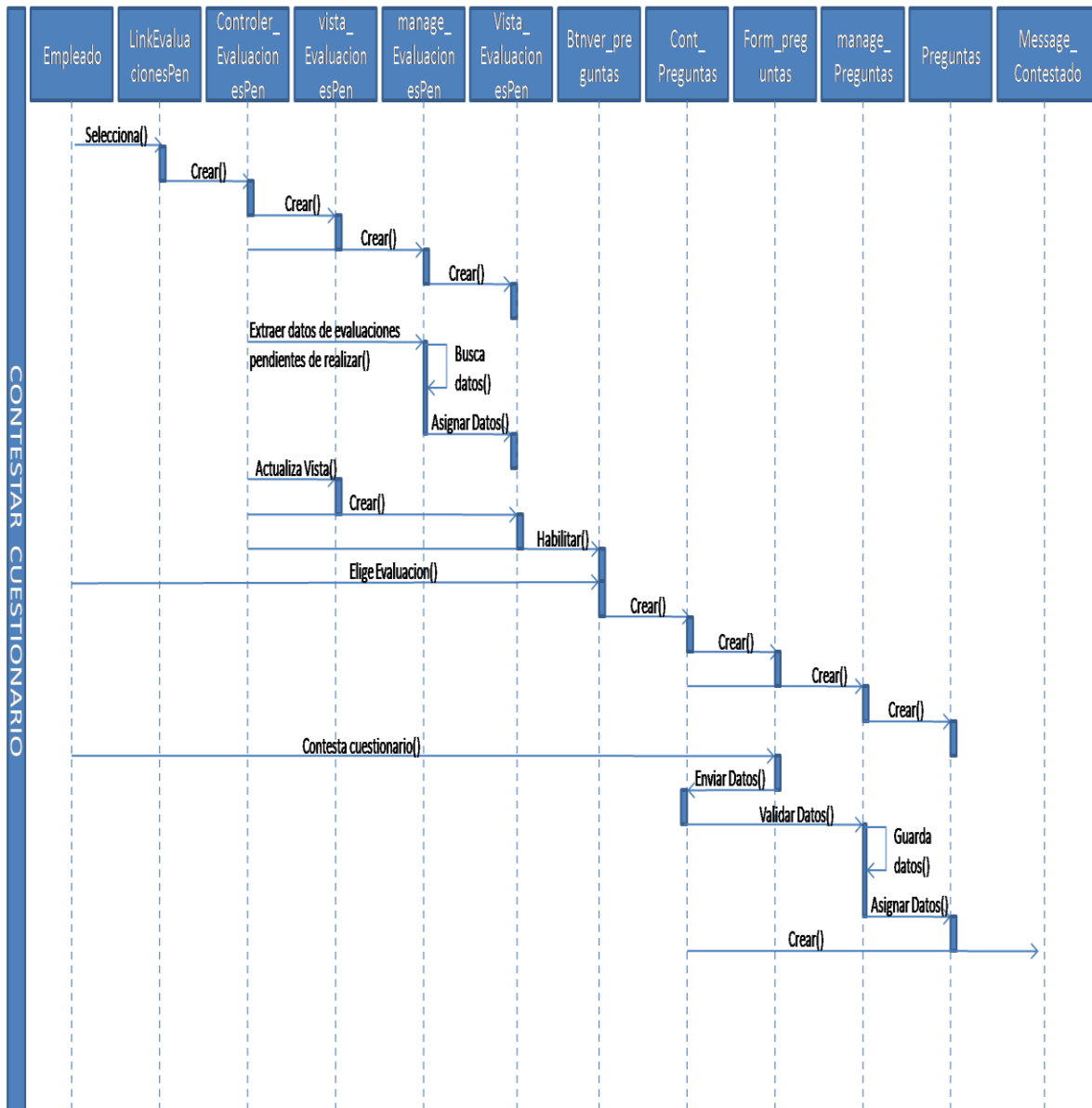




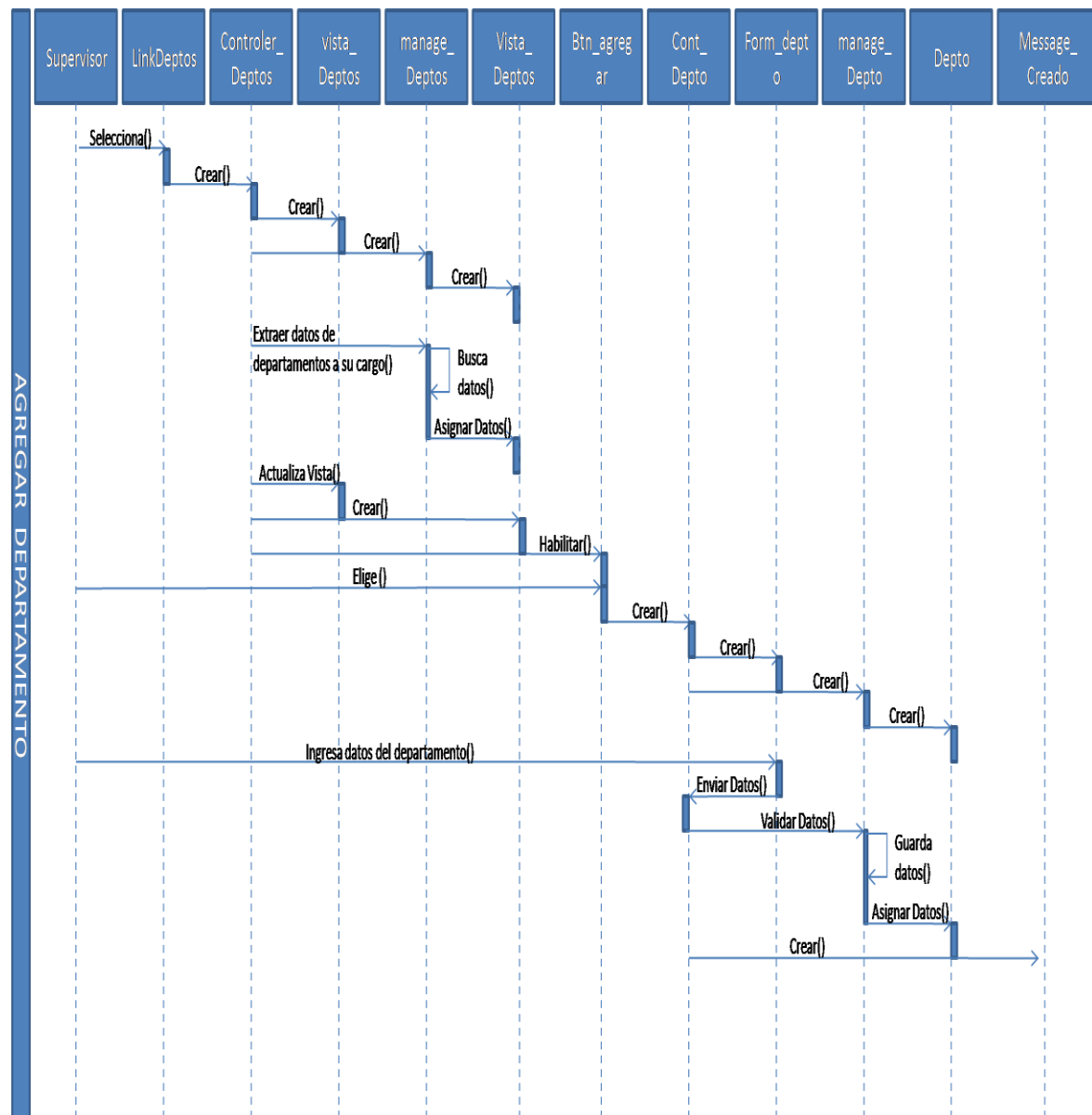


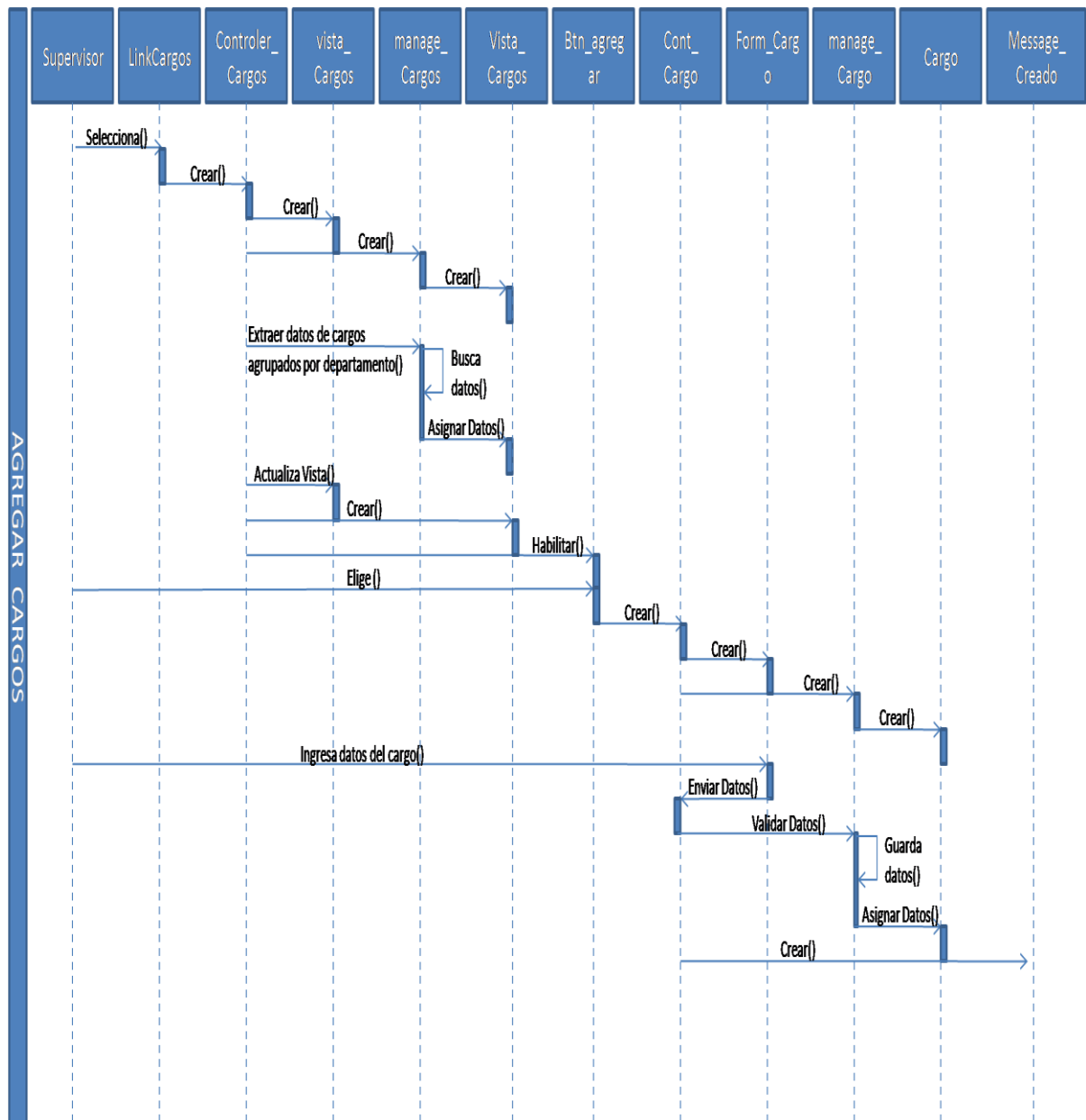


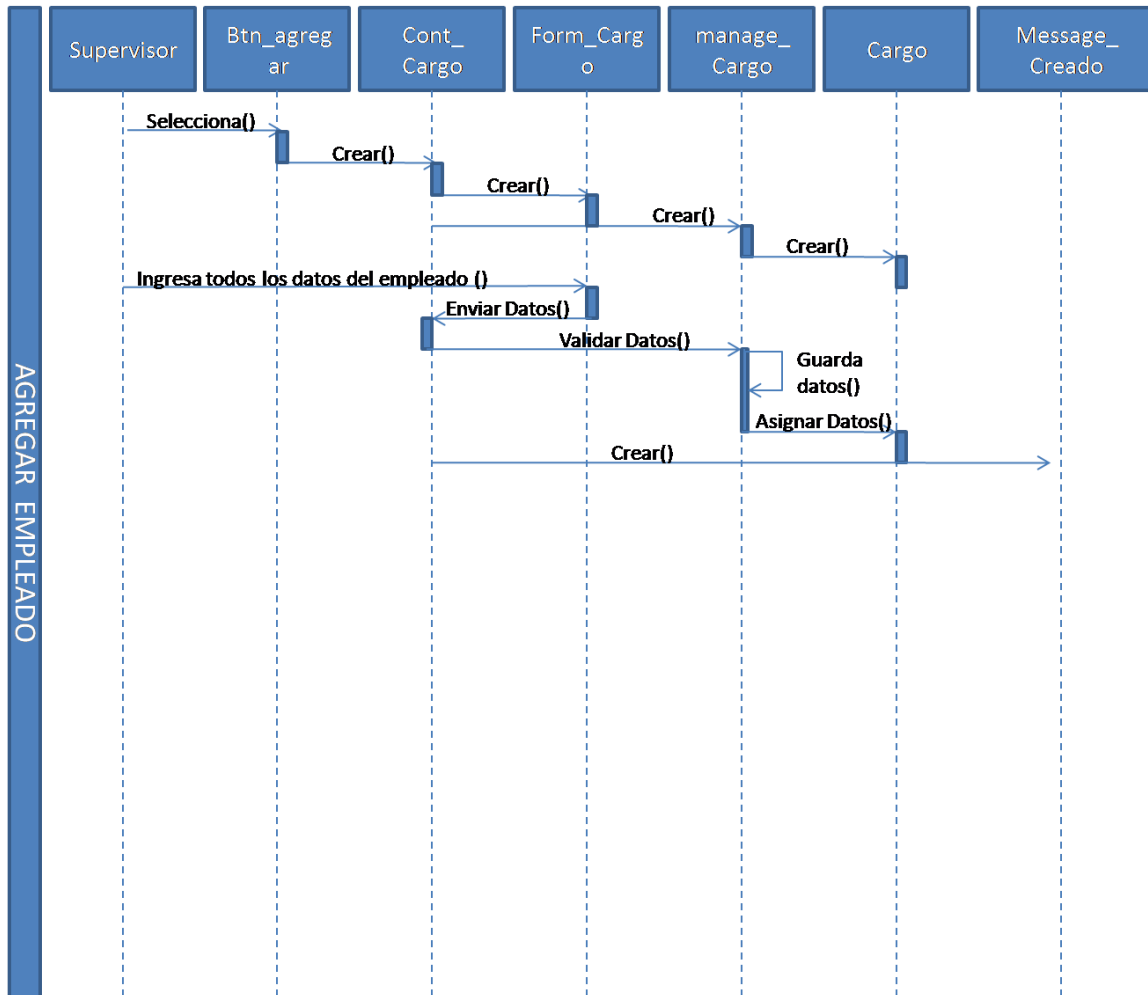




MODULO DE SEGURIDAD

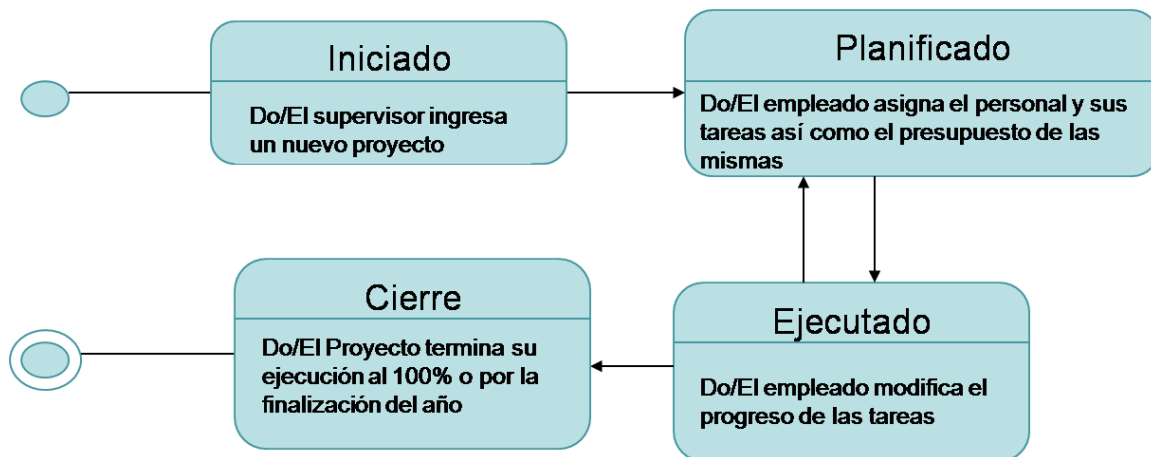




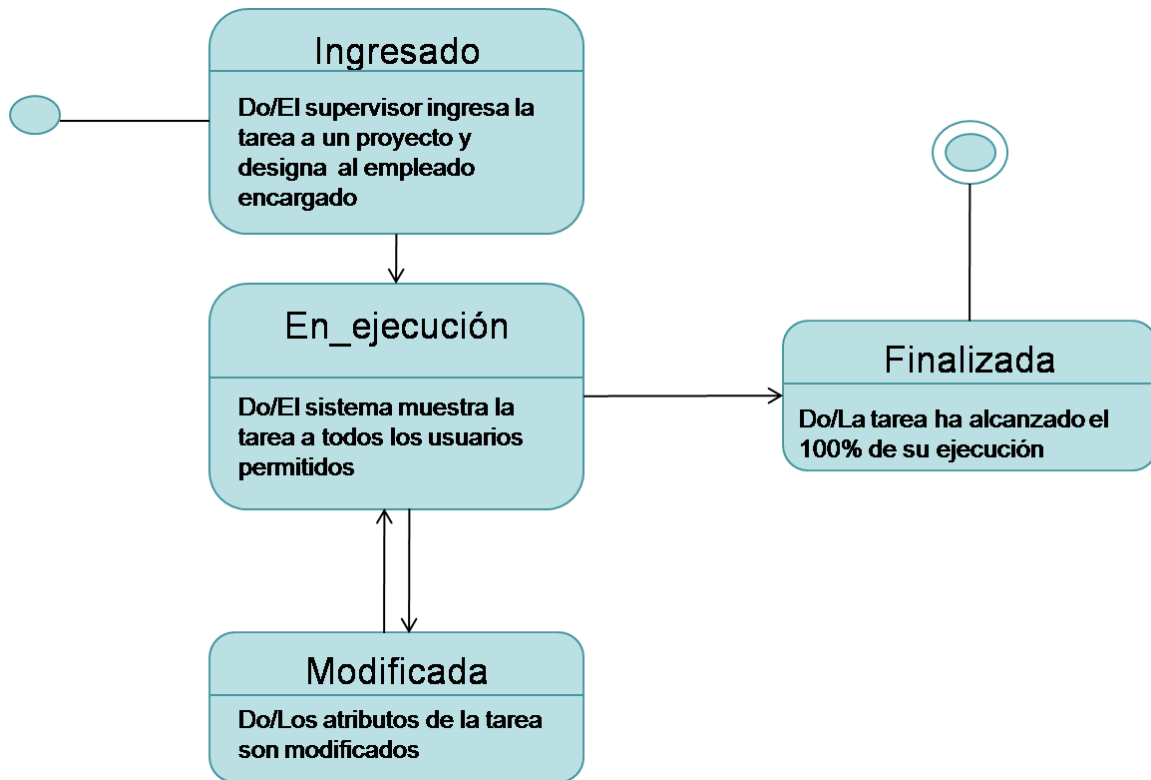


5.1.16.4 Gráficos de estado

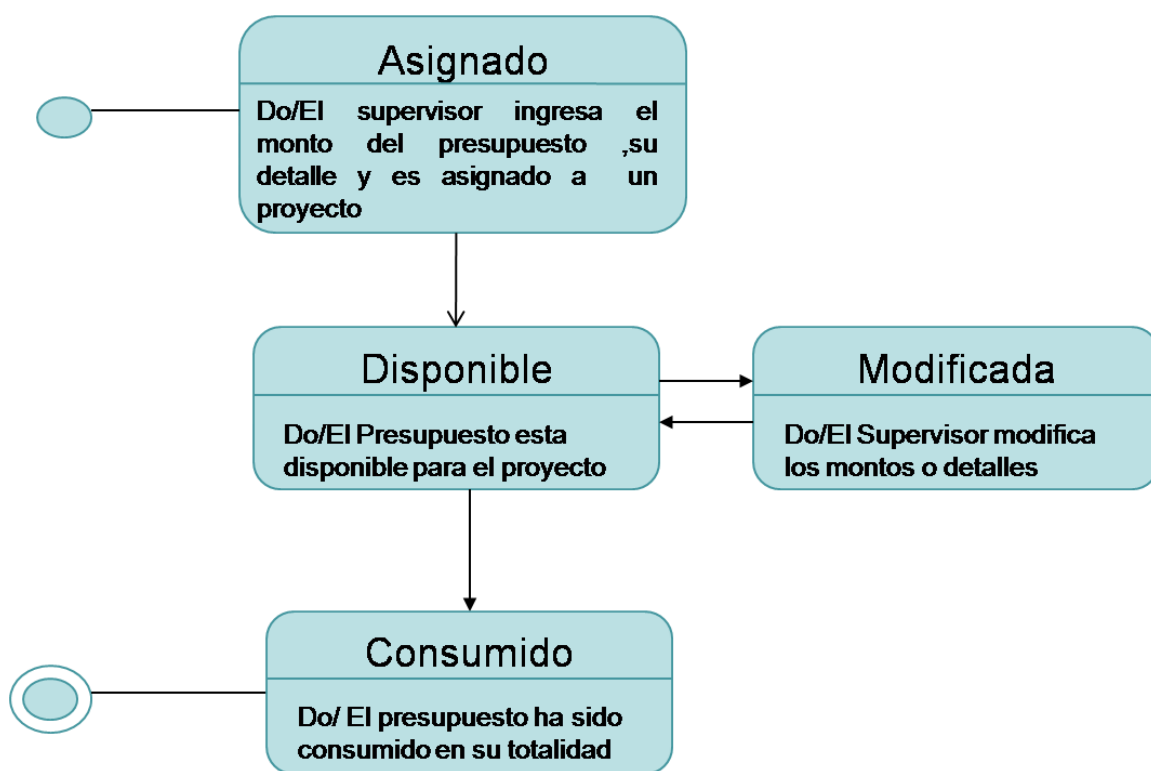
PROYECTO



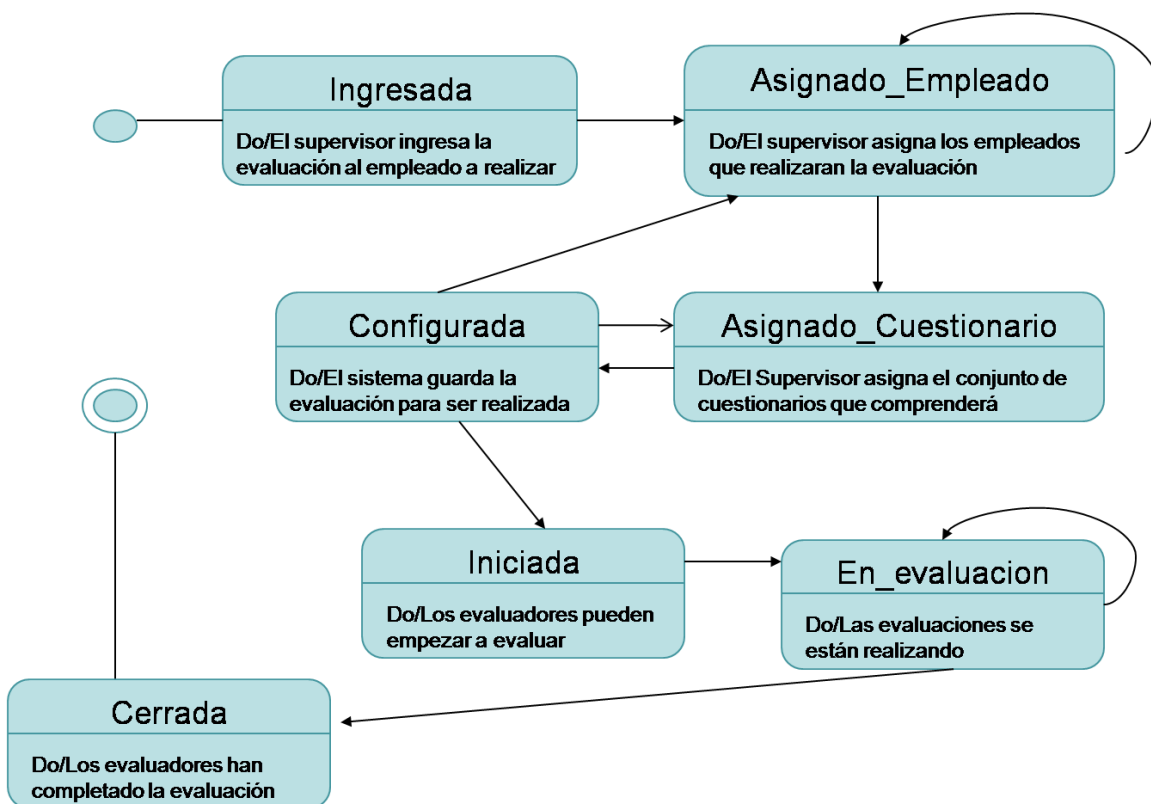
ACTIVIDAD



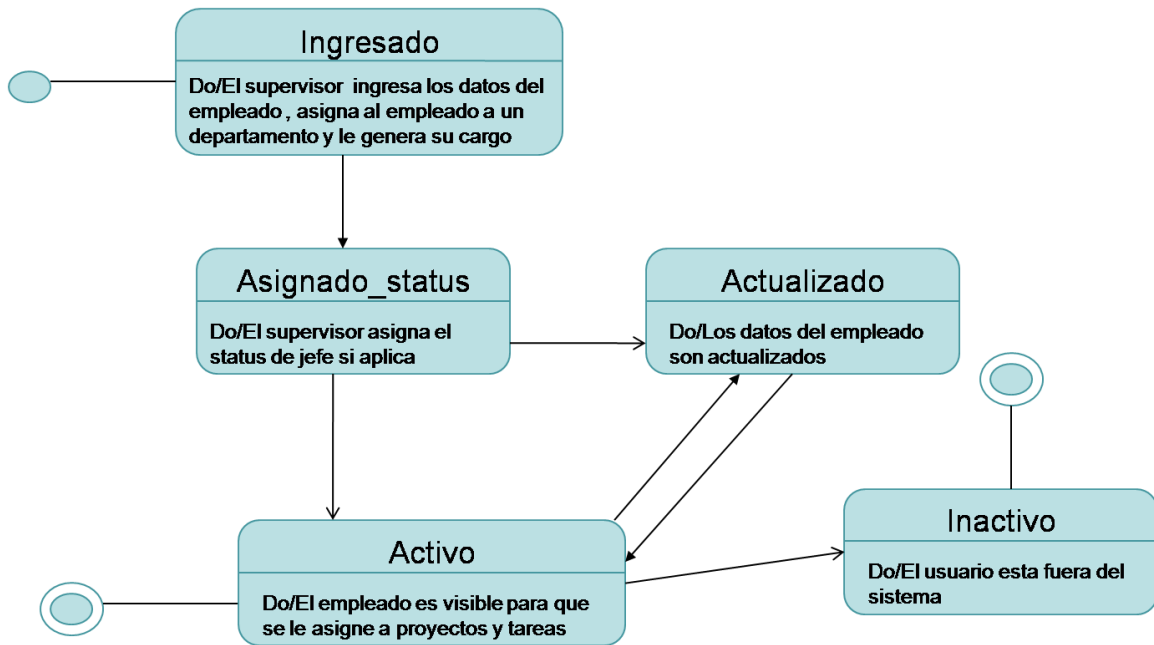
PRESUPUESTO



EVALUACION



EMPLEADO



5.1.16.5 Glosario general de términos UML

Objetivo

Proporcionar una descripción clara de cada uno de los términos relacionados al proyecto.

Glosario

Actividad

En un diagrama de actividad, una actividad representa acciones que el sistema realiza.

Abstracción

Es una simplificación o modelo de un proceso, objeto de la vida real o concepto complejo. Resalta el comportamiento y las características relevantes de aquello que es demasiado complicado para entender como un todo. La abstracción nos permite simplificar nuestro entendimiento del mundo que nos rodea, de tal manera que nuestro entendimiento se vuelve útil sin convertirse en algo abrumador.

Actor

Representa un grupo de personas que obtienen un beneficio al interactuar con el sistema. Puede representar un sistema externo que obtiene o provee un beneficio a la aplicación.

Caso de uso

Describe un conjunto de interacciones entre un actor y el sistema que proveen un beneficio concreto al actor, el beneficio debe ser significativo por si mismo. Proporciona un solo beneficio y este es lo suficientemente significativo.

1.1. Nombre

Frase breve que describe el propósito del caso de uso.

1.2. Descripción

Breve enunciado que explica el propósito del caso de uso con énfasis en el beneficio de los actores. Si no es posible explicar el propósito en un breve párrafo, probablemente, el caso de uso no esté correctamente enfocado.

1.3. Condiciones Previas

Párrafo corto que enumera los casos de uso que deben ser completados antes que el caso de uso pueda ser iniciado, y su dependencia.

1.4. Restricciones de la implementación

Párrafo Breve que describe como el sistema será usado para completar el caso de uso. Por ejemplo un caso de uso puede ser iniciado por un actor que se encuentra tras un firewall. Descuidar esta clase de restricciones puede tener serias consecuencias, por lo que es importante obtener esta información desde el inicio del proyecto.

1.5. Flujo normal de eventos

Es una lista ordenada de las interacciones del actor y el sistema, que describe todas las entradas al sistema y las respuestas de este, que constituyen el flujo normal del caso de uso. Muestra el propósito del caso de uso mostrando las interacciones con el sistema cuando todo ocurre de acuerdo con lo planeado.

1.6. Flujo alterno de eventos

Es una lista ordenada de interacciones que describe todas las entradas y respuestas del sistema que constituyen una única ruta alternativa a través del caso de uso. El flujo alterno de eventos muestra la respuesta del sistema a variaciones del usuario, tales como el ingreso de datos inválidos o el intento

de realizar pasos en un orden inusual. Esta sección se repite por cada flujo alterno.

1.7. Excepciones o flujo de errores de eventos

Una lista ordenada de interacciones que describe todas las entradas y respuestas del sistema que constituyen una única ruta de excepción a través del sistema. Muestra la respuesta del sistema ante los errores, tales como la falta de disponibilidad de recursos internos o externos. Esta sección se repite por cada flujo de excepción en el sistema.

1.8. Diagrama de actividad

Muestra todos los flujos de eventos para el caso de uso en un solo diagrama. Complementa los flujos de eventos y provee una valiosa herramienta para medir la complejidad del caso de uso.

1.9. Requerimientos no funcionales

Es un corto párrafo que describe cualquier criterio para el caso de uso que no pueda ser descrito fácilmente en el flujo de eventos. Por ejemplo, el sistema debiera proporcionar la respuesta del caso de uso en menos de 5 segundos; o que exista un límite máximo de 7 clics del ratón para terminar el caso de uso.

1.10. Notas

Es una lista opcional de casos propuestos que no encajan correctamente en ninguna otra categoría. Puede incluir restricciones en la funcionalidad del sistema.

1.11. Casos abiertos

Lista opcional de preguntas para los usuarios o clientes.

Un caso de uso debe cumplir con:

1. El caso de uso proporciona una sola ventaja concreta.
2. Se puede describir su beneficio en 20 o 30 palabras, sin utilizar muchos “y” u “o”

3. Es independiente.
4. El actor tiende a completar el caso de uso en una sola sesión.
5. Puede el caso de uso ser un único caso de prueba en un Test Plan.

De lo contrario debe considerarse dividir el caso de uso.

Un caso de uso debe cumplir:

1. El actor obtiene una ganancia o cambia el sistema significativamente.
2. El actor puede detener su actividad en el sistema cuando finaliza con un caso de uso.

De lo contrario debe ser considerado incluir el caso de uso dentro de otro caso de uso.

Análisis Orientado a Objetos

Debido a que los objetos realizan el trabajo en los sistemas orientados a objetos, el análisis orientado a objetos pretende descubrir los objetos que formaran al sistema, describir sus responsabilidades, y determinar sus interacciones. Este proceso es independiente a la tecnología de desarrollo que utilizará los objetos.

Casos de uso complementarios

Casos adicionales que soportan a los casos principales identificados en la investigación preliminar.

Clase

Una clase representa a un conjunto de objetos comunes entre sí. Representa una abstracción y provee una plantilla para la creación de objetos.

Cliente

Entidad que adquiere el sistema. Destinatario del servicio; a menudo el cliente es responsable del sistema y su costo, ya sea directamente a través del cargo o en términos de una necesidad de negocio demostrable.

Encapsulamiento

En los sistemas orientados a objetos los datos y las especificaciones de la lógica del funcionamiento se encuentran ocultos dentro de cada tipo de objeto. El encapsulamiento nos permite ocultar todos los detalles engorrosos dentro de un objeto. Un objeto encapsulado permite que otros objetos dentro del sistema puedan usarlo sin depender de ningún detalle interno. Consecuentemente, los nuevos requerimientos pueden ser efectuados a través de un cambio de los detalles encapsulados, si que esto afecte el código que utiliza al objeto.

Herencia

Es el mecanismo a través del cual elementos específicos incorporan la estructura y comportamiento de elementos más generales.

Interfaz

Una interfaz define un conjunto de características relacionadas, pero no especifica su implementación. Específicamente una interfaz especifica la firma de uno o más métodos, sus parámetros y tipo de dato de retorno, pero sin los detalles de implementación.

Mensajes

Los objetos interactúan llamando métodos de otros objetos. Los objetos utilizan este mecanismo para transmitir información, solicitar un servicio o solicitar información. Estas interacciones son llamadas mensajes, los mensajes son enviados por un objeto y recibidos por otro objeto. Los mensajes son nombrados desde la perspectiva del objeto que realiza la petición, sin embargo la implementación se encuentra en el objeto que recibe la petición.

Multiplicidad

Un objeto puede tener una asociación con un único objeto, con cierto número de objetos, o con un ilimitado número de objetos.

Navegabilidad

La navegabilidad se refiere a la capacidad de mando entre los objetos. Las relaciones entre objetos a menudo son unidireccionales. Por ejemplo, el objeto automóvil controla a los objetos neumático, pero los objetos neumático son incapaces de controlar al carro. Hay ocasiones en que las relaciones son bidireccionales, es decir ambos objetos en la relación puede invocar métodos en el otro.

Objeto

Un objeto es un elemento particular y finito dentro de un modelo. Puede representar un objeto tangible del mundo real, como por ejemplo un automóvil. O bien puede representar algo intangible como una transacción bancaria.

Objetos de Ciclo de vida de Clases

Los objetos de ciclo de vida de clases mantienen un registro de los objetos entidad. Ya que la lógica para crear, encontrar y destruir un tipo particular de objeto entidad puede ser usada en diferentes casos de uso, es más cómodo tener un tipo de objeto de ciclo de vida para cada tipo de objeto entidad. Como regla general existe un objeto ciclo de vida por cada objeto entidad.

Objetos de Control

Los objetos de control proveen servicios de control de flujo y sesión a otros objetos. Los objetos de control unen la complicada serie de solicitudes a los objetos entidad en un flujo común que es fácilmente accedido por los objetos limítrofes. Un mensaje de alto nivel de un objeto limítrofe aun objeto control es transformado en una serie de mensajes desde el objeto control al objeto entidad. Como regla cada caso de uso tiene un tipo de objeto de control en el modelo de análisis.

Objetos Entidad

Encapsula los datos y la lógica de negocio del sistema.

Objetos Limítrofes

Son aquellos objetos que interactúan con los actores.

Polimorfismo

El polimorfismo se refiere a múltiples implementaciones de una única abstracción. Las abstracciones son definidas en las clases y en las interfaces. Por lo que podemos conseguir el polimorfismo teniendo más de una clase heredada de una clase base. Cada clase puede sobre escribir la implementación por defecto de la

clase base. También podemos conseguir el polimorfismo teniendo más de una clase que implemente una interfaz.

Relaciones entre casos de uso

Include: Permite que un caso de uso siempre incluya otro caso de uso. Esta relación es representada por una flecha punteada que inicia en el caso de uso base y finaliza en el caso de uso subordinado junto con la etiqueta “<<Include>>”. En esta relación el flujo de eventos se ejecuta a lo largo del caso de uso base hasta que el punto de inclusión es alcanzado. En este punto se ejecuta el flujo de eventos del caso subordinado. La inclusión no es opcional, debe ejecutarse siempre al alcanzar el punto de inclusión.

Extend: Permite que un caso de uso sea incluido opcionalmente dentro de otro caso de uso. Esta relación es representada por una flecha punteada que inicia en el caso de uso extendido y finaliza en el caso de uso base. Junto con la etiqueta “<<Extend>>”. El caso de uso base puede definir uno o más puntos de extensión en su flujo de eventos, cada caso de uso especifica el punto de extensión donde será incluido, junto con la especificación de las condiciones que definen cuando se ejecutara la extensión.

Relaciones entre objetos

Los sistemas orientados a objetos se constituyen por distintos objetos que cooperan para realizar varias tareas. Cada objeto posee un conjunto limitado de responsabilidades, por lo que deben trabajar juntos para alcanzar las metas colectivas. Para cooperar los objetos deben tener relaciones que les permitan comunicarse unos con otros. Existen cuatro tipos de relaciones, estos son:

1. Relación de dependencia

Relación de corto termino con otro objeto, durante la cual el objeto dependiente puede llamar métodos en el otro objeto para solicitar servicios o configurar al objeto.

2. Relación de asociación

Relación de largo término entre objetos, durante la cual un objeto mantiene una referencia al otro objeto y puede llamar a sus métodos cuando sea necesario.

3. Relación de agregado

Indica que un objeto es parte de algo más grande. El objeto puede formar parte en más de una relación de agregado o puede existir como un objeto independiente.

4. Relación de composición

Indica que un objeto es propiedad de algo más grande. El objeto no puede participar en más de una relación de composición y no puede existir como un objeto independiente. El objeto es creado como parte de un todo y es destruido cuando el todo de destruye.

Transición

En un diagrama de actividad una transición representa un cambio de actividad ocasionado por la intervención de un actor en el sistema.

UML

Lenguaje de Modelado Unificado (Unified Modeling Language) es un lenguaje para especificar, visualizar, construir y documentar los componentes de los sistemas de software. Provee una notación precisa que se necesita al modelar sistemas de software

Usuario

Persona que utiliza directamente el sistema.

6 GLOSARIO GENERAL.

Actividad. Un elemento de trabajo desarrollado durante el curso de un proyecto. Una actividad normalmente tiene una duración esperada, un costo esperado, y unos requerimientos esperados de recursos. Las actividades generalmente se subdividen en tareas.

Administración: Es el proceso de organizar, planear, dirigir y controlar actividades y recursos con el fin de lograr un objetivo común.

Administración de Calidad del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto va a satisfacer las necesidades para las cuales fue encomendado. Y consiste de planeación de la calidad, aseguramiento de la calidad, y control de calidad.

Administración Total de Calidad (TQM). Una aproximación común para implementar un programa de mejoramiento de la calidad dentro de una organización.

Administración de Costos del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto se termina dentro del presupuesto aprobado. Esta consiste de planeación de recursos, estimación de costos, presupuesto de costos, y control de costos.

Administración de la Integración del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para asegurar que los elementos varios del proyecto están adecuadamente coordinados. Y consiste de desarrollo del plan del proyecto, ejecución del plan de proyecto, y control de cambios general.

Administración de las Comunicaciones del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para asegurar la colección y disseminación adecuada de la información del proyecto. Esta consiste de planeación de las comunicaciones, distribución de la información, reportes de desempeño, y cierre administrativo.

Administración de Proyectos. Es la aplicación de conocimientos, habilidades, herramientas, y técnicas a las actividades del proyecto de manera que se cumplan o excedan las necesidades y expectativas que los partidos interesados tengan en el proyecto.

Administración de Riesgo del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos concernientes a identificar, analizar, y responder al riesgo del proyecto. Y consiste de identificación de riesgo, cuantificación de riesgo, desarrollo de respuesta al riesgo, y control de respuesta al riesgo.

Administración del Alcance del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto incluye todo el trabajo requerido, y solo el trabajo requerido, para terminar el proyecto de manera exitosa. Y consiste de iniciación, planeación del alcance, definición del alcance, verificación del alcance, y control de cambios al alcance.

Administración del Contrato. Es la administración la relación con el vendedor.

Administración del Recurso Humano del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para hacer el uso más efectivo de las personas involucradas en el proyecto.

Administración del Tiempo del Proyecto. Es un subgrupo de la administración de proyectos que incluye los procesos requeridos para una terminación oportuna del proyecto. Y consiste de definición de actividades, secuencia de actividades, estimación de duración de actividades, desarrollo de la programación, y control de la programación.

Administrador de Proyecto (PM). Es el individuo responsable por la administración del proyecto.

Alcance. Es la suma de productos y servicios que serán proveídos por el proyecto.

Análisis de sistemas: Estudio de los sistemas detalladamente con el fin de mejorarlos o implantarlos.

Analista de sistemas: Persona que realiza el proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados, en este caso computacionales.

Aplicación: Programa o conjuntos de programas diseñados para la realización de una tarea concreta, como puede ser una aplicación comercial, contable, etc.

Archivo: Sinónimo de fichero y esencia del almacenamiento informático. Delimitado por una cabecera y una marca de final, lugar donde puede estar ubicada cualquier tipo de información, ya sea texto, programas, imágenes, sonidos, etc.

Área de Aplicación. Es una categoría de proyectos que tienen elementos en común que no están presentes en todos los proyectos. Las áreas de aplicación están usualmente definidas en términos del producto del proyecto o por el tipo de cliente

Aseguramiento de la Calidad (QA). (1) Es el proceso de evaluar el desempeño general del proyecto de manera regular para proveer la confianza de que el proyecto va a satisfacer los estándares relevantes de calidad. (2) Es la unidad organizacional a la que se le asigna la responsabilidad por la aseguramiento de la calidad.

Base de Datos: Conjunto de ficheros dedicados a guardar información relacionada entre sí, con referencias entre ellos de manera que se complementen con el principio de no duplicidad de datos. Dependiendo de cómo se vinculen dan lugar a B.D. jerárquicas, relacionales, etc. Un caso especial de éstas son las documentales, que, como su nombre indica, están diseñadas para almacenar volúmenes grandes de documentos, lo que genera una problemática distinta por los sistemas de búsqueda.

Capital: Dinero o bienes invertidos, hacienda de una persona o empresa.

Ciclo de Vida del Proyecto. Es una colección de fases de proyecto generalmente secuenciales cuyos nombres y números están determinados por las necesidades de control de organización u organizaciones involucradas en el proyecto.

Contrato. Un contrato es un acuerdo mutuamente relacionado, que obliga al vendedor a proveer el producto especificado y obliga al comprador a pagar por él.

Control de Calidad (QC). (1) Es el proceso de monitorear resultados específicos del proyecto para determinar si estos cumplen los estándares relevantes de calidad e identificar maneras de eliminar causantes de desempeño no satisfactorios. (2) Es la unidad organizacional a la que se le asigna la responsabilidad por el control de la calidad.

Control de Cambios General. Coordina cambios a través de todo el proyecto.

Control de Respuesta al Riesgo. Es responder a cambios en los riesgos sobre la vida del proyecto.

Control. Es el proceso de comparar el rendimiento real con el planeado, analizar varianzas, evaluar posibles alternativas, y tomar la acción correctiva apropiada en la medida que se necesite.

Costeo de Ciclo de Vida. Es el concepto de incluir los costos de adquisición, operación, y eliminación cuando se evalúan varias alternativas.

Costo Presupuestado del Trabajo Realizado (BCWP). Es la suma de los estimados presupuestales aprobados (incluyendo cualquier provisión para los costos administrativos) para actividades (o porciones de actividades) programadas para ser ejecutadas durante un periodo dado (usualmente el proyecto-hasta-la fecha). Véase también valor ganado.

Costo Real de Trabajo Realizado (ACWP). Son los costos en los que se incurre (directo e indirecto) al realizar trabajos en un periodo dado. Véase también valor ganado.

Costos de la Calidad. Son los costos en los que se incurre para asegurar la calidad. El costo de la calidad incluye la planeación de la calidad, aseguramiento de la calidad, y rehacer trabajo.

Desarrollo de Respuesta al Riesgo. Es definir los pasos de mejoramiento para oportunidades y los pasos de mitigación para las amenazas.

Desarrollo del Plan de Proyecto. Es tomar los resultados de los otros procesos de planeación y colocarlos en un solo documento consistente y coherente.

Ejecución del Plan de Proyecto. Es llevar a cabo el plan del proyecto al ejecutar las actividades incluidas en el.

Estimación de Costos. Es estimar el costo de los recursos que se necesitan para completar las actividades del proyecto.

Estimación de la Duración de la Actividad. Es estimar el número de periodos de trabajo que se necesitan para actividades individuales.

Estimado. Es la evaluación del resultado cuantitativo probable. Usualmente se aplica a los costos y duraciones del proyecto y siempre deberá incluir algún indicador de precisión.

Estructura de Desglose de Trabajo (WBS). Es una agrupación orientada por entregas de los elementos de proyecto que organiza y define el alcance total del proyecto. Cada categoría descendiente representa un grado mayor de detalle y definición de los componentes del proyecto, Los componentes del proyecto pueden ser productos o servicios.

Evento de Riesgo. Una ocurrencia discreta que puede afectar el proyecto para mejor o peor.

Fases del Proyecto. Es una colección de actividades relacionadas de manera lógica, que usualmente culminan en la terminación de una entrega principal.

Identificación de Riesgo. Es determinar que eventos de riesgo pueden probablemente afectar el proyecto.

Línea de Base. El plan original (para un proyecto, para un paquete de trabajo, o una actividad), más o menos los cambios autorizados.

Lucro: Ganancia o provecho que se saca de alguna cosa.

Miembros del Equipo de Proyecto. Son las personas que reportan de manera directa o indirecta al administrador del proyecto.

Plan del Proyecto. Es un documento formal, aprobado usado para guiar tanto la ejecución como el control del proyecto. Los usos primarios del plan de proyecto son documentar las suposiciones de la planeación y toma de decisiones, de facilitar la comunicación entre los partidos interesados del proyecto, y de documentar los cambios aprobados a la línea de base del alcance, costos, y programación. U plan de proyecto puede ser detallado o concatenado.

Presupuesto de Costos. Es la asignación de los costos estimados a los componentes individuales del proyecto.

Programa. Es un grupo de proyectos relacionados administrados de una forma coordinada. Los programas usualmente incluyen un elemento de actividad en ejecución.

Programación del Proyecto. Son las fechas planeadas para la ejecución de actividades y las fechas planeadas para el cumplimiento de hitos.

Proyecto. Es un esfuerzo temporal emprendido para crear un servicio o producto único.

Sistemas de Información: Se debe considerar un sistema de computación e información como el conjunto de componentes físicos (hardware), lógicos (software), de comunicación (bien redes de cualquier tipo o Internet) y medios humanos (lo que ahora llaman humanware), todo ello unido permite el tratamiento de la información.

Software de Administración de Proyectos. Es una categoría de aplicaciones para [computadoras](#) diseñadas especialmente para asistir con la planeación y control de la programación y costos de los proyectos.

Tangibles: Que se puede tocar, que se percibe de forma real.

7 CONCLUSION

Después de haber realizado este trabajo hemos llegado a concluir:

En la actualidad, todos los desarrollos de software de sistemas se enfocan en esta época en asegurar que la información es igual de valiosa que los recursos financieros, recursos humanos y recursos materiales en cualquier empresa. Por lo que enfocamos en particular a la utilización de métodos para poder cumplir con sus objetivos y alcances, una de sus aplicaciones que se culminó en este documento fue proveer una descripción general y aplicación de los conceptos de fundamentos de la Dirección de proyectos basados en la PMBOK, con lo que se desarrolló un proyecto de software llamado “Mecanización del proceso de evaluación general de desempeño” para el Instituto Tecnológico Centroamericano.

Y su propósito final fue la elaboración de un documento que se pueda utilizar como guía para las mejores prácticas de dirección de proyectos a un proyecto de desarrollo de software para todas aquellas personas que deseen elaborar o iniciar proyectos a un proyecto de desarrollo de software en caso particular.

8 Fuentes de consulta

8.1 Sitios de internet visitados

- <http://www.liderdeproyecto.com>

8.2 Bibliografía

- The Art of Project Management
Scott Berkun
O'Reilly
2005
- Fundamentals of Project Management
James P. Lewis
AMACOM
1995
- Project Management Body of Knowledge
Project Management Institute, Inc.
2004
- The Project Management Life Cycle
Jason Westland
2006

- Enterprise Java With UML

C.T. Arlington

John Wiley & Sons, Inc.

2005

- Java Server Faces

Hans Bergsten

O'Reilly

2005

- Preparing for the Project Management Professional Certification Exam

Michael W. Newell

AMACON

2005

NOTA: LOS APENDICES ESTAN INCLUIDOS EN EL CD.

- 1- APENDICE A. MANUAL DEL PROGRAMADOR**
- 2- APENDICE B. MANUAL DE INSTALACION.**
- 3- APENDICE C. MANUAL DEL USUARIO.**

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
INGENIERIA EN COMPUTACION



**“GUIA PARA LA IMPLEMENTACION DE LAS MEJORES
PRACTICAS EN LA DIRECCION DE PROYECTOS APLICADO EN
UN PROYECTO DE DESARROLLO DE SOFTWARE PARA EL
INSTITUTO TECNOLOGICO CENTROAMERICANO”**

Apéndice A

Trabajo de graduación para optar al título de:
Ingeniero en Ciencias de la Computación

Presentado por:

Guillermo Arturo Cerrato Gámez

Salvador Fonseca Rivas

Asesor:

Ing. Carlos López Linares, MSc, MBA

Soyapango, 25 Marzo del 2008

Índice

APÉNDICE A - MANUAL DEL PROGRAMADOR	3
1.1 TABLAS	3
1.1.1 Diccionario de datos.....	3
1.1.2 Diagrama Entidad Relación	14
1.2 PROCEDIMIENTOS ALMACENADOS	15
1.2.1 Procesos.....	15
1.2.1.1 spActivaEvaluaciones	15
1.2.2 Reportes.....	17
1.2.2.1 rptEvaluacionDesempenio	17
1.2.2.2 rptPresupuesto	22
1.2.2.3 rptTarea	23
1.3 JOBS.....	25
1.4 VISTAS	26
1.5 PROGRAMAS.....	30
1.5.1 Agregar proyecto.....	30
1.5.2 Agregar Tarea.....	39
1.5.3 Agregar Presupuesto	58
1.5.4 Agregar Cuestionario y preguntas.....	73
1.5.4.1 Agregar Cuestionario	74
1.5.4.2 Agregar Pregunta	82
1.5.5 Configuración de la evaluación.....	93
1.5.6 Contestar cuestionario	112
1.5.7 Agregar Departamento	119
1.5.8 Agregar Empleado	127
1.5.9 Ingreso al sistema.....	141
1.5.10 Cambio de contraseña	148

Apéndice A - Manual del programador

1.1 Tablas

1.1.1 Diccionario de datos

Nombre de la tabla: tarea

Descripción: Almacena el porcentaje de avance mensual planificado para las actividades de: Planificación Anual, Tareas no planificadas, Desarrollo Profesional y Presupuesto.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idTarea	bigint	8	No		Si	Código de la Tarea
idProyecto	bigint	8	Si		No	Proyecto al que pertenece la tarea
idEmpleado	varchar(15)	15	No		No	Empleado a quien se le asigna la actividad
fechaAsignacion	datetime	8	No	(getdate())	No	Fecha en que se asignó la actividad
tarea	varchar(75)	75	No		No	Nombre corto que describe la tarea
comentario	varchar(500)	500	Si		No	Comentario del requerimiento
cumplimientoMes1	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Enero
cumplimientoMes2	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Febrero
cumplimientoMes3	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Marzo
cumplimientoMes4	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Abril
cumplimientoMes5	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Mayo
cumplimientoMes6	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes

						Junio
cumplimientoMes7	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Julio
cumplimientoMes8	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Agosto
cumplimientoMes9	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Septiembre
cumplimientoMes10	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Octubre
cumplimientoMes11	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Noviembre
cumplimientoMes12	numeric(6, 2)	5	No	((0))	No	porcentaje del avance de la asignación Mes Diciembre
real	numeric(6, 2)	5	No	((0))	No	avance a la fecha actual
estado	char(1)	1	No	('a')	No	Estado de la asignación. Los valores que puede tomar son: 'A' => Asignado (sin iniciar); 'E' => en ejecución; 'F' => Finalizado; 'T' => Atrasado; 'R' => Finalizado con demora
fechaFinalizacion	datetime	8	Si		No	Fecha en que se finalizo la asignación
ultimoUsuario	varchar(50)	50	No	(suser_sname())	No	Ultimo usuario que actualizo el registro
ultimaFecha	datetime	8	No	(getdate())	No	Fecha de la última actualización
ultimoOrdenador	varchar(50)	50	No	(host_name())	No	Computadora donde se realizo la última actualización
tipo	char(1)	1	No		No	Tipo de tarea. Los valores que puede tomar son: A = > Planificación Anual; D = > Desarrollo Profesional; N = > Tareas no Planificadas; P= >

						Presupuesto

Nombre de la Tabla: Cargo

Descripción: Guarda la estructura del organigrama de la empresa.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idCargo	int	4	No		Si	Código del cargo
idSuperior	int	4	No		No	Superior o jefe del cargo
idDepartamento	int	4	No		No	Departamento al que está asignado el cargo
cargo	varchar(50)	50	No		No	Nombre del cargo
nivel	int	4	No		No	Nivel del cargo dentro del organigrama
ultimoUsuario	varchar(50)	50	No	(user_sname())	No	Ultimo usuario que actualizo el registro
ultimaFecha	datetime	8	No	(getdate())	No	Fecha de la última actualización
ultimoOrdenador	varchar(50)	50	No	(host_name())	No	Computadora donde se realizo la última actualización

Nombre de la Tabla: Departamento

Descripción: Es un catálogo de los departamentos de la empresa

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idDepartamento	int	4	No		Si	Código del departamento
departamento	varchar(50)	50	No		No	Nombre del departamento

Nombre de la tabla: Proyecto

Descripción: Catálogo de proyectos

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idProyecto	bigint	8	No		Si	Código del proyecto.
idDepartamento	int	4	No		No	Departamento responsable del proyecto.
proyecto	varchar(100)	100	No		No	Nombre del proyecto.
anio	int	4	No	(datepart(year,getdate()))	No	Año de ejecución del proyecto
estado	char(1)	1	No	('A')	No	Estado del proyecto.
ultimoUsuario	varchar(50)	50	No	(suser_name())	No	Último usuario que actualizo el registro
ultimaFecha	datetime	8	No	(getdate())	No	Fecha de la última actualización
ultimoOrdenador	varchar(50)	50	No	(host_name())	No	Computadora donde se realizo la última actualización

Nombre de la tabla: evaluación

Descripción: Datos generales de la Evaluación Anual de los empleados

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
fechaEvaluacion	datetime	8	No		No	Fecha de inicio de la evaluación.
idCuestionario	varchar(10)	10	No		No	Cuestionario utilizado en la evaluación.
idEmpleado	varchar(15)	15	No		No	El valor de este campo coincide con el login del usuario en la BDD.
estado	char(1)	1	No	('P')	No	Estado de la evaluación. Este campo puede contener los valores: A=>Evaluación pendiente; E=>Evaluación en proceso; F=>Evaluación Finalizada

Nombre de la tabla: evaluacionDetalle

Descripción: Detalle de la Evaluación Anual del empleado. Contiene la calificación de cada pregunta del cuestionario contestado por los evaluadores.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
fechaEvaluacion	datetime	8	No		No	Fecha de inicio de la evaluación.
idCuestionario	varchar(10)	10	No		No	Cuestionario utilizado en la evaluación.
idEmpleado	varchar(15)	15	No		No	Empleado que está siendo evaluado.
idEvaluador	varchar(15)	15	No		No	Empleado que esta contestando el cuestionario. (Evaluador)
idPregunta	int	4	No		No	Código de la pregunta en el cuestionario
calificacion	tinyint	1	No	((0))	No	Calificación asignada por el evaluador. Este campo puede tomar los valores: 0=>NA;1=>NM;2=>R;3=>B;4=>MB;5=>E
contestado	char(1)	1	Si	('N')	No	Indica si la pregunta ya ha sido contestada por el evaluador, o si está pendiente de ser contestada. Puede tomar los valores: N=>NO; S=SI

Nombre de la tabla: cuestionario

Descripción: Catálogo de los cuestionarios que la empresa dispone para la Evaluación Anual de Desempeño.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idCuestionario	varchar(10)	10	No		No	Código de desempeño.
cuestionario	varchar(100)	100	No		No	Nombre del cuestionario

Nombre de la tabla: pregunta

Descripción: Detalle de las preguntas en cada uno de los cuestionarios de la Evaluación Anual de Desempeño.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idPregunta	int	4	No		No	Código de la pregunta.
idCuestionario	varchar(10)	10	No		No	Cuestionario al que pertenece la pregunta.
idAreaEvaluacion	int	4	No		No	Área de evaluación de la pregunta.
pregunta	varchar(100)	100	No		No	Pregunta.

Nombre de la tabla: areaEvaluacion

Descripción: Catálogo de Áreas de Evaluación, estas son categorías para las preguntas dentro de los cuestionarios de Evaluación de Desempeño.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idAreaEvaluacion	int	4	No		No	Código del Área de Evaluación
areaEvaluacion	varchar(100)	100	No		No	Nombre del Área de Evaluación

Nombre de la tabla: valoración

Descripción: Guarda el conjunto de calificaciones que un evaluador puede asignar a un empleado en cada una de las preguntas de los cuestionarios de evaluación anual.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idValoracion	tinyint	1	No		No	
valoracion	varchar(2)	2	No		No	

Nombre de la tabla: empleado

Descripción: Catálogo de empleados de la institución

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idEmpleado	varchar(15)	15	No		No	Código del empleado. El valor de este campo coincide con el login del usuario en la BDD.
idCargo	int	4	No		No	Cargo del empleado. (Según organigrama de la empresa)
nombres	varchar(50)	50	No		No	Nombres del empleado.
apellidos	varchar(50)	50	No		No	Apellidos del empleado.
email	varchar(75)	75	Yes		No	Dirección de correo electrónico del empleado.
estado	char(1)	1	No	('A')	No	Estado del empleado dentro de la institución. Este campo puede tomar los valores: 'A' => Activo; 'I' => Inactivo
contrasenia	varbinary(50)	50	Yes	(EncryptByPassPhrase('S EGD','itca'))	No	Este campo almacena la contraseña del usuario de manera encriptada en un campo tipo binario. El método de encriptamiento

						es sincrónico con una llave de encriptamiento de 128 bits.
tipo	char(1)	1	No	('E')	No	Este campo indica el tipo de usuario (Jefe, Empleado normal). Puede tomar los valores: 'S' => Jefe o supervisor; 'E' => empleado normal.
primerLogin	bit	1	No	((1))	No	Bandera booleana, que indica si es la primera vez que el empleado entra al sistema. Este campo se utiliza para forzar a que el usuario cambie su contraseña por defecto.
ultimoUsuario	varchar(50)	50	No	(suser_name())	No	Último usuario que actualizó el registro
ultimaFecha	datetime	8	No	(getdate())	No	Fecha de la última actualización
ultimoOrdenador	varchar(50)	50	No	(host_name())	No	Computadora donde se realizó la última actualización

Nombre de la tabla: empleadoTipo

Descripción: Catálogo de tipos de usuario, 'S' si el empleado tiene personas a su cargo, es decir si es un gerente, jefe o supervisor; 'E' si es un empleado normal que no tiene personas a su cargo. La finalidad de esta tabla es hacer más sencillas las consultas dentro del programa.

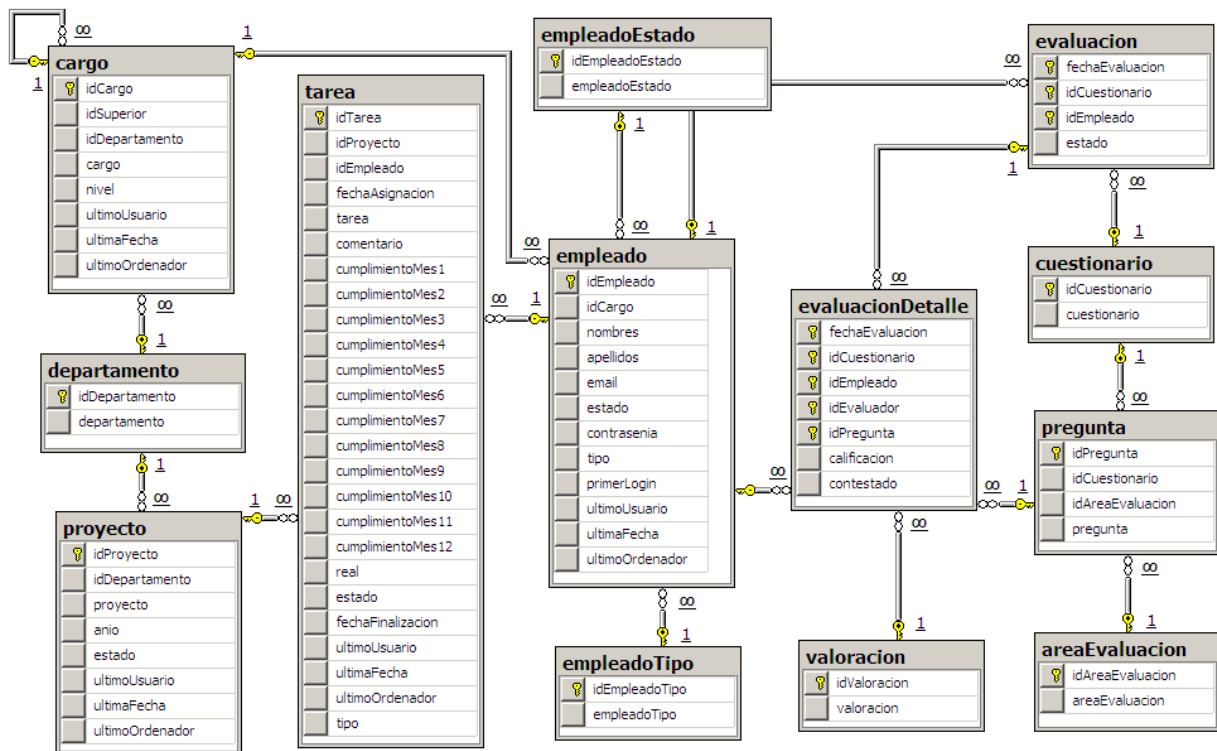
Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idEmpleadoTipo	char(1)	1	No		No	Código del tipo de empleado.
empleadoTipo	varchar(25)	25	No		No	Nombre del tipo de empleado.

Nombre de la tabla: empleadoEstado

Descripción: Catálogo de los diferentes estados que un empleado puede tener dentro de la institución: Activo ('A'), Inactivo ('I'). La finalidad de esta tabla es hacer más sencillas las consultas dentro del programa.

Nombre del Campo	Tipo de Dato	Longitud	Acepta Nulos	Valor por Defecto	Auto-numérico	Descripción
idEmpleadoEstado	char(1)	1	No		No	Código del estado del empleado
empleadoEstado	varchar(25)	25	No		No	Nombre del estado del empleado

1.1.2 Diagrama Entidad Relación



1.2 Procedimientos almacenados

1.2.1 Procesos

1.2.1.1 spActivaEvaluaciones

Este procedimiento almacenado es ejecutado todos los días a las 12:30 AM por un job llamado: jbActivaEvaluaciones. Se encarga de buscar y activar las evaluaciones programas para el día en curso (según la fecha y hora configurada en el servidor MS SQLServer), por lo que cuando encuentra una evaluación que debe ser iniciada el día en curso, cambia el estado de la evaluación de 'A' (Estado inicial) a 'E' (en ejecución).

Así mismo notifica por correo electrónico a cada evaluador del inicio de cada evaluación a la que está asignado y del cuestionario que tiene que llenar.

Parámetros de entrada:

- No tiene parámetros de entrada.

Salidas del procedimiento almacenado:

- Correo electrónico para cada evaluador, informando el inicio de la evaluación.

Código Fuente:

```
CREATE PROCEDURE [dbo].[spActivaEvaluaciones]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @FECHA EVALUACION DATETIME
            , @IDCUESTIONARIO VARCHAR(10)
            , @IDEMPLEADO VARCHAR(15)

    DECLARE @EMPLEADO AS VARCHAR(250)
            , @EVALUADOR AS VARCHAR(250)
```

```

-- VARIABLES PARA ENVIO DE CORREO
DECLARE @PROFILE VARCHAR(MAX)
        , @MENSAJE VARCHAR(MAX)
        , @DESTINATARIOS VARCHAR(MAX)
        , @TITULO VARCHAR(1000)

SET @TITULO = 'NOTIFICACION DE EVALUACION'

DECLARE CEVALUACION CURSOR FAST_FORWARD FOR
SELECT FECHA EVALUACION
      , IDCUESTIONARIO
      , IDEMPLEADO
FROM EVALUACION
WHERE FECHA EVALUACION <= GETDATE()
      AND ESTADO = 'A'
OPEN CEVALUACION
FETCH NEXT FROM CEVALUACION INTO @FECHA EVALUACION, @IDCUESTIONARIO,
@IDEMPLEADO
WHILE @@FETCH_STATUS = 0 BEGIN
    DECLARE CEVALUADOR CURSOR FAST_FORWARD FOR
    SELECT DISTINCT
          E2.NOMBRES + ' ' + E2.APELLIDOS AS 'EMPLEADO'
        , E.NOMBRES + ' ' + E.APELLIDOS AS 'EVALUADOR'
        , E.EMAIL
    FROM EVALUACION DETALLE D
    INNER JOIN EMPLEADO E ON E.IDEMPLEADO = D.IDEVALUADOR
    INNER JOIN EMPLEADO E2 ON E2.IDEMPLEADO = D.IDEMPLEADO
    WHERE D.FECHA EVALUACION = @FECHA EVALUACION
          AND D.IDCUESTIONARIO = @IDCUESTIONARIO
          AND D.IDEMPLEADO = @IDEMPLEADO
          AND E.EMAIL IS NOT NULL
          AND E.EMAIL <> ''
    OPEN CEVALUADOR
    FETCH NEXT FROM CEVALUADOR INTO @EMPLEADO, @EVALUADOR,
@DESTINATARIOS
    WHILE @@FETCH_STATUS = 0 BEGIN
        -- ENVIA CORREO ELECTRONICO PARA NOTIFICAR EL INICIO DE LA
EVALUACION
        SET @MENSAJE = 'ESTIMADO ' + @EVALUADOR + ', SE LE
NOTIFICA QUE ESTE DIA HA INICIADO LA EVALUACION'
        SET @MENSAJE = @MENSAJE + ' DEL EMPLEADO(A) ' + @EMPLEADO
+ '. POR FAVOR INGRESE AL MODULO DE EVALUACION PARA PROCEDER A CONTESTAR EL
RESPECTIVO CUESTIONARIO.'
        EXEC MSDB.DBO.SP_SEND_DBMAIL @PROFILE_NAME = NULL --
@PROFILE
        , @RECIPIENTS = @DESTINATARIOS
        , @BODY = @MENSAJE
        , @SUBJECT = @TITULO
        FETCH NEXT FROM CEVALUADOR INTO @EMPLEADO, @EVALUADOR,
@DESTINATARIOS
    END
    CLOSE CEVALUADOR
    DEALLOCATE CEVALUADOR

    -- CAMBIA EL ESTADO DE LA EVALUACION A 'E' (EVALUACION EN
PROCESO)
    UPDATE EVALUACION

```



```

        SET ESTADO = 'E'
        WHERE FECHA EVALUACION = @FECHA EVALUACION
              AND IDCUESTIONARIO = @IDCUESTIONARIO
              AND IDEMPLEADO = @IDEMPLEADO

        FETCH NEXT FROM CEVALUACION INTO @FECHA EVALUACION,
        @IDCUESTIONARIO, @IDEMPLEADO
    END
    CLOSE CEVALUACION
    DEALLOCATE CEVALUACION
END

```

1.2.2 Reportes

1.2.2.1 rptEvaluacionDesempenio

Este procedimiento almacenado se encarga de generar la consulta que muestra los resultados de la evaluación realizada a un empleado. Es utilizado por el reporte de evaluación de Desempeño.

Parámetros de entrada:

- @dtFechaEvaluacion:

Tipo: Datetime

Descripción: Fecha de inicio de la evaluación.

- @idCuestionario:

Tipo: varchar(10):

Descripción: Código del cuestionario con el que se evaluó al empleado.

- @idEmpleado:

Tipo: varchar(15)

Descripción: Código del empleado que se evaluó.

Código fuente:

```
CREATE PROCEDURE [dbo].[rptEvaluacionDesempenio]
    @dtFechaEvaluacion datetime
    , @idCuestionario varchar(10)
    , @idEmpleado varchar(15)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @intCuentaEvaluadores INT
        , @IDEVALUADOR VARCHAR(15)

    SET @intCuentaEvaluadores = 0

    CREATE TABLE #tblEvaluacion (
        CUESTIONARIO VARCHAR(100) NULL
        , EMPLEADO VARCHAR(101) NULL
        , IDAREAEVALUACION INT NULL
        , AREAEVALUACION VARCHAR(100) NULL
        , IDPREGUNTA INT NULL
        , PREGUNTA VARCHAR(100) NULL
        , CARGO VARCHAR(50)
        , DEPARTAMENTO VARCHAR(50)
        , JEFE VARCHAR(101) NULL
    )

    INSERT INTO #tblEvaluacion
    SELECT C.CUESTIONARIO
        , U.NOMBRES + ' ' + U.APELLIDOS AS 'EMPLEADO'
        , A.IDAREAEVALUACION
        , A.AREAEVALUACION
        , P.IDPREGUNTA
        , P.PREGUNTA
        , CA.CARGO
        , D.DEPARTAMENTO AS 'AREA'
        , J.NOMBRES + ' ' + J.APELLIDOS AS 'JEFE'
    FROM EVALUACION E
    INNER JOIN CUESTIONARIO C
        ON C.IDCUESTIONARIO = E.IDCUESTIONARIO
    INNER JOIN EMPLEADO U
        ON U.IDEMPLEADO = E.IDEMPLEADO
    INNER JOIN PREGUNTA P
        ON P.IDCUESTIONARIO = E.IDCUESTIONARIO
    INNER JOIN AREAEVALUACION A
        ON A.IDAREAEVALUACION = P.IDAREAEVALUACION
    INNER JOIN CARGO CA ON CA.IDCARGO = U.IDCARGO
    INNER JOIN DEPARTAMENTO D ON D.IDDEPARTAMENTO = CA.IDDEPARTAMENTO
    INNER JOIN EMPLEADO J ON J.IDCARGO = CA.IDSUPERIOR
    WHERE E.FECHA EVALUACION = @dtFechaEvaluacion
        AND E.IDCUESTIONARIO = @idCuestionario
        AND E.IDEMPLEADO = @idEmpleado
```

```

DECLARE CEVALUADOR CURSOR FAST_FORWARD FOR
SELECT DISTINCT IDEVALUADOR
FROM EVALUACION E
INNER JOIN EVALUACIONDETALLE D
    ON D.FECHA EVALUACION = E.FECHA EVALUACION
    AND D.IDCUESTIONARIO = E.IDCUESTIONARIO
    AND D.IDEMPLEADO = E.IDEMPLEADO
WHERE E.FECHA EVALUACION = @dtFechaEvaluacion
    AND E.IDCUESTIONARIO = @idCuestionario
    AND E.IDEMPLEADO = @idEmpleado
OPEN CEVALUADOR
FETCH NEXT FROM CEVALUADOR INTO @IDEVALUADOR
WHILE @@FETCH_STATUS = 0 BEGIN
    SET @intCuentaEvaluadores = @intCuentaEvaluadores + 1

    IF @intCuentaEvaluadores = 1 BEGIN
        ALTER TABLE #tblEvaluacion ADD VALORACION1 TINYINT DEFAULT

0
        UPDATE #tblEvaluacion
        SET VALORACION1 = ED.CALIFICACION
        FROM #tblEvaluacion TT
        INNER JOIN EVALUACIONDETALLE ED
            ON ED.IDPREGUNTA = TT.IDPREGUNTA
            AND ED.FECHA EVALUACION = @dtFechaEvaluacion
            AND ED.IDCUESTIONARIO = @idCuestionario
            AND ED.IDEMPLEADO = @idEmpleado
            AND ED.IDEVALUADOR = @IDEVALUADOR
        END

    IF @intCuentaEvaluadores = 2 BEGIN
        ALTER TABLE #tblEvaluacion ADD VALORACION2 TINYINT DEFAULT

0
        UPDATE #tblEvaluacion
        SET VALORACION2 = ED.CALIFICACION
        FROM #tblEvaluacion TT
        INNER JOIN EVALUACIONDETALLE ED
            ON ED.IDPREGUNTA = TT.IDPREGUNTA
            AND ED.FECHA EVALUACION = @dtFechaEvaluacion
            AND ED.IDCUESTIONARIO = @idCuestionario
            AND ED.IDEMPLEADO = @idEmpleado
            AND ED.IDEVALUADOR = @IDEVALUADOR
        END

    IF @intCuentaEvaluadores = 3 BEGIN
        ALTER TABLE #tblEvaluacion ADD VALORACION3 TINYINT DEFAULT

0
        UPDATE #tblEvaluacion
        SET VALORACION3 = ED.CALIFICACION
        FROM #tblEvaluacion TT
        INNER JOIN EVALUACIONDETALLE ED
            ON ED.IDPREGUNTA = TT.IDPREGUNTA
            AND ED.FECHA EVALUACION = @dtFechaEvaluacion
            AND ED.IDCUESTIONARIO = @idCuestionario
            AND ED.IDEMPLEADO = @idEmpleado

```

```

                                AND ED.IDEVALUADOR = @IDEVALUADOR
END

IF @intCuentaEvaluadores = 4 BEGIN
    ALTER TABLE #tblEvaluacion ADD VALORACION4 TINYINT DEFAULT

0

    UPDATE #tblEvaluacion
    SET VALORACION4 = ED.CALIFICACION
    FROM #tblEvaluacion TT
    INNER JOIN EVALUACIONDETALLE ED
        ON ED.IDPREGUNTA = TT.IDPREGUNTA
        AND ED.FECHA EVALUACION = @dtFechaEvaluacion
        AND ED.IDCUESTIONARIO = @idCuestionario
        AND ED.IDEMPLEADO = @idEmpleado
        AND ED.IDEVALUADOR = @IDEVALUADOR
END

IF @intCuentaEvaluadores = 5 BEGIN
    ALTER TABLE #tblEvaluacion ADD VALORACION5 TINYINT DEFAULT

0

    UPDATE #tblEvaluacion
    SET VALORACION5 = ED.CALIFICACION
    FROM #tblEvaluacion TT
    INNER JOIN EVALUACIONDETALLE ED
        ON ED.IDPREGUNTA = TT.IDPREGUNTA
        AND ED.FECHA EVALUACION = @dtFechaEvaluacion
        AND ED.IDCUESTIONARIO = @idCuestionario
        AND ED.IDEMPLEADO = @idEmpleado
        AND ED.IDEVALUADOR = @IDEVALUADOR
END

IF @intCuentaEvaluadores = 6 BEGIN
    ALTER TABLE #tblEvaluacion ADD VALORACION6 TINYINT DEFAULT

0

    UPDATE #tblEvaluacion
    SET VALORACION6 = ED.CALIFICACION
    FROM #tblEvaluacion TT
    INNER JOIN EVALUACIONDETALLE ED
        ON ED.IDPREGUNTA = TT.IDPREGUNTA
        AND ED.FECHA EVALUACION = @dtFechaEvaluacion
        AND ED.IDCUESTIONARIO = @idCuestionario
        AND ED.IDEMPLEADO = @idEmpleado
        AND ED.IDEVALUADOR = @IDEVALUADOR
END

    FETCH NEXT FROM CEVALUADOR INTO @IDEVALUADOR
END
CLOSE CEVALUADOR
DEALLOCATE CEVALUADOR

WHILE (@intCuentaEvaluadores < 6) BEGIN
    SET @intCuentaEvaluadores = @intCuentaEvaluadores + 1

    IF @intCuentaEvaluadores = 1 BEGIN

```

```

ALTER TABLE #tblEvaluacion ADD VALORACION1 TINYINT
DEFAULT (0)
UPDATE #tblEvaluacion SET VALORACION1 = 0
END

IF @intCuentaEvaluadores = 2 BEGIN
ALTER TABLE #tblEvaluacion ADD VALORACION2 TINYINT
DEFAULT (0)
UPDATE #tblEvaluacion SET VALORACION2 = 0
END

IF @intCuentaEvaluadores = 3 BEGIN
ALTER TABLE #tblEvaluacion ADD VALORACION3 TINYINT
DEFAULT (0)
UPDATE #tblEvaluacion SET VALORACION3 = 0
END

IF @intCuentaEvaluadores = 4 BEGIN
ALTER TABLE #tblEvaluacion ADD VALORACION4 TINYINT
DEFAULT (0)
UPDATE #tblEvaluacion SET VALORACION4 = 0
END

IF @intCuentaEvaluadores = 5 BEGIN
ALTER TABLE #tblEvaluacion ADD VALORACION5 TINYINT
DEFAULT (0)
UPDATE #tblEvaluacion SET VALORACION5 = 0
END

IF @intCuentaEvaluadores = 6 BEGIN
ALTER TABLE #tblEvaluacion ADD VALORACION6 TINYINT
DEFAULT (0)
UPDATE #tblEvaluacion SET VALORACION6 = 0
END

END

SELECT
T.CUESTIONARIO
, T.EMPLEADO
, T.AREA-EVALUACION
, T.IDPREGUNTA
, T.PREGUNTA
, T.VALORACION1
, T.VALORACION2
, T.VALORACION3
, T.VALORACION4
, T.VALORACION5
, T.VALORACION6
, T.CARGO
, T.DEPARTAMENTO
, T.JEFE
, CONVERT(CHAR(10), @dtFechaEvaluacion, 103) AS 'FECHA'
FROM #tblEvaluacion T
ORDER BY T.IDAREA-EVALUACION, T.IDPREGUNTA

DROP TABLE #tblEvaluacion
END

```

1.2.2.2 rptPresupuesto

Procedimiento encargado de generar los datos para el reporte de presupuesto.

Parámetros de entrada:

- @intProyecto

Tipo: bigint

Descripción: Código del proyecto al que se esta consultando el presupuesto.

Código fuente:

```
CREATE PROCEDURE [dbo].[rptPresupuesto]
    @intProyecto BIGINT
AS
BEGIN
    SET NOCOUNT ON;

    SELECT P.PROYECTO
        , T.IDPROYECTO
        , T.TAREA
        , T.CUMPLIMIENTOMES1
        , T.CUMPLIMIENTOMES2
        , T.CUMPLIMIENTOMES3
        , T.CUMPLIMIENTOMES4
        , T.CUMPLIMIENTOMES5
        , T.CUMPLIMIENTOMES6
        , T.CUMPLIMIENTOMES7
        , T.CUMPLIMIENTOMES8
        , T.CUMPLIMIENTOMES9
        , T.CUMPLIMIENTOMES10
        , T.CUMPLIMIENTOMES11
        , T.CUMPLIMIENTOMES12
        , T.COMENTARIO
    FROM TAREA T
    INNER JOIN PROYECTO P ON P.IDPROYECTO = T.IDPROYECTO
    WHERE T.ESTADO = 'A'
        AND T.TIPO = 'P'
        AND T.IDPROYECTO = @intProyecto
    ORDER BY P.PROYECTO
        , T.TAREA
END
```

1.2.2.3 rptTarea

Este reporte es el encargado de generar los reportes de tareas para los módulos de:

- Planificación Anual
- Desarrollo Profesional
- Tareas no planificadas

Parámetros de entrada:

- @chrTipo

Tipo: char(1)

Descripción: Identifica el módulo que se está consultando:

- 'A': Planificación Anual
- 'D': Desarrollo Profesional
- 'N': Tareas no Planificadas

- @chrIdEmpleado

Tipo: varchar(12)

Descripción: Código del empleado al que se le consulta las actividades.

Código Fuente:

```
CREATE PROCEDURE [dbo].[rptTarea]
    @chrTipo CHAR(1)
    , @chrIdEmpleado VARCHAR(12)
AS
BEGIN
    SET NOCOUNT ON;

    SELECT P.PROYECTO
        , T.TAREA
        , T.CUMPLIMIENTOMES1
        , T.CUMPLIMIENTOMES2
        , T.CUMPLIMIENTOMES3
        , T.CUMPLIMIENTOMES4
```

```

, T.CUMPLIMIENTOMES5
, T.CUMPLIMIENTOMES6
, T.CUMPLIMIENTOMES7
, T.CUMPLIMIENTOMES8
, T.CUMPLIMIENTOMES9
, T.CUMPLIMIENTOMES10
, T.CUMPLIMIENTOMES11
, T.CUMPLIMIENTOMES12
, (T.CUMPLIMIENTOMES1
    + T.CUMPLIMIENTOMES2
    + T.CUMPLIMIENTOMES3
    + T.CUMPLIMIENTOMES4
    + T.CUMPLIMIENTOMES5
    + T.CUMPLIMIENTOMES6
    + T.CUMPLIMIENTOMES7
    + T.CUMPLIMIENTOMES8
    + T.CUMPLIMIENTOMES9
    + T.CUMPLIMIENTOMES10
    + T.CUMPLIMIENTOMES11
    + T.CUMPLIMIENTOMES12) AS 'ESP.'
, T.REAL
, (T.REAL - (T.CUMPLIMIENTOMES1
    + T.CUMPLIMIENTOMES2
    + T.CUMPLIMIENTOMES3
    + T.CUMPLIMIENTOMES4
    + T.CUMPLIMIENTOMES5
    + T.CUMPLIMIENTOMES6
    + T.CUMPLIMIENTOMES7
    + T.CUMPLIMIENTOMES8
    + T.CUMPLIMIENTOMES9
    + T.CUMPLIMIENTOMES10
    + T.CUMPLIMIENTOMES11
    + T.CUMPLIMIENTOMES12)) AS 'DIF.'
, T.COMENTARIO
, E.NOMBRES + ' ' + E.APELLIDOS AS 'NOMBRE'
, C.CARGO
, D.DEPARTAMENTO AS 'AREA'
, J.NOMBRES + ' ' + J.APELLIDOS AS 'JEFE'
, CASE @chrTipo
    WHEN 'A' THEN 'PLANIFICACION ANUAL'
    WHEN 'D' THEN 'DESARROLLO PROFESIONAL'
    WHEN 'N' THEN 'TAREAS NO PLANIFICADAS'
    WHEN 'P' THEN 'PRESUPUESTO'
    ELSE '' END AS 'TITULO'
FROM TAREA T
INNER JOIN PROYECTO P ON P.IDPROYECTO = T.IDPROYECTO
INNER JOIN EMPLEADO E ON E.IDEMPLEADO = T.IDEMPLEADO
INNER JOIN CARGO C ON C.IDCARGO = E.IDCARGO
INNER JOIN DEPARTAMENTO D ON D.IDDEPARTAMENTO = C.IDDEPARTAMENTO
INNER JOIN EMPLEADO J ON J.IDCARGO = C.IDSUPERIOR
WHERE T.ESTADO = 'A'
AND T.TIPO = @chrTipo
AND T.IDEMPLEADO = @chrIdEmpleado
ORDER BY P.PROYECTO
, T.TAREA

```

END

1.3 Jobs

jbActivaEvaluaciones

Este Job se encarga de ejecutar todos los días a las 12:30 AM el Stored Procedure llamado: spActivaEvaluaciones. Este Stored Procedure es el encargado de buscar y activar las evaluaciones programadas para el día en curso (según la fecha y hora configurada en el Servidor de MS SQLServer).

Código fuente:

```
USE [msdb]
GO
/***** Object:  Job [jbActivaEvaluaciones]      Script Date: 03/16/2008
16:06:54 *****/
BEGIN TRANSACTION
DECLARE @ReturnCode INT
SELECT @ReturnCode = 0
/***** Object:  JobCategory [[Uncategorized (Local)]]      Script Date:
03/16/2008 16:06:55 *****/
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE
name=N'[Uncategorized (Local)]' AND category_class=1)
BEGIN
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL',
@name=N'[Uncategorized (Local)]'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)
EXEC @ReturnCode =  msdb.dbo.sp_add_job @job_name=N'jbActivaEvaluaciones',
        @enabled=1,
        @notify_level_eventlog=0,
        @notify_level_email=0,
        @notify_level_netsend=0,
        @notify_level_page=0,
        @delete_level=0,
        @description=N'ejecuta spActivaEvaluaciones, para activar
evaluaciones programadas',
        @category_name=N'[Uncategorized (Local)]',
        @owner_login_name=N'sa', @job_id = @jobId OUTPUT
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object:  Step [EjecutarSP]      Script Date: 03/16/2008 16:06:56
*****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId,
@step_name=N'EjecutarSP',
        @step_id=1,
        @cmdexec_success_code=0,
        @on_success_action=1,
        @on_success_step_id=0,
```

```

        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'TSQL',
        @command=N'exec spActivaEvaluaciones',
        @database_name=N'SEGD',
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id =
1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId,
@name=N'scActivaEvaluaciones',
        @enabled=1,
        @freq_type=4,
        @freq_interval=1,
        @freq_subday_type=1,
        @freq_subday_interval=0,
        @freq_relative_interval=0,
        @freq_recurrence_factor=0,
        @active_start_date=20080131,
        @active_end_date=99991231,
        @active_start_time=1000,
        @active_end_time=235959
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name
= N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
    IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:

```

1.4 Vistas

vSubAlternos

Esta vista almacena una consulta que selecciona a todos los empleados junto con el código de sus superiores.

Código fuente:

```

USE [SEGD]
GO
/***** Object: View [dbo].[vSubAlternos]      Script Date: 03/16/2008
17:28:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE VIEW [dbo].[vSubAlternos]
AS
SELECT TOP (100) PERCENT
    dbo.empleado.idEmpleado
    , dbo.empleado.nombres + ' ' + dbo.empleado.apellidos + ' - ' +
    dbo.departamento.departamento AS empleado
    , dbo.cargo.idSuperior
FROM dbo.empleado
INNER JOIN dbo.cargo
    ON dbo.empleado.idCargo = dbo.cargo.idCargo
INNER JOIN dbo.departamento
    ON dbo.cargo.idDepartamento = dbo.departamento.idDepartamento
WHERE dbo.empleado.estado = 'A'
ORDER BY dbo.cargo.idDepartamento
    , dbo.cargo.nivel

GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPanel1',
@value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
    Begin PaneConfigurations =
        Begin PaneConfiguration = 0
            NumPanes = 4
            Configuration = "(H (1[40] 4[20] 2[20] 3) )"
        End
        Begin PaneConfiguration = 1
            NumPanes = 3
            Configuration = "(H (1 [50] 4 [25] 3))"
        End
        Begin PaneConfiguration = 2
            NumPanes = 3
            Configuration = "(H (1 [50] 2 [25] 3))"
        End
        Begin PaneConfiguration = 3
            NumPanes = 3
            Configuration = "(H (4 [30] 2 [40] 3))"
        End
        Begin PaneConfiguration = 4
            NumPanes = 2
            Configuration = "(H (1 [56] 3))"
        End
        Begin PaneConfiguration = 5
            NumPanes = 2
            Configuration = "(H (2 [66] 3))"
        End
        Begin PaneConfiguration = 6
            NumPanes = 2
            Configuration = "(H (4 [50] 3))"
        End
        Begin PaneConfiguration = 7
            NumPanes = 1
            Configuration = "(V (3))"
        End
        Begin PaneConfiguration = 8
            NumPanes = 3
            Configuration = "(H (1[56] 4[18] 2) )"
        End
        Begin PaneConfiguration = 9

```

```

        NumPanels = 2
        Configuration = "(H (1 [75] 4))"
    End
    Begin PaneConfiguration = 10
        NumPanels = 2
        Configuration = "(H (1[66] 2) )"
    End
    Begin PaneConfiguration = 11
        NumPanels = 2
        Configuration = "(H (4 [60] 2))"
    End
    Begin PaneConfiguration = 12
        NumPanels = 1
        Configuration = "(H (1) )"
    End
    Begin PaneConfiguration = 13
        NumPanels = 1
        Configuration = "(V (4))"
    End
    Begin PaneConfiguration = 14
        NumPanels = 1
        Configuration = "(V (2))"
    End
    ActivePaneConfig = 0
End
Begin DiagramPane =
    Begin Origin =
        Top = 0
        Left = 0
    End
    Begin Tables =
        Begin Table = "empleado"
            Begin Extent =
                Top = 6
                Left = 38
                Bottom = 121
                Right = 190
            End
            DisplayFlags = 280
            TopColumn = 0
        End
        Begin Table = "cargo"
            Begin Extent =
                Top = 6
                Left = 228
                Bottom = 121
                Right = 389
            End
            DisplayFlags = 280
            TopColumn = 0
        End
        Begin Table = "departamento"
            Begin Extent =
                Top = 6
                Left = 427
                Bottom = 91
                Right = 585
            End
        End
    End
End

```

```

        DisplayFlags = 280
        TopColumn = 0
    End
End
Begin SQLPane =
End
Begin DataPane =
    Begin ParameterDefaults = ""
    End
    Begin ColumnWidths = 9
        Width = 284
        Width = 1500
        Width = 4155
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
    End
End
Begin CriteriaPane =
    Begin ColumnWidths = 11
        Column = 1440
        Alias = 900
        Table = 1170
        Output = 720
        Append = 1400
        NewValue = 1170
        SortType = 1350
        SortOrder = 1410
        GroupBy = 1350
        Filter = 1350
        Or = 1350
        Or = 1350
        Or = 1350
    End
End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vSubAlternos'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vSubAlternos'

```

1.5 Programas

1.5.1 Agregar proyecto

Programa encargado de agregar, modificar y eliminar proyectos en el sistema.

Nombre del programa: /PlanAnual/paProyecto.java

Diagrama de actividad:

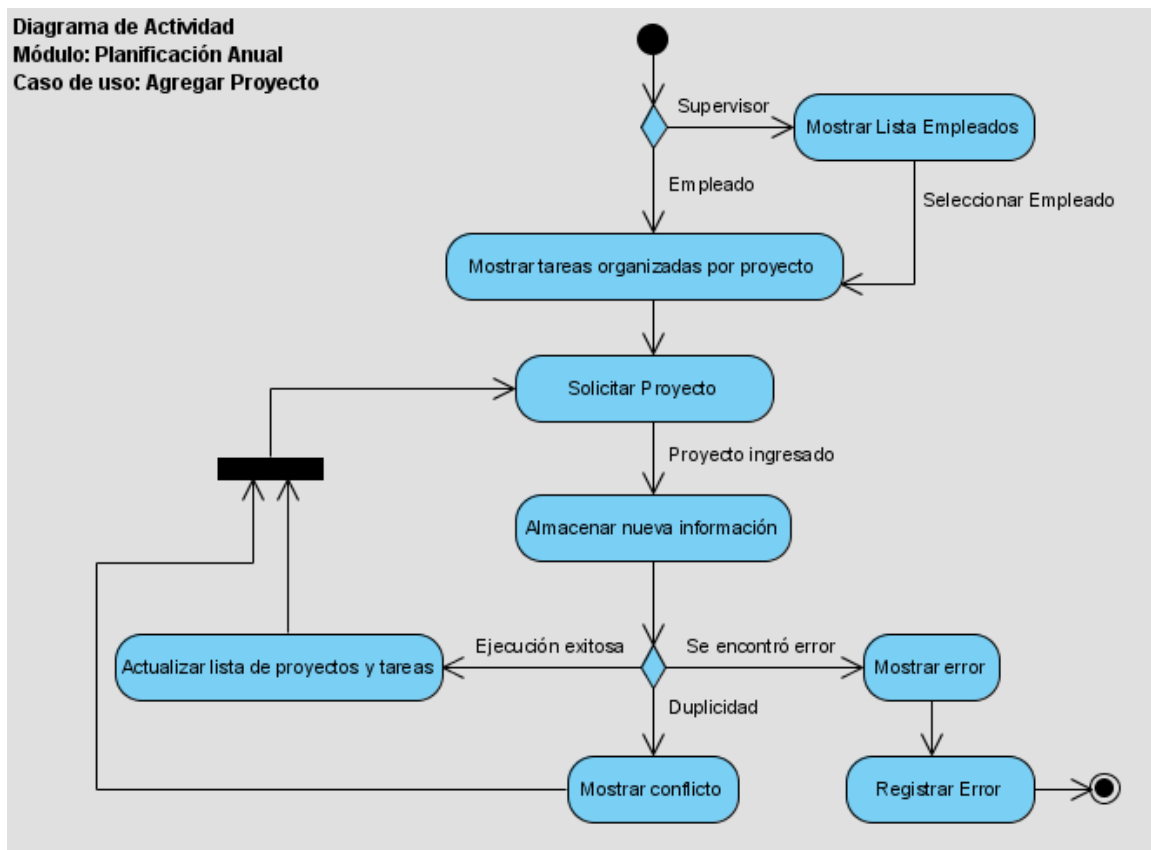


Diagrama de actividad para el caso de uso: Agregar Proyecto.

Código fuente:

```

/*
 * paProyecto.java
 *
 * Created on January 21, 2008, 10:28 PM
 * Copyright sfons0
 */
package segd.PlanAnual;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextField;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class paProyecto extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
     <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
     inserted
     * here is subject to being replaced.</p>
     */
    private void _init() throws Exception {

        proyectoManDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) getVa
        lue("#{SessionBean1.proyectoManRowSet}"));
    }

```

```

private Page page1 = new Page();

public Page getPage1() {
    return page1;
}

public void setPage1(Page p) {
    this.page1 = p;
}

private Html html1 = new Html();

public Html getHtml1() {
    return html1;
}

public void setHtml1(Html h) {
    this.html1 = h;
}

private Head head1 = new Head();

public Head getHead1() {
    return head1;
}

public void setHead1(Head h) {
    this.head1 = h;
}

private Link link1 = new Link();

public Link getLink1() {
    return link1;
}

public void setLink1(Link l) {
    this.link1 = l;
}

private Body body1 = new Body();

public Body getBody1() {
    return body1;
}

public void setBody1(Body b) {
    this.body1 = b;
}

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

```



```

    }

    private Label lblTituloPagina = new Label();

    public Label getLblTituloPagina() {
        return lblTituloPagina;
    }

    public void setLblTituloPagina(Label l) {
        this.lblTituloPagina = l;
    }

    private Label lblTituloPagina2 = new Label();

    public Label getLblTituloPagina2() {
        return lblTituloPagina2;
    }

    public void setLblTituloPagina2(Label l) {
        this.lblTituloPagina2 = l;
    }

    private Table table1 = new Table();

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

    private TextField txtAnio = new TextField();

    public TextField getTxtAnio() {
        return txtAnio;
    }

    public void setTxtAnio(TextField tf) {
        this.txtAnio = tf;
    }

    private Button btnGuardar1 = new Button();

    public Button getBtnGuardar1() {
        return btnGuardar1;
    }

```

```

public void setBtnGuardar1(Button b) {
    this.btnGuardar1 = b;
}

private Button btnAgregar1 = new Button();

public Button getBtnAgregar1() {
    return btnAgregar1;
}

public void setBtnAgregar1(Button b) {
    this.btnAgregar1 = b;
}

private Button btnDeshacer1 = new Button();

public Button getBtnDeshacer1() {
    return btnDeshacer1;
}

public void setBtnDeshacer1(Button b) {
    this.btnDeshacer1 = b;
}

private CachedRowSetDataProvider proyectoManDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getProyectoManDataProvider() {
    return proyectoManDataProvider;
}

public void setProyectoManDataProvider(CachedRowSetDataProvider crsdp) {
    this.proyectoManDataProvider = crsdp;
}

private TableColumn tableColumn1 = new TableColumn();

public TableColumn getTableColumn1() {
    return tableColumn1;
}

public void setTableColumn1(TableColumn tc) {
    this.tableColumn1 = tc;
}

private TextField textField1 = new TextField();

public TextField getTextField1() {
    return textField1;
}

public void setTextField1(TextField tf) {
    this.textField1 = tf;
}

private Button btnActualizar = new Button();

public Button getBtnActualizar() {

```

```

        return btnActualizar;
    }

    public void setBtnActualizar(Button b) {
        this.btnActualizar = b;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

    public MessageGroup getMessageGroup1() {
        return messageGroup1;
    }

    public void setMessageGroup1(MessageGroup mg) {
        this.messageGroup1 = mg;
    }

    private TableColumn tableColumn2 = new TableColumn();

    public TableColumn getTableColumn2() {
        return tableColumn2;
    }

    public void setTableColumn2(TableColumn tc) {
        this.tableColumn2 = tc;
    }

    private Button btnBorrar = new Button();

    public Button getBtnBorrar() {
        return btnBorrar;
    }

    public void setBtnBorrar(Button b) {
        this.btnBorrar = b;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>
     */
    public paProyecto() {
    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property
     * values of the components do <strong>not</strong> represent any
     * values submitted with this request. Instead, they represent the
     * property values that were saved for this view when it was
     rendered.</p>
     */

```

```

public void init() {
    // Perform initializations inherited from our superclass
    super.init();
    // Perform application initialization that must complete
    // *before* managed components are initialized
    // TODO - add your own initialization code here

    // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
    // Initialize automatically managed components
    // *Note* - this logic should NOT be modified
    try {
        _init();
    } catch (Exception e) {
        log("paProyecto Initialization Failure", e);
        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }

    // </editor-fold>
    // Perform application initialization that must complete
    // *after* managed components are initialized
    // TODO - add your own initialization code here
}

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {
}

/**
 * <p>Callback method that is called just before rendering takes place.
 * This method will <strong>only</strong> be called for the page that
 * will actually be rendered (and not, for example, on a page that
 * handled a postback and then navigated to a different page).
Customize
 * this method to allocate resources that will be required for rendering
 * this page.</p>
 */
public void prerender() {
    ExternalContext externalContext =
getFacesContext().getExternalContext();
    try{
        if (getSessionBean1().getIdUsuario() == null){
            externalContext.redirect("faces/login.jsp");
        } else {
            if (txtAnio.getText() == null){
                int Anio;

                Date d = new Date();
                GregorianCalendar gc = new GregorianCalendar();
                gc.setTime(d);
                Anio = gc.get(Calendar.YEAR);
            }
        }
    }
}

```

```

        txtAnio.setText(String.valueOf(Anio));
        getSessionBean1().getProyectoManRowSet().setObject(1,
getSessionBean1().getIntDepartamento());
        getSessionBean1().getProyectoManRowSet().setObject(2,
Anio);

        proyectoManDataProvider.refresh();
    }
}
} catch (Exception ex) {
    log("Error :", ex);
    error("Error :" + ex);
}
}

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,
 * <code>preprocess()</code>, or <code>prerender()</code> methods (or
 * acquired during execution of an event handler).</p>
 */
public void destroy() {
    proyectoManDataProvider.close();
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected SessionBean1 getSessionBean1() {
    return (SessionBean1)getBean("SessionBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected ApplicationBean1 getApplicationBean1() {
    return (ApplicationBean1)getBean("ApplicationBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected RequestBean1 getRequestBean1() {
    return (RequestBean1)getBean("RequestBean1");
}

public String btnGuardar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        proyectoManDataProvider.commitChanges();
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :"+ex.getMessage());
    }
}

```

```

    }
    return null;
}

public String btnAgregar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = proyectoManDataProvider.appendRow();
        proyectoManDataProvider.setCursorRow(rk);

        proyectoManDataProvider.setValue("idDepartamento",
getSessionBean1().getIntDepartamento());
        proyectoManDataProvider.setValue("anio",
Integer.parseInt((String) txtAnio.getText()));
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnDeshacer_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        proyectoManDataProvider.refresh();

    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnActualizar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        getSessionBean1().getProyectoManRowSet().setObject(1,
getSessionBean1().getIntDepartamento());
        getSessionBean1().getProyectoManRowSet().setObject(2,
Integer.parseInt((String) txtAnio.getText()));
        proyectoManDataProvider.refresh();
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :" + ex);
    }

    return null;
}

public String btnBorrar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        if (rk != null) {

```

```

        proyectoManDataProvider.removeRow(rk);
        proyectoManDataProvider.commitChanges();
        proyectoManDataProvider.refresh();}
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}
}

```

1.5.2 Agregar Tarea

Este programa se encarga de agregar, modificar y eliminar las tareas o actividades para los módulos de:

- Planificación Anual
- Desarrollo Profesional
- Actividades no Planificadas

Nombre del programa: /PlanAnual/paTarea.java

Diagrama de actividad:

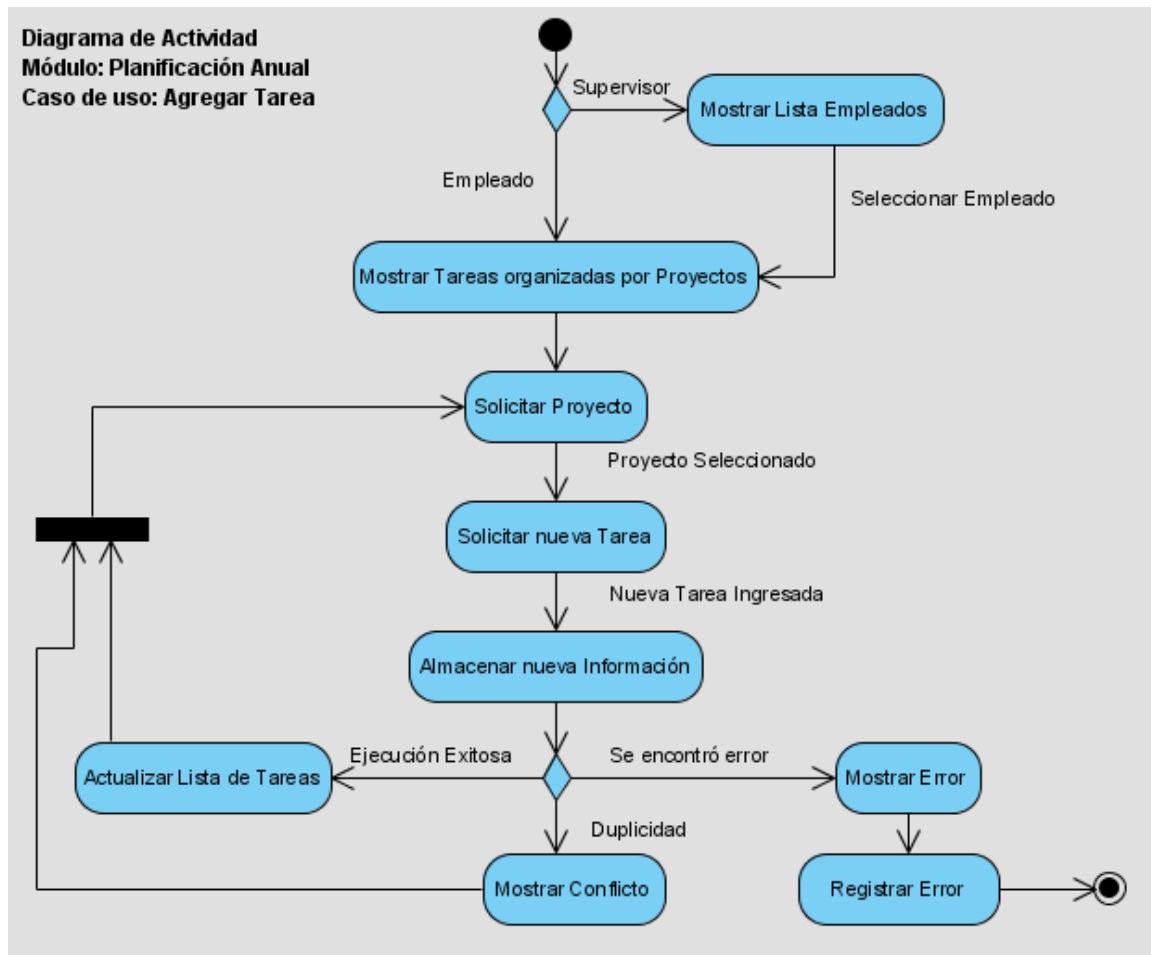


Diagrama de actividad para el caso de uso: Agregar Tarea.

Código fuente:

```
/*
 * paTarea.java
 *
 * Created on January 16, 2008, 9:59 PM
 * Copyright sfons0
 */
package segd.PlanAnual;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.DropDown;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.HiddenField;
import com.sun.webui.jsf.component.Html;
```



```

import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextArea;
import com.sun.webui.jsf.component.TextField;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import javax.faces.convert.LongConverter;
import javax.faces.event.ValueChangeEvent;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class paTarea extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
        private int __placeholder;

        /**
         * <p>Automatically managed component initialization.
         <strong>WARNING:</strong>
         * This method is automatically generated, so any user-specified code
         inserted
         * here is subject to being replaced.</p>
         */
        private void _init() throws Exception {

            subalterno2DataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue("#{SessionBean1.subalterno2RowSet}"));

            tareaDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue("#{SessionBean1.tareaRowSet}"));

            proyectoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue("#{SessionBean1.proyectoRowSet}"));
        }

        private Page page1 = new Page();

        public Page getPage1() {
            return page1;
        }

        public void setPage1(Page p) {
            this.page1 = p;
        }

```

```

private Html html1 = new Html();

public Html getHtml1() {
    return html1;
}

public void setHtml1(Html h) {
    this.html1 = h;
}

private Head head1 = new Head();

public Head getHead1() {
    return head1;
}

public void setHead1(Head h) {
    this.head1 = h;
}

private Link link1 = new Link();

public Link getLink1() {
    return link1;
}

public void setLink1(Link l) {
    this.link1 = l;
}

private Body body1 = new Body();

public Body getBody1() {
    return body1;
}

public void setBody1(Body b) {
    this.body1 = b;
}

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private MessageGroup messageGroup1 = new MessageGroup();

public MessageGroup getMessageGroup1() {
    return messageGroup1;
}

public void setMessageGroup1(MessageGroup mg) {
    this.messageGroup1 = mg;
}

```

```

    }

    private DropDown ddlSubAlterno = new DropDown();

    public DropDown getDdlSubAlterno() {
        return ddlSubAlterno;
    }

    public void setDdlSubAlterno(DropDown dd) {
        this.ddlSubAlterno = dd;
    }

    private CachedRowSetDataProvider subalterno2DataProvider = new
    CachedRowSetDataProvider();

    public CachedRowSetDataProvider getSubalterno2DataProvider() {
        return subalterno2DataProvider;
    }

    public void setSubalterno2DataProvider(CachedRowSetDataProvider crsdp) {
        this.subalterno2DataProvider = crsdp;
    }

    private CachedRowSetDataProvider tareaDataProvider = new
    CachedRowSetDataProvider();

    public CachedRowSetDataProvider getTareaDataProvider() {
        return tareaDataProvider;
    }

    public void setTareaDataProvider(CachedRowSetDataProvider crsdp) {
        this.tareaDataProvider = crsdp;
    }

    private Table table1 = new Table();

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

    private TableColumn tableColumn1 = new TableColumn();

    public TableColumn getTableColumn1() {
        return tableColumn1;
    }

```

```

    }

    public void setTableColumn1(TableColumn tc) {
        this.tableColumn1 = tc;
    }

    private TableColumn tableColumn2 = new TableColumn();

    public TableColumn getTableColumn2() {
        return tableColumn2;
    }

    public void setTableColumn2(TableColumn tc) {
        this.tableColumn2 = tc;
    }

    private TableColumn tableColumn3 = new TableColumn();

    public TableColumn getTableColumn3() {
        return tableColumn3;
    }

    public void setTableColumn3(TableColumn tc) {
        this.tableColumn3 = tc;
    }

    private TableColumn tableColumn4 = new TableColumn();

    public TableColumn getTableColumn4() {
        return tableColumn4;
    }

    public void setTableColumn4(TableColumn tc) {
        this.tableColumn4 = tc;
    }

    private TableColumn tableColumn5 = new TableColumn();

    public TableColumn getTableColumn5() {
        return tableColumn5;
    }

    public void setTableColumn5(TableColumn tc) {
        this.tableColumn5 = tc;
    }

    private TableColumn tableColumn6 = new TableColumn();

    public TableColumn getTableColumn6() {
        return tableColumn6;
    }

    public void setTableColumn6(TableColumn tc) {
        this.tableColumn6 = tc;
    }

    private TableColumn tableColumn7 = new TableColumn();

```

```

public TableColumn getTableColumn7() {
    return tableColumn7;
}

public void setTableColumn7(TableColumn tc) {
    this.tableColumn7 = tc;
}

private TableColumn tableColumn8 = new TableColumn();

public TableColumn getTableColumn8() {
    return tableColumn8;
}

public void setTableColumn8(TableColumn tc) {
    this.tableColumn8 = tc;
}

private TableColumn tableColumn9 = new TableColumn();

public TableColumn getTableColumn9() {
    return tableColumn9;
}

public void setTableColumn9(TableColumn tc) {
    this.tableColumn9 = tc;
}

private TableColumn tableColumn10 = new TableColumn();

public TableColumn getTableColumn10() {
    return tableColumn10;
}

public void setTableColumn10(TableColumn tc) {
    this.tableColumn10 = tc;
}

private TableColumn tableColumn11 = new TableColumn();

public TableColumn getTableColumn11() {
    return tableColumn11;
}

public void setTableColumn11(TableColumn tc) {
    this.tableColumn11 = tc;
}

private TableColumn tableColumn12 = new TableColumn();

public TableColumn getTableColumn12() {
    return tableColumn12;
}

public void setTableColumn12(TableColumn tc) {
    this.tableColumn12 = tc;
}

```

```

private TableColumn tableColumn13 = new TableColumn();

public TableColumn getTableColumn13() {
    return tableColumn13;
}

public void setTableColumn13(TableColumn tc) {
    this.tableColumn13 = tc;
}

private TableColumn tableColumn14 = new TableColumn();

public TableColumn getTableColumn14() {
    return tableColumn14;
}

public void setTableColumn14(TableColumn tc) {
    this.tableColumn14 = tc;
}

private TableColumn tableColumn15 = new TableColumn();

public TableColumn getTableColumn15() {
    return tableColumn15;
}

public void setTableColumn15(TableColumn tc) {
    this.tableColumn15 = tc;
}

private DropDown ddlIdProyecto = new DropDown();

public DropDown getDdlIdProyecto() {
    return ddlIdProyecto;
}

public void setDdlIdProyecto(DropDown dd) {
    this.ddlIdProyecto = dd;
}

private TextField txtCumplimientoMes1 = new TextField();

public TextField getTxtCumplimientoMes1() {
    return txtCumplimientoMes1;
}

public void setTxtCumplimientoMes1(TextField tf) {
    this.txtCumplimientoMes1 = tf;
}

private TextField txtCumplimientoMes2 = new TextField();

public TextField getTxtCumplimientoMes2() {
    return txtCumplimientoMes2;
}

public void setTxtCumplimientoMes2(TextField tf) {
    this.txtCumplimientoMes2 = tf;
}

```

```

    }

    private TextField txtCumplimientoMes3 = new TextField();

    public TextField getTxtCumplimientoMes3() {
        return txtCumplimientoMes3;
    }

    public void setTxtCumplimientoMes3(TextField tf) {
        this.txtCumplimientoMes3 = tf;
    }

    private TextField txtCumplimientoMes4 = new TextField();

    public TextField getTxtCumplimientoMes4() {
        return txtCumplimientoMes4;
    }

    public void setTxtCumplimientoMes4(TextField tf) {
        this.txtCumplimientoMes4 = tf;
    }

    private TextField txtCumplimientoMes5 = new TextField();

    public TextField getTxtCumplimientoMes5() {
        return txtCumplimientoMes5;
    }

    public void setTxtCumplimientoMes5(TextField tf) {
        this.txtCumplimientoMes5 = tf;
    }

    private TextField txtCumplimientoMes6 = new TextField();

    public TextField getTxtCumplimientoMes6() {
        return txtCumplimientoMes6;
    }

    public void setTxtCumplimientoMes6(TextField tf) {
        this.txtCumplimientoMes6 = tf;
    }

    private TextField txtCumplimientoMes7 = new TextField();

    public TextField getTxtCumplimientoMes7() {
        return txtCumplimientoMes7;
    }

    public void setTxtCumplimientoMes7(TextField tf) {
        this.txtCumplimientoMes7 = tf;
    }

    private TextField txtCumplimientoMes8 = new TextField();

    public TextField getTxtCumplimientoMes8() {
        return txtCumplimientoMes8;
    }
}

```

```

public void setTxtCumplimientoMes8(TextField tf) {
    this.txtCumplimientoMes8 = tf;
}

private TextField txtCumplimientoMes9 = new TextField();

public TextField getTxtCumplimientoMes9() {
    return txtCumplimientoMes9;
}

public void setTxtCumplimientoMes9(TextField tf) {
    this.txtCumplimientoMes9 = tf;
}

private TextField txtCumplimientoMes10 = new TextField();

public TextField getTxtCumplimientoMes10() {
    return txtCumplimientoMes10;
}

public void setTxtCumplimientoMes10(TextField tf) {
    this.txtCumplimientoMes10 = tf;
}

private TextField txtCumplimientoMes11 = new TextField();

public TextField getTxtCumplimientoMes11() {
    return txtCumplimientoMes11;
}

public void setTxtCumplimientoMes11(TextField tf) {
    this.txtCumplimientoMes11 = tf;
}

private TextField txtCumplimientoMes12 = new TextField();

public TextField getTxtCumplimientoMes12() {
    return txtCumplimientoMes12;
}

public void setTxtCumplimientoMes12(TextField tf) {
    this.txtCumplimientoMes12 = tf;
}

private TextArea txtComentario = new TextArea();

public TextArea getTxtComentario() {
    return txtComentario;
}

public void setTxtComentario(TextArea ta) {
    this.txtComentario = ta;
}

private CachedRowSetDataProvider proyectoDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getProyectoDataProvider() {

```



```

        return proyectoDataProvider;
    }

    public void setProyectoDataProvider(CachedRowSetDataProvider crsdp) {
        this.proyectoDataProvider = crsdp;
    }

    private LongConverter ddlIdProyectoConverter = new LongConverter();

    public LongConverter getDdlIdProyectoConverter() {
        return ddlIdProyectoConverter;
    }

    public void setDdlIdProyectoConverter(LongConverter lc) {
        this.ddlIdProyectoConverter = lc;
    }

    private Button btnAgregar = new Button();

    public Button getBtnAgregar() {
        return btnAgregar;
    }

    public void setBtnAgregar(Button b) {
        this.btnAgregar = b;
    }

    private Button btnDeshacer = new Button();

    public Button getBtnDeshacer() {
        return btnDeshacer;
    }

    public void setBtnDeshacer(Button b) {
        this.btnDeshacer = b;
    }

    private Button btnGuardar = new Button();

    public Button getBtnGuardar() {
        return btnGuardar;
    }

    public void setBtnGuardar(Button b) {
        this.btnGuardar = b;
    }

    private TextArea textArea1 = new TextArea();

    public TextArea getTextArea1() {
        return textArea1;
    }

    public void setTextArea1(TextArea ta) {
        this.textArea1 = ta;
    }

    private HiddenField strTipoModulo = new HiddenField();

```

```

public HiddenField getStrTipoModulo() {
    return strTipoModulo;
}

public void setStrTipoModulo(HiddenField hf) {
    this.strTipoModulo = hf;
}

private Label lblTituloPagina = new Label();

public Label getLblTituloPagina() {
    return lblTituloPagina;
}

public void setLblTituloPagina(Label l) {
    this.lblTituloPagina = l;
}

private Label lblSubTitulo = new Label();

public Label getLblSubTitulo() {
    return lblSubTitulo;
}

public void setLblSubTitulo(Label l) {
    this.lblSubTitulo = l;
}

private TableColumn tableColumn16 = new TableColumn();

public TableColumn getTableColumn16() {
    return tableColumn16;
}

public void setTableColumn16(TableColumn tc) {
    this.tableColumn16 = tc;
}

private Button btnBorrar = new Button();

public Button getBtnBorrar() {
    return btnBorrar;
}

public void setBtnBorrar(Button b) {
    this.btnBorrar = b;
}

private TableColumn tableColumn17 = new TableColumn();

public TableColumn getTableColumn17() {
    return tableColumn17;
}

public void setTableColumn17(TableColumn tc) {
    this.tableColumn17 = tc;
}

```

```

private TextField textField1 = new TextField();

public TextField getTextField1() {
    return textField1;
}

public void setTextField1(TextField tf) {
    this.textField1 = tf;
}

private Button btnReporte = new Button();

public Button getBtnReporte() {
    return btnReporte;
}

public void setBtnReporte(Button b) {
    this.btnReporte = b;
}

// </editor-fold>

/**
 * <p>Construct a new Page bean instance.</p>
 */
public paTarea() {
}

/**
 * <p>Callback method that is called whenever a page is navigated to,
 * either directly via a URL, or indirectly via page navigation.
 * Customize this method to acquire resources that will be needed
 * for event handlers and lifecycle methods, whether or not this
 * page is performing post back processing.</p>
 *
 * <p>Note that, if the current request is a postback, the property
 * values of the components do <strong>not</strong> represent any
 * values submitted with this request. Instead, they represent the
 * property values that were saved for this view when it was
 rendered.</p>
 */
public void init() {
    // Perform initializations inherited from our superclass
    super.init();
    // Perform application initialization that must complete
    // *before* managed components are initialized
    // TODO - add your own initialization code here

    // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
    // Initialize automatically managed components
    // *Note* - this logic should NOT be modified
    try {
        _init();
    } catch (Exception e) {
        log("paTarea Initialization Failure", e);
    }
}

```

```

        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }

    // </editor-fold>
    // Perform application initialization that must complete
    // *after* managed components are initialized
    // TODO - add your own initialization code here
}

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {
}

/**
 * <p>Callback method that is called just before rendering takes place.
 * This method will <strong>only</strong> be called for the page that
 * will actually be rendered (and not, for example, on a page that
 * handled a postback and then navigated to a different page).
Customize
 * this method to allocate resources that will be required for rendering
 * this page.</p>
 */
//     private String strTipo;
//     private Object objTipo;

    public void prerender() {
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        try{
            strTipo = (String)
getExternalContext().getRequestParameterMap().get("strTipoModulo");

            if (strTipo != null) {
                strTipoModulo.setText((String) strTipo);
            } else {
                strTipo = (String)strTipoModulo.getText();
            }

            if (strTipo.compareTo("A") == 0){
                strTipoMod = "Tarea";
                strTitulo = "Planificación Anual";
                strSubTitulo = "Agregar Tarea";
            }

            if (strTipo.compareTo("D") == 0){
                strTipoMod = "Asignación académica";
                strTitulo = "Desarrollo Profesional";
                strSubTitulo = "Agregar Asignación académica";

                tableColumn1.setVisible(false);
            }
}

```

```

        if (strTipo.compareTo("N") == 0){
            strTipoMod = "Tarea no planificada";
            strTitulo = "Tareas no Planificadas";
            strSubTitulo = "Agregar Tarea no Planificada";
        }

        if (strTipo.compareTo("P") == 0){
            strTipoMod = "Presupuesto";
            strTitulo = "Planificación Anual";
            strSubTitulo = "Agregar Presupuesto";
        }

        if (getSessionBean1().getIdUsuario() == null){
            externalContext.redirect("faces/login.jsp");
        } else {
            if ((getSessionBean1().getTipo().compareTo("E") == 0 ||
strTipo.compareTo("P") == 0)){
                ddlSubAlterno.setVisible(false);
                getSessionBean1().getTareaRowSet().setObject(1,
getSessionBean1().getIdUsuario());
                getSessionBean1().getTareaRowSet().setObject(2,
(String)strTipoModulo.getText());

                tareaDataProvider.refresh();
            }
            if (ddlSubAlterno.getSelected() == null ) {
                getSessionBean1().getSubalternos2RowSet().setObject(1,
getSessionBean1().getIdCargo());
                getSessionBean1().getSubalternos2RowSet().setObject(2,
getSessionBean1().getIdUsuario());
                subalternos2DataProvider.refresh();
                Object firstSubAlterno = null;
                subalternos2DataProvider.cursorFirst();
                firstSubAlterno =
subalternos2DataProvider.getValue("idEmpleado");
                ddlSubAlterno.setSelected(firstSubAlterno);
                getSessionBean1().getTareaRowSet().setObject(1,
firstSubAlterno);
                getSessionBean1().getTareaRowSet().setObject(2,
(String)strTipoModulo.getText());

                tareaDataProvider.refresh();
            } else {
                // Synchronize data provider with current selection

                subalternos2DataProvider.setCursorRow(subalternos2DataProvider.findFirst("idEm
pleado", ddlSubAlterno.getSelected()));
            }
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :" + ex);
    }
}

```

```

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,
 * <code>preprocess()</code>, or <code>prerender()</code> methods (or
 * acquired during execution of an event handler).</p>
 */
public void destroy() {
    subalterno2DataProvider.close();
    tareaDataProvider.close();
    proyectoDataProvider.close();
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected RequestBean1 getRequestBean1() {
    return (RequestBean1) getBean("RequestBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected SessionBean1 getSessionBean1() {
    return (SessionBean1) getBean("SessionBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected ApplicationBean1 getApplicationBean1() {
    return (ApplicationBean1) getBean("ApplicationBean1");
}

public void ddlSubAlterno_processValueChange(ValueChangeEvent event) {
    // TODO: Replace with your code
    Object selectedSubAlterno = ddlSubAlterno.getSelected();
    try {

subalterno2DataProvider.setCursorRow(subalterno2DataProvider.findFirst("idEm
pleado", selectedSubAlterno));
        getSessionBean1().getTareaRowSet().setObject(1,
selectedSubAlterno);
        getSessionBean1().getTareaRowSet().setObject(2,
(String) strTipoModulo.getText());
        tareaDataProvider.refresh();
        //form1.discardSubmittedValues("vfGuardar");
    } catch (Exception e) {
        error("No pude seleccionar " + selectedSubAlterno);
        log("No pude seleccionar " + selectedSubAlterno, e);
    }
}

public String btnAgregar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.

```

```

        try {
            RowKey rk = tareaDataProvider.appendRow();
            tareaDataProvider.setCursorRow(rk);

            tareaDataProvider.setValue("tipo", new
String((String)strTipoModulo.getText()));
            tareaDataProvider.setValue("comentario", new String(""));
            if (getSessionBean1().getTipo().compareTo("E") == 0){
                tareaDataProvider.setValue("idEmpleado",
getSessionBean1().getIdUsuario());
            } else{
                if (ddlSubAlterno.getSelected() == null ) {
                    Object firstSubAlterno = null;
                    subalterno2DataProvider.cursorFirst();
                    firstSubAlterno =
subalterno2DataProvider.getValue("idEmpleado");
                    ddlSubAlterno.setSelected(firstSubAlterno);
                    tareaDataProvider.setValue("idEmpleado",
firstSubAlterno);
                } else {
                    tareaDataProvider.setValue("idEmpleado",
ddlSubAlterno.getSelected());
                }
            }
        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnDeshacer_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            tareaDataProvider.refresh();

        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnGuardar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            tareaDataProvider.commitChanges();
        } catch (Exception ex) {
            log("Error :", ex);
            error("Error :"+ex.getMessage());
        }
        return null;
    }

    /**
     * Holds value of property strTipoMod.

```

```

    */
private String strTipoMod;

/**
 * Getter for property strTipoMod.
 * @return Value of property strTipoMod.
 */
public String getStrTipoMod() {
    return this.strTipoMod;
}

/**
 * Setter for property strTipoMod.
 * @param strTipoMod New value of property strTipoMod.
 */
public void setStrTipoMod(String strTipoMod) {
    this.strTipoMod = strTipoMod;
}

/**
 * Holds value of property strTitulo.
 */
private String strTitulo;

/**
 * Getter for property strTitulo.
 * @return Value of property strTitulo.
 */
public String getStrTitulo() {
    return this.strTitulo;
}

/**
 * Setter for property strTitulo.
 * @param strTitulo New value of property strTitulo.
 */
public void setStrTitulo(String strTitulo) {
    this.strTitulo = strTitulo;
}

/**
 * Holds value of property strSubTitulo.
 */
private String strSubTitulo;

/**
 * Getter for property strSubTitulo.
 * @return Value of property strSubTitulo.
 */
public String getStrSubTitulo() {
    return this.strSubTitulo;
}

/**
 * Setter for property strSubTitulo.
 * @param strSubTitulo New value of property strSubTitulo.
 */
public void setStrSubTitulo(String strSubTitulo) {

```



```

        this.strSubTitulo = strSubTitulo;
    }

    /**
     * Holds value of property strTipo.
     */
    private String strTipo;

    /**
     * Getter for property strTipo.
     * @return Value of property strTipo.
     */
    public String getStrTipo() {
        return this.strTipo;
    }

    /**
     * Setter for property strTipo.
     * @param strTipo New value of property strTipo.
     */
    public void setStrTipo(String strTipo) {
        this.strTipo = strTipo;
    }

    public String btnBorrar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            RowKey rk = tableRowGroup1.getRowKey();
            if (rk != null) {
                tareaDataProvider.removeRow(rk);
                tareaDataProvider.commitChanges();
                tareaDataProvider.refresh();
            }
        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnReporte_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        try {
            if ((getSessionBean1().getTipo().compareTo("E") == 0 ||
strTipoModulo.getText().toString().compareTo("P") == 0)){

externalContext.redirect("http://localhost/ReportServer/?/ITCA/rptTarea&rs:C
ommand=Render&rc:Parameters=false&chrTipo=" +
(String)strTipoModulo.getText() + "&chrIdEmpleado=" +
getSessionBean1().getIdUsuario());
            } else {
                Object selectedSubAlterno = ddlSubAlterno.getSelected();

```

```

subalterno2DataProvider.setCursorRow(subalterno2DataProvider.findFirst("idEmpleado", selectedSubAlterno));

externalContext.redirect("http://localhost/ReportServer/?/ITCA/rptTarea&rs:Command=Render&rc:Parameters=false&chrTipo=" +
(String)strTipoModulo.getText() + "&chrIdEmpleado=" +
selectedSubAlterno.toString());
    }
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}
}

```

1.5.3 Agregar Presupuesto

Este programa se encarga de agregar, modificar y eliminar los elementos del presupuesto de un proyecto específico.

Nombre del programa: /PlanAnual/paPresupuesto.java

Diagrama de actividad:

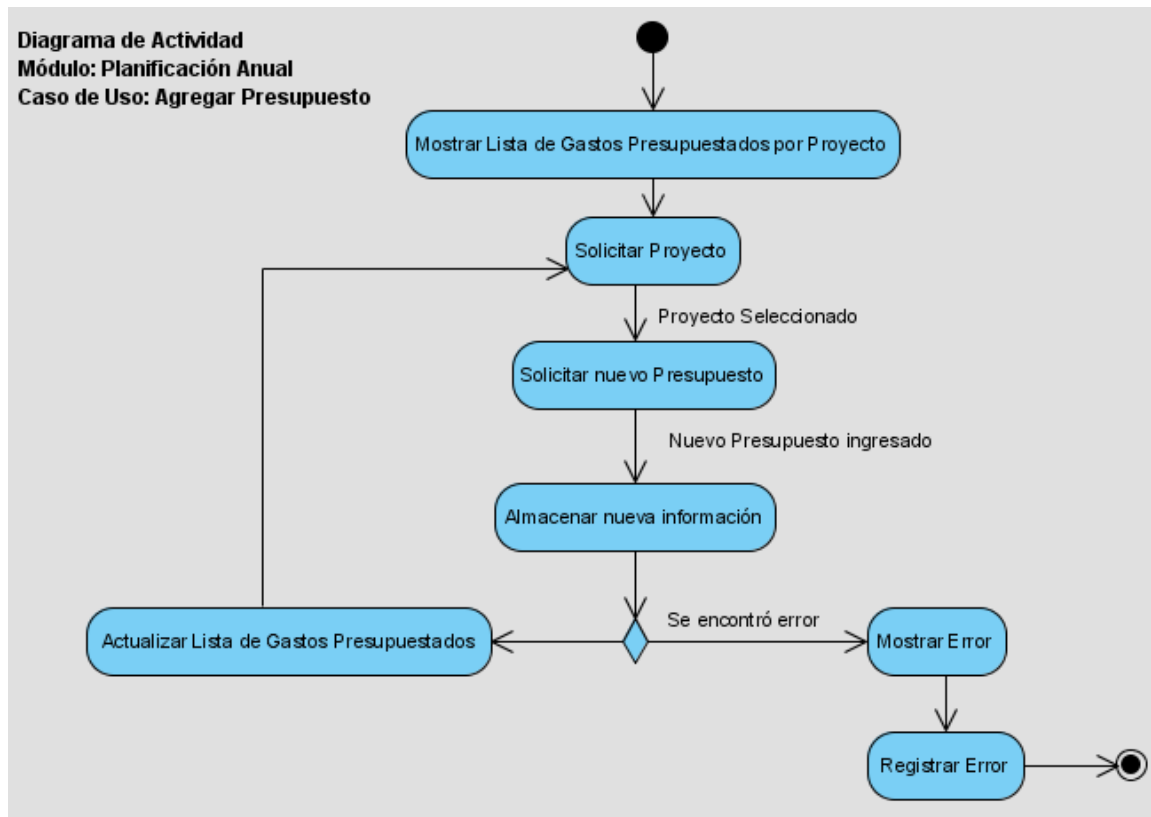


Diagrama de actividad para el caso de uso: Agregar Presupuesto.

Código fuente:

```
/*
 * paPresupuesto.java
 *
 * Created on January 27, 2008, 3:50 PM
 * Copyright sfons0
 */
package segd.PlanAnual;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.DropDown;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.Table;
```

```

import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextArea;
import com.sun.webui.jsf.component.TextField;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import javax.faces.convert.LongConverter;
import javax.faces.event.ValueChangeEvent;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class paPresupuesto extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
        private int __placeholder;

        /**
         * <p>Automatically managed component initialization.
         <strong>WARNING:</strong>
         * This method is automatically generated, so any user-specified code
         inserted
         * here is subject to being replaced.</p>
         */
        private void _init() throws Exception {

            tareaProyectoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) get
            Value("#{SessionBean1.tareaProyectoRowSet}"));

            proyectoDeptoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) get
            Value("#{SessionBean1.proyectoDeptoRowSet}"));
        }

        private Page page1 = new Page();

        public Page getPage1() {
            return page1;
        }

        public void setPage1(Page p) {
            this.page1 = p;
        }

        private Html html1 = new Html();

        public Html getHtml1() {
            return html1;
        }

        public void setHtml1(Html h) {
            this.html1 = h;
        }

```

```

    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

    public void setHead1(Head h) {
        this.head1 = h;
    }

    private Link link1 = new Link();

    public Link getLink1() {
        return link1;
    }

    public void setLink1(Link l) {
        this.link1 = l;
    }

    private Body body1 = new Body();

    public Body getBody1() {
        return body1;
    }

    public void setBody1(Body b) {
        this.body1 = b;
    }

    private Form form1 = new Form();

    public Form getForm1() {
        return form1;
    }

    public void setForm1(Form f) {
        this.form1 = f;
    }

    private Table table1 = new Table();

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

```

```

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

    private CachedRowSetDataProvider tareaProyectoDataProvider = new
    CachedRowSetDataProvider();

    public CachedRowSetDataProvider getTareaProyectoDataProvider() {
        return tareaProyectoDataProvider;
    }

    public void setTareaProyectoDataProvider(CachedRowSetDataProvider crsdp)
    {
        this.tareaProyectoDataProvider = crsdp;
    }

    private TableColumn tableColumn1 = new TableColumn();

    public TableColumn getTableColumn1() {
        return tableColumn1;
    }

    public void setTableColumn1(TableColumn tc) {
        this.tableColumn1 = tc;
    }

    private TextField textField1 = new TextField();

    public TextField getTextField1() {
        return textField1;
    }

    public void setTextField1(TextField tf) {
        this.textField1 = tf;
    }

    private TableColumn tableColumn2 = new TableColumn();

    public TableColumn getTableColumn2() {
        return tableColumn2;
    }

    public void setTableColumn2(TableColumn tc) {
        this.tableColumn2 = tc;
    }

    private TextField textField2 = new TextField();

    public TextField getTextField2() {
        return textField2;
    }

    public void setTextField2(TextField tf) {
        this.textField2 = tf;
    }

    private TableColumn tableColumn3 = new TableColumn();

```

```

public TableColumn getTableColumn3() {
    return tableColumn3;
}

public void setTableColumn3(TableColumn tc) {
    this.tableColumn3 = tc;
}

private TextField textField3 = new TextField();

public TextField getTextField3() {
    return textField3;
}

public void setTextField3(TextField tf) {
    this.textField3 = tf;
}

private TableColumn tableColumn4 = new TableColumn();

public TableColumn getTableColumn4() {
    return tableColumn4;
}

public void setTableColumn4(TableColumn tc) {
    this.tableColumn4 = tc;
}

private TextField textField4 = new TextField();

public TextField getTextField4() {
    return textField4;
}

public void setTextField4(TextField tf) {
    this.textField4 = tf;
}

private TableColumn tableColumn5 = new TableColumn();

public TableColumn getTableColumn5() {
    return tableColumn5;
}

public void setTableColumn5(TableColumn tc) {
    this.tableColumn5 = tc;
}

private TextField textField5 = new TextField();

public TextField getTextField5() {
    return textField5;
}

public void setTextField5(TextField tf) {
    this.textField5 = tf;
}

```

```

private TableColumn tableColumn6 = new TableColumn();

public TableColumn getTableColumn6() {
    return tableColumn6;
}

public void setTableColumn6(TableColumn tc) {
    this.tableColumn6 = tc;
}

private TextField textField6 = new TextField();

public TextField getTextField6() {
    return textField6;
}

public void setTextField6(TextField tf) {
    this.textField6 = tf;
}

private TableColumn tableColumn7 = new TableColumn();

public TableColumn getTableColumn7() {
    return tableColumn7;
}

public void setTableColumn7(TableColumn tc) {
    this.tableColumn7 = tc;
}

private TextField textField7 = new TextField();

public TextField getTextField7() {
    return textField7;
}

public void setTextField7(TextField tf) {
    this.textField7 = tf;
}

private TableColumn tableColumn8 = new TableColumn();

public TableColumn getTableColumn8() {
    return tableColumn8;
}

public void setTableColumn8(TableColumn tc) {
    this.tableColumn8 = tc;
}

private TextField textField8 = new TextField();

public TextField getTextField8() {
    return textField8;
}

public void setTextField8(TextField tf) {
    this.textField8 = tf;
}

```



```

    }

    private TableColumn tableColumn9 = new TableColumn();

    public TableColumn getTableColumn9() {
        return tableColumn9;
    }

    public void setTableColumn9(TableColumn tc) {
        this.tableColumn9 = tc;
    }

    private TextField textField9 = new TextField();

    public TextField getTextField9() {
        return textField9;
    }

    public void setTextField9(TextField tf) {
        this.textField9 = tf;
    }

    private TableColumn tableColumn10 = new TableColumn();

    public TableColumn getTableColumn10() {
        return tableColumn10;
    }

    public void setTableColumn10(TableColumn tc) {
        this.tableColumn10 = tc;
    }

    private TextField textField10 = new TextField();

    public TextField getTextField10() {
        return textField10;
    }

    public void setTextField10(TextField tf) {
        this.textField10 = tf;
    }

    private TableColumn tableColumn11 = new TableColumn();

    public TableColumn getTableColumn11() {
        return tableColumn11;
    }

    public void setTableColumn11(TableColumn tc) {
        this.tableColumn11 = tc;
    }

    private TextField textField11 = new TextField();

    public TextField getTextField11() {
        return textField11;
    }

```

```

public void setTextField11(TextField tf) {
    this.textField11 = tf;
}

private TableColumn tableColumn12 = new TableColumn();

public TableColumn getTableColumn12() {
    return tableColumn12;
}

public void setTableColumn12(TableColumn tc) {
    this.tableColumn12 = tc;
}

private TextField textField12 = new TextField();

public TextField getTextField12() {
    return textField12;
}

public void setTextField12(TextField tf) {
    this.textField12 = tf;
}

private TableColumn tableColumn13 = new TableColumn();

public TableColumn getTableColumn13() {
    return tableColumn13;
}

public void setTableColumn13(TableColumn tc) {
    this.tableColumn13 = tc;
}

private TextField textField13 = new TextField();

public TextField getTextField13() {
    return textField13;
}

public void setTextField13(TextField tf) {
    this.textField13 = tf;
}

private TableColumn tableColumn14 = new TableColumn();

public TableColumn getTableColumn14() {
    return tableColumn14;
}

public void setTableColumn14(TableColumn tc) {
    this.tableColumn14 = tc;
}

private TextArea textArea1 = new TextArea();

public TextArea getTextArea1() {
    return textArea1;
}

```

```

    }

    public void setTextArea1(TextArea ta) {
        this.textArea1 = ta;
    }

    private TableColumn tableColumn15 = new TableColumn();

    public TableColumn getTableColumn15() {
        return tableColumn15;
    }

    public void setTableColumn15(TableColumn tc) {
        this.tableColumn15 = tc;
    }

    private Button btnBorrar = new Button();

    public Button getBtnBorrar() {
        return btnBorrar;
    }

    public void setBtnBorrar(Button b) {
        this.btnBorrar = b;
    }

    private DropDown ddlIdProyecto = new DropDown();

    public DropDown getDdlIdProyecto() {
        return ddlIdProyecto;
    }

    public void setDdlIdProyecto(DropDown dd) {
        this.ddlIdProyecto = dd;
    }

    private Label lblSubTitulo = new Label();

    public Label getLblSubTitulo() {
        return lblSubTitulo;
    }

    public void setLblSubTitulo(Label l) {
        this.lblSubTitulo = l;
    }

    private Label lblTituloPagina = new Label();

    public Label getLblTituloPagina() {
        return lblTituloPagina;
    }

    public void setLblTituloPagina(Label l) {
        this.lblTituloPagina = l;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

```

```

public MessageGroup getMessageGroup1() {
    return messageGroup1;
}

public void setMessageGroup1(MessageGroup mg) {
    this.messageGroup1 = mg;
}

private CachedRowSetDataProvider proyectoDeptoDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getProyectoDeptoDataProvider() {
    return proyectoDeptoDataProvider;
}

public void setProyectoDeptoDataProvider(CachedRowSetDataProvider crsdp)
{
    this.proyectoDeptoDataProvider = crsdp;
}

private LongConverter ddlIdProyectoConverter = new LongConverter();

public LongConverter getDdlIdProyectoConverter() {
    return ddlIdProyectoConverter;
}

public void setDdlIdProyectoConverter(LongConverter lc) {
    this.ddlIdProyectoConverter = lc;
}

private Button btnAgregar = new Button();

public Button getBtnAgregar() {
    return btnAgregar;
}

public void setBtnAgregar(Button b) {
    this.btnAgregar = b;
}

private Button btnDeshacer = new Button();

public Button getBtnDeshacer() {
    return btnDeshacer;
}

public void setBtnDeshacer(Button b) {
    this.btnDeshacer = b;
}

private Button btnGuardar = new Button();

public Button getBtnGuardar() {
    return btnGuardar;
}

public void setBtnGuardar(Button b) {
    this.btnGuardar = b;
}

```

```

    }

    private Button btnReporte = new Button();

    public Button getBtnReporte() {
        return btnReporte;
    }

    public void setBtnReporte(Button b) {
        this.btnReporte = b;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>
     */
    public paPresupuesto() {
    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property
     * values of the components do <strong>not</strong> represent any
     * values submitted with this request. Instead, they represent the
     * property values that were saved for this view when it was
     rendered.</p>
     */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
        Initialization">
        // Initialize automatically managed components
        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {
            log("paPresupuesto Initialization Failure", e);
            throw e instanceof FacesException ? (FacesException) e: new
        FacesException(e);
        }

        // </editor-fold>
        // Perform application initialization that must complete
        // *after* managed components are initialized
        // TODO - add your own initialization code here
    }

```

```

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {
}

/**
 * <p>Callback method that is called just before rendering takes place.
 * This method will <strong>only</strong> be called for the page that
 * will actually be rendered (and not, for example, on a page that
 * handled a postback and then navigated to a different page).
Customize
 * this method to allocate resources that will be required for rendering
 * this page.</p>
 */
public void prerender() {
    ExternalContext externalContext =
getFacesContext().getExternalContext();
    try{
        if (getSessionBean1().getIdUsuario() == null){
            externalContext.redirect("faces/login.jsp");
        } else {
            if (ddlIdProyecto.getSelected() == null ) {
                getSessionBean1().getProyectoDeptoRowSet().setObject(1,
getSessionBean1().getIdCargo());
                proyectoDeptoDataProvider.refresh();
                Object firstProyecto = null;
                proyectoDeptoDataProvider.cursorFirst();
                firstProyecto =
proyectoDeptoDataProvider.getValue("idProyecto");
                ddlIdProyecto.setSelected(firstProyecto);
                getSessionBean1().getTareaProyectoRowSet().setObject(1,
firstProyecto);

                tareaProyectoDataProvider.refresh();
            } else {
                // Synchronize data provider with current selection

proyectoDeptoDataProvider.setCursorRow(proyectoDeptoDataProvider.findFirst("
idProyecto", ddlIdProyecto.getSelected()));
            }
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :" + ex);
    }
}

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,

```

```

    * <code>preprocess()</code>, or <code>prerender()</code> methods (or
    * acquired during execution of an event handler).</p>
    */
    public void destroy() {
        tareaProyectoDataProvider.close();
        proyectoDeptoDataProvider.close();
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected RequestBean1 getRequestBean1() {
        return (RequestBean1) getBean("RequestBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1) getBean("SessionBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1) getBean("ApplicationBean1");
    }

    public void ddlIdProyecto_processValueChange(ValueChangeEvent event) {
        // TODO: Replace with your code
        Object selectedProyecto = ddlIdProyecto.getSelected();
        try {

proyectoDeptoDataProvider.setCursorRow(proyectoDeptoDataProvider.findFirst("
idProyecto", selectedProyecto));
            getSessionBean1().getTareaProyectoRowSet().setObject(1,
selectedProyecto);
            tareaProyectoDataProvider.refresh();
            //form1.discardSubmittedValues("vfGuardar");
        } catch (Exception e) {
            error("No pude seleccionar " + selectedProyecto);
            log("No pude seleccionar " + selectedProyecto, e);
        }
    }

    public String btnAgregar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            RowKey rk = tareaProyectoDataProvider.appendRow();
            tareaProyectoDataProvider.setCursorRow(rk);

            tareaProyectoDataProvider.setValue("idProyecto",
ddlIdProyecto.getSelected());
            tareaProyectoDataProvider.setValue("tipo", "P");
            tareaProyectoDataProvider.setValue("comentario", new
String(""));

```

```

        tareaProyectoDataProvider.setValue("idEmpleado",
getSessionBean1().getIdUsuario());
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnDeshacer_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        tareaProyectoDataProvider.refresh();

    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnGuardar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        tareaProyectoDataProvider.commitChanges();
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :"+ex.getMessage());
    }
    return null;
}

public String btnBorrar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        if (rk != null) {
            tareaProyectoDataProvider.removeRow(rk);
            tareaProyectoDataProvider.commitChanges();
            tareaProyectoDataProvider.refresh();
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnReporte_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    ExternalContext externalContext =
getFacesContext().getExternalContext();
    Object selectedProyecto = ddlIdProyecto.getSelected();
    try {

```



```

externalContext.redirect("http://localhost/ReportServer/?/ITCA/rptPresupuest
o&rs:Command=Render&rc:Parameters=false&intProyecto=" +
selectedProyecto.toString());
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}
}
}

```

1.5.4 Agregar Cuestionario y preguntas

Son un conjunto de dos programas encargados de administrar el catálogo de cuestionarios y preguntas en la base de datos.

Diagrama de actividad:

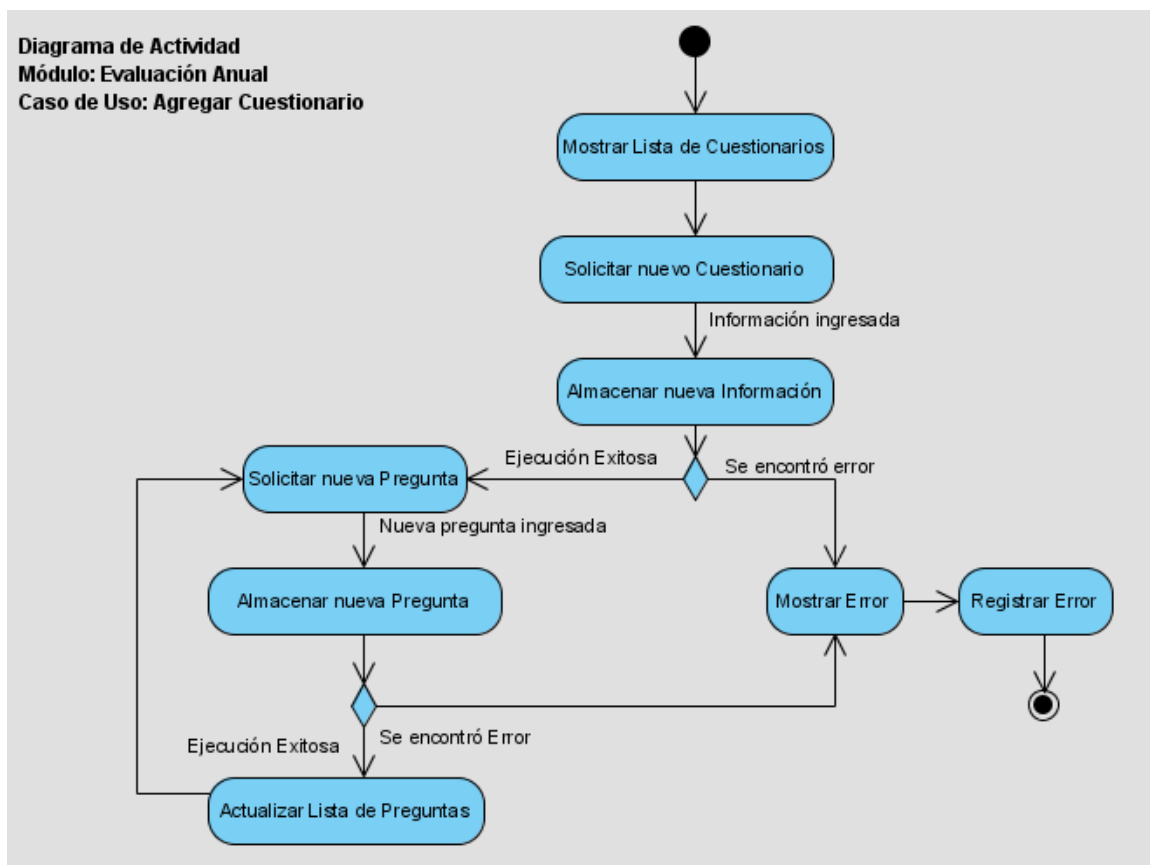


Diagrama de actividad para el caso de uso: Agregar Cuestionario.

1.5.4.1 Agregar Cuestionario

Se encarga de agregar, modificar y eliminar los encabezados de cuestionarios en la base de datos.

Nombre del programa: /cuestionario.java

Código fuente:

```
/*
 * cuestionario.java
 *
 * Created on November 17, 2007, 8:15 PM
 * Copyright sfons0
 */
package segd;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.Message;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextField;
import java.io.IOException;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class cuestionario extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
     <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
     inserted
     * here is subject to being replaced.</p>

```

```

    */
    private void _init() throws Exception {

cuestionarioDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) getV
alue("#{SessionBean1.cuestionarioRowSet}"));
    }

    private Page page1 = new Page();

    public Page getPage1() {
        return page1;
    }

    public void setPage1(Page p) {
        this.page1 = p;
    }

    private Html html1 = new Html();

    public Html getHtml1() {
        return html1;
    }

    public void setHtml1(Html h) {
        this.html1 = h;
    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

    public void setHead1(Head h) {
        this.head1 = h;
    }

    private Link link1 = new Link();

    public Link getLink1() {
        return link1;
    }

    public void setLink1(Link l) {
        this.link1 = l;
    }

    private Body body1 = new Body();

    public Body getBody1() {
        return body1;
    }

    public void setBody1(Body b) {
        this.body1 = b;
    }

    private Form form1 = new Form();

```

```

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private MessageGroup messageGroup1 = new MessageGroup();

public MessageGroup getMessageGroup1() {
    return messageGroup1;
}

public void setMessageGroup1(MessageGroup mg) {
    this.messageGroup1 = mg;
}

private Table table1 = new Table();

public Table getTable1() {
    return table1;
}

public void setTable1(Table t) {
    this.table1 = t;
}

private TableRowGroup tableRowGroup1 = new TableRowGroup();

public TableRowGroup getTableRowGroup1() {
    return tableRowGroup1;
}

public void setTableRowGroup1(TableRowGroup trg) {
    this.tableRowGroup1 = trg;
}

private CachedRowSetDataProvider cuestionarioDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getCuestionarioDataProvider() {
    return cuestionarioDataProvider;
}

public void setCuestionarioDataProvider(CachedRowSetDataProvider crsdp)
{
    this.cuestionarioDataProvider = crsdp;
}

private TableColumn tableColumn1 = new TableColumn();

public TableColumn getTableColumn1() {
    return tableColumn1;
}

public void setTableColumn1(TableColumn tc) {

```

```

        this.tableColumn1 = tc;
    }

    private TableColumn tableColumn2 = new TableColumn();

    public TableColumn getTableColumn2() {
        return tableColumn2;
    }

    public void setTableColumn2(TableColumn tc) {
        this.tableColumn2 = tc;
    }

    private TextField textField1 = new TextField();

    public TextField getTextField1() {
        return textField1;
    }

    public void setTextField1(TextField tf) {
        this.textField1 = tf;
    }

    private TextField textField2 = new TextField();

    public TextField getTextField2() {
        return textField2;
    }

    public void setTextField2(TextField tf) {
        this.textField2 = tf;
    }

    private Button btnGuardar = new Button();

    public Button getBtnGuardar() {
        return btnGuardar;
    }

    public void setBtnGuardar(Button b) {
        this.btnGuardar = b;
    }

    private Button btnAgrega = new Button();

    public Button getBtnAgrega() {
        return btnAgrega;
    }

    public void setBtnAgrega(Button b) {
        this.btnAgrega = b;
    }

    private Button btnDeshacer = new Button();

    public Button getBtnDeshacer() {
        return btnDeshacer;
    }

```

```

public void setBtnDeshacer(Button b) {
    this.btnDeshacer = b;
}

private Label lblTituloPagina = new Label();

public Label getLblTituloPagina() {
    return lblTituloPagina;
}

public void setLblTituloPagina(Label l) {
    this.lblTituloPagina = l;
}

private Message message1 = new Message();

public Message getMessage1() {
    return message1;
}

public void setMessage1(Message m) {
    this.message1 = m;
}

private Message message2 = new Message();

public Message getMessage2() {
    return message2;
}

public void setMessage2(Message m) {
    this.message2 = m;
}

private TableColumn tableColumn3 = new TableColumn();

public TableColumn getTableColumn3() {
    return tableColumn3;
}

public void setTableColumn3(TableColumn tc) {
    this.tableColumn3 = tc;
}

private Button btnBorrar = new Button();

public Button getBtnBorrar() {
    return btnBorrar;
}

public void setBtnBorrar(Button b) {
    this.btnBorrar = b;
}

private TableColumn tableColumn4 = new TableColumn();

public TableColumn getTableColumn4() {

```

```

        return tableColumn4;
    }

    public void setTableColumn4(TableColumn tc) {
        this.tableColumn4 = tc;
    }

    private Button btnPreguntas = new Button();

    public Button getBtnPreguntas() {
        return btnPreguntas;
    }

    public void setBtnPreguntas(Button b) {
        this.btnPreguntas = b;
    }

    private Label lblTituloPagina1 = new Label();

    public Label getLblTituloPagina1() {
        return lblTituloPagina1;
    }

    public void setLblTituloPagina1(Label l) {
        this.lblTituloPagina1 = l;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>
     */
    public cuestionario() {
    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property
     * values of the components do <strong>not</strong> represent any
     * values submitted with this request. Instead, they represent the
     * property values that were saved for this view when it was
     rendered.</p>
     */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
        Initialization">
        // Initialize automatically managed components

```

```

        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {
            log("cuestionario Initialization Failure", e);
            throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
        }

        // </editor-fold>
        // Perform application initialization that must complete
        // *after* managed components are initialized
        // TODO - add your own initialization code here
    }

    /**
     * <p>Callback method that is called after the component tree has been
     * restored, but before any event processing takes place. This method
     * will <strong>only</strong> be called on a postback request that
     * is processing a form submit. Customize this method to allocate
     * resources that will be required in your event handlers.</p>
     */
    public void preprocess() {
    }

    /**
     * <p>Callback method that is called just before rendering takes place.
     * This method will <strong>only</strong> be called for the page that
     * will actually be rendered (and not, for example, on a page that
     * handled a postback and then navigated to a different page).
     Customize
     * this method to allocate resources that will be required for rendering
     * this page.</p>
     */
    public void prerender() {
        ExternalContext externalContext =
getFacesContext().getExternalContext();

        if (getSessionBean1().getIdUsuario() == null){
            try {
                externalContext.redirect("faces/login.jsp");
            } catch (IOException ioe) {
                log("Error :", ioe);
                error("Error :" + ioe);
            }
        }
    }

    /**
     * <p>Callback method that is called after rendering is completed for
     * this request, if <code>init()</code> was called (regardless of
whether
     * or not this was the page that was actually rendered). Customize this
     * method to release resources acquired in the <code>init()</code>,
     * <code>preprocess()</code>, or <code>prerender()</code> methods (or
     * acquired during execution of an event handler).</p>
     */
    public void destroy() {

```



```

        cuestionarioDataProvider.close();
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1)getBean("ApplicationBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1)getBean("SessionBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected RequestBean1 getRequestBean1() {
        return (RequestBean1)getBean("RequestBean1");
    }

    public String btnAgregar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            RowKey rk = cuestionarioDataProvider.appendRow();
            cuestionarioDataProvider.setCursorRow(rk);

            // cuestionarioDataProvider.setValue("TRIP.TRIPID", new
            Integer(0));
            // cuestionarioDataProvider.setValue("TRIP.PERSONID",
            personDD.getSelected());
            // cuestionarioDataProvider.setValue("TRIP.TRIPTYPEID", new
            Integer(1));
        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnGuardar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            cuestionarioDataProvider.commitChanges();
        } catch (Exception ex) {
            log("Error :", ex);
            error("Error :"+ex.getMessage());
        }
        return null;
    }

    public String btnBorrar_action() {
        // TODO: Process the action. Return value is a navigation

```

```

        // case name where null will return to the same page.
        try {
            RowKey rk = tableRowGroup1.getRowKey();
            if (rk != null) {
                cuestionarioDataProvider.removeRow(rk);
                cuestionarioDataProvider.commitChanges();
                cuestionarioDataProvider.refresh();
            } catch (Exception ex) {
                log("Error :", ex);
                error(ex.getMessage());
            }
            return null;
        }

        public String btnDeshacer_action() {
            // TODO: Process the action. Return value is a navigation
            // case name where null will return to the same page.
            try {
                cuestionarioDataProvider.refresh();

            } catch (Exception ex) {
                log("Error :", ex);
                error(ex.getMessage());
            }
            return null;
        }

        public String btnPreguntas_action() {
            // TODO: Process the action. Return value is a navigation
            // case name where null will return to the same page.
            RowKey rk = tableRowGroup1.getRowKey();

            getRequestBody().setIdCuestionario((String)cuestionarioDataProvider.getValue("idCuestionario", rk));

            return "case1";
        }
    }
}

```

1.5.4.2 Agregar Pregunta

Este programa se encarga de agregar, modificar y eliminar las preguntas para cada uno de los cuestionarios utilizados en las evaluaciones.

Nombre del programa: /preguntas.java

```

/*
 * preguntas.java
 *
 * Created on November 17, 2007, 11:58 PM
 * Copyright sfons0
 */
package segd;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;

```

```

import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.DropDown;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.Message;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.StaticText;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextField;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import javax.faces.convert.IntegerConverter;
import javax.faces.event.ValueChangeEvent;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class preguntas extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
        private int __placeholder;

        /**
         * <p>Automatically managed component initialization.
         <strong>WARNING:</strong>
         * This method is automatically generated, so any user-specified code
         inserted
         * here is subject to being replaced.</p>
         */
        private void _init() throws Exception {

            cuestionarioDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue(
                "#{SessionBean1.cuestionarioRowSet}"));

            preguntaDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue(
                "#{SessionBean1.preguntaRowSet}"));

            areaevaluacionDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)ge
                tValue("#{SessionBean1.areaevaluacionRowSet}"));
        }

        private Page page1 = new Page();

        public Page getPage1() {
            return page1;
        }
    }

```

```

public void setPage1(Page p) {
    this.page1 = p;
}

private Html html1 = new Html();

public Html getHtml1() {
    return html1;
}

public void setHtml1(Html h) {
    this.html1 = h;
}

private Head head1 = new Head();

public Head getHead1() {
    return head1;
}

public void setHead1(Head h) {
    this.head1 = h;
}

private Link link1 = new Link();

public Link getLink1() {
    return link1;
}

public void setLink1(Link l) {
    this.link1 = l;
}

private Body body1 = new Body();

public Body getBody1() {
    return body1;
}

public void setBody1(Body b) {
    this.body1 = b;
}

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private Label lblTituloPagina = new Label();

public Label getLblTituloPagina() {

```

```

        return lblTituloPagina;
    }

    public void setLblTituloPagina(Label l) {
        this.lblTituloPagina = l;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

    public MessageGroup getMessageGroup1() {
        return messageGroup1;
    }

    public void setMessageGroup1(MessageGroup mg) {
        this.messageGroup1 = mg;
    }

    private StaticText staticText1 = new StaticText();

    public StaticText getStaticText1() {
        return staticText1;
    }

    public void setStaticText1(StaticText st) {
        this.staticText1 = st;
    }

    private DropDown ddlCuestionario = new DropDown();

    public DropDown getDdlCuestionario() {
        return ddlCuestionario;
    }

    public void setDdlCuestionario(DropDown dd) {
        this.ddlCuestionario = dd;
    }

    private Table table1 = new Table();

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

```

```

    private CachedRowSetDataProvider cuestionarioDataProvider = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getCuestionarioDataProvider() {
        return cuestionarioDataProvider;
    }

    public void setCuestionarioDataProvider(CachedRowSetDataProvider crsdp)
{
        this.cuestionarioDataProvider = crsdp;
    }

    private CachedRowSetDataProvider preguntaDataProvider = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getPreguntaDataProvider() {
        return preguntaDataProvider;
    }

    public void setPreguntaDataProvider(CachedRowSetDataProvider crsdp) {
        this.preguntaDataProvider = crsdp;
    }

    private TableColumn tableColumn3 = new TableColumn();

    public TableColumn getTableColumn3() {
        return tableColumn3;
    }

    public void setTableColumn3(TableColumn tc) {
        this.tableColumn3 = tc;
    }

    private TableColumn tableColumn4 = new TableColumn();

    public TableColumn getTableColumn4() {
        return tableColumn4;
    }

    public void setTableColumn4(TableColumn tc) {
        this.tableColumn4 = tc;
    }

    private TextField txtPregunta = new TextField();

    public TextField getTxtPregunta() {
        return txtPregunta;
    }

    public void setTxtPregunta(TextField tf) {
        this.txtPregunta = tf;
    }

    private DropDown ddlIdAreaEvaluacion = new DropDown();

    public DropDown getDdlIdAreaEvaluacion() {
        return ddlIdAreaEvaluacion;
    }

```

```

    public void setDdlIdAreaEvaluacion(DropDown dd) {
        this.ddlIdAreaEvaluacion = dd;
    }

    private CachedRowSetDataProvider areaevaluacionDataProvider = new
    CachedRowSetDataProvider();

    public CachedRowSetDataProvider getAreaevaluacionDataProvider() {
        return areaevaluacionDataProvider;
    }

    public void setAreaevaluacionDataProvider(CachedRowSetDataProvider
    crsdp) {
        this.areaevaluacionDataProvider = crsdp;
    }

    private IntegerConverter ddlIdAreaEvaluacionConverter = new
    IntegerConverter();

    public IntegerConverter getDdlIdAreaEvaluacionConverter() {
        return ddlIdAreaEvaluacionConverter;
    }

    public void setDdlIdAreaEvaluacionConverter(IntegerConverter ic) {
        this.ddlIdAreaEvaluacionConverter = ic;
    }

    private Button btnDeshacer = new Button();

    public Button getBtnDeshacer() {
        return btnDeshacer;
    }

    public void setBtnDeshacer(Button b) {
        this.btnDeshacer = b;
    }

    private Button btnGuardar = new Button();

    public Button getBtnGuardar() {
        return btnGuardar;
    }

    public void setBtnGuardar(Button b) {
        this.btnGuardar = b;
    }

    private Button btnAgregar = new Button();

    public Button getBtnAgregar() {
        return btnAgregar;
    }

    public void setBtnAgregar(Button b) {
        this.btnAgregar = b;
    }

```

```

private Message message1 = new Message();

public Message getMessage1() {
    return message1;
}

public void setMessage1(Message m) {
    this.message1 = m;
}

private TableColumn tableColumn1 = new TableColumn();

public TableColumn getTableColumn1() {
    return tableColumn1;
}

public void setTableColumn1(TableColumn tc) {
    this.tableColumn1 = tc;
}

private Button btnBorrar = new Button();

public Button getBtnBorrar() {
    return btnBorrar;
}

public void setBtnBorrar(Button b) {
    this.btnBorrar = b;
}

private StaticText lblIdCuestionario = new StaticText();

public StaticText getLblIdCuestionario() {
    return lblIdCuestionario;
}

public void setLblIdCuestionario(StaticText st) {
    this.lblIdCuestionario = st;
}

// </editor-fold>

/**
 * <p>Construct a new Page bean instance.</p>
 */
public preguntas() {
}

/**
 * <p>Callback method that is called whenever a page is navigated to,
 * either directly via a URL, or indirectly via page navigation.
 * Customize this method to acquire resources that will be needed
 * for event handlers and lifecycle methods, whether or not this
 * page is performing post back processing.</p>
 *
 * <p>Note that, if the current request is a postback, the property
 * values of the components do <strong>not</strong> represent any
 * values submitted with this request. Instead, they represent the

```



```

        * property values that were saved for this view when it was
        rendered.</p>
        */
        public void init() {
            // Perform initializations inherited from our superclass
            super.init();
            // Perform application initialization that must complete
            // *before* managed components are initialized
            // TODO - add your own initialization code here

            // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
            // Initialize automatically managed components
            // *Note* - this logic should NOT be modified
            try {
                _init();
            } catch (Exception e) {
                log("preguntas Initialization Failure", e);
                throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
            }

            // </editor-fold>
            // Perform application initialization that must complete
            // *after* managed components are initialized
            // TODO - add your own initialization code here
        }

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {
}

/**
 * <p>Callback method that is called just before rendering takes place.
 * This method will <strong>only</strong> be called for the page that
 * will actually be rendered (and not, for example, on a page that
 * handled a postback and then navigated to a different page).
Customize
 * this method to allocate resources that will be required for rendering
 * this page.</p>
 */
public void prerender() {
    ExternalContext externalContext =
getFacesContext().getExternalContext();
    try{
        if (getSessionBean1().getIdUsuario() == null){
            externalContext.redirect("faces/login.jsp");
        } else {
            if (getRequestBean1().getIdCuestionario() != null){
                ddlCuestionario.setSelected(getRequestBean1().getIdCuestionario());
            }
        }
    } catch (Exception e) {
        log("Error in prerender method", e);
    }
}

```

```

        getSessionBean1().getPreguntaRowSet().setObject(1,
getRequestBean1().getIdCuestionario());
        preguntaDataProvider.refresh();
    }

    if (ddlCuestionario.getSelected() == null ) {
        Object firstIdCuestionario = null;
        cuestionarioDataProvider.cursorFirst();
        firstIdCuestionario =
cuestionarioDataProvider.getValue("idCuestionario");
        ddlCuestionario.setSelected(firstIdCuestionario);
        getSessionBean1().getPreguntaRowSet().setObject(1,
firstIdCuestionario);
        preguntaDataProvider.refresh();
    } else {
        // Synchronize data provider with current selection

cuestionarioDataProvider.setCursorRow(cuestionarioDataProvider.findFirst("id
Cuestionario", ddlCuestionario.getSelected()));
    }
}
} catch (Exception ex) {
    log("Error :", ex);
    error("Error :" + ex);
}
}

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,
 * <code>preprocess()</code>, or <code>prerender()</code> methods (or
 * acquired during execution of an event handler).</p>
 */
public void destroy() {
    cuestionarioDataProvider.close();
    preguntaDataProvider.close();
    areaevaluacionDataProvider.close();
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected SessionBean1 getSessionBean1() {
    return (SessionBean1)getBean("SessionBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected RequestBean1 getRequestBean1() {
    return (RequestBean1)getBean("RequestBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>

```

```

    */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1) getBean("ApplicationBean1");
    }

    public void ddlCuestionario_processValueChange(ValueChangeEvent event) {
        // TODO: Replace with your code
        Object selectedIdCuestionario = ddlCuestionario.getSelected();
        try {

cuestionarioDataProvider.setCursorRow(cuestionarioDataProvider.findFirst("id
Cuestionario", selectedIdCuestionario));
            getSessionBean1().getPreguntaRowSet().setObject(1,
selectedIdCuestionario);
            preguntaDataProvider.refresh();
            form1.discardSubmittedValues("vfGuardar");
        } catch (Exception e) {
            error("No pude seleccionar " + selectedIdCuestionario);
            log("No pude seleccionar " + selectedIdCuestionario, e);
        }
    }

    public String btnDeshacer_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        form1.discardSubmittedValues("vfGuardar");
        try {
            preguntaDataProvider.refresh();

        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnGuardar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {

            // Get the next key, using result of query on MaxTrip data
provider
            CachedRowSetDataProvider maxIdPregunta =
getSessionBean1().getMaxPreguntaDataProvider();
            maxIdPregunta.refresh();
            maxIdPregunta.cursorFirst();
            int newIdPregunta = ((Integer)
maxIdPregunta.getValue("maxIdPregunta")).intValue();

            // Navigate through rows with data provider
            if (preguntaDataProvider.getRowCount() > 0) {
                preguntaDataProvider.cursorFirst();
                do {
                    if
(preguntaDataProvider.getValue("idPregunta").equals(new Integer(0))) {
                        preguntaDataProvider.setValue("idPregunta", new
Integer(newIdPregunta));

```

```

        newIdPregunta++;
    }

    } while (preguntaDataProvider.cursorNext());
    };
    preguntaDataProvider.commitChanges();
} catch (Exception ex) {
    log("Error :", ex);
    error("Error :"+ex.getMessage());
}
return null;
}

public String btnAgregar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = preguntaDataProvider.appendRow();
        preguntaDataProvider.setCursorRow(rk);

        preguntaDataProvider.setValue("idPregunta", new Integer(0));
        preguntaDataProvider.setValue("idCuestionario",
ddlCuestionario.getSelected());
        preguntaDataProvider.setValue("idAreaEvaluacion", new
Integer(1));
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnBorrar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    form1.discardSubmittedValues("vfGuardar");
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        if (rk != null) {
            preguntaDataProvider.removeRow(rk);
            preguntaDataProvider.commitChanges();
            preguntaDataProvider.refresh();
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}
}
}

```

1.5.5 Configuración de la evaluación

Son dos programas que se encargan de configurar la evaluación de un empleado, esto incluye:

- La fecha de inicio de evaluación.
- El tipo de cuestionario a utilizar en la evaluación.
- La selección de los evaluadores.

Diagrama de actividad:

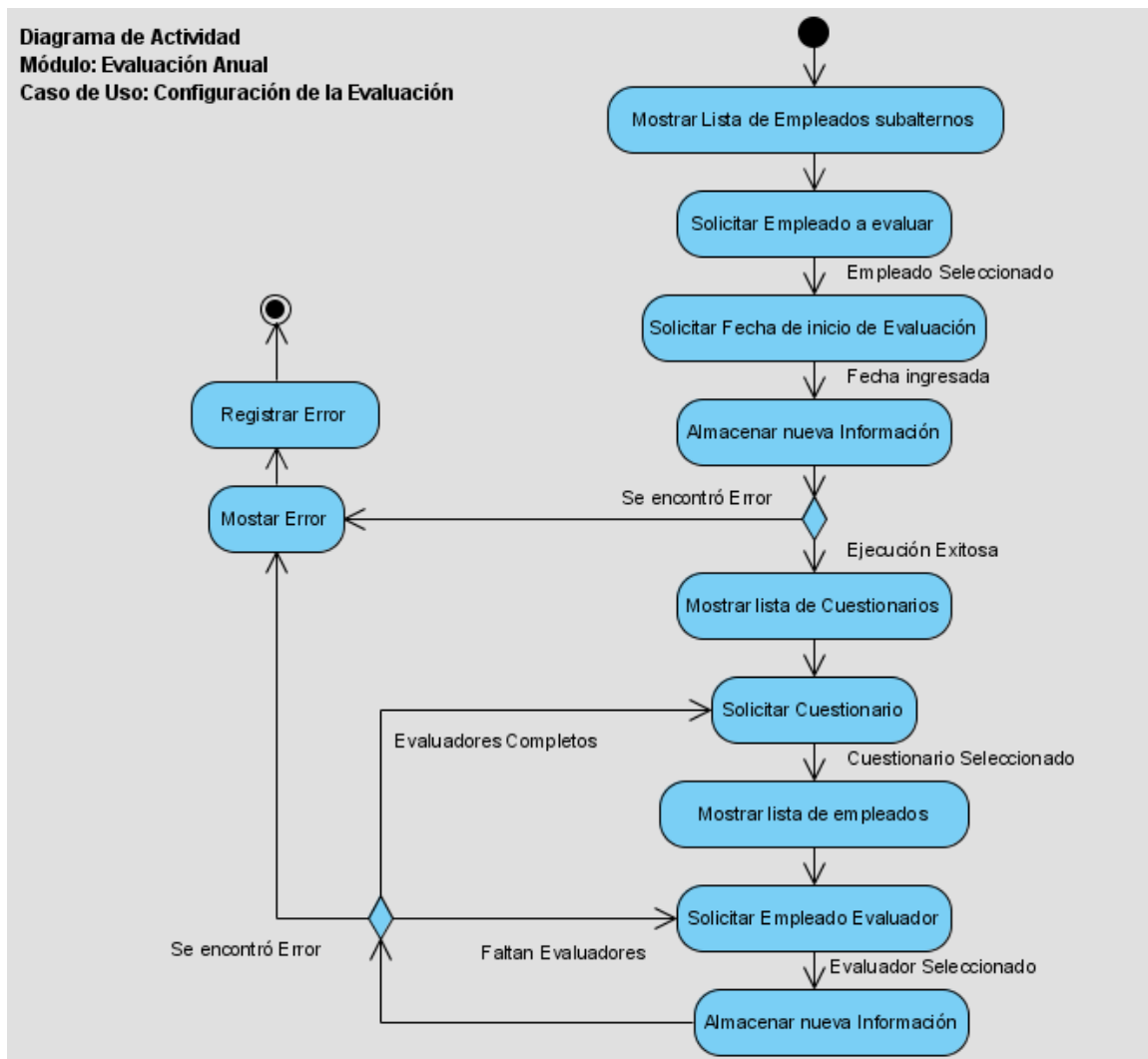


Diagrama de actividad para el caso de uso: Agregar Tarea.

Nombre del programa: /eaConfiguración.java

Código fuente:

```
/*
 * eaConfiguracion.java
 *
 * Created on November 20, 2007, 8:18 PM
 * Copyright sfons0
 */
package segd;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Calendar;
import com.sun.webui.jsf.component.DropDown;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.Message;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.StaticText;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import java.text.DateFormat;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class eaConfiguracion extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
     <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
     inserted
     * here is subject to being replaced.</p>
     */
    private void _init() throws Exception {
```

```
cuestionarioDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue("#{SessionBean1.cuestionarioRowSet}"));
```

```
subalternoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue("#{SessionBean1.subalternoRowSet}"));
}
```

```
private Page page1 = new Page();
```

```
public Page getPage1() {
    return page1;
}
```

```
public void setPage1(Page p) {
    this.page1 = p;
}
```

```
private Html html1 = new Html();
```

```
public Html getHtml1() {
    return html1;
}
```

```
public void setHtml1(Html h) {
    this.html1 = h;
}
```

```
private Head head1 = new Head();
```

```
public Head getHead1() {
    return head1;
}
```

```
public void setHead1(Head h) {
    this.head1 = h;
}
```

```
private Link link1 = new Link();
```

```
public Link getLink1() {
    return link1;
}
```

```
public void setLink1(Link l) {
    this.link1 = l;
}
```

```
private Body body1 = new Body();
```

```
public Body getBody1() {
    return body1;
}
```

```
public void setBody1(Body b) {
    this.body1 = b;
}
```

```

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private Calendar calFechaEvaluacion = new Calendar();

public Calendar getCalFechaEvaluacion() {
    return calFechaEvaluacion;
}

public void setCalFechaEvaluacion(Calendar c) {
    this.calFechaEvaluacion = c;
}

private Message message1 = new Message();

public Message getMessage1() {
    return message1;
}

public void setMessage1(Message m) {
    this.message1 = m;
}

private DropDown ddlCuestionario = new DropDown();

public DropDown getDdlCuestionario() {
    return ddlCuestionario;
}

public void setDdlCuestionario(DropDown dd) {
    this.ddlCuestionario = dd;
}

private CachedRowSetDataProvider cuestionarioDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getCuestionarioDataProvider() {
    return cuestionarioDataProvider;
}

public void setCuestionarioDataProvider(CachedRowSetDataProvider crsdp)
{
    this.cuestionarioDataProvider = crsdp;
}

private Label lblTituloPagina = new Label();

public Label getLblTituloPagina() {
    return lblTituloPagina;
}

```



```

public void setLblTituloPagina(Label l) {
    this.lblTituloPagina = l;
}

private Label lblTituloPagina2 = new Label();

public Label getLblTituloPagina2() {
    return lblTituloPagina2;
}

public void setLblTituloPagina2(Label l) {
    this.lblTituloPagina2 = l;
}

private MessageGroup messageGroup1 = new MessageGroup();

public MessageGroup getMessageGroup1() {
    return messageGroup1;
}

public void setMessageGroup1(MessageGroup mg) {
    this.messageGroup1 = mg;
}

private Table table1 = new Table();

public Table getTable1() {
    return table1;
}

public void setTable1(Table t) {
    this.table1 = t;
}

private TableRowGroup tableRowGroup1 = new TableRowGroup();

public TableRowGroup getTableRowGroup1() {
    return tableRowGroup1;
}

public void setTableRowGroup1(TableRowGroup trg) {
    this.tableRowGroup1 = trg;
}

private CachedRowSetDataProvider subalternoDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getSubalternoDataProvider() {
    return subalternoDataProvider;
}

public void setSubalternoDataProvider(CachedRowSetDataProvider crsdp) {
    this.subalternoDataProvider = crsdp;
}

private TableColumn tableColumn3 = new TableColumn();

public TableColumn getTableColumn3() {

```

```

        return tableColumn3;
    }

    public void setTableColumn3(TableColumn tc) {
        this.tableColumn3 = tc;
    }

    private StaticText staticText3 = new StaticText();

    public StaticText getStaticText3() {
        return staticText3;
    }

    public void setStaticText3(StaticText st) {
        this.staticText3 = st;
    }

    private TableColumn tableColumn4 = new TableColumn();

    public TableColumn getTableColumn4() {
        return tableColumn4;
    }

    public void setTableColumn4(TableColumn tc) {
        this.tableColumn4 = tc;
    }

    private StaticText staticText4 = new StaticText();

    public StaticText getStaticText4() {
        return staticText4;
    }

    public void setStaticText4(StaticText st) {
        this.staticText4 = st;
    }

    private TableColumn tableColumn9 = new TableColumn();

    public TableColumn getTableColumn9() {
        return tableColumn9;
    }

    public void setTableColumn9(TableColumn tc) {
        this.tableColumn9 = tc;
    }

    private StaticText staticText9 = new StaticText();

    public StaticText getStaticText9() {
        return staticText9;
    }

    public void setStaticText9(StaticText st) {
        this.staticText9 = st;
    }

    private TableColumn tableColumn10 = new TableColumn();

```

```

public TableColumn getTableColumn10() {
    return tableColumn10;
}

public void setTableColumn10(TableColumn tc) {
    this.tableColumn10 = tc;
}

private StaticText staticText10 = new StaticText();

public StaticText getStaticText10() {
    return staticText10;
}

public void setStaticText10(StaticText st) {
    this.staticText10 = st;
}

private TableColumn tableColumn1 = new TableColumn();

public TableColumn getTableColumn1() {
    return tableColumn1;
}

public void setTableColumn1(TableColumn tc) {
    this.tableColumn1 = tc;
}

private Button btnConfigurar = new Button();

public Button getBtnConfigurar() {
    return btnConfigurar;
}

public void setBtnConfigurar(Button b) {
    this.btnConfigurar = b;
}

// </editor-fold>

/**
 * <p>Construct a new Page bean instance.</p>
 */
public eaConfiguracion() {
}

/**
 * <p>Callback method that is called whenever a page is navigated to,
 * either directly via a URL, or indirectly via page navigation.
 * Customize this method to acquire resources that will be needed
 * for event handlers and lifecycle methods, whether or not this
 * page is performing post back processing.</p>
 *
 * <p>Note that, if the current request is a postback, the property
 * values of the components do <strong>not</strong> represent any
 * values submitted with this request. Instead, they represent the

```

```

        * property values that were saved for this view when it was
rendered.</p>
    */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
        // Initialize automatically managed components
        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {
            log("eaConfiguracion Initialization Failure", e);
            throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
        }

        // </editor-fold>
        // Perform application initialization that must complete
        // *after* managed components are initialized
        // TODO - add your own initialization code here
    }

    /**
     * <p>Callback method that is called after the component tree has been
     * restored, but before any event processing takes place. This method
     * will <strong>only</strong> be called on a postback request that
     * is processing a form submit. Customize this method to allocate
     * resources that will be required in your event handlers.</p>
     */
    public void preprocess() {
    }

    /**
     * <p>Callback method that is called just before rendering takes place.
     * This method will <strong>only</strong> be called for the page that
     * will actually be rendered (and not, for example, on a page that
     * handled a postback and then navigated to a different page).
     Customize
     * this method to allocate resources that will be required for rendering
     * this page.</p>
     */
    public void prerender() {
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        try{
            if (getSessionBean1().getIdUsuario() == null){
                externalContext.redirect("faces/login.jsp");
            } else {
                if (getSessionBean1().getIdCargo() != null){
                    getSessionBean1().getSubalternoRowSet().setObject(1,
getSessionBean1().getIdCargo());
                    subalternoDataProvider.refresh();
                }
            }
        } catch (Exception e) {
            log("Error in prerender method", e);
        }
    }

```

```

        }
    }
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :" + ex);
    }
}

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,
 * <code>preprocess()</code>, or <code>prerender()</code> methods (or
 * acquired during execution of an event handler).</p>
 */
public void destroy() {
    cuestionarioDataProvider.close();
    subalternoDataProvider.close();
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected RequestBean1 getRequestBean1() {
    return (RequestBean1) getBean("RequestBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected SessionBean1 getSessionBean1() {
    return (SessionBean1) getBean("SessionBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected ApplicationBean1 getApplicationBean1() {
    return (ApplicationBean1) getBean("ApplicationBean1");
}

public String btnConfigurar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        //calFechaEvaluacion.getSelectedDate().

        getSessionBean1().setStrEmpleadoEvaluacion((String)
subalternoDataProvider.getValue("idEmpleado", rk));
        getSessionBean1().setStrCuestionarioEvaluacion((String)
ddlCuestionario.getSelected());
        getSessionBean1().setStrFechaEvaluacion((String)
DateFormat.getDateInstance(DateFormat.SHORT).format(calFechaEvaluacion.getSe
lectedDate()));
        // externalContext.redirect("faces/eaEvaluadores.jsp");

```

```

        } catch (Exception ex) {
            log("Error :", ex);
            error("Error :" + ex);
        }
        return "case1";
    }
}

```

Nombre del programa: /eaEvaouadores.java

Código fuente:

```

/*
 * eaEvaluadores.java
 *
 * Created on November 20, 2007, 10:48 PM
 * Copyright sfons0
 */
package segd;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.sql.rowset.CachedRowSetXImpl;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Checkbox;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.StaticText;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import java.io.IOException;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import java.sql.*;
import java.util.Iterator;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class eaEvaluadores extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
        private int __placeholder;

```

```

    /**
     * <p>Automatically managed component initialization.
<strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
inserted
     * here is subject to being replaced.</p>
    */
    private void _init() throws Exception {
        evaluadorRowSet.setDataSourceName("java:comp/env/jdbc/dataSource4");
        evaluadorRowSet.setCommand("SELECT ALL dbo.empleado.nombres, \n
dbo.empleado.apellidos, \n                dbo.empleado.email, \n
dbo.cargo.cargo, \n                dbo.departamento.departamento, \n
dbo.empleado.idEmpleado \nFROM dbo.empleado\n                INNER JOIN dbo.cargo
ON dbo.empleado.idCargo = dbo.cargo.idCargo\n                INNER JOIN
dbo.departamento ON dbo.cargo.idDepartamento =
dbo.departamento.idDepartamento\nWHERE dbo.empleado.estado = 'A'\nORDER BY
dbo.cargo.idDepartamento ASC, \n                dbo.cargo.nivel ASC ");
        evaluadorRowSet.setTableName("empleado");

evaluadorDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValu
e("#{eaEvaluadores.evaluadorRowSet}"));
    }

    private Page page1 = new Page();

    public Page getPage1() {
        return page1;
    }

    public void setPage1(Page p) {
        this.page1 = p;
    }

    private Html html1 = new Html();

    public Html getHtml1() {
        return html1;
    }

    public void setHtml1(Html h) {
        this.html1 = h;
    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

    public void setHead1(Head h) {
        this.head1 = h;
    }

    private Link link1 = new Link();

    public Link getLink1() {
        return link1;
    }

```

```

public void setLink1(Link l) {
    this.link1 = l;
}

private Body body1 = new Body();

public Body getBody1() {
    return body1;
}

public void setBody1(Body b) {
    this.body1 = b;
}

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private Label lblTituloPagina3 = new Label();

public Label getLblTituloPagina3() {
    return lblTituloPagina3;
}

public void setLblTituloPagina3(Label l) {
    this.lblTituloPagina3 = l;
}

private Label lblTituloPagina2 = new Label();

public Label getLblTituloPagina2() {
    return lblTituloPagina2;
}

public void setLblTituloPagina2(Label l) {
    this.lblTituloPagina2 = l;
}

private CachedRowSetDataProvider evaluadorDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getEvaluadorDataProvider() {
    return evaluadorDataProvider;
}

public void setEvaluadorDataProvider(CachedRowSetDataProvider crsdp) {
    this.evaluadorDataProvider = crsdp;
}

private CachedRowSetXImpl evaluadorRowSet = new CachedRowSetXImpl();

```



```

public CachedRowSetXImpl getEvaluadorRowSet() {
    return evaluadorRowSet;
}

public void setEvaluadorRowSet(CachedRowSetXImpl crsxi) {
    this.evaluadorRowSet = crsxi;
}

private Table table1 = new Table();

public Table getTable1() {
    return table1;
}

public void setTable1(Table t) {
    this.table1 = t;
}

private TableRowGroup tableRowGroup1 = new TableRowGroup();

public TableRowGroup getTableRowGroup1() {
    return tableRowGroup1;
}

public void setTableRowGroup1(TableRowGroup trg) {
    this.tableRowGroup1 = trg;
}

private TableColumn tableColumn1 = new TableColumn();

public TableColumn getTableColumn1() {
    return tableColumn1;
}

public void setTableColumn1(TableColumn tc) {
    this.tableColumn1 = tc;
}

private StaticText staticText1 = new StaticText();

public StaticText getStaticText1() {
    return staticText1;
}

public void setStaticText1(StaticText st) {
    this.staticText1 = st;
}

private TableColumn tableColumn2 = new TableColumn();

public TableColumn getTableColumn2() {
    return tableColumn2;
}

public void setTableColumn2(TableColumn tc) {
    this.tableColumn2 = tc;
}

```

```

private StaticText staticText2 = new StaticText();

public StaticText getStaticText2() {
    return staticText2;
}

public void setStaticText2(StaticText st) {
    this.staticText2 = st;
}

private TableColumn tableColumn3 = new TableColumn();

public TableColumn getTableColumn3() {
    return tableColumn3;
}

public void setTableColumn3(TableColumn tc) {
    this.tableColumn3 = tc;
}

private StaticText staticText3 = new StaticText();

public StaticText getStaticText3() {
    return staticText3;
}

public void setStaticText3(StaticText st) {
    this.staticText3 = st;
}

private TableColumn tableColumn4 = new TableColumn();

public TableColumn getTableColumn4() {
    return tableColumn4;
}

public void setTableColumn4(TableColumn tc) {
    this.tableColumn4 = tc;
}

private StaticText staticText4 = new StaticText();

public StaticText getStaticText4() {
    return staticText4;
}

public void setStaticText4(StaticText st) {
    this.staticText4 = st;
}

private TableColumn tableColumn5 = new TableColumn();

public TableColumn getTableColumn5() {
    return tableColumn5;
}

public void setTableColumn5(TableColumn tc) {
    this.tableColumn5 = tc;
}

```

```

    }

    private Checkbox chkConfigurar = new Checkbox();

    public Checkbox getChkConfigurar() {
        return chkConfigurar;
    }

    public void setChkConfigurar(Checkbox c) {
        this.chkConfigurar = c;
    }

    private Button btnConfigurar = new Button();

    public Button getBtnConfigurar() {
        return btnConfigurar;
    }

    public void setBtnConfigurar(Button b) {
        this.btnConfigurar = b;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

    public MessageGroup getMessageGroup1() {
        return messageGroup1;
    }

    public void setMessageGroup1(MessageGroup mg) {
        this.messageGroup1 = mg;
    }

    private StaticText lblEvaluado = new StaticText();

    public StaticText getLblEvaluado() {
        return lblEvaluado;
    }

    public void setLblEvaluado(StaticText st) {
        this.lblEvaluado = st;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>
     */
    public eaEvaluadores() {
    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property

```

```

    * values of the components do <strong>not</strong> represent any
    * values submitted with this request. Instead, they represent the
    * property values that were saved for this view when it was
    rendered.</p>
    */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
        // Initialize automatically managed components
        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {
            log("eaEvaluadores Initialization Failure", e);
            throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
        }

        // </editor-fold>
        // Perform application initialization that must complete
        // *after* managed components are initialized
        // TODO - add your own initialization code here
    }

    /**
     * <p>Callback method that is called after the component tree has been
     * restored, but before any event processing takes place. This method
     * will <strong>only</strong> be called on a postback request that
     * is processing a form submit. Customize this method to allocate
     * resources that will be required in your event handlers.</p>
     */
    public void preprocess() {
    }

    /**
     * <p>Callback method that is called just before rendering takes place.
     * This method will <strong>only</strong> be called for the page that
     * will actually be rendered (and not, for example, on a page that
     * handled a postback and then navigated to a different page).
     Customize
     * this method to allocate resources that will be required for rendering
     * this page.</p>
     */
    public void prerender() {
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        try{
            if (getSessionBean1().getIdUsuario() == null){
                externalContext.redirect("faces/login.jsp");
            } else{
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

```

```

        String strConnectionUrl = "jdbc:sqlserver://localhost:1433;"
+
        "databaseName=SEGD;user=sa;password=sql@fruit";
        String StrSQL;
        StrSQL = "select nombres + ' ' + apellidos as 'nombre' from
empleado where idEmpleado = '" +
getSessionBean1().getStrEmpleadoEvaluacion() + "'";

        Connection objConexion =
DriverManager.getConnection(strConnectionUrl);
        Statement stmt = objConexion.createStatement();
        ResultSet objResultSet = stmt.executeQuery(StrSQL);
        objResultSet.next();
        lblEvaluado.setText(objResultSet.getString("nombre"));

        objResultSet.close();
        stmt.close();

    }
} catch (Exception ex) {
    log("Error :", ex);
    error("Error :" + ex);
}
}

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,
 * <code>preprocess()</code>, or <code>prerender()</code> methods (or
 * acquired during execution of an event handler).</p>
 */
public void destroy() {
    evaluadorDataProvider.close();
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected SessionBean1 getSessionBean1() {
    return (SessionBean1)getBean("SessionBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected ApplicationBean1 getApplicationBean1() {
    return (ApplicationBean1)getBean("ApplicationBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected RequestBean1 getRequestBean1() {
    return (RequestBean1)getBean("RequestBean1");
}
}

```

```

    public String btnConfigurar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        String StrSQL;
        StrSQL = "";
        try {
            if (evaluadorDataProvider.getRowCount() > 0) {

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
                String strConnectionUrl = "jdbc:sqlserver://localhost:1433;"
+
                    "databaseName=SEGD;user=sa;password=sql@fruit;";

                Connection objConexion =
DriverManager.getConnection(strConnectionUrl);
                Statement stmt = objConexion.createStatement();
                Statement stmt2 = objConexion.createStatement();
                Statement stmt3 = objConexion.createStatement();
                StrSQL = "insert into evaluacion (fechaEvaluacion,
idCuestionario, idEmpleado, estado)"
                    + " values ('" +
getSessionBean1().getStrFechaEvaluacion()
                    + "',''" +
getSessionBean1().getStrCuestionarioEvaluacion()
                    + "',''" +
getSessionBean1().getStrEmpleadoEvaluacion() + "','a')";

                stmt.execute(StrSQL);

                Iterator rows =
getSessionBean1().getSelectedRows().iterator();
                while (rows.hasNext()) {
                    String rowId = (String) rows.next();
                    RowKey rowKey = evaluadorDataProvider.getRowKey(rowId);

//                    objResultSet.first();
                    StrSQL = "select idPregunta, idCuestionario,
idAreaEvaluacion, pregunta from pregunta "
                        + " where idCuestionario = '" +
getSessionBean1().getStrCuestionarioEvaluacion() + "'";
                    ResultSet objResultSet = stmt2.executeQuery(StrSQL);
                    while (objResultSet.next()) {
                        StrSQL = "insert into evaluacionDetalle
(fechaEvaluacion, idCuestionario, idEmpleado"
                            + ", idEvaluador, idPregunta)"
                            + " values ('" +
getSessionBean1().getStrFechaEvaluacion()
                            + "',''" +
getSessionBean1().getStrCuestionarioEvaluacion()
                            + "',''" +
getSessionBean1().getStrEmpleadoEvaluacion()
                            + "',''" +
evaluadorDataProvider.getValue("idEmpleado", rowKey)
                            + "',''" +
objResultSet.getString("idPregunta")
                            + "')";
                        stmt3.execute(StrSQL);

```

```

        }
        objResultSet.close();
    }

    stmt.close();
    stmt2.close();
    stmt3.close();

    try {
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        externalContext.redirect("faces/Page1.jsp?pmtMensaje=La
evaluación ha sido configurada satisfactoriamente.");
    } catch (IOException ioe) {
        log("Error :", ioe);
        error("Error :" + ioe);
    }
    } else {
        log(" Error : Por selecciones por lo menos a un
evaluador.");
        error(" Error : Por selecciones por lo menos a un
evaluador.");
    }
    } catch (SQLException e) {
        //System.out.println("SQL Exception: "+ e.toString());
        log(" Error :", e);
        error(" Error :" + e);
    } catch (ClassNotFoundException cE) {
        //System.out.println("Class Not Found Exception: "+
cE.toString());
        log(" Error :", cE);
        error(" Error :" + cE);
    }
    return null;
}
}

```

1.5.6 Contestar cuestionario

Esta parte del programa se encarga de proporcionar al evaluador un medio a través del cual pueda contestar cada una de las preguntas en el cuestionario de Evaluación Anual de Desempeño.

Nombre del programa: /EvaluacionAnual/eaContestar.java

Diagrama de actividad:

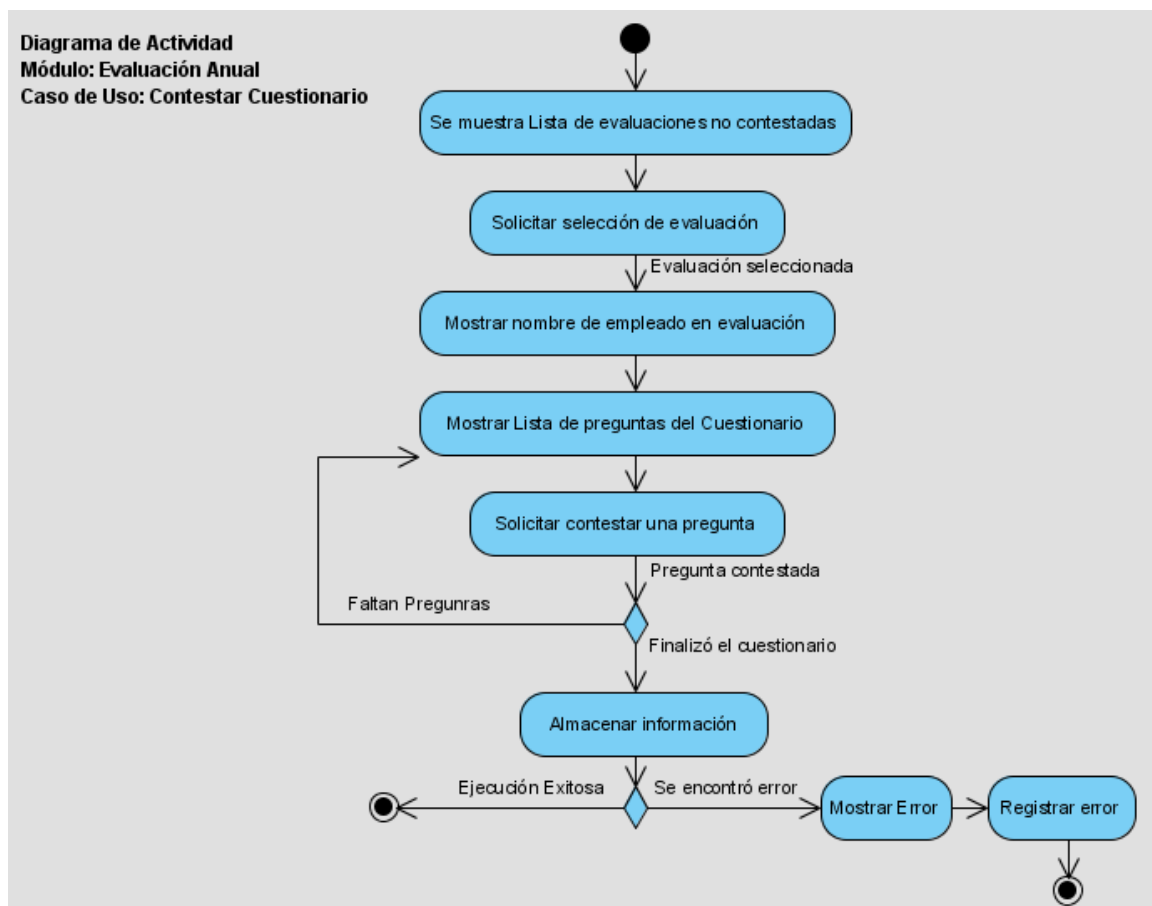


Diagrama de actividad para el caso de uso: Contestar Cuestionario.

Código fuente:

```
/*
 * eaContestar.java
 *
 * Created on January 12, 2008, 9:27 PM
 * Copyright sfons0
```



```

    */
package segd.EvaluacionAnual;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.sql.rowset.CachedRowSetXImpl;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.StaticText;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.model.DefaultTableDataProvider;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class eaContestar extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
    <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
    inserted
     * here is subject to being replaced.</p>
     */
    private void _init() throws Exception {

        evaluacionesPendientesDataProvider.setCachedRowSet((javax.sql.rowset.CachedR
        owSet)getValue("#{SessionBean1.evaluacionesPendientesRowSet}"));
    }

    private Page page1 = new Page();

    public Page getPage1() {
        return page1;
    }
}

```

```

public void setPage1(Page p) {
    this.page1 = p;
}

private Html html1 = new Html();

public Html getHtml1() {
    return html1;
}

public void setHtml1(Html h) {
    this.html1 = h;
}

private Head head1 = new Head();

public Head getHead1() {
    return head1;
}

public void setHead1(Head h) {
    this.head1 = h;
}

private Link link1 = new Link();

public Link getLink1() {
    return link1;
}

public void setLink1(Link l) {
    this.link1 = l;
}

private Body body1 = new Body();

public Body getBody1() {
    return body1;
}

public void setBody1(Body b) {
    this.body1 = b;
}

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private Label lblTituloPagina1 = new Label();

public Label getLblTituloPagina1() {
    return lblTituloPagina1;
}

```

```

    }

    public void setLblTituloPagina1(Label l) {
        this.lblTituloPagina1 = l;
    }

    private Label lblTituloPagina2 = new Label();

    public Label getLblTituloPagina2() {
        return lblTituloPagina2;
    }

    public void setLblTituloPagina2(Label l) {
        this.lblTituloPagina2 = l;
    }

    private Table table1 = new Table();

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

    public MessageGroup getMessageGroup1() {
        return messageGroup1;
    }

    public void setMessageGroup1(MessageGroup mg) {
        this.messageGroup1 = mg;
    }

    private CachedRowSetDataProvider evaluacionesPendientesDataProvider =
new CachedRowSetDataProvider();

    public CachedRowSetDataProvider getEvaluacionesPendientesDataProvider()
{
        return evaluacionesPendientesDataProvider;
    }

    public void
setEvaluacionesPendientesDataProvider(CachedRowSetDataProvider crsdp) {
        this.evaluacionesPendientesDataProvider = crsdp;
    }

```

```

private TableColumn tableColumn1 = new TableColumn();

public TableColumn getTableColumn1() {
    return tableColumn1;
}

public void setTableColumn1(TableColumn tc) {
    this.tableColumn1 = tc;
}

private StaticText staticText1 = new StaticText();

public StaticText getStaticText1() {
    return staticText1;
}

public void setStaticText1(StaticText st) {
    this.staticText1 = st;
}

private TableColumn tableColumn2 = new TableColumn();

public TableColumn getTableColumn2() {
    return tableColumn2;
}

public void setTableColumn2(TableColumn tc) {
    this.tableColumn2 = tc;
}

private StaticText staticText2 = new StaticText();

public StaticText getStaticText2() {
    return staticText2;
}

public void setStaticText2(StaticText st) {
    this.staticText2 = st;
}

private TableColumn tableColumn3 = new TableColumn();

public TableColumn getTableColumn3() {
    return tableColumn3;
}

public void setTableColumn3(TableColumn tc) {
    this.tableColumn3 = tc;
}

private StaticText staticText3 = new StaticText();

public StaticText getStaticText3() {
    return staticText3;
}

public void setStaticText3(StaticText st) {

```

```

        this.staticText3 = st;
    }

    private TableColumn tableColumn4 = new TableColumn();

    public TableColumn getTableColumn4() {
        return tableColumn4;
    }

    public void setTableColumn4(TableColumn tc) {
        this.tableColumn4 = tc;
    }

    private Button btnEvaluar = new Button();

    public Button getBtnEvaluar() {
        return btnEvaluar;
    }

    public void setBtnEvaluar(Button b) {
        this.btnEvaluar = b;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>
     */
    public eaContestar() {
    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property
     * values of the components do <strong>not</strong> represent any
     * values submitted with this request. Instead, they represent the
     * property values that were saved for this view when it was
     rendered.</p>
     */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
        // Initialize automatically managed components
        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {

```

```

        log("eaContestar Initialization Failure", e);
        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }

    // </editor-fold>
    // Perform application initialization that must complete
    // *after* managed components are initialized
    // TODO - add your own initialization code here
}

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {
}

/**
 * <p>Callback method that is called just before rendering takes place.
 * This method will <strong>only</strong> be called for the page that
 * will actually be rendered (and not, for example, on a page that
 * handled a postback and then navigated to a different page).
Customize
 * this method to allocate resources that will be required for rendering
 * this page.</p>
 */
public void prerender() {
    ExternalContext externalContext =
getFacesContext().getExternalContext();
    try{
        if (getSessionBean1().getIdUsuario() == null){
            externalContext.redirect("faces/login.jsp");
        } else {

getSessionBean1().getEvaluacionesPendientesRowSet().setObject(1,
getSessionBean1().getIdUsuario());
            evaluacionesPendientesDataProvider.refresh();
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :" + ex);
    }
}

/**
 * <p>Callback method that is called after rendering is completed for
 * this request, if <code>init()</code> was called (regardless of
whether
 * or not this was the page that was actually rendered). Customize this
 * method to release resources acquired in the <code>init()</code>,
 * <code>preprocess()</code>, or <code>prerender()</code> methods (or
 * acquired during execution of an event handler).</p>
 */
public void destroy() {

```

```

        evaluacionesPendientesDataProvider.close();
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1) getBean("ApplicationBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected RequestBean1 getRequestBean1() {
        return (RequestBean1) getBean("RequestBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1) getBean("SessionBean1");
    }

    public String btnEvaluar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        RowKey rk = tableRowGroup1.getRowKey();

        getSessionBean1().setStrEaCuestionario((String) evaluacionesPendientesDataPro
            vider.getValue("idCuestionario", rk));

        getSessionBean1().setStrEaEmpleado((String) evaluacionesPendientesDataProvide
            r.getValue("idEmpleado", rk));

        getSessionBean1().setStrEaFecha((String) evaluacionesPendientesDataProvider.g
            etValue("fecha", rk));

        return "case1";
    }
}

```

1.5.7 Agregar Departamento

Mantenimiento del catálogo de departamentos de la empresa, este catálogo también incluye a las áreas de los departamentos.

Nombre del programa: /Seguridad/sgDepartamento.java

Diagrama de actividad:

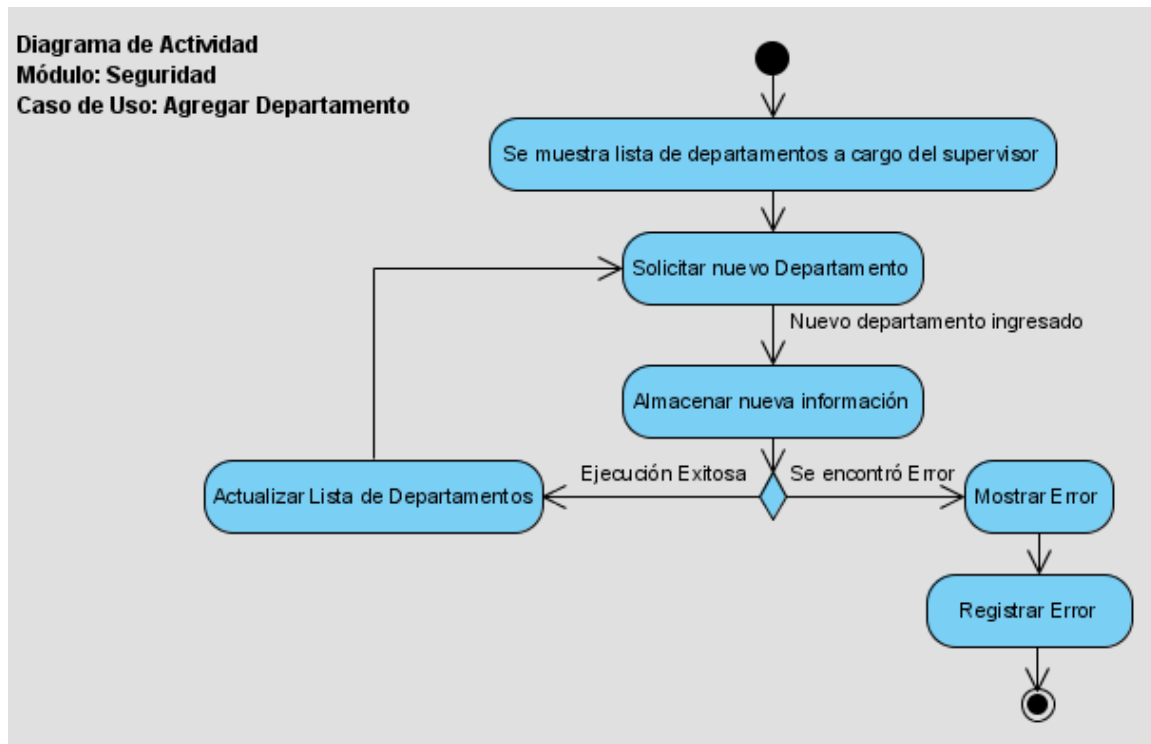


Diagrama de actividad para el caso de uso: Agregar Departamento.

Código Fuente:

```
/*
 * sgDepartamento.java
 *
 * Created on January 27, 2008, 7:02 PM
 * Copyright sfons0
 */
package segd.Seguridad;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
```



```

import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextField;
import java.io.IOException;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class sgDepartamento extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
    <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
    inserted
     * here is subject to being replaced.</p>
    */
    private void _init() throws Exception {

        departamentoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) getV
        alue("#{SessionBean1.departamentoRowSet}"));
    }

    private Page page1 = new Page();

    public Page getPage1() {
        return page1;
    }

    public void setPage1(Page p) {
        this.page1 = p;
    }

    private Html html1 = new Html();

    public Html getHtml1() {
        return html1;
    }

    public void setHtml1(Html h) {
        this.html1 = h;
    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

```

```

    }

    public void setHead1(Head h) {
        this.head1 = h;
    }

    private Link link1 = new Link();

    public Link getLink1() {
        return link1;
    }

    public void setLink1(Link l) {
        this.link1 = l;
    }

    private Body body1 = new Body();

    public Body getBody1() {
        return body1;
    }

    public void setBody1(Body b) {
        this.body1 = b;
    }

    private Form form1 = new Form();

    public Form getForm1() {
        return form1;
    }

    public void setForm1(Form f) {
        this.form1 = f;
    }

    private Label lblTituloPagina = new Label();

    public Label getLblTituloPagina() {
        return lblTituloPagina;
    }

    public void setLblTituloPagina(Label l) {
        this.lblTituloPagina = l;
    }

    private Label lblTituloPagina1 = new Label();

    public Label getLblTituloPagina1() {
        return lblTituloPagina1;
    }

    public void setLblTituloPagina1(Label l) {
        this.lblTituloPagina1 = l;
    }

    private Table table1 = new Table();

```

```

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

    private CachedRowSetDataProvider departamentoDataProvider = new
    CachedRowSetDataProvider();

    public CachedRowSetDataProvider getDepartamentoDataProvider() {
        return departamentoDataProvider;
    }

    public void setDepartamentoDataProvider(CachedRowSetDataProvider crsdp)
    {
        this.departamentoDataProvider = crsdp;
    }

    private TableColumn tableColumn2 = new TableColumn();

    public TableColumn getTableColumn2() {
        return tableColumn2;
    }

    public void setTableColumn2(TableColumn tc) {
        this.tableColumn2 = tc;
    }

    private TextField textField1 = new TextField();

    public TextField getTextField1() {
        return textField1;
    }

    public void setTextField1(TextField tf) {
        this.textField1 = tf;
    }

    private TableColumn tableColumn3 = new TableColumn();

    public TableColumn getTableColumn3() {
        return tableColumn3;
    }

    public void setTableColumn3(TableColumn tc) {
        this.tableColumn3 = tc;
    }

```

```

    }

    private Button btnBorrar = new Button();

    public Button getBtnBorrar() {
        return btnBorrar;
    }

    public void setBtnBorrar(Button b) {
        this.btnBorrar = b;
    }

    private Button btnDeshacer = new Button();

    public Button getBtnDeshacer() {
        return btnDeshacer;
    }

    public void setBtnDeshacer(Button b) {
        this.btnDeshacer = b;
    }

    private Button btnGuardar = new Button();

    public Button getBtnGuardar() {
        return btnGuardar;
    }

    public void setBtnGuardar(Button b) {
        this.btnGuardar = b;
    }

    private Button btnAgregar = new Button();

    public Button getBtnAgregar() {
        return btnAgregar;
    }

    public void setBtnAgregar(Button b) {
        this.btnAgregar = b;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

    public MessageGroup getMessageGroup1() {
        return messageGroup1;
    }

    public void setMessageGroup1(MessageGroup mg) {
        this.messageGroup1 = mg;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>
     */
    public sgDepartamento() {

```

```

    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property
     * values of the components do <strong>not</strong> represent any
     * values submitted with this request. Instead, they represent the
     * property values that were saved for this view when it was
     rendered.</p>
     */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
        Initialization">
        // Initialize automatically managed components
        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {
            log("sgDepartamento Initialization Failure", e);
            throw new FacesException ? (FacesException) e: new
FacesException(e);
        }

        // </editor-fold>
        // Perform application initialization that must complete
        // *after* managed components are initialized
        // TODO - add your own initialization code here
    }

    /**
     * <p>Callback method that is called after the component tree has been
     * restored, but before any event processing takes place. This method
     * will <strong>only</strong> be called on a postback request that
     * is processing a form submit. Customize this method to allocate
     * resources that will be required in your event handlers.</p>
     */
    public void preprocess() {
    }

    /**
     * <p>Callback method that is called just before rendering takes place.
     * This method will <strong>only</strong> be called for the page that
     * will actually be rendered (and not, for example, on a page that
     * handled a postback and then navigated to a different page).
     Customize
     * this method to allocate resources that will be required for rendering
     * this page.</p>

```

```

    */
    public void prerender() {
        ExternalContext externalContext =
getFacesContext().getExternalContext();

        if (getSessionBean1().getIdUsuario() == null){
            try {
                externalContext.redirect("faces/login.jsp");
            } catch(IOException ioe) {
                log("Error :", ioe);
                error("Error :" + ioe);
            }
        }
    }

    /**
     * <p>Callback method that is called after rendering is completed for
     * this request, if <code>init()</code> was called (regardless of
whether
     * or not this was the page that was actually rendered). Customize this
     * method to release resources acquired in the <code>init()</code>,
     * <code>preprocess()</code>, or <code>prerender()</code> methods (or
     * acquired during execution of an event handler).</p>
    */
    public void destroy() {
        departamentoDataProvider.close();
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
    */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1)getBean("ApplicationBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
    */
    protected RequestBean1 getRequestBean1() {
        return (RequestBean1)getBean("RequestBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
    */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1)getBean("SessionBean1");
    }

    public String btnDeshacer_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            departamentoDataProvider.refresh();

        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
    }

```

```

    }
    return null;
}

public String btnGuardar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        departamentoDataProvider.commitChanges();
    } catch (Exception ex) {
        log("Error :", ex);
        error("Error :"+ex.getMessage());
    }
    return null;
}

public String btnAgregar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = departamentoDataProvider.appendRow();
        departamentoDataProvider.setCursorRow(rk);

    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnBorrar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        if (rk != null) {
            departamentoDataProvider.removeRow(rk);
            departamentoDataProvider.commitChanges();
            departamentoDataProvider.refresh();
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}
}

```

1.5.8 Agregar Empleado

Mantenimiento del catálogo de empleados, permite, agregar nuevos empleados, modificar datos de los empleados existentes, y reestablecer contraseñas.

Nombre del Programa: /Seguridad/sgEmpleado.java

Diagrama de actividad:

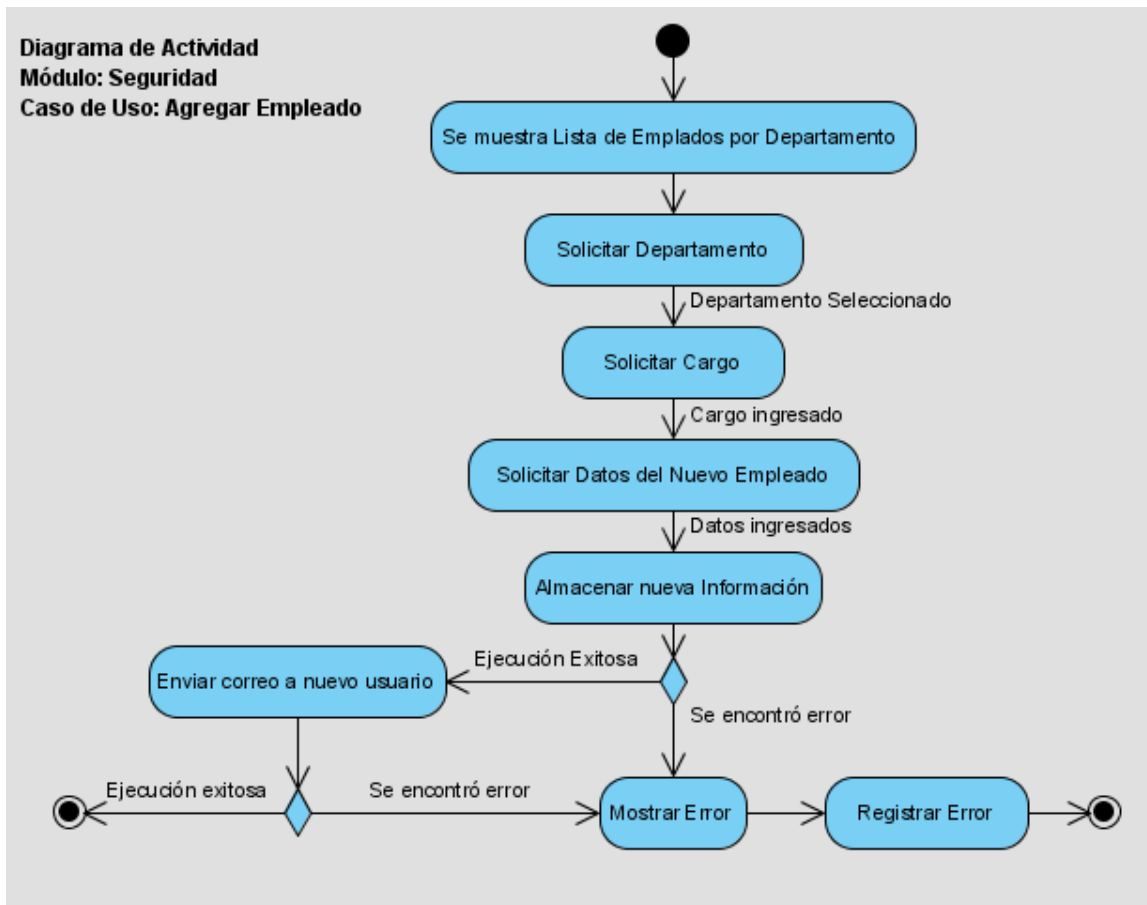


Diagrama de actividad para el caso de uso: Agregar Empleado.

Código Fuente:

```
/*
 * sgEmpleado.java
 *
 * Created on January 29, 2008, 8:42 PM
 * Copyright sfons0
 */
package segd.Seguridad;

import com.sun.data.provider.RowKey;
import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.DropDown;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
```



```

import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.Table;
import com.sun.webui.jsf.component.TableColumn;
import com.sun.webui.jsf.component.TableRowGroup;
import com.sun.webui.jsf.component.TextField;
import java.io.IOException;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;
import javax.faces.convert.IntegerConverter;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class sgEmpleado extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
        private int __placeholder;

        /**
         * <p>Automatically managed component initialization.
         <strong>WARNING:</strong>
         * This method is automatically generated, so any user-specified code
         inserted
         * here is subject to being replaced.</p>
         */
        private void _init() throws Exception {

empleadoManDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) getVa
lue("#{SessionBean1.empleadoManRowSet}"));

cargoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) getValue("#{
SessionBean1.cargoRowSet}"));

cargoDataProvider1.setCachedRowSet((javax.sql.rowset.CachedRowSet) getValue("#
#{SessionBean1.cargoRowSet}"));

empleadoestadoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) ge
tValue("#{SessionBean1.empleadoestadoRowSet}"));

empleadotipoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet) getV
alue("#{SessionBean1.empleadotipoRowSet}"));
        }

        private Page page1 = new Page();

        public Page getPage1() {
            return page1;
        }

        public void setPage1(Page p) {

```

```

        this.page1 = p;
    }

    private Html html1 = new Html();

    public Html getHtml1() {
        return html1;
    }

    public void setHtml1(Html h) {
        this.html1 = h;
    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

    public void setHead1(Head h) {
        this.head1 = h;
    }

    private Link link1 = new Link();

    public Link getLink1() {
        return link1;
    }

    public void setLink1(Link l) {
        this.link1 = l;
    }

    private Body body1 = new Body();

    public Body getBody1() {
        return body1;
    }

    public void setBody1(Body b) {
        this.body1 = b;
    }

    private Form form1 = new Form();

    public Form getForm1() {
        return form1;
    }

    public void setForm1(Form f) {
        this.form1 = f;
    }

    private CachedRowSetDataProvider empleadoManDataProvider = new
    CachedRowSetDataProvider();

    public CachedRowSetDataProvider getEmpleadoManDataProvider() {
        return empleadoManDataProvider;
    }

```

```

    }

    public void setEmpleadoManDataProvider(CachedRowSetDataProvider crsdp) {
        this.empleadoManDataProvider = crsdp;
    }

    private Table table1 = new Table();

    public Table getTable1() {
        return table1;
    }

    public void setTable1(Table t) {
        this.table1 = t;
    }

    private TableRowGroup tableRowGroup1 = new TableRowGroup();

    public TableRowGroup getTableRowGroup1() {
        return tableRowGroup1;
    }

    public void setTableRowGroup1(TableRowGroup trg) {
        this.tableRowGroup1 = trg;
    }

    private TableColumn tableColumn1 = new TableColumn();

    public TableColumn getTableColumn1() {
        return tableColumn1;
    }

    public void setTableColumn1(TableColumn tc) {
        this.tableColumn1 = tc;
    }

    private TableColumn tableColumn2 = new TableColumn();

    public TableColumn getTableColumn2() {
        return tableColumn2;
    }

    public void setTableColumn2(TableColumn tc) {
        this.tableColumn2 = tc;
    }

    private TableColumn tableColumn3 = new TableColumn();

    public TableColumn getTableColumn3() {
        return tableColumn3;
    }

    public void setTableColumn3(TableColumn tc) {
        this.tableColumn3 = tc;
    }

    private TableColumn tableColumn4 = new TableColumn();

```

```

public TableColumn getTableColumn4() {
    return tableColumn4;
}

public void setTableColumn4(TableColumn tc) {
    this.tableColumn4 = tc;
}

private TableColumn tableColumn5 = new TableColumn();

public TableColumn getTableColumn5() {
    return tableColumn5;
}

public void setTableColumn5(TableColumn tc) {
    this.tableColumn5 = tc;
}

private TableColumn tableColumn6 = new TableColumn();

public TableColumn getTableColumn6() {
    return tableColumn6;
}

public void setTableColumn6(TableColumn tc) {
    this.tableColumn6 = tc;
}

private TableColumn tableColumn7 = new TableColumn();

public TableColumn getTableColumn7() {
    return tableColumn7;
}

public void setTableColumn7(TableColumn tc) {
    this.tableColumn7 = tc;
}

private TableColumn tableColumn8 = new TableColumn();

public TableColumn getTableColumn8() {
    return tableColumn8;
}

public void setTableColumn8(TableColumn tc) {
    this.tableColumn8 = tc;
}

private TextField txtIdEmpleado = new TextField();

public TextField getTxtIdEmpleado() {
    return txtIdEmpleado;
}

public void setTxtIdEmpleado(TextField tf) {
    this.txtIdEmpleado = tf;
}

```

```

private DropDown ddlIdCargo = new DropDown();

public DropDown getDdlIdCargo() {
    return ddlIdCargo;
}

public void setDdlIdCargo(DropDown dd) {
    this.ddlIdCargo = dd;
}

private TextField txtNombres = new TextField();

public TextField getTxtNombres() {
    return txtNombres;
}

public void setTxtNombres(TextField tf) {
    this.txtNombres = tf;
}

private TextField txtApellidos = new TextField();

public TextField getTxtApellidos() {
    return txtApellidos;
}

public void setTxtApellidos(TextField tf) {
    this.txtApellidos = tf;
}

private TextField txtEmail = new TextField();

public TextField getTxtEmail() {
    return txtEmail;
}

public void setTxtEmail(TextField tf) {
    this.txtEmail = tf;
}

private DropDown ddleestado = new DropDown();

public DropDown getDdleestado() {
    return ddleestado;
}

public void setDdleestado(DropDown dd) {
    this.ddleestado = dd;
}

private DropDown ddlTipo = new DropDown();

public DropDown getDdlTipo() {
    return ddlTipo;
}

public void setDdlTipo(DropDown dd) {
    this.ddlTipo = dd;
}

```

```

    }

    private CachedRowSetDataProvider cargoDataProvider = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getCargoDataProvider() {
        return cargoDataProvider;
    }

    public void setCargoDataProvider(CachedRowSetDataProvider crsdp) {
        this.cargoDataProvider = crsdp;
    }

    private CachedRowSetDataProvider cargoDataProvider1 = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getCargoDataProvider1() {
        return cargoDataProvider1;
    }

    public void setCargoDataProvider1(CachedRowSetDataProvider crsdp) {
        this.cargoDataProvider1 = crsdp;
    }

    private IntegerConverter ddlIdCargoConverter = new IntegerConverter();

    public IntegerConverter getDdlIdCargoConverter() {
        return ddlIdCargoConverter;
    }

    public void setDdlIdCargoConverter(IntegerConverter ic) {
        this.ddlIdCargoConverter = ic;
    }

    private CachedRowSetDataProvider empleadoestadoDataProvider = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getEmpleadoestadoDataProvider() {
        return empleadoestadoDataProvider;
    }

    public void setEmpleadoestadoDataProvider(CachedRowSetDataProvider
crsdp) {
        this.empleadoestadoDataProvider = crsdp;
    }

    private CachedRowSetDataProvider empleadotipoDataProvider = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getEmpleadotipoDataProvider() {
        return empleadotipoDataProvider;
    }

    public void setEmpleadotipoDataProvider(CachedRowSetDataProvider crsdp)
{
        this.empleadotipoDataProvider = crsdp;
    }

```

```

private Button btnDeshacer = new Button();

public Button getBtnDeshacer() {
    return btnDeshacer;
}

public void setBtnDeshacer(Button b) {
    this.btnDeshacer = b;
}

private Button btnAgrega = new Button();

public Button getBtnAgrega() {
    return btnAgrega;
}

public void setBtnAgrega(Button b) {
    this.btnAgrega = b;
}

private Button btnGuardar = new Button();

public Button getBtnGuardar() {
    return btnGuardar;
}

public void setBtnGuardar(Button b) {
    this.btnGuardar = b;
}

private Label lblTituloPagina1 = new Label();

public Label getLblTituloPagina1() {
    return lblTituloPagina1;
}

public void setLblTituloPagina1(Label l) {
    this.lblTituloPagina1 = l;
}

private Label lblTituloPagina = new Label();

public Label getLblTituloPagina() {
    return lblTituloPagina;
}

public void setLblTituloPagina(Label l) {
    this.lblTituloPagina = l;
}

private MessageGroup messageGroup1 = new MessageGroup();

public MessageGroup getMessageGroup1() {
    return messageGroup1;
}

public void setMessageGroup1(MessageGroup mg) {
    this.messageGroup1 = mg;
}

```

```

    }

    private Button btnContrasenia = new Button();

    public Button getBtnContrasenia() {
        return btnContrasenia;
    }

    public void setBtnContrasenia(Button b) {
        this.btnContrasenia = b;
    }

    private TableColumn tableColumn9 = new TableColumn();

    public TableColumn getTableColumn9() {
        return tableColumn9;
    }

    public void setTableColumn9(TableColumn tc) {
        this.tableColumn9 = tc;
    }

    private Button btnBorrar = new Button();

    public Button getBtnBorrar() {
        return btnBorrar;
    }

    public void setBtnBorrar(Button b) {
        this.btnBorrar = b;
    }

    private TextField txtBuscarIdEmpleado = new TextField();

    public TextField getTxtBuscarIdEmpleado() {
        return txtBuscarIdEmpleado;
    }

    public void setTxtBuscarIdEmpleado(TextField tf) {
        this.txtBuscarIdEmpleado = tf;
    }

    private Button btnBuscar = new Button();

    public Button getBtnBuscar() {
        return btnBuscar;
    }

    public void setBtnBuscar(Button b) {
        this.btnBuscar = b;
    }

    private Button btnLimpiarBusqueda = new Button();

    public Button getBtnLimpiarBusqueda() {
        return btnLimpiarBusqueda;
    }
}

```



```

public void setBtnLimpiarBusqueda(Button b) {
    this.btnLimpiarBusqueda = b;
}

// </editor-fold>

/**
 * <p>Construct a new Page bean instance.</p>
 */
public sgEmpleado() {
}

/**
 * <p>Callback method that is called whenever a page is navigated to,
 * either directly via a URL, or indirectly via page navigation.
 * Customize this method to acquire resources that will be needed
 * for event handlers and lifecycle methods, whether or not this
 * page is performing post back processing.</p>
 *
 * <p>Note that, if the current request is a postback, the property
 * values of the components do <strong>not</strong> represent any
 * values submitted with this request. Instead, they represent the
 * property values that were saved for this view when it was
 rendered.</p>
 */
public void init() {
    // Perform initializations inherited from our superclass
    super.init();
    // Perform application initialization that must complete
    // *before* managed components are initialized
    // TODO - add your own initialization code here

    // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
    // Initialize automatically managed components
    // *Note* - this logic should NOT be modified
    try {
        _init();
    } catch (Exception e) {
        log("sgEmpleado Initialization Failure", e);
        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }

    // </editor-fold>
    // Perform application initialization that must complete
    // *after* managed components are initialized
    // TODO - add your own initialization code here
}

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {

```

```

    }

    /**
     * <p>Callback method that is called just before rendering takes place.
     * This method will <strong>only</strong> be called for the page that
     * will actually be rendered (and not, for example, on a page that
     * handled a postback and then navigated to a different page).
    Customize
     * this method to allocate resources that will be required for rendering
     * this page.</p>
     */
    public void prerender() {
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        try{
            if (getSessionBean1().getIdUsuario() == null){
                try {
                    externalContext.redirect("faces/login.jsp");
                } catch (IOException ioe) {
                    log("Error :", ioe);
                    error("Error :" + ioe);
                }
            } else {
                if (txtBuscarIdEmpleado.getText() == null){
                    getSessionBean1().getEmpleadoManRowSet().setObject(1,
1);

                    getSessionBean1().getEmpleadoManRowSet().setObject(2,
"1");

                    empleadoManDataProvider.refresh();
                }
            }
        } catch (Exception ex) {
            log("Error :", ex);
            error("Error :" + ex);
        }
    }

    /**
     * <p>Callback method that is called after rendering is completed for
     * this request, if <code>init()</code> was called (regardless of
whether
     * or not this was the page that was actually rendered). Customize this
     * method to release resources acquired in the <code>init()</code>,
     * <code>preprocess()</code>, or <code>prerender()</code> methods (or
     * acquired during execution of an event handler).</p>
     */
    public void destroy() {
        empleadoManDataProvider.close();
        cargoDataProvider.close();
        cargoDataProvider1.close();
        empleadoestadoDataProvider.close();
        empleadotipoDataProvider.close();
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected RequestBean1 getRequestBean1() {

```

```

        return (RequestBean1)getBean("RequestBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1)getBean("ApplicationBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1)getBean("SessionBean1");
    }

    public String btnDeshacer_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            empleadoManDataProvider.refresh();

        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnAgregar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            RowKey rk = empleadoManDataProvider.appendRow();
            empleadoManDataProvider.setCursorRow(rk);

            // cuestionarioDataProvider.setValue("TRIP.TRIPID", new
            Integer(0));
            // cuestionarioDataProvider.setValue("TRIP.PERSONID",
            personDD.getSelected());
            // cuestionarioDataProvider.setValue("TRIP.TRIPTYPEID", new
            Integer(1));
        } catch (Exception ex) {
            log("Error :", ex);
            error(ex.getMessage());
        }
        return null;
    }

    public String btnGuardar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try {
            if (empleadoManDataProvider.getRowCount() > 0) {
                empleadoManDataProvider.cursorFirst();
                do {

```

```

//          if
(empleadoManDataProvider.getValue("TRIP.TRIPID").equals
//          (new Integer(0))) {
//          empleadoManDataProvider.setValue("TRIP.TRIPID",
//          new Integer(newTripId));
//          newTripId++;
//          }

        } while (empleadoManDataProvider.cursorNext());
    };
    empleadoManDataProvider.commitChanges();
} catch (Exception ex) {
    log("Error :", ex);
    error("Error :"+ex.getMessage());
}
return null;
}

public String btnContrasenia_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    ExternalContext externalContext =
getFacesContext().getExternalContext();

    try {
        RowKey rk = tableRowGroup1.getRowKey();
//        getRequestBean1().setIdCuestionario();

        externalContext.redirect("sgContraseniaMan.jsp?pmtIdEmpleado=" +
(String)empleadoManDataProvider.getValue("idEmpleado",rk) + "&pmtEmpleado="
+ (String)empleadoManDataProvider.getValue("nombres",rk) + " " +
(String)empleadoManDataProvider.getValue("apellidos",rk));
    } catch (IOException ioe) {
        log("Error :", ioe);
        error("Error :"+ioe);
    }
    return null;
}

public String btnBorrar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        if (rk != null) {
            empleadoManDataProvider.removeRow(rk);
            empleadoManDataProvider.commitChanges();
            empleadoManDataProvider.refresh();
        }
    } catch (Exception ex) {
        log("Error :", ex);
        error(ex.getMessage());
    }
    return null;
}

public String btnBuscar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.

```

```

        try {
            getSessionBean1().getEmpleadoManRowSet().setObject(1, 0);
            getSessionBean1().getEmpleadoManRowSet().setObject(2,
txtBuscarIdEmpleado.getText().toString());
            empleadoManDataProvider.refresh();

//            error(txtBuscarIdEmpleado.getText().toString());
        } catch (Exception ex) {
            log("Error :", ex);
            error("Error :" + ex);
        }

//empleadoManDataProvider.setCursorRow(empleadoManDataProvider.findFirst("id
Empleado", txtBuscarIdEmpleado.getText()));

//        if (empleadoManDataProvider.getRowCount() > 0) {
//            empleadoManDataProvider.cursorFirst();
//            do {
//                if (empleadoManDataProvider.getValue("TRIP.TRIPID").equals
//                    (new Integer(0))) {
//                    empleadoManDataProvider.setValue("TRIP.TRIPID",
//                        new Integer(newTripId));
//                    newTripId++;
//                }
//            } while (empleadoManDataProvider.cursorNext());
//        };

        return null;
    }

    public String btnLimpiarBusqueda_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        try{
            getSessionBean1().getEmpleadoManRowSet().setObject(1, 1);
            getSessionBean1().getEmpleadoManRowSet().setObject(2, "1");
            empleadoManDataProvider.refresh();
        } catch (Exception ex) {
            log("Error :", ex);
            error("Error :" + ex);
        }
        return null;
    }
}

```

1.5.9 Ingreso al sistema

Este programa controla el acceso al sistema, validando si el usuario tiene derechos para ingresar, a través de un nombre de usuario y una contraseña debidamente encriptada en la Base de Datos.

Nombre del programa: /login.java

Diagrama de actividad:

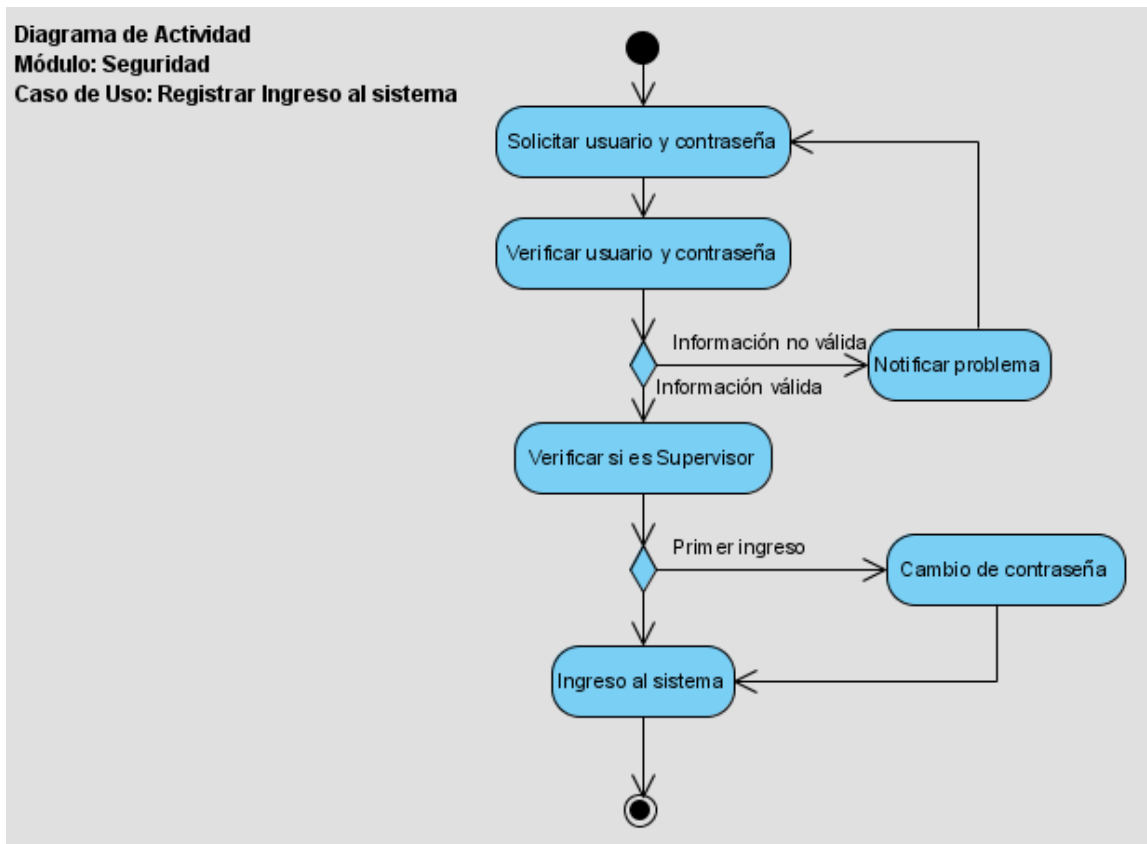


Diagrama de actividad para el caso de uso: Registrar Ingreso.

Código Fuente:

```
/*
 * login.java
 *
 * Created on November 19, 2007, 9:44 PM
 * Copyright sfons0
 */
package segd;

import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Hyperlink;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.PasswordField;
```

```

import com.sun.webui.jsf.component.TextField;
import javax.faces.FacesException;
import javax.faces.context.ExternalContext;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class login extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
    <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
    inserted
     * here is subject to being replaced.</p>
     */
    private void _init() throws Exception {

        empleadoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue
        ("#{SessionBean1.empleadoRowSet}"));
    }

    private Page page1 = new Page();

    public Page getPage1() {
        return page1;
    }

    public void setPage1(Page p) {
        this.page1 = p;
    }

    private Html html1 = new Html();

    public Html getHtml1() {
        return html1;
    }

    public void setHtml1(Html h) {
        this.html1 = h;
    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

    public void setHead1(Head h) {
        this.head1 = h;
    }
}

```

```

private Link link1 = new Link();

public Link getLink1() {
    return link1;
}

public void setLink1(Link l) {
    this.link1 = l;
}

private Body body1 = new Body();

public Body getBody1() {
    return body1;
}

public void setBody1(Body b) {
    this.body1 = b;
}

private Form form1 = new Form();

public Form getForm1() {
    return form1;
}

public void setForm1(Form f) {
    this.form1 = f;
}

private Label lblUsuario = new Label();

public Label getLblUsuario() {
    return lblUsuario;
}

public void setLblUsuario(Label l) {
    this.lblUsuario = l;
}

private Label lblContrasenia = new Label();

public Label getLblContrasenia() {
    return lblContrasenia;
}

public void setLblContrasenia(Label l) {
    this.lblContrasenia = l;
}

private TextField txtIdEmpleado = new TextField();

public TextField getTxtIdEmpleado() {
    return txtIdEmpleado;
}

public void setTxtIdEmpleado(TextField tf) {

```



```

        this.txtIdEmpleado = tf;
    }

    private PasswordField txtContrasenia = new PasswordField();

    public PasswordField getTxtContrasenia() {
        return txtContrasenia;
    }

    public void setTxtContrasenia(PasswordField pf) {
        this.txtContrasenia = pf;
    }

    private Button btnAceptar = new Button();

    public Button getBtnAceptar() {
        return btnAceptar;
    }

    public void setBtnAceptar(Button b) {
        this.btnAceptar = b;
    }

    private Button button2 = new Button();

    public Button getButton2() {
        return button2;
    }

    public void setButton2(Button b) {
        this.button2 = b;
    }

    private CachedRowSetDataProvider empleadoDataProvider = new
CachedRowSetDataProvider();

    public CachedRowSetDataProvider getEmpleadoDataProvider() {
        return empleadoDataProvider;
    }

    public void setEmpleadoDataProvider(CachedRowSetDataProvider crsdp) {
        this.empleadoDataProvider = crsdp;
    }

    private MessageGroup messageGroup1 = new MessageGroup();

    public MessageGroup getMessageGroup1() {
        return messageGroup1;
    }

    public void setMessageGroup1(MessageGroup mg) {
        this.messageGroup1 = mg;
    }

    // </editor-fold>

    /**
     * <p>Construct a new Page bean instance.</p>

```

```

    */
    public login() {
    }

    /**
     * <p>Callback method that is called whenever a page is navigated to,
     * either directly via a URL, or indirectly via page navigation.
     * Customize this method to acquire resources that will be needed
     * for event handlers and lifecycle methods, whether or not this
     * page is performing post back processing.</p>
     *
     * <p>Note that, if the current request is a postback, the property
     * values of the components do <strong>not</strong> represent any
     * values submitted with this request. Instead, they represent the
     * property values that were saved for this view when it was
     rendered.</p>
     */
    public void init() {
        // Perform initializations inherited from our superclass
        super.init();
        // Perform application initialization that must complete
        // *before* managed components are initialized
        // TODO - add your own initialization code here

        // <editor-fold defaultstate="collapsed" desc="Managed Component
        Initialization">
        // Initialize automatically managed components
        // *Note* - this logic should NOT be modified
        try {
            _init();
        } catch (Exception e) {
            log("login Initialization Failure", e);
            throw e instanceof FacesException ? (FacesException) e: new
        FacesException(e);
        }

        // </editor-fold>
        // Perform application initialization that must complete
        // *after* managed components are initialized
        // TODO - add your own initialization code here
    }

    /**
     * <p>Callback method that is called after the component tree has been
     * restored, but before any event processing takes place. This method
     * will <strong>only</strong> be called on a postback request that
     * is processing a form submit. Customize this method to allocate
     * resources that will be required in your event handlers.</p>
     */
    public void preprocess() {
    }

    /**
     * <p>Callback method that is called just before rendering takes place.
     * This method will <strong>only</strong> be called for the page that
     * will actually be rendered (and not, for example, on a page that
     * handled a postback and then navigated to a different page).
     Customize

```

```

    * this method to allocate resources that will be required for rendering
    * this page.</p>
    */
    public void prerender() {
        pfEncabezadoGeneral pfEncabezadoGeneralFragmentBean =
(pfEncabezadoGeneral) getBean("pfEncabezadoGeneral");
        Hyperlink homeLink = pfEncabezadoGeneralFragmentBean.getLnkMenu();
        Hyperlink salirLink = pfEncabezadoGeneralFragmentBean.getLnkSalir();
        Hyperlink contraseniaLink =
pfEncabezadoGeneralFragmentBean.getLnkContrasenia();

        homeLink.setDisabled(true);
        homeLink.setVisible(false);

        salirLink.setDisabled(true);
        salirLink.setVisible(false);

        contraseniaLink.setDisabled(true);
        contraseniaLink.setVisible(false);

        getSessionBean1().setIdUsuario(null);

pfEncabezadoGeneralFragmentBean.getLblUsuarioContectado().setVisible(false);
    }

    /**
    * <p>Callback method that is called after rendering is completed for
    * this request, if <code>init()</code> was called (regardless of
whether
    * or not this was the page that was actually rendered). Customize this
    * method to release resources acquired in the <code>init()</code>,
    * <code>preprocess()</code>, or <code>prerender()</code> methods (or
    * acquired during execution of an event handler).</p>
    */
    public void destroy() {
        empleadoDataProvider.close();
    }

    /**
    * <p>Return a reference to the scoped data bean.</p>
    */
    protected RequestBean1 getRequestBean1() {
        return (RequestBean1) getBean("RequestBean1");
    }

    /**
    * <p>Return a reference to the scoped data bean.</p>
    */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1) getBean("ApplicationBean1");
    }

    /**
    * <p>Return a reference to the scoped data bean.</p>
    */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1) getBean("SessionBean1");
    }
}

```

```

    public String btnAceptar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        ExternalContext externalContext =
getFacesContext().getExternalContext();
        try {
            getSessionBean1().getEmpleadoRowSet().setObject(1,
txtIdEmpleado.getText());
            empleadoDataProvider.refresh();
            empleadoDataProvider.cursorFirst();
            if
(empleadoDataProvider.getValue("contrasenia").equals(txtContrasenia.getText(
))) {
                getSessionBean1().setIdUsuario((String)
txtIdEmpleado.getText());
                getSessionBean1().setIdCargo((String)
empleadoDataProvider.getValue("idCargo").toString());
                getSessionBean1().setTipo((String)
empleadoDataProvider.getValue("tipo"));
                getSessionBean1().setIntDepartamento((Integer)
empleadoDataProvider.getValue("idDepartamento"));
                getSessionBean1().setStrUsuarioConectado((String)
txtIdEmpleado.getText());

error(empleadoDataProvider.getValue("primerLogin").toString());
                if
(empleadoDataProvider.getValue("primerLogin").toString().equals("true")) {
externalContext.redirect("faces/Seguridad/sgContrasenia.jsp?pmtMensaje=Por
favor cambie su contraseña");
                }
                else {
                    externalContext.redirect("faces/Page1.jsp");
                }

                } else {
                    error("Usuario y/o Contraseña no validos.");
                }
            } catch (Exception ex) {
                log("Error :", ex);
                error("Error :" + ex);
            }
            return null;
        }
    }
}

```

1.5.10 Cambio de contraseña

Este programa le permite a un usuario del sistema cambiar su contraseña, o bien a un jefe reestablecer la contraseña de uno de sus empleados.

Nombre del programa: /Seguridad/sgContrasenia.java

Diagrama de actividad:

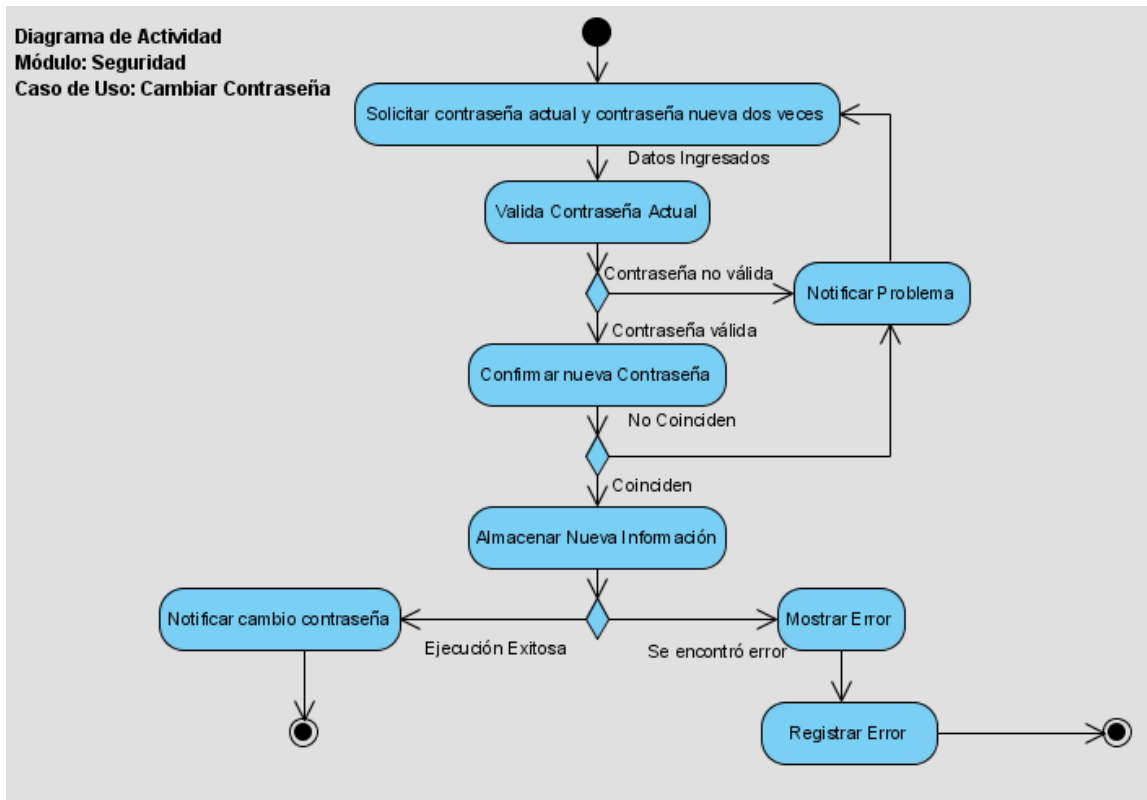


Diagrama de actividad para el caso de uso: Cambiar Contraseña.

Código Fuente:

```
/*
 * sgContrasenia.java
 *
 * Created on January 27, 2008, 10:39 PM
 * Copyright sfons0
 */
package segd.Seguridad;

import com.sun.data.provider.impl.CachedRowSetDataProvider;
import com.sun.rave.web.ui.appbase.AbstractPageBean;
import com.sun.webui.jsf.component.Body;
import com.sun.webui.jsf.component.Button;
import com.sun.webui.jsf.component.Form;
import com.sun.webui.jsf.component.Head;
import com.sun.webui.jsf.component.Html;
import com.sun.webui.jsf.component.Label;
import com.sun.webui.jsf.component.Link;
import com.sun.webui.jsf.component.MessageGroup;
import com.sun.webui.jsf.component.Page;
import com.sun.webui.jsf.component.PasswordField;
```

```

import com.sun.webui.jsf.component.StaticText;
import java.io.IOException;
import javax.faces.FacesException;
import segd.ApplicationBean1;
import segd.RequestBean1;
import segd.SessionBean1;
import java.sql.*;
import javax.faces.context.ExternalContext;

/**
 * <p>Page bean that corresponds to a similarly named JSP page. This
 * class contains component definitions (and initialization code) for
 * all components that you have defined on this page, as well as
 * lifecycle methods and event handlers where you may add behavior
 * to respond to incoming events.</p>
 */
public class sgContrasenia extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Managed Component
    Definition">
    private int __placeholder;

    /**
     * <p>Automatically managed component initialization.
    <strong>WARNING:</strong>
     * This method is automatically generated, so any user-specified code
    inserted
     * here is subject to being replaced.</p>
     */
    private void _init() throws Exception {

        empleadoDataProvider.setCachedRowSet((javax.sql.rowset.CachedRowSet)getValue
        ("#{SessionBean1.empleadoRowSet}"));
    }

    private Page page1 = new Page();

    public Page getPage1() {
        return page1;
    }

    public void setPage1(Page p) {
        this.page1 = p;
    }

    private Html html1 = new Html();

    public Html getHtml1() {
        return html1;
    }

    public void setHtml1(Html h) {
        this.html1 = h;
    }

    private Head head1 = new Head();

    public Head getHead1() {
        return head1;
    }

```

```

    }

    public void setHead1(Head h) {
        this.head1 = h;
    }

    private Link link1 = new Link();

    public Link getLink1() {
        return link1;
    }

    public void setLink1(Link l) {
        this.link1 = l;
    }

    private Body body1 = new Body();

    public Body getBody1() {
        return body1;
    }

    public void setBody1(Body b) {
        this.body1 = b;
    }

    private Form form1 = new Form();

    public Form getForm1() {
        return form1;
    }

    public void setForm1(Form f) {
        this.form1 = f;
    }

    private Label lblTituloPagina = new Label();

    public Label getLblTituloPagina() {
        return lblTituloPagina;
    }

    public void setLblTituloPagina(Label l) {
        this.lblTituloPagina = l;
    }

    private Button btnAceptar = new Button();

    public Button getBtnAceptar() {
        return btnAceptar;
    }

    public void setBtnAceptar(Button b) {
        this.btnAceptar = b;
    }

    private Button btnCancelar = new Button();

```

```

public Button getBtnCancelar() {
    return btnCancelar;
}

public void setBtnCancelar(Button b) {
    this.btnCancelar = b;
}

private PasswordField pwfContraseniaActual = new PasswordField();

public PasswordField getPwfContraseniaActual() {
    return pwfContraseniaActual;
}

public void setPwfContraseniaActual(PasswordField pf) {
    this.pwfContraseniaActual = pf;
}

private PasswordField pwfContrasenia1 = new PasswordField();

public PasswordField getPwfContrasenia1() {
    return pwfContrasenia1;
}

public void setPwfContrasenia1(PasswordField pf) {
    this.pwfContrasenia1 = pf;
}

private PasswordField pwfContrasenia2 = new PasswordField();

public PasswordField getPwfContrasenia2() {
    return pwfContrasenia2;
}

public void setPwfContrasenia2(PasswordField pf) {
    this.pwfContrasenia2 = pf;
}

private CachedRowSetDataProvider empleadoDataProvider = new
CachedRowSetDataProvider();

public CachedRowSetDataProvider getEmpleadoDataProvider() {
    return empleadoDataProvider;
}

public void setEmpleadoDataProvider(CachedRowSetDataProvider crsdp) {
    this.empleadoDataProvider = crsdp;
}

private MessageGroup messageGroup1 = new MessageGroup();

public MessageGroup getMessageGroup1() {
    return messageGroup1;
}

public void setMessageGroup1(MessageGroup mg) {
    this.messageGroup1 = mg;
}

```



```

private StaticText lblMensaje = new StaticText();

public StaticText getLblMensaje() {
    return lblMensaje;
}

public void setLblMensaje(StaticText st) {
    this.lblMensaje = st;
}

// </editor-fold>

/**
 * <p>Construct a new Page bean instance.</p>
 */
public sgContrasenia() {
}

/**
 * <p>Callback method that is called whenever a page is navigated to,
 * either directly via a URL, or indirectly via page navigation.
 * Customize this method to acquire resources that will be needed
 * for event handlers and lifecycle methods, whether or not this
 * page is performing post back processing.</p>
 *
 * <p>Note that, if the current request is a postback, the property
 * values of the components do <strong>not</strong> represent any
 * values submitted with this request. Instead, they represent the
 * property values that were saved for this view when it was
 rendered.</p>
 */
public void init() {
    // Perform initializations inherited from our superclass
    super.init();
    // Perform application initialization that must complete
    // *before* managed components are initialized
    // TODO - add your own initialization code here

    // <editor-fold defaultstate="collapsed" desc="Managed Component
Initialization">
    // Initialize automatically managed components
    // *Note* - this logic should NOT be modified
    try {
        _init();
    } catch (Exception e) {
        log("sgContrasenia Initialization Failure", e);
        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }

    // </editor-fold>
    // Perform application initialization that must complete
    // *after* managed components are initialized
    // TODO - add your own initialization code here
}

/**

```

```

    * <p>Callback method that is called after the component tree has been
    * restored, but before any event processing takes place. This method
    * will <strong>only</strong> be called on a postback request that
    * is processing a form submit. Customize this method to allocate
    * resources that will be required in your event handlers.</p>
    */
    public void preprocess() {
    }

    /**
    * <p>Callback method that is called just before rendering takes place.
    * This method will <strong>only</strong> be called for the page that
    * will actually be rendered (and not, for example, on a page that
    * handled a postback and then navigated to a different page).
    Customize
    * this method to allocate resources that will be required for rendering
    * this page.</p>
    */
    public void prerender() {
        ExternalContext externalContext =
        getFacesContext().getExternalContext();

        if (getSessionBean1().getIdUsuario() == null){
            try {
                externalContext.redirect("faces/login.jsp");
            } catch(IOException ioe) {
                log("Error :", ioe);
                error("Error :" + ioe);
            }
        }
        else {
            String pmtMensaje = (String)
            getExternalContext().getRequestParameterMap().get("pmtMensaje");
            if (pmtMensaje != null){
                getLblMensaje().setText(pmtMensaje);
                btnCancelar.setDisabled(true);
            }

            String strIdEmpleado = (String)
            getExternalContext().getRequestParameterMap().get("pmtIdEmpleado");
        }
    }

    /**
    * <p>Callback method that is called after rendering is completed for
    * this request, if <code>init()</code> was called (regardless of
    whether
    * or not this was the page that was actually rendered). Customize this
    * method to release resources acquired in the <code>init()</code>,
    * <code>preprocess()</code>, or <code>prerender()</code> methods (or
    * acquired during execution of an event handler).</p>
    */
    public void destroy() {
        empleadoDataProvider.close();
    }

    /**
    * <p>Return a reference to the scoped data bean.</p>

```

```

    */
    protected SessionBean1 getSessionBean1() {
        return (SessionBean1) getBean("SessionBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected RequestBean1 getRequestBean1() {
        return (RequestBean1) getBean("RequestBean1");
    }

    /**
     * <p>Return a reference to the scoped data bean.</p>
     */
    protected ApplicationBean1 getApplicationBean1() {
        return (ApplicationBean1) getBean("ApplicationBean1");
    }

    public String btnAceptar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        String strSQL;
        strSQL = "";
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            String strConnectionUrl = "jdbc:sqlserver://localhost:1433;" +
                "databaseName=SEGD;user=sa;password=sql@fruit;";

            Connection objConexion =
                DriverManager.getConnection(strConnectionUrl);
            Statement stmt = objConexion.createStatement();

            if (strIdEmpleado != null){
                getSessionBean1().getEmpleadoRowSet().setObject(1,
strIdEmpleado);
            }
            else {
                getSessionBean1().getEmpleadoRowSet().setObject(1,
getSessionBean1().getIdUsuario());
            }
            //      empleadoDataProvider.refresh();
            //      empleadoDataProvider.cursorFirst();
            if
(empleadoDataProvider.getValue("contrasenia").equals(pwfContraseniaActual.ge
tText())){
                if
(pwfContrasenia1.getText().toString().compareTo(pwfContrasenia2.getText().to
String()) == 0){
                    if (strIdEmpleado != null){
                        strSQL = "update empleado set primerLogin = 0,
contrasenia = EncryptByPassPhrase('SEGD', '"' +
pwfContrasenia1.getText().toString() + "') where idEmpleado = '" +
String.valueOf(strIdEmpleado) + "'";
                    }
                    else {

```

```

        StrSQL = "update empleado set primerLogin = 0,
contrasenia = EncryptByPassPhrase('SEGD', '"' +
pwfContrasenia1.getText().toString() + "') where idEmpleado = '" +
String.valueOf(getSessionBean1().getIdUsuario()) + "'";
    }

    stmt.execute(StrSQL);

    stmt.close();

    try {
        ExternalContext externalContext =
getFacesContext().getExternalContext();

externalContext.redirect("../faces/Pagel.jsp?pmtMensaje=La Contraseña ha
sido cambiada satisfactoriamente.");
    } catch (IOException ioe) {
        log("Error :", ioe);
        error("Error :" + ioe);
    }
    } else {
        log("Error : La nueva contraseña y su confirmación no
son iguales.");
        error("Error : La nueva contraseña y su confirmación no
son iguales.");
    }
    } else {
        log("Error : Contraseña actual no valida.");
        error("Error : Contraseña actual no valida.");
    }
}

} catch (SQLException e) {
    //System.out.println("SQL Exception: "+ e.toString());
    log(" Error :", e);
    error(" Error :" + e);
} catch (ClassNotFoundException cE) {
    //System.out.println("Class Not Found Exception: "+
cE.toString());
    log(" Error :", cE);
    error(" Error :" + cE);
}

return null;
}

public String btnCancelar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.

    return "case1";
}

/**
 * Holds value of property strIdEmpleado.
 */
private String strIdEmpleado;

```

```
/**
 * Getter for property strIdEmpleado.
 * @return Value of property strIdEmpleado.
 */
public String getStrIdEmpleado() {
    return this.strIdEmpleado;
}

/**
 * Setter for property strIdEmpleado.
 * @param strIdEmpleado New value of property strIdEmpleado.
 */
public void setStrIdEmpleado(String strIdEmpleado) {
    this.strIdEmpleado = strIdEmpleado;
}
}
```

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
INGENIERIA EN COMPUTACION



**“GUIA PARA LA IMPLEMENTACION DE LAS MEJORES
PRACTICAS EN LA DIRECCION DE PROYECTOS APLICADO EN
UN PROYECTO DE DESARROLLO DE SOFTWARE PARA EL
INSTITUTO TECNOLOGICO CENTROAMERICANO”**

Apéndice B

Trabajo de graduación para optar al título de:
Ingeniero en Ciencias de la Computación

Presentado por:

Guillermo Arturo Cerrato Gámez

Salvador Fonseca Rivas

Asesor:

Ing. Carlos López Linares, MSc, MBA

Soyapango, 25 Marzo del 2008

Índice

APÉNDICE B – MANUAL DE INSTALACIÓN	3
1.1 PRERREQUISITOS	3
1.1.1 Hardware	3
1.1.2 Software	3
1.2 INSTALACIÓN DE JDK – JAVA (TM) SE DEVELOPMENT KIT.	3
1.3 INSTALACIÓN DE SUN JAVA APPLICATION SERVER	7
1.4 INSTALACIÓN DEL SISTEMA DE EVALUACIÓN GENERAL DE DESEMPEÑO (SEGD).....	16

Apéndice B – Manual de Instalación

Este manual de instalación es una guía paso a paso que ilustra la forma correcta de preparar un servidor para que El Sistema de Evaluación General de Desempeño pueda ser utilizado en el.

1.1 Prerrequisitos

1.1.1 Hardware

- El servidor debe tener por lo menos 4GB de memoria RAM.
- Un mínimo de 10 GB libres de memoria en Disco Duro.
- Microprocesador Dual Core Intel Xeon 5110 o superior.
- Dual Embedded Broadcom® NetXtreme II 5708 Gigabit Ethernet NIC o superior.

1.1.2 Software

- JDK – Java (TM) SE Development Kit 6 Update 3 o superior.
- Sun Java Application Server 9.1 o superior.

1.2 Instalación de JDK – Java (TM) SE Development Kit.

El JDK contiene el FrameWork de Java y las herramientas necesarias para que el contenedor de servlets (Sun Java Application Server) pueda administrar una aplicación web desarrollada en Java.

Para instalar el JDK en nuestro servidor primero descargamos los programas necesarios desde el sitio: <http://java.sun.com/>

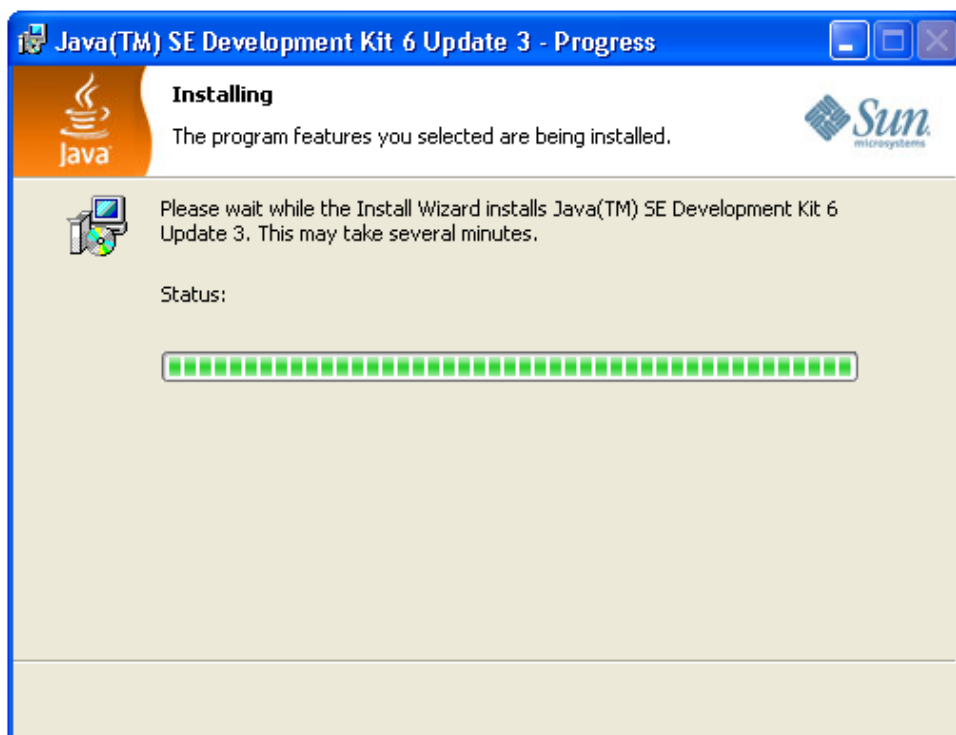
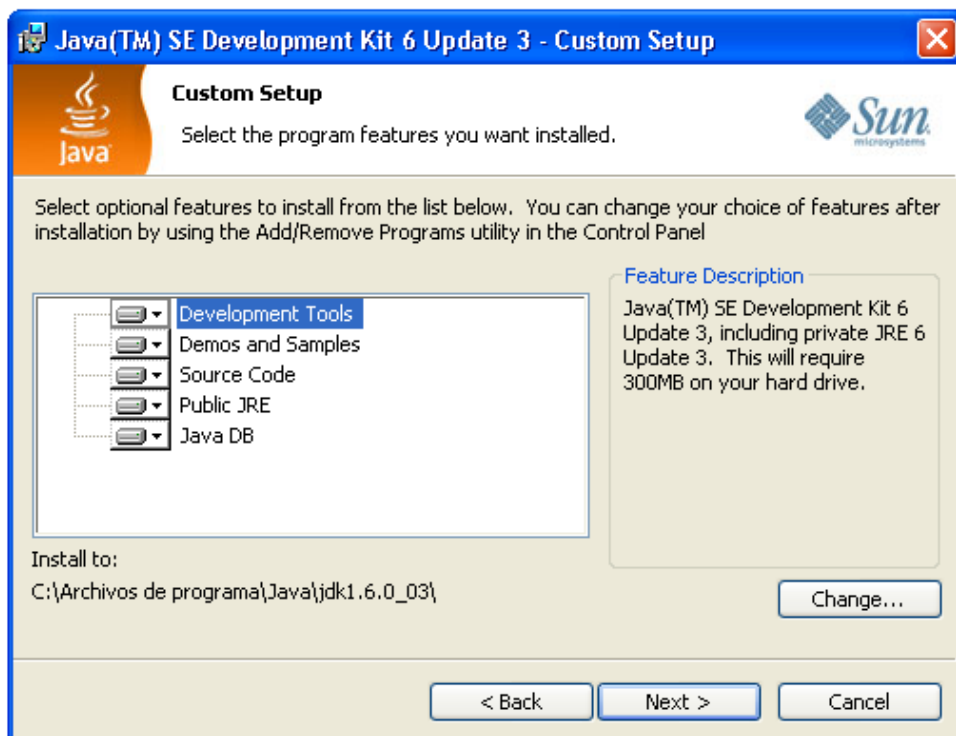
Una vez descargado el programa, procedemos a ejecutar el archivo.



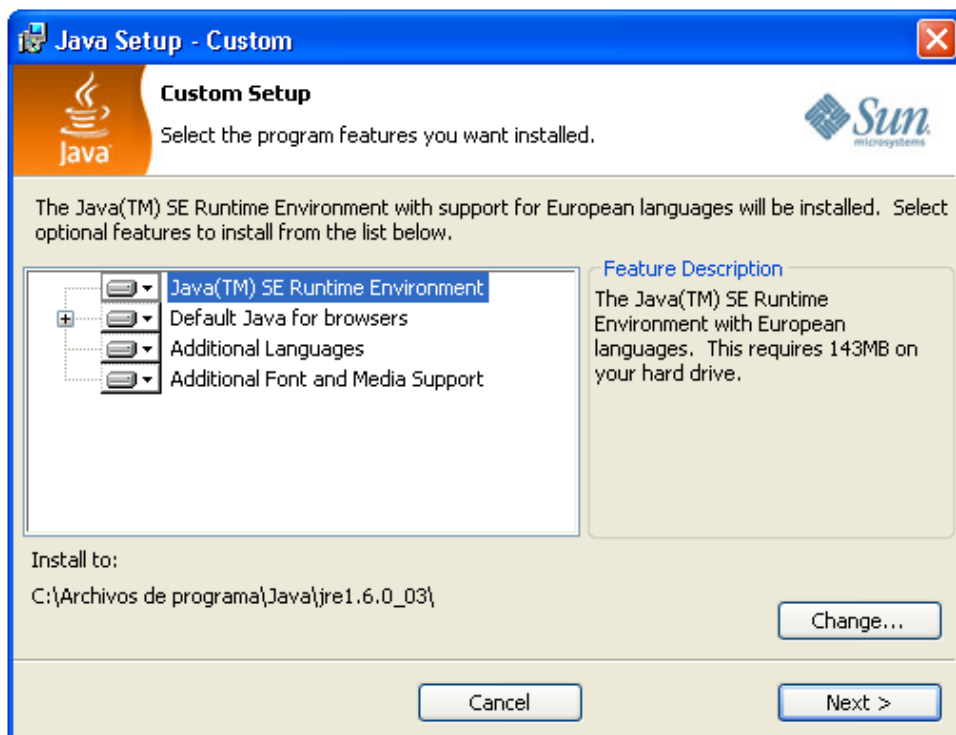
En la siguiente pantalla nos muestra los términos de la licencia que protege la aplicación del JDK. Procedemos a dar clic en el botón "Accept".



La siguiente pantalla nos permite personalizar los elementos opcionales del JDK que se instalarán en nuestro servidor, dejaremos esta pantalla con las opciones predeterminadas. Damos un clic en el botón “Next”.



Esta pantalla nos permite seleccionar las características que tendrá nuestra instalación del JDK como: el SE Runtime Environment o soporte para lenguajes adicionales.

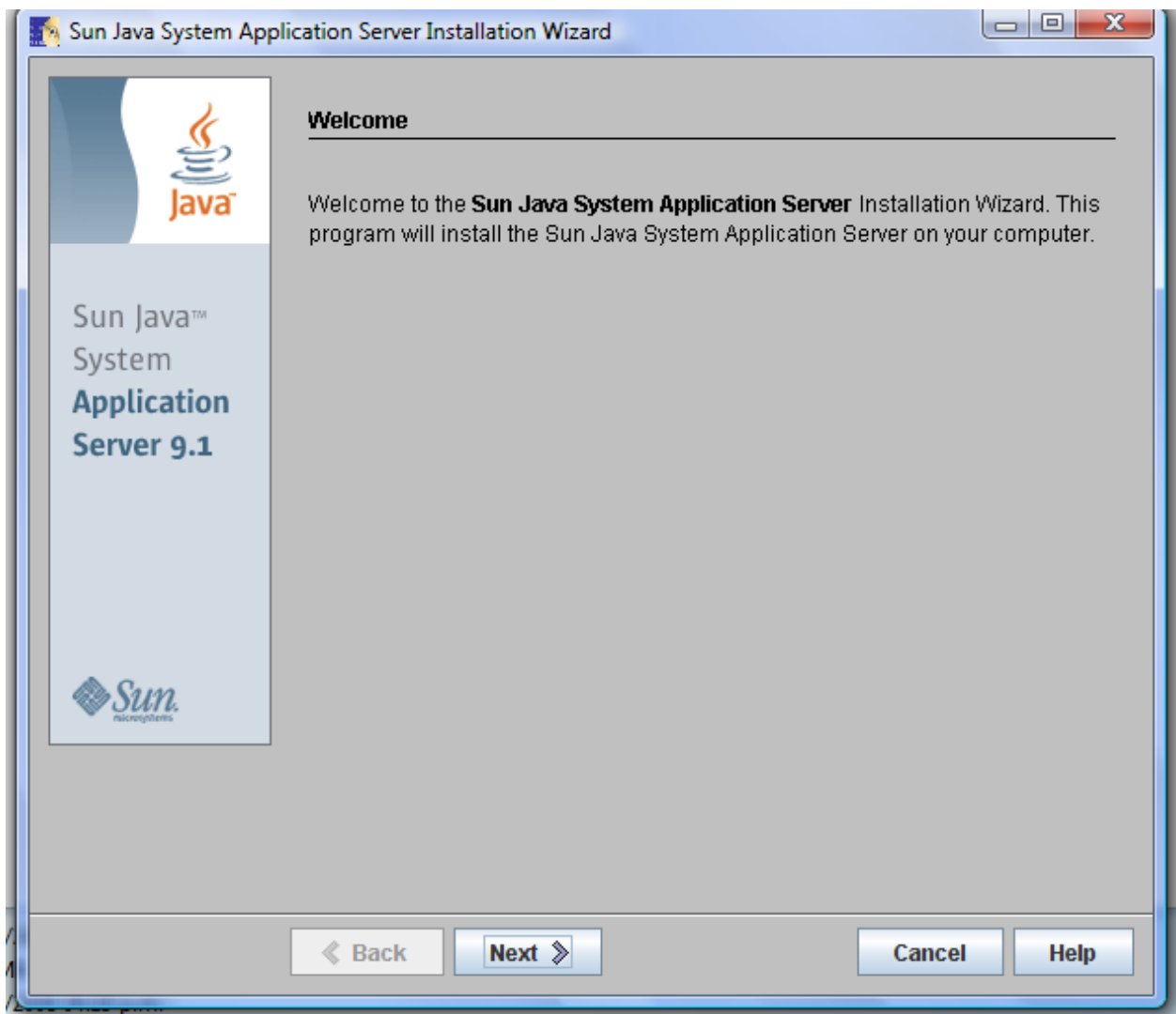


1.3 Instalación de Sun Java Application Server

Sun Java Application Server es el contenedor de servlets con más amplio soporte para el estándar de Java, y el servidor calificado como el de más sencilla administración.

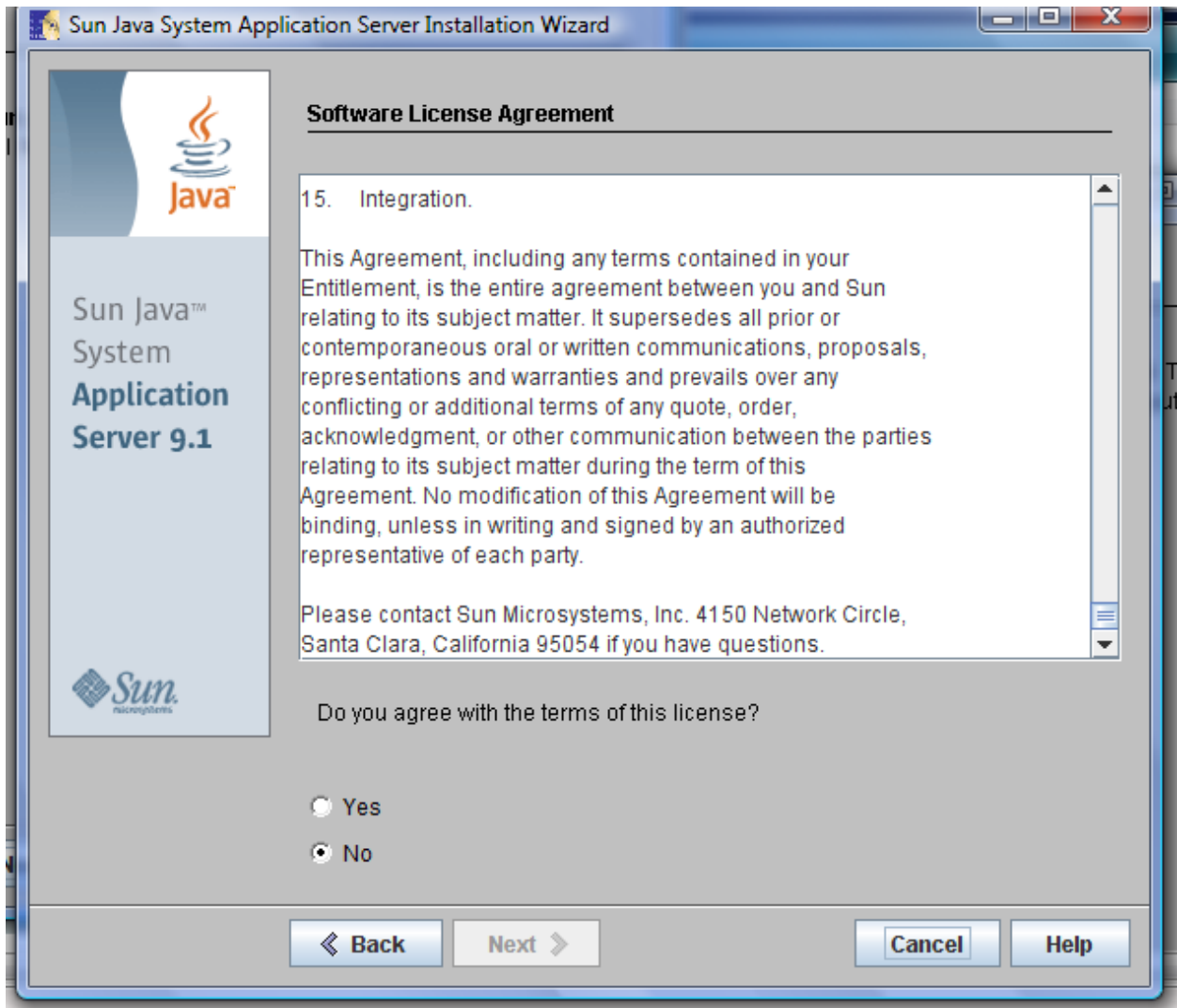
Para instalar el Sun Java Application Server primero bajaremos el archivo de instalación desde el sitio web: <http://java.sun.com/javase/downloads/index.jsp>

Una vez descargado el programa ejecutamos el archivo en nuestro servidor.

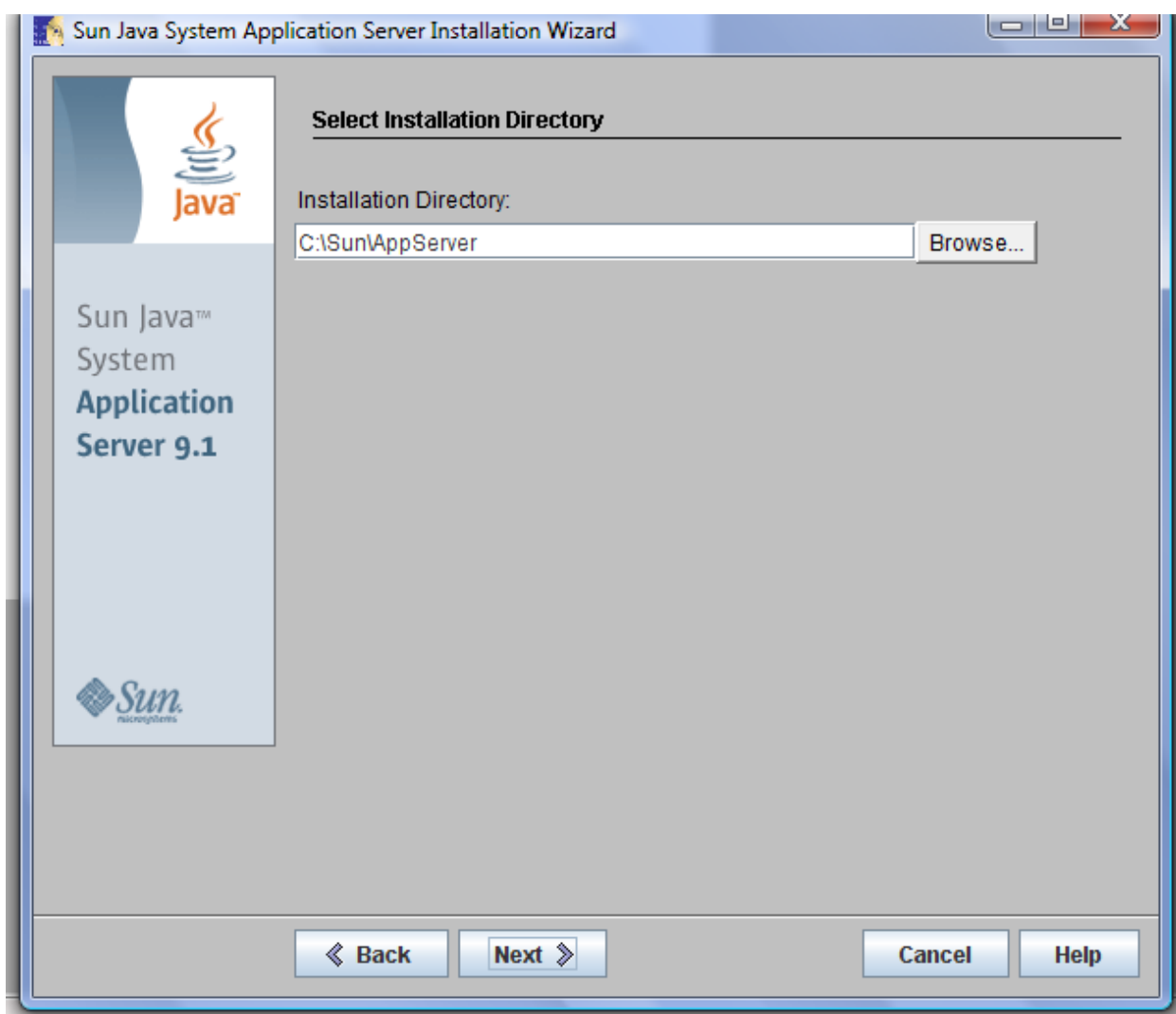


Damos clic en el botón “Next”.

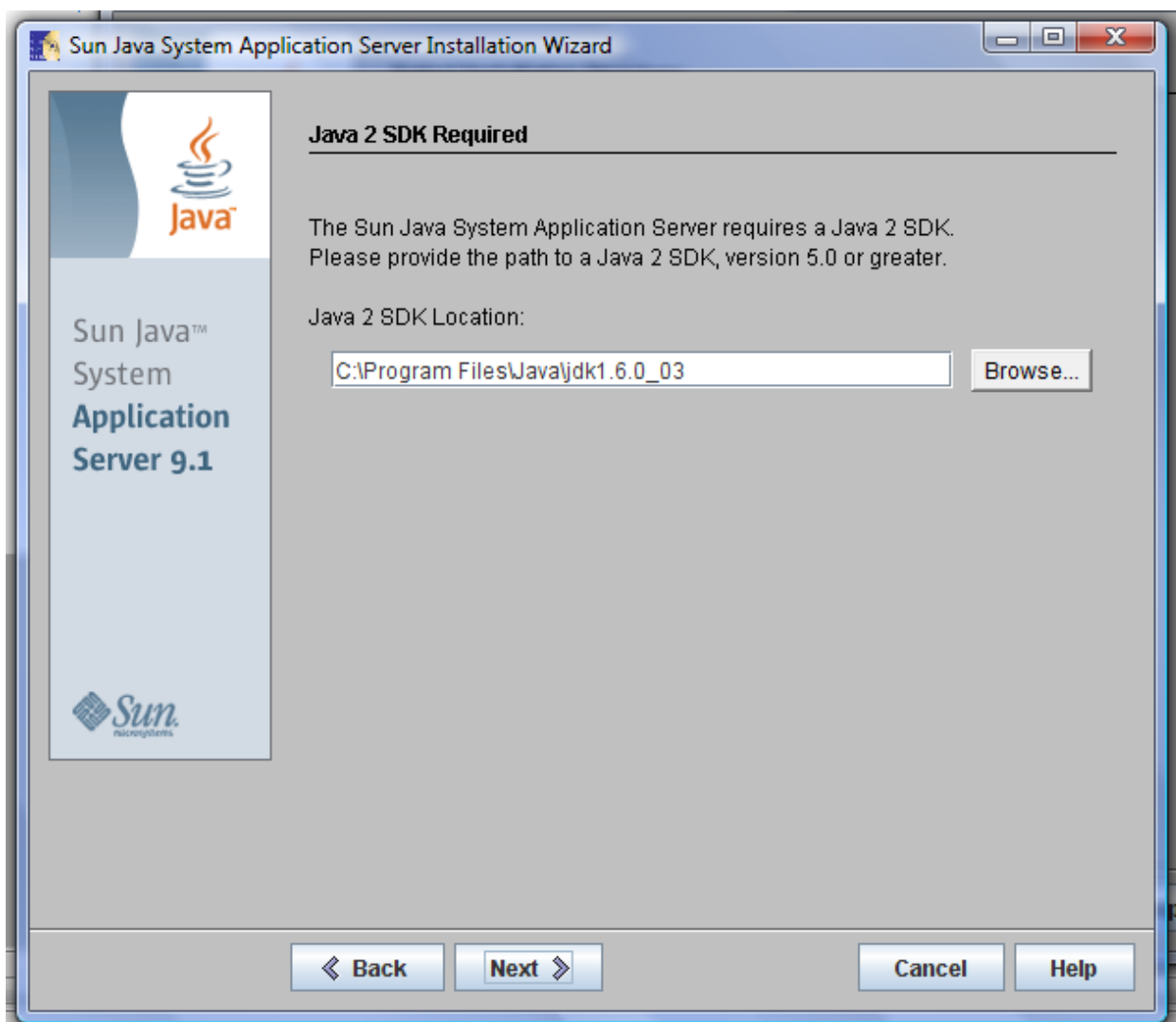
Primero nos muestra los términos de la licencia de Sun Java Application Server. Seleccionamos el botón de selección “Yes” que se encuentra bajo la viñeta “Do you agree with the terms of the license?”. Damos clic en el botón “Next”.



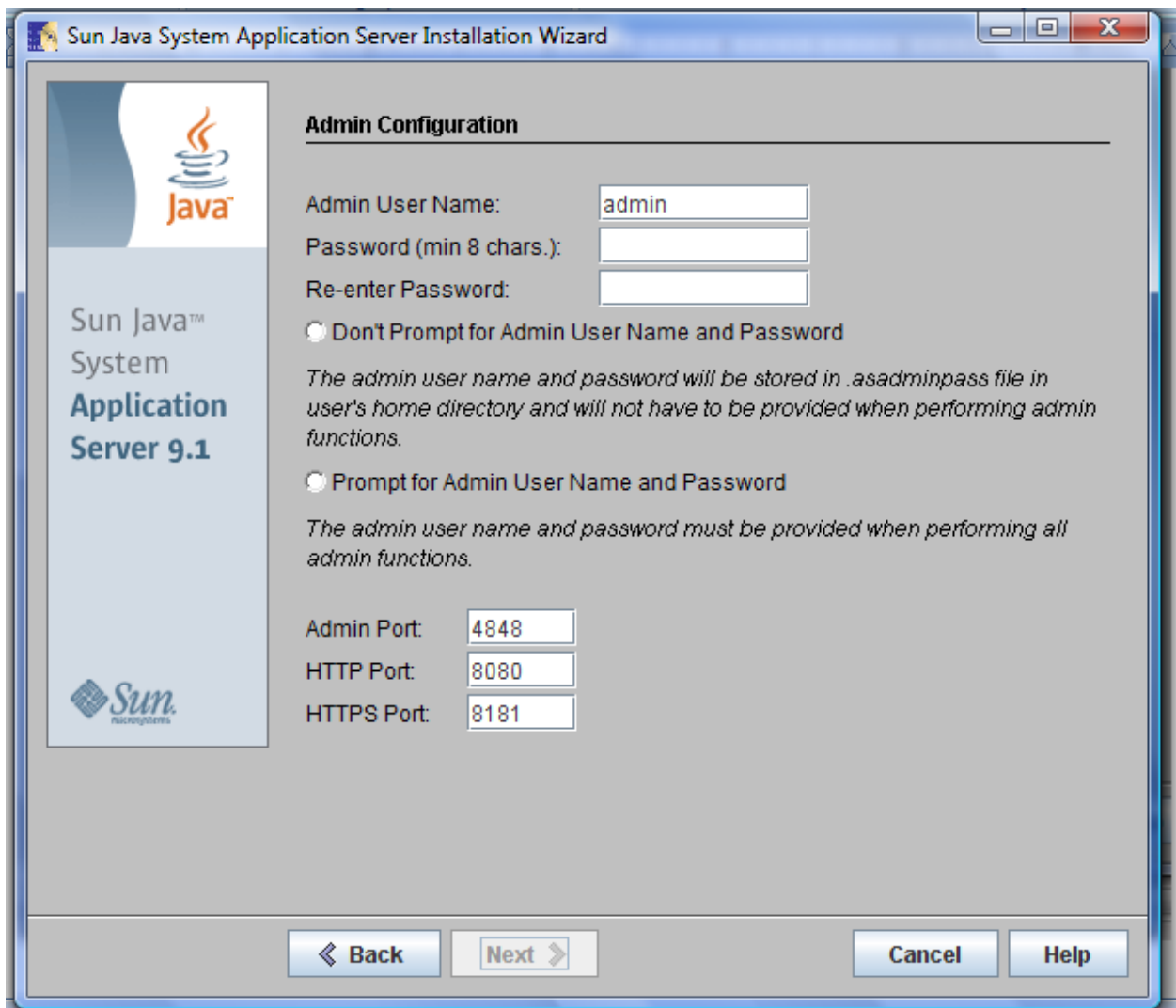
En esta pantalla, el programa de instalación nos pregunta la ubicación dentro de nuestro sistema de archivos, en la que deseamos almacenar el Sun Java Application Server. Si desamos almacenar la aplicación en una partición diferente o en un disco duro secundario, damos un clic al botón “Browse” y seleccionamos la ubicación deseada. En nuestro caso dejamos la ubicación por defecto. Si es la primera vez que instalamos el Sun Java Application Server en nuestro servidor no existe el directorio seleccionado, el programa de instalación nos pedirá confirmación para crear la ubicación nueva dentro del sistema de archivos, en este caso solo damos clic en el botón: “Create Directory”.



Ahora el programa nos pregunta la ubicación de nuestro JDK instalado en numeral anterior en este mismo manual. Si no modificamos la ruta por defecto de instalación de nuestro JDK durante su instalación, el programa de instalación será capaz de detectar la ultima versión del JDK instalado en nuestro servidor y utilizara esa versión durante la configuración y ejecución del Sun Java Application Server. En nuestro caso no modificamos la ruta de instalación del JDK por lo que dejamos la ruta sugerida clic en el botón “Next”. Si fuera el caso contrario, deberemos seleccionar el botón “Browse”, y seleccionar la ruta correcta de nuestra instalación del JDK.



Aquí el programa de instalación nos permite configurar el nombre y contraseña del administrador del Sun Java Application Server así como los puertos que este utiliza. Dejamos los puertos por defecto y colocamos una contraseña mínima de 8 caracteres, seleccionamos la opción “Prompt for admin User Name and Password” para mayor seguridad en nuestro servidor, ya que de lo contrario el nombre de usuario administrador y su contraseña se almacenaran en el disco duro y cualquiera con acceso al servidor podrá modificar la configuración si entra a la consola de administración.

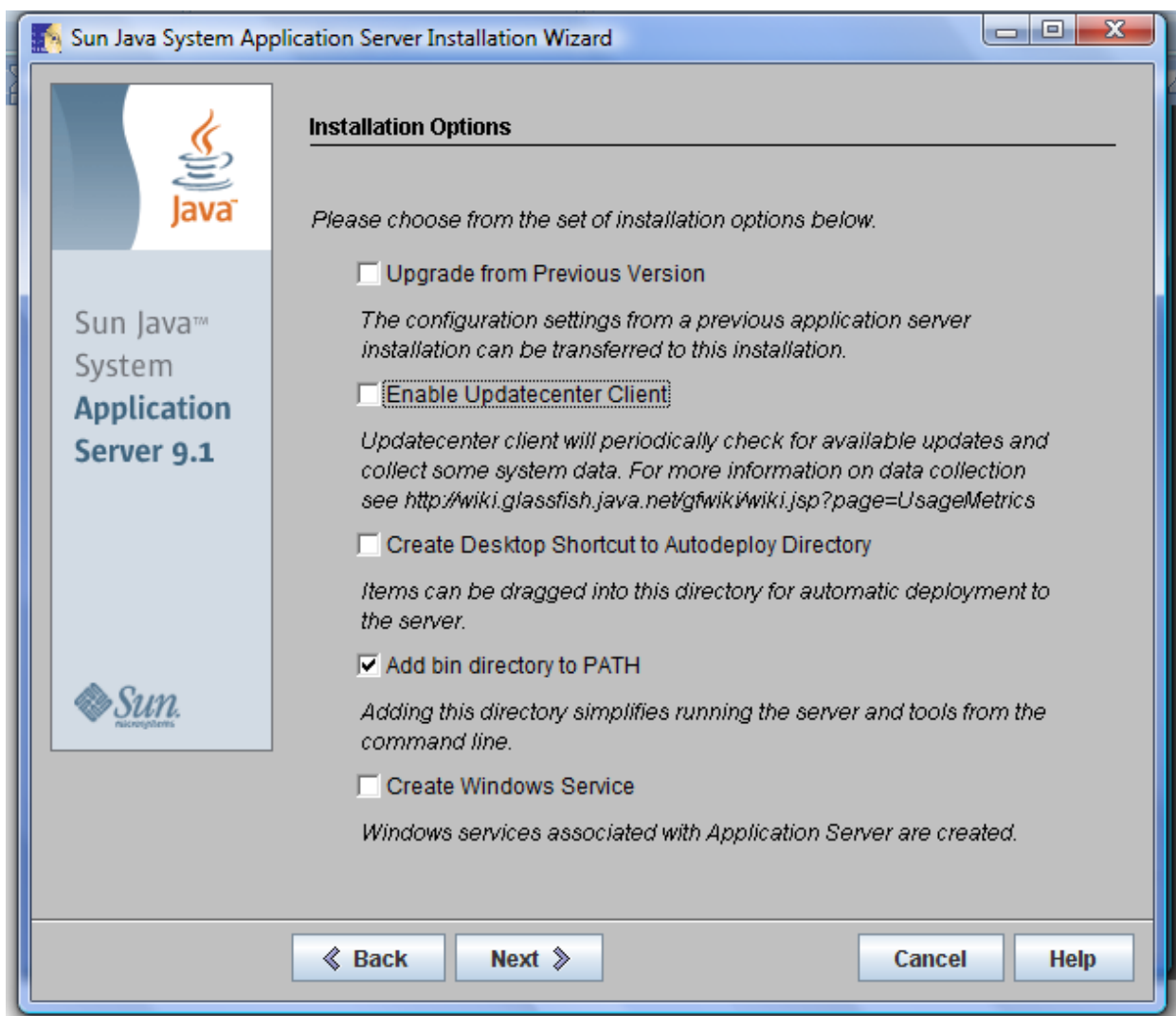


The screenshot shows the 'Sun Java System Application Server Installation Wizard' window. The title bar reads 'Sun Java System Application Server Installation Wizard'. On the left side, there is a vertical panel with the Java logo at the top, followed by the text 'Sun Java™ System Application Server 9.1', and the Sun Microsystems logo at the bottom. The main area is titled 'Admin Configuration'. It contains the following fields and options:

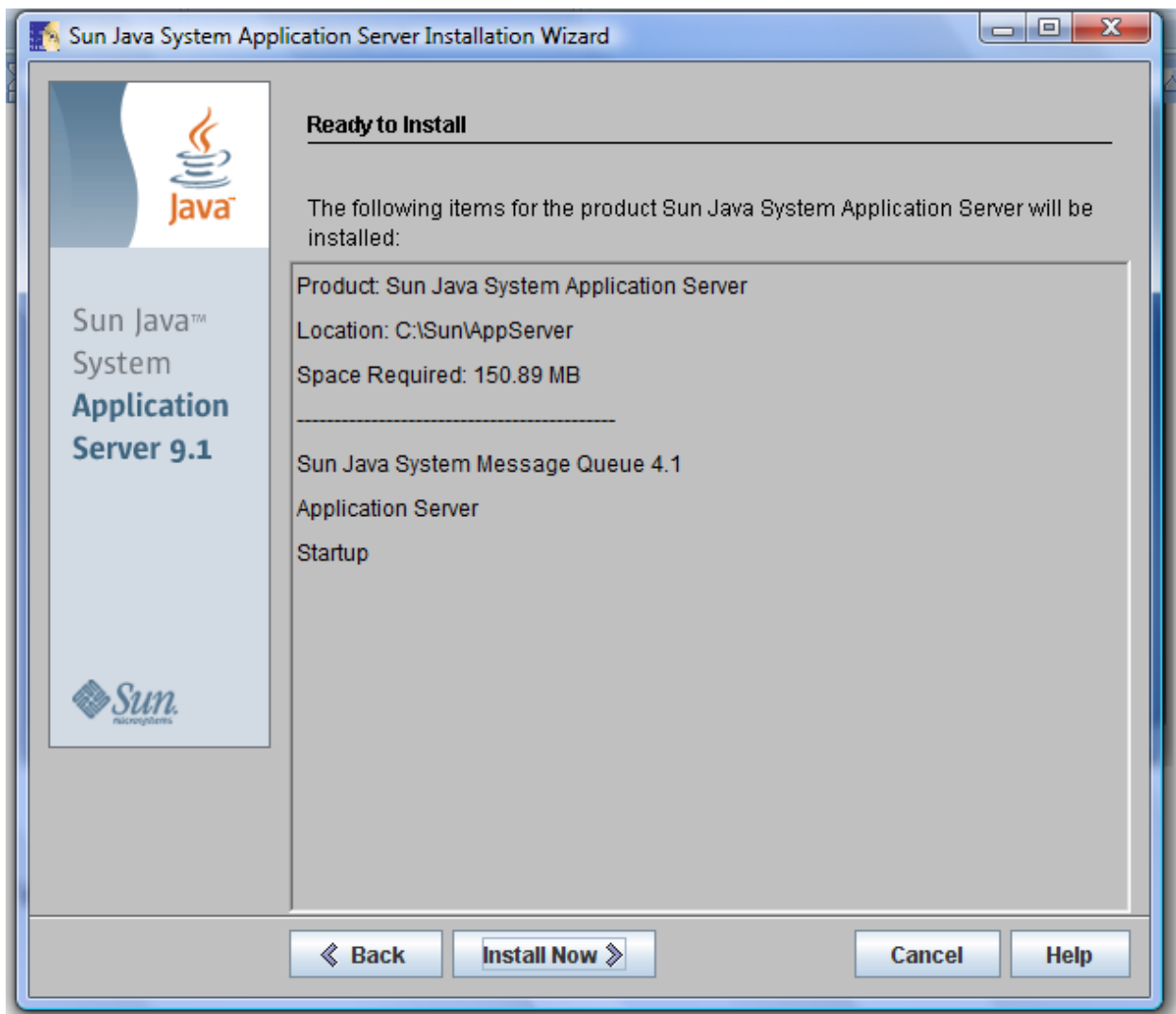
- Admin User Name:** A text box containing 'admin'.
- Password (min 8 chars.):** An empty text box.
- Re-enter Password:** An empty text box.
- Radio buttons:**
 - ☐ Don't Prompt for Admin User Name and Password
 - ☒ Prompt for Admin User Name and Password
- Descriptions:**
 - Under the first option: *The admin user name and password will be stored in .asadminpass file in user's home directory and will not have to be provided when performing admin functions.*
 - Under the second option: *The admin user name and password must be provided when performing all admin functions.*
- Ports:**
 - Admin Port:** 4848
 - HTTP Port:** 8080
 - HTTPS Port:** 8181

At the bottom of the window, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

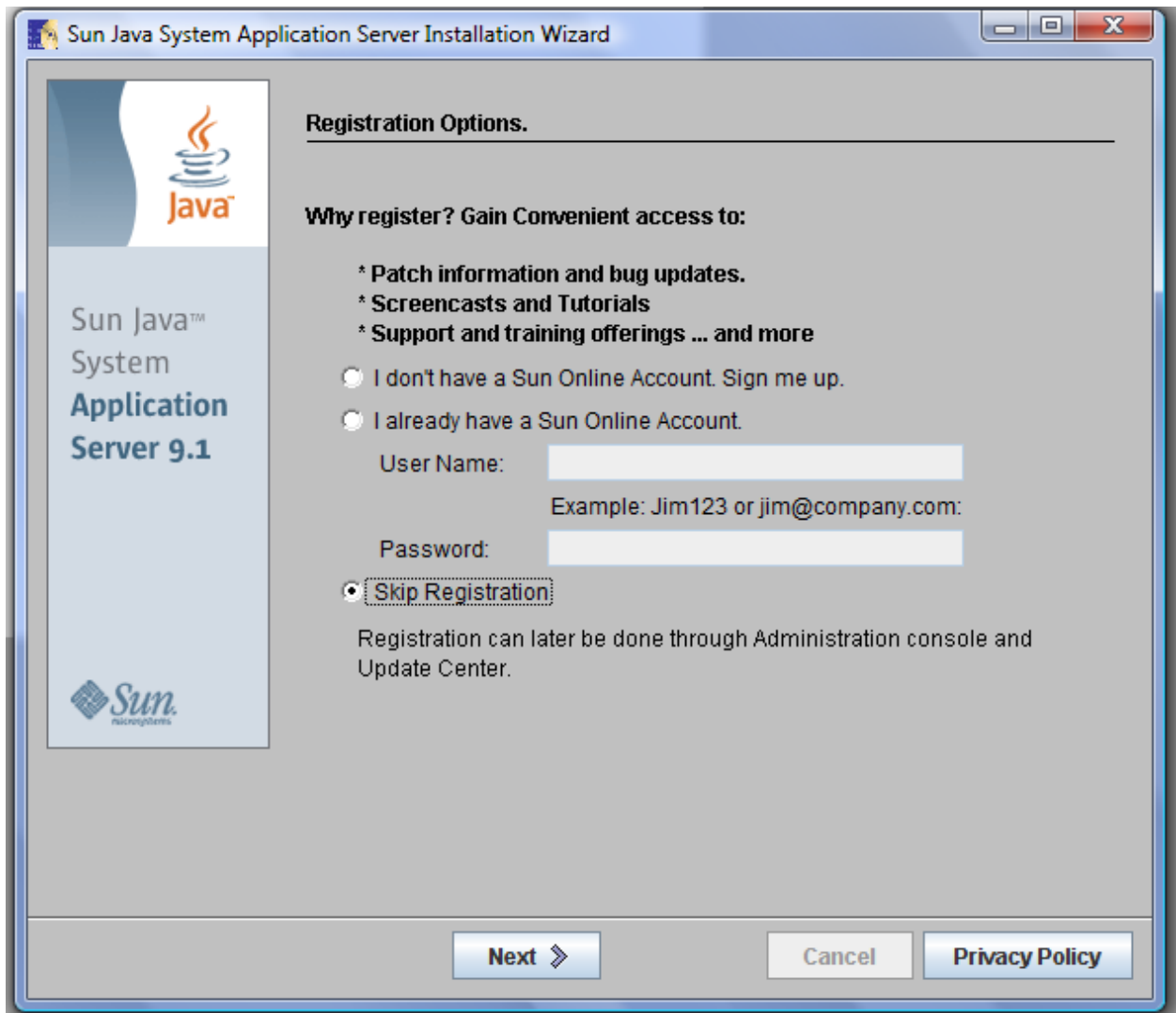
En la siguiente pantalla para evitar sobrecargas innecesarias en nuestra red solamente seleccionaremos la opción “Add bin directory to PATH”. Damos clic en el botón “Next”.



Después de configurar todos los elementos de la instalación, damos clic en el botón “Install Now”.



Ahora el programa de instalación preguntara si deseamos crear una cuenta en el sitio de Sun Java, seleccionamos "Skip Registration" para evitar correo innecesario. Y seleccionamos el botón "Next".



The image shows a screenshot of the "Sun Java System Application Server Installation Wizard" window. The title bar reads "Sun Java System Application Server Installation Wizard". On the left side, there is a vertical panel with the Java logo at the top, followed by the text "Sun Java™ System Application Server 9.1", and the Sun Microsystems logo at the bottom. The main area of the window is titled "Registration Options." and contains the following text: "Why register? Gain Convenient access to:" followed by a bulleted list: "* Patch information and bug updates.", "* Screencasts and Tutorials", and "* Support and training offerings ... and more". Below this list are two radio button options: "I don't have a Sun Online Account. Sign me up." and "I already have a Sun Online Account.". Under the second option, there are input fields for "User Name:" and "Password:", with an example "Example: Jim123 or jim@company.com:" provided. The "Skip Registration" option is selected with a radio button. Below this, a note states: "Registration can later be done through Administration console and Update Center." At the bottom of the window, there are three buttons: "Next >", "Cancel", and "Privacy Policy".

Sun Java™ System Application Server 9.1

Registration Options.

Why register? Gain Convenient access to:

- * Patch information and bug updates.
- * Screencasts and Tutorials
- * Support and training offerings ... and more

☐ I don't have a Sun Online Account. Sign me up.

☐ I already have a Sun Online Account.

User Name:

Example: Jim123 or jim@company.com:

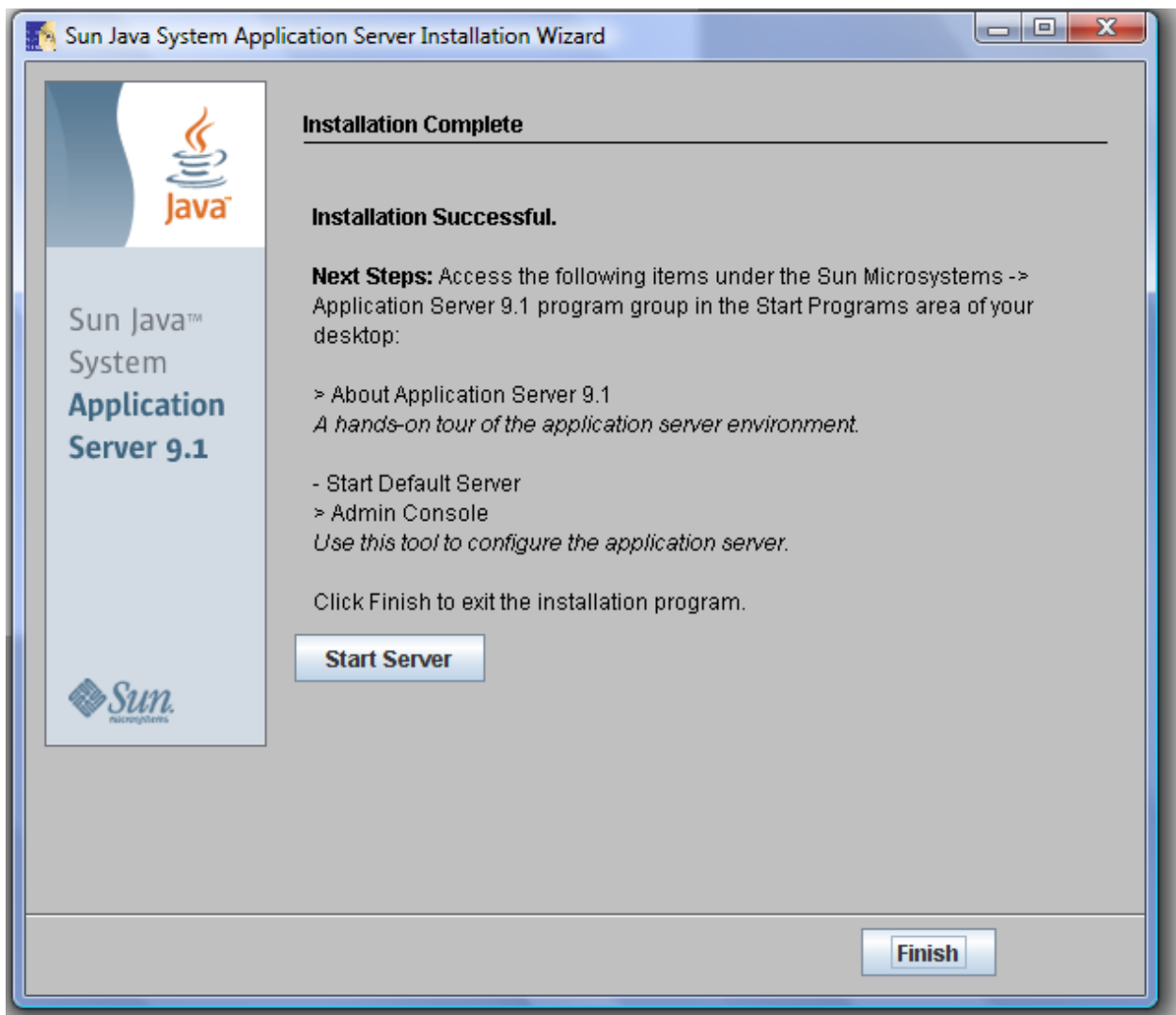
Password:

☒ Skip Registration

Registration can later be done through Administration console and Update Center.

Next > Cancel Privacy Policy

Una vez finalizada la instalación seleccionamos el botón “Start Server”, esto iniciará los servicios de nuestro contenedor de servlet. Damos clic en el botón “Finish”.

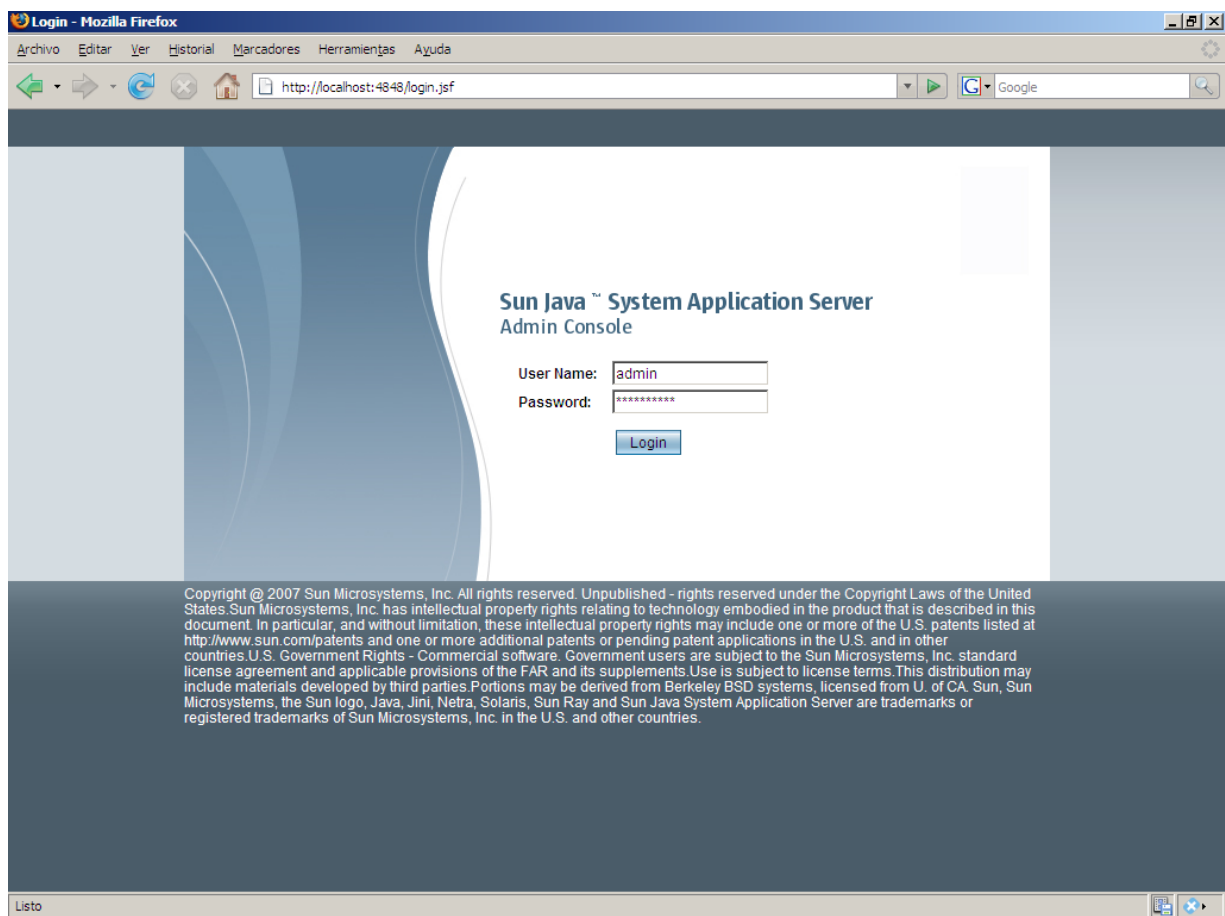


1.4 Instalación del Sistema de Evaluación General de Desempeño (SEGD)

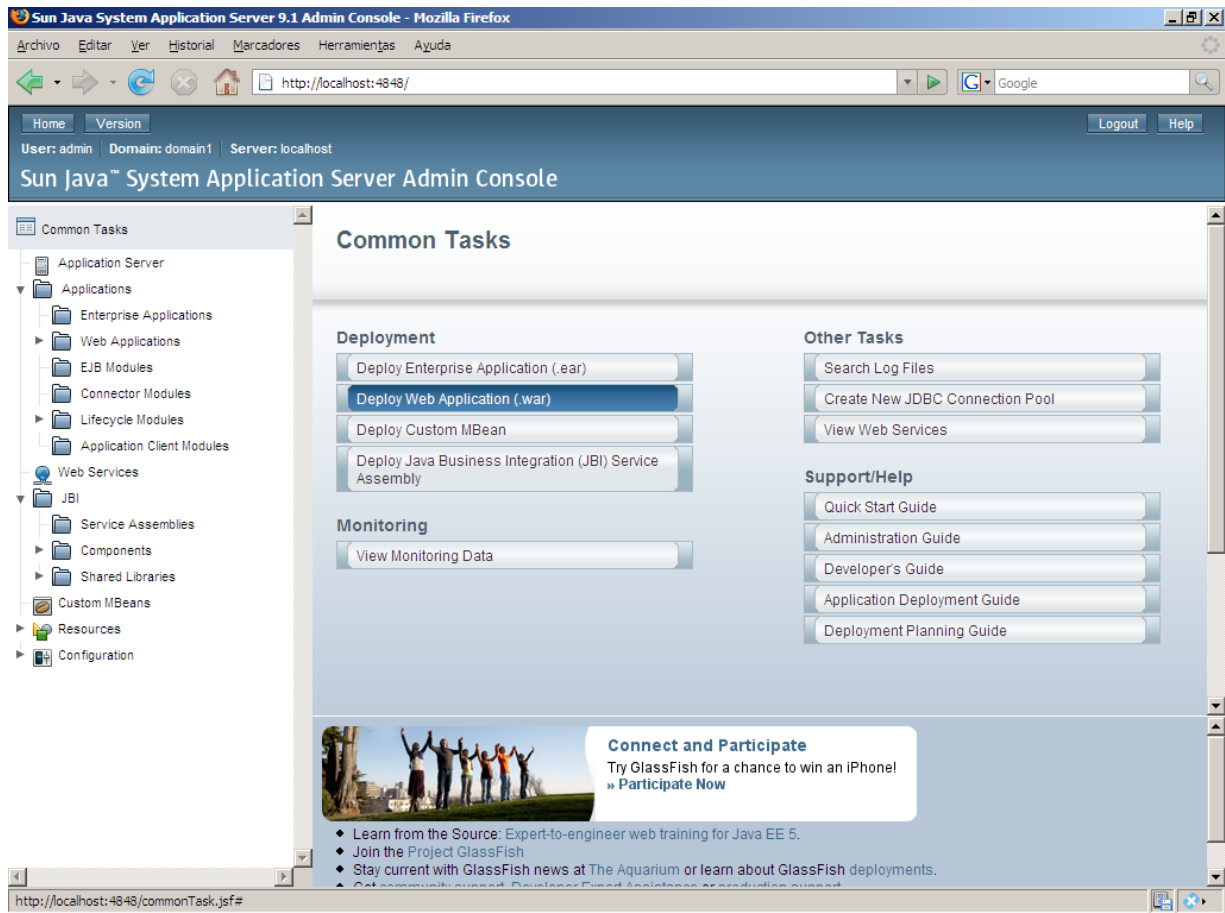
La distribución del sistema será a través de un archivo WAR, es archivo WAR es un archivo comprimido que contiene todos los archivos con el código fuente y otros recursos necesarios, como imágenes y librerías. Todos estos archivos deben organizarse bajo una estructura de archivos estándar definida en el estándar de Sun Java. En nuestro caso el IDE utilizado para desarrollar el SEG D (Netbeans) crea automáticamente la estructura requerida por los archivos WAR.

Para instalar el archivo WAR en nuestro contenedor de servlet, primero entramos a la consola de administración a través del URL: <http://localhost:4848/>

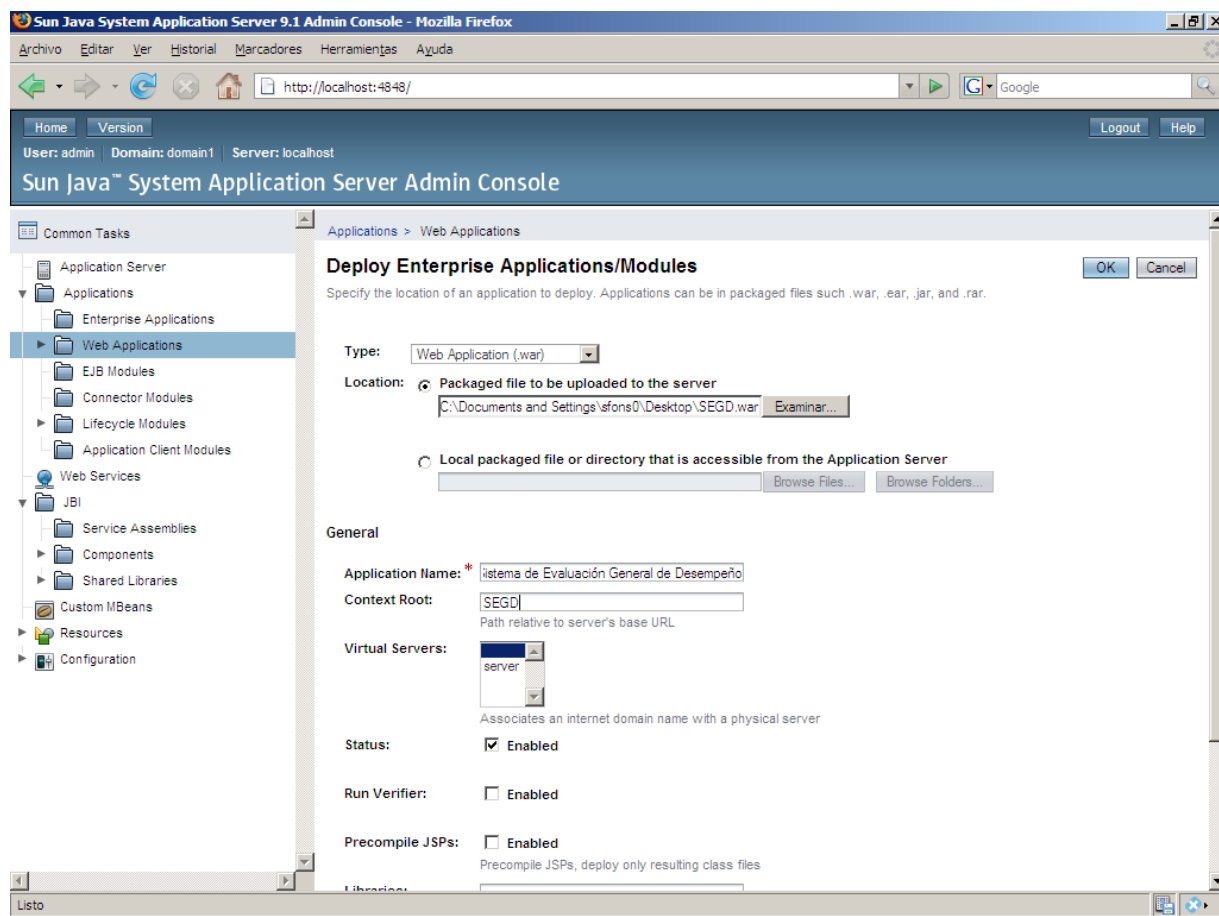
Deberemos digitar el usuario administrador y su contraseña definidos durante la instalación del Sun Java Application Server.



Ahora seleccionaremos la opción “Deploy Web application (.WAR)”.



Ahora damos clic en el botón “Examinar” y buscamos la ubicación de nuestro archivo WAR (SEGD.war). Luego definimos un nombre de aplicación, para esto escribimos “Sistema de Evaluación General de Desempeño” dentro de la opción llamada “Application Name”, dejamos las otras opciones con sus valores por defecto y damos un clic al botón “OK”.



UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
INGENIERIA EN COMPUTACION



**“GUIA PARA LA IMPLEMENTACION DE LAS MEJORES
PRACTICAS EN LA DIRECCION DE PROYECTOS APLICADO EN
UN PROYECTO DE DESARROLLO DE SOFTWARE PARA EL
INSTITUTO TECNOLOGICO CENTROAMERICANO”**

Apéndice C

Trabajo de graduación para optar al título de:
Ingeniero en Ciencias de la Computación

Presentado por:

Guillermo Arturo Cerrato Gámez

Salvador Fonseca Rivas

Asesor:

Ing. Carlos López Linares, MSc, MBA

Soyapango, 25 Marzo del 2008

Índice

APÉNDICE C - MANUAL DEL USUARIO.....	3
1.1 PANTALLAS GENERALES.....	3
1.1.1 Pantalla de ingreso al sistema	3
1.1.2 Menú principal	4
1.1.3 Cambio de contraseña	5
1.2 MÓDULO DE PLANIFICACIÓN ANUAL	7
1.2.1 Agregar tarea	7
1.2.2 Agregar proyecto.....	8
1.2.3 Agregar presupuesto.....	10
1.3 REPORTES.....	12
1.3.1 Planificación Anual	12
1.3.2 Presupuesto	13
1.4 MÓDULO DE DESARROLLO PROFESIONAL	14
1.4.1 Agregar tarea	14
1.4.2 Reporte de Desarrollo Profesional	16
1.5 MÓDULO DE ACTIVIDADES NO PLANIFICADAS	16
1.5.1 Agregar tarea	16
1.5.2 Reporte de Actividades no planificadas.....	19
1.6 MÓDULO DE EVALUACIÓN ANUAL.....	20
1.6.1 Mantenimientos.....	20
1.6.1.1 Agregar cuestionarios.....	20
1.6.1.2 Agregar preguntas	21
1.6.2 Procesos.....	23
1.6.2.1 Configurar evaluación	23
1.6.2.2 Selección de evaluadores.....	24
1.6.2.3 Iniciar evaluación	24
1.6.2.4 Evaluaciones Pendientes.....	26
1.6.3 Reportes.....	28
1.6.3.1 Resultados.....	28
1.7 MÓDULO DE SEGURIDAD.....	32
1.7.1 Agregar departamentos	32
1.7.2 Agregar cargo de empleados	33
1.7.3 Agregar empleado.	35

Apéndice C - Manual del Usuario

Este manual es una referencia para el uso del Sistema de Evaluación General de Desempeño. Este manual será especialmente útil para los usuarios nuevos del sistema.

1.1 Pantallas Generales

1.1.1 Pantalla de ingreso al sistema

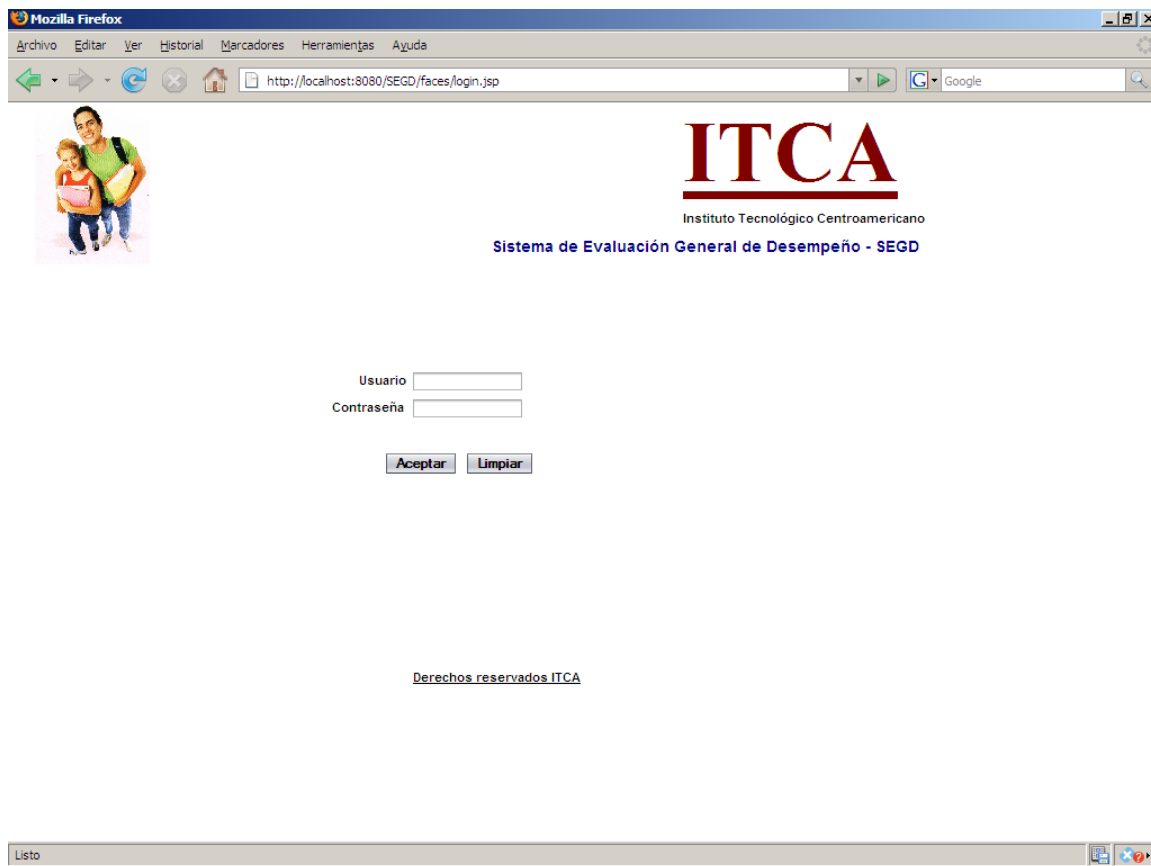
Descripción: Permite que el empleado acceda a las funciones del sistema.

Condiciones previas:

1. El empleado debe estar registrado en el sistema. <http://www.itca.edu.sv/segd/>

Pasos

1. En el campo de **Usuario** será asignado por el administrador del área y se utilizara el mismo de la red por requerimiento del Instituto.
2. El campo de **Contraseña** se puede utilizar una contraseña genérica temporal asignada por el administrador del área y que posteriormente el usuario puede cambiarla a una personal.
3. Se presionara el botón de **Aceptar** para continuar con el desarrollo del sistema o puede presionar el botón de **Limpiar** en caso de corregir cualquiera de los dos campos anteriores.



1.1.2 Menú principal

Descripción: Permite que el empleado ingrese a las opciones del sistema.

Pasos:

Las dos primeras opciones serán:

1. **Salir:** En la cual puede el usuario salir del sistema.
2. **Contraseña:** Cambio de contraseña genérica temporal a una personal.

Se continuara con un menú de opciones con el cual se desea trabajar. El menú constara de cuatro puntos importantes en su utilización.

- 1. Planificación Anual**
- 2. Desarrollo profesional**
- 3. Actividades no planificadas**
- 4. Evaluación Anual**
- 5. Seguridad**



1.1.3 Cambio de contraseña

Descripción: Permite que el empleado cambie su contraseña del sistema.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.

Pasos.

Esta etapa del sistema será capaz de cambiar la contraseña, se puede mantener la contraseña temporal para ingresar al sistema o se asigna una nueva. Con lo cual se recomienda lo siguiente:

1. El empleado proporciona su contraseña en el campo **Contraseña Actual**.
2. Se digita la nueva contraseña. Utiliza el campo de **Nueva Contraseña**.
3. Se confirma la nueva contraseña.
4. Se presiona la opción de **Aceptar** para efectuar su cambio.
5. Puede seleccionar la opción **Menú** para continuar.



The screenshot shows a web browser window (Mozilla Firefox) displaying the ITCA (Instituto Tecnológico Centroamericano) system interface. The URL bar shows `http://localhost:8080/SEGD/faces/Seguridad/sgContraseña.jsp`. The page header includes the ITCA logo and the text "Instituto Tecnológico Centroamericano". Below the logo, there are navigation links: "Salir", "Contraseña", and "Menú". The main title of the page is "Sistema de Evaluación General de Desempeño - SEGD", and it indicates "Usuario conectado: clopez".

The main content area is titled "Cambio de Contraseña" in purple. It contains three input fields for password management:

- Contraseña Actual
- Nueva Contraseña
- Confirmar Contraseña

Below these fields are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

1.2 Módulo de Planificación Anual

1.2.1 Agregar tarea

Descripción: El empleado utiliza esta fase para *Agregar tarea* para registrar todas las actividades que le han sido asignadas por proyecto y programadas en su planificación anual. Una vez la actividad ha sido ingresada, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe registrar su ingreso al sistema.
2. El proyecto correspondiente debe estar creado.

Pasos

Cuando el empleado desea agregar al sistema una actividad o tarea se le presenta una pantalla con las opciones siguientes:

1. En el campo llamado **Subalternos**, se decide a la persona que se le asignara las nuevas tareas o actividades
2. Luego el empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
3. El empleado selecciona un proyecto. Utiliza el botón de opción denominada **Proyectos**.
4. El empleado agrega una tarea en el proyecto seleccionado para su planificación anual. La nueva tarea es mostrada en la lista de tareas por proyectos. Presiona el botón de **Agregar Tarea**.
5. Seguido el empleado opta por los botones de **Guardar Cambios o deshacer cambios**.
6. Si necesita reporte se generan con el botón de **Reporte**.

ITCA
Instituto Tecnológico Centroamericano

Sistema de Evaluación General de Desempeño - SEG D
Usuario conectado: clopez

Planificación Anual
Agregar Tarea

Subalterno:

Proyecto	Tarea	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5
Implementación de tecnología JAVA como Estándar de programación	Tomar curso general de JAVA	10.00	10.00	10.00	10.00	10.00
Implementación de tecnología JAVA como Estándar de programación	Tomar curso de JSP	0.00	0.00	0.00	0.00	0.00
Implementación de UML como Estándar para la recolección de requerimientos	Elaborar Estándar	10.00	50.00	10.00	10.00	10.00

1.2.2 Agregar proyecto

Descripción: El empleado utiliza esta fase Agregar proyecto para registrar los proyectos que le han sido asignados.

Condiciones previas:

1. El empleado debe registrar su ingreso al sistema.

Pasos

Cuando el empleado desea agregar al sistema un proyecto se le presenta una pantalla con las opciones siguientes:

1. Se inicia con seleccionar el año en el cual se agregaran los proyectos y luego se presiona el botón **Actualizar** para que presente los proyectos actuales de ese año.
2. Se tendrá la opción que el empleado ve una lista de sus tareas asignadas, las tareas están organizadas por proyectos.
3. Luego el empleado agrega el proyecto nuevo. Presiona el botón **Agregar Tareas**.
4. Si desea borrar algún proyecto que ya no utilizará se presionara el botón de **Borrar**.
5. Y por último, el empleado presiona los botones de **Guardar Cambios o Deshacer Cambios**.



1.2.3 Agregar presupuesto

Descripción: El empleado utiliza esta fase Agregar Presupuesto para registrar todos los gastos presupuestados de un proyecto dentro de su planificación anual. Una vez el gasto ha sido ingresado, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Pasos

Cuando el empleado desea agregar al sistema un gasto presupuestado se le presenta una pantalla con las siguientes opciones:

1. Se inicia con que el empleado ve una lista de proyectos y decide cual escogerá.
2. Continúa que el empleado deberá seleccionar un proyecto.
3. Luego el empleado ve una lista de sus gastos presupuestados asignados para un proyecto asignado.
4. El empleado agrega un gasto en el presupuesto de un proyecto dentro de su planificación anual. Se utiliza el botón de **Agregar Presupuesto** para tal caso.
5. Y por último, el empleado presiona el botón **Guarda Cambios o Deshacer Cambios**.
6. Si el empleado necesita un reporte de este, selecciona el botón correspondiente **Reporte**.

Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost:8080/SEGD/faces/PlanAnual/paPresupuesto.jsp?strTipoModulo=P

Google



Salir Contraseña Menú

ITCA
Instituto Tecnológico Centroamericano

Sistema de Evaluación General de Desempeño - SEGD

Usuario conectado: clopez

Agregar Presupuesto
Planificación Anual

Proyecto: Implementación de tecnología JAVA como Estándar de programación

Guardar Cambios Agregar Presupuesto Deshacer Cambios Reporte

Presupuesto	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9
Curso de JAVA	1200.00	1200.00	1200.00	0.00	0.00	0.00	0.00	0.00	0.00

Listo

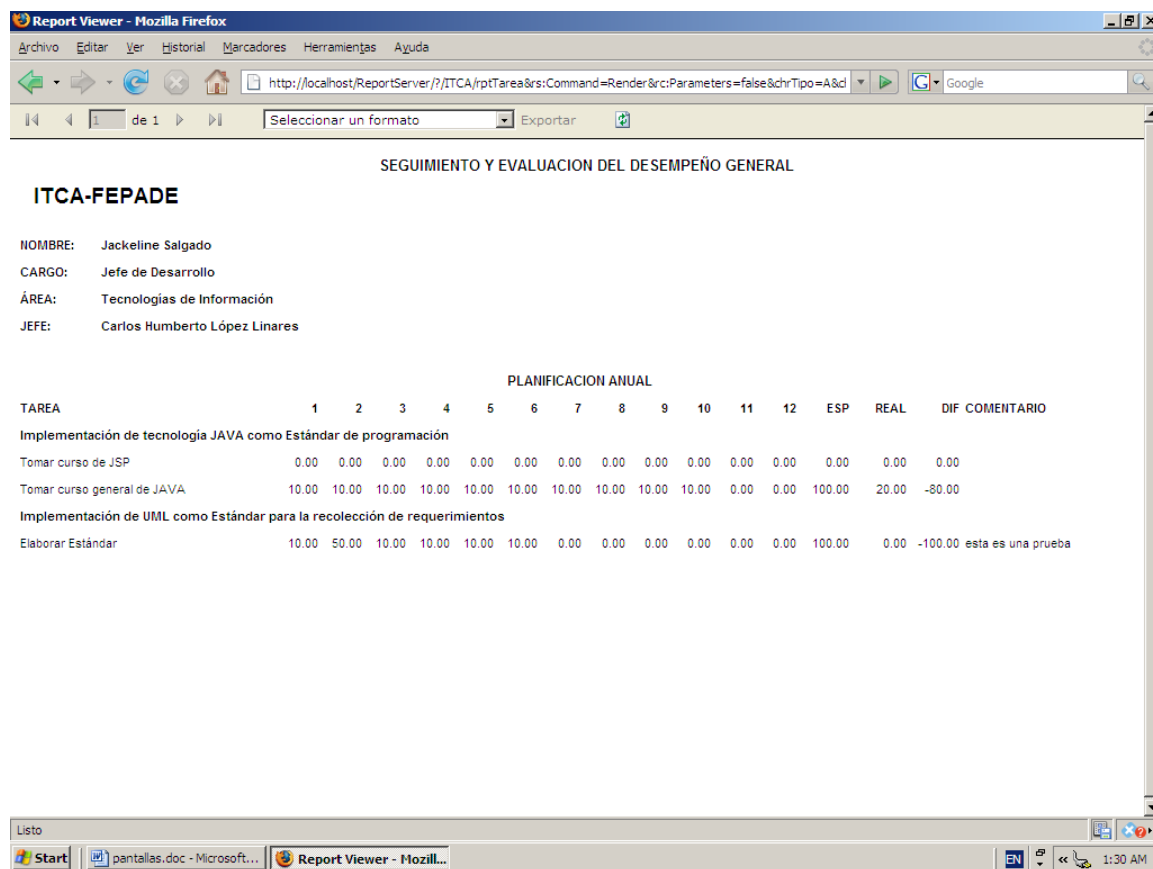
Start pantallas.doc - Microsoft... Mozilla Firefox

1:32 AM

1.3 Reportes

Se desarrollo para cada fase del proyecto un reporte de salida, ya definidos por los requerimientos del Instituto con el objetivo de facilitar el uso y lectura de su información.

1.3.1 Planificación Anual



Report Viewer - Mozilla Firefox

http://localhost/ReportServer/?/ITCA/rptTarea&rs:Command=Render&rc:Parameters=false&chrTipo=A&d

SEGUIMIENTO Y EVALUACION DEL DESEMPEÑO GENERAL

ITCA-FEPADE

NOMBRE: Jackeline Salgado
CARGO: Jefe de Desarrollo
ÁREA: Tecnologías de Información
JEFE: Carlos Humberto López Linares

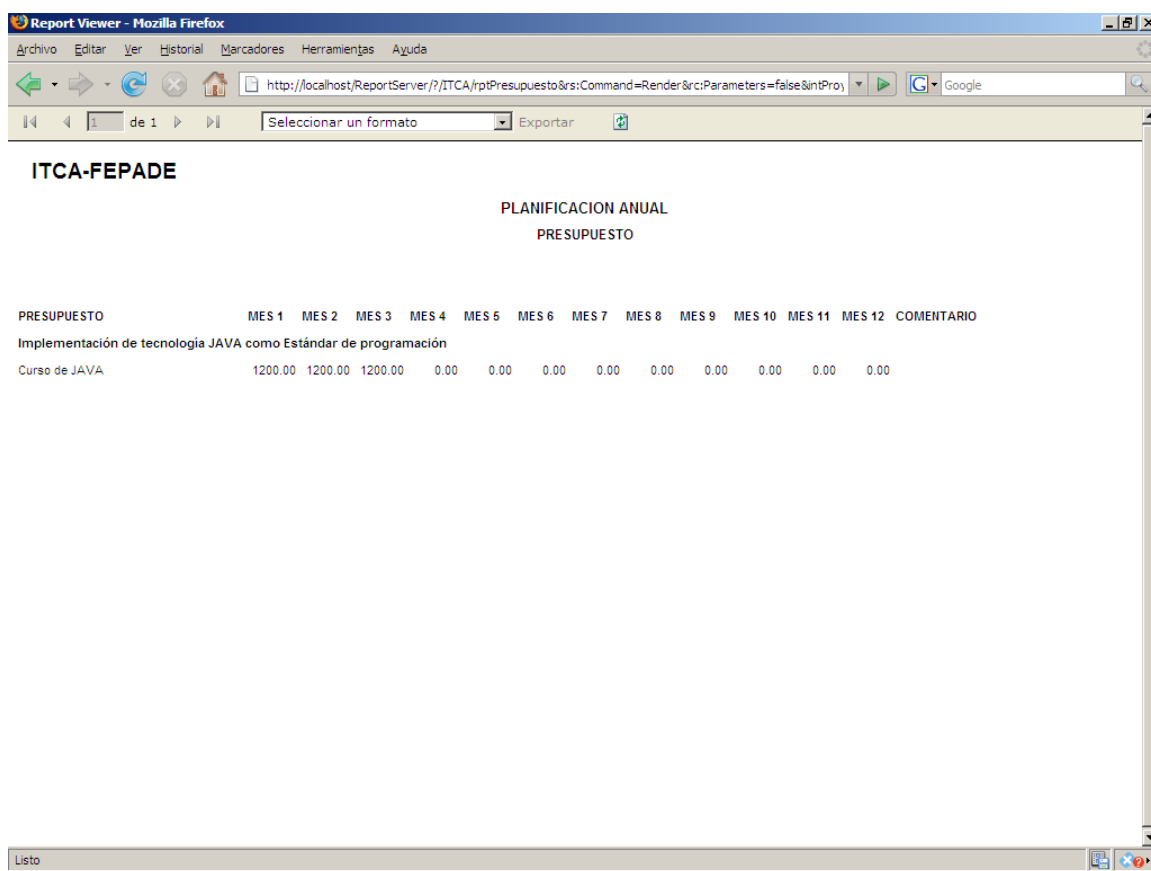
PLANIFICACION ANUAL

TAREA	1	2	3	4	5	6	7	8	9	10	11	12	ESP	REAL	DIF	COMENTARIO
Implementación de tecnología JAVA como Estándar de programación																
Tomar curso de JSP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Tomar curso general de JAVA	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	0.00	0.00	100.00	20.00	-80.00	
Implementación de UML como Estándar para la recolección de requerimientos																
Elaborar Estándar	10.00	50.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	-100.00	esta es una prueba

Listo

Start | pantallas.doc - Microsoft... | Report Viewer - Mozill... | EN | 1:30 AM

1.3.2 Presupuesto



Report Viewer - Mozilla Firefox

http://localhost/ReportServer/?/ITCA/rptPresupuesto&rs:Command=Render&rc:Parameters=false&intProy

ITCA-FEPADE

PLANIFICACION ANUAL
PRESUPUESTO

PRESUPUESTO	MES 1	MES 2	MES 3	MES 4	MES 5	MES 6	MES 7	MES 8	MES 9	MES 10	MES 11	MES 12	COMENTARIO
Implementación de tecnología JAVA como Estándar de programación													
Curso de JAVA	1200.00	1200.00	1200.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

Listo

1.4 Módulo de Desarrollo profesional

1.4.1 Agregar tarea

Descripción: El empleado utiliza esta etapa Agregar tarea para registrar todas las actividades de estudio que le han sido asignadas para su desarrollo profesional.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.

Pasos

Se desarrollo en esta parte se detalla como agregar al sistema una actividad o tarea de asignación académica en la fase de desarrollo profesional, con lo que se define en los pasos siguientes para llevarla a cabo:

1. Se inicia con la selección del personal en el botón de **Subalterno**.
2. Luego, el empleado ve una lista de sus tareas asignadas.
3. El empleado agrega una tarea en su planificación de desarrollo profesional.
Presiona botón **Agregar Tarea**.
4. Por último el empleado puede seleccionar los botones de **Guardar cambios o Deshacer cambios**.
5. Si desea ver reportes, selecciona botón **Reporte**.

Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost:8080/SEGD/faces/PlanAnual/psTarea.jsp

Google



Salir Contraseña Menú

ITCA

Instituto Tecnológico Centroamericano

Sistema de Evaluación General de Desempeño - SEGD

Usuario conectado: clopez

Desarrollo Profesional

Agregar Asignación académica

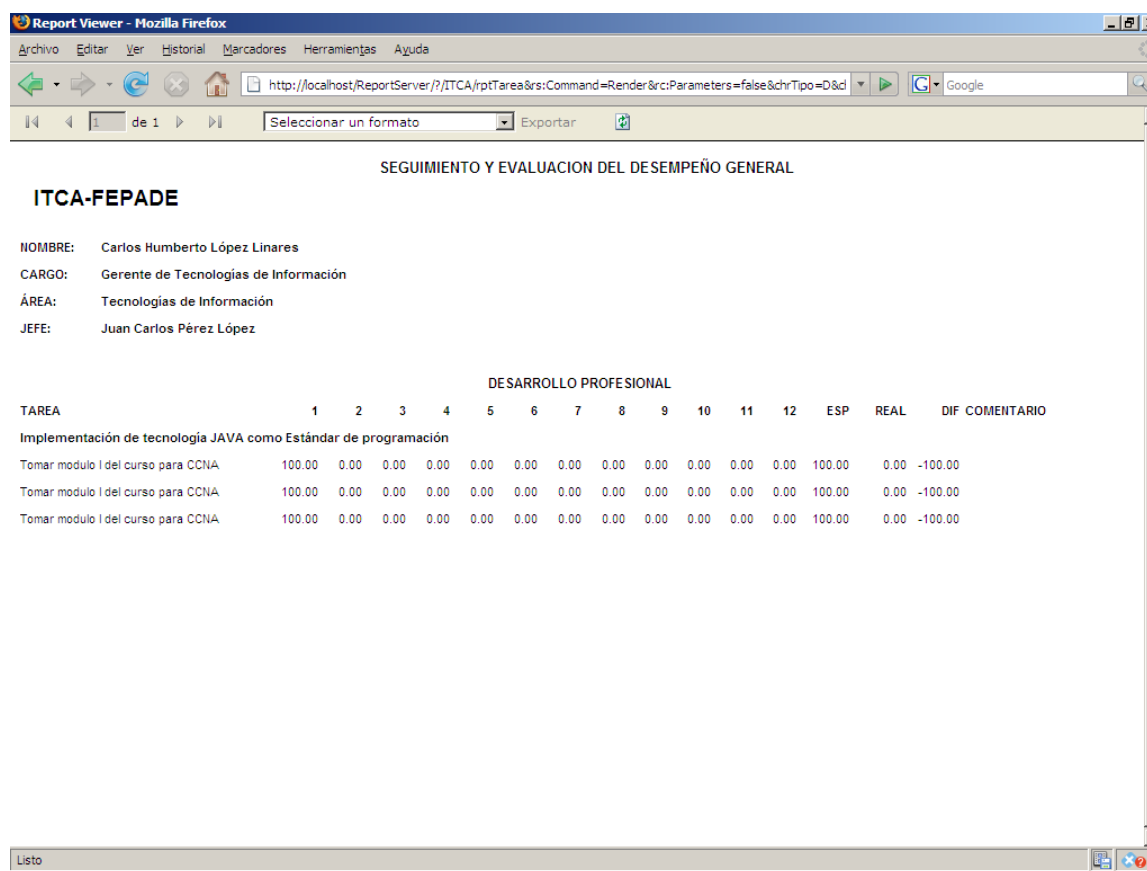
Subalterno: Carlos Humberto López Linares - Tecnologías de Información

Guardar Cambios Agregar Tarea Deshacer Cambios Reporte

Asignación académica	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8
Tomar modulo I del curso para CCNA	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Listo

1.4.2 Reporte de Desarrollo Profesional



Report Viewer - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost/ReportServer/?/ITCA/rptTarea&rs:Command=Render&rs:Parameters=false&chrTipo=D&d

Google

Seleccionar un formato Exportar

SEGUIMIENTO Y EVALUACION DEL DESEMPEÑO GENERAL

ITCA-FEPADE

NOMBRE: Carlos Humberto López Linares
CARGO: Gerente de Tecnologías de Información
ÁREA: Tecnologías de Información
JEFE: Juan Carlos Pérez López

DESARROLLO PROFESIONAL

TAREA	1	2	3	4	5	6	7	8	9	10	11	12	ESP	REAL	DIF	COMENTARIO
Implementación de tecnología JAVA como Estándar de programación																
Tomar modulo I del curso para CCNA	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	-100.00	
Tomar modulo I del curso para CCNA	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	-100.00	
Tomar modulo I del curso para CCNA	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	-100.00	

Listo

1.5 Módulo de Actividades no Planificadas

1.5.1 Agregar tarea

Descripción: El empleado utiliza esta etapa Agregar tarea para registrar todas las actividades que no han sido programadas en su planificación anual. La actividad de asocia a un proyecto y una vez ha sido ingresada, es visible para el empleado y sus superiores.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Pasos

Se desarrollo en esta parte se detalla como agregar al sistema una actividad o tarea no planificada, con lo que se define en los pasos siguientes para llevarla a cabo:

1. Se inicia con la selección del personal en el botón de **Subalterno**
2. Luego, el empleado ve una lista de sus tareas no planificadas.
3. El empleado agrega una tarea o actividad no planificada. Presiona botón **Agregar Tarea**.
4. Por último el empleado puede seleccionar los botones de **Guardar cambios o Deshacer cambios**.
5. Si desea ver reportes, selecciona botón **Reporte**.



ITCA

Instituto Tecnológico Centroamericano

Sistema de Evaluación General de Desempeño - SEGD

Usuario conectado: clopez

Salir Contraseña Menú

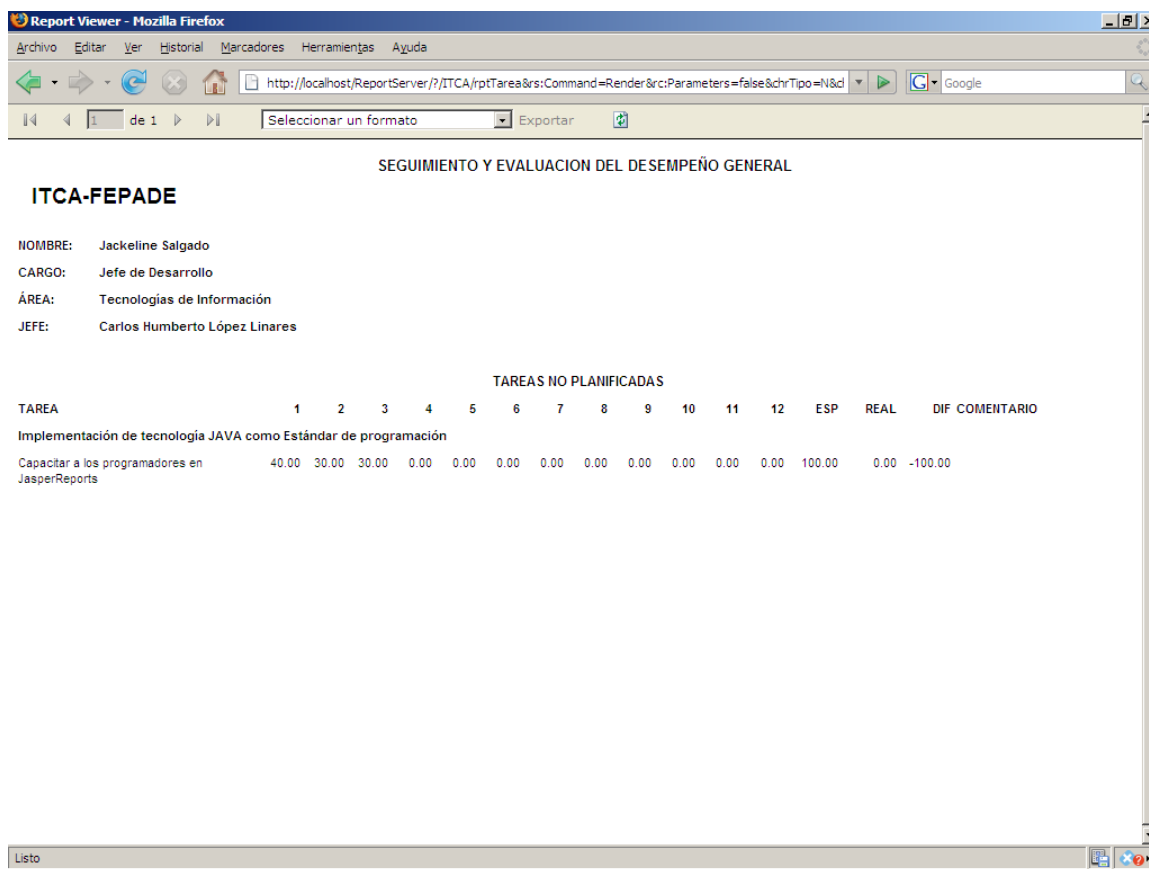
Tareas no Planificadas

Agregar Tarea no Planificada

Subalterno Jackeline Salgado - Tecnologías de Información

Tareas		Mes 1	Mes 2	Mes 3	Mes 4	Mes 5
Proyecto Implementación de tecnología JAVA como Estándar de programación	Tarea no planificada Capacitar a los programadores en JasperReports	40	30	30	0	0

1.5.2 Reporte de Actividades no planificadas.



Report Viewer - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost/ReportServer/?/ITCA/rptTarea&rs:Command=Render&rs:Parameters=false&chrTipo=N&d

Google

Seleccionar un formato Exportar

SEGUIMIENTO Y EVALUACION DEL DESEMPEÑO GENERAL

ITCA-FEPADE

NOMBRE: Jackeline Salgado
CARGO: Jefe de Desarrollo
ÁREA: Tecnologías de Información
JEFE: Carlos Humberto López Linares

TAREAS NO PLANIFICADAS

TAREA	1	2	3	4	5	6	7	8	9	10	11	12	ESP	REAL	DIF	COMENTARIO
Implementación de tecnología JAVA como Estándar de programación																
Capacitar a los programadores en JasperReports	40.00	30.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	-100.00	

Listo

1.6 Módulo de Evaluación Anual

1.6.1 Mantenimientos

1.6.1.1 Agregar cuestionarios

Descripción: El supervisor utiliza la etapa de Agregar cuestionario para ingresar al sistema las preguntas de un cuestionario nuevo. Una vez el cuestionario ha sido ingresado, es visible para todos los supervisores.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.
3. El proyecto correspondiente debe estar creado.

Pasos

Se desarrollo en esta parte se como agregar al sistema un cuestionario nuevo en la fase de Modulo de evaluación Anual, con lo que se define en los pasos siguientes para llevarla a cabo:

1. El supervisor ve una lista de los cuestionarios existentes, los cuestionarios están ordenados alfabéticamente.
2. Si adiciona preguntas utilice el botón de **Preguntas**. En el caso de omitir de ellas utilice el botón **Borrar**.
3. El supervisor agrega un nuevo cuestionario especificando su nombre. El supervisor ve la lista de preguntas en blanco del cuestionario. Presiona el botón **Agregar Cuestionario**.
4. Por último el empleado puede seleccionar los botones de **Guardar cambios o Deshacer cambios**.



1.6.1.2 Agregar preguntas

Descripción: El supervisor utiliza esta etapa de Agregar preguntas para ingresar al sistema las preguntas de un cuestionario nuevo.

Condiciones previas:

4. El supervisor debe estar registrado en el sistema.
5. El supervisor debe registrar su ingreso al sistema.
6. El proyecto correspondiente debe estar creado.

Pasos

Se desarrollo en esta parte se como agregar al sistema una pregunta en la fase de Modulo de evaluación Anual, con lo que se define en los pasos siguientes para llevarla a cabo:

1. El supervisor ve una lista de los cuestionarios existentes, los cuestionarios están ordenados alfabéticamente.
2. El supervisor decidirá en que área de evaluación se realizara los cambios, estas ya están definidas por el Instituto. Presione el botón de opción **Área de Evaluación**.
3. Si adiciona preguntas utilice el botón **Agregar Preguntas**. En el caso de omitir de ellas utilice el botón **Borrar**.
4. Por último el empleado puede seleccionar los botones de **Guardar cambios o Deshacer cambios**.

ITCA
Instituto Tecnológico Centroamericano
Sistema de Evaluación General de Desempeño - SEGD
Usuario conectado: clopez

Agregar Preguntas

Cuestionario: FRHCO-19
FRHCO-19 REVISION DEL DESEMPEÑO CARGOS INTERRELACIONADOS

Guardar Cambios Agregar Pregunta Deshacer Cambios

Área de Evaluación	Pregunta	
CAPACIDAD	Domina su área trabajo	Borrar
CAPACIDAD	Incorpora nuevos conocimientos en su trabajo	Borrar
CAPACIDAD	Es una fuente de nuevas ideas	Borrar
CAPACIDAD	Resuelve adecuadamente los problemas	Borrar
CAPACIDAD	Logra el óptimo desempeño de sus funciones	Borrar
CAPACIDAD	Revisa y mejora la efectividad de su trabajo	Borrar
CAPACIDAD	Se esmera porque sus clientes internos y externos estén satisfechos con sus servicios	Borrar
CAPACIDAD	Realiza su trabajo con calidad	Borrar
CAPACIDAD	Planifica adecuadamente su trabajo	Borrar
COMUNICACIÓN Y PARTICIPACION	Demuestra capacidad para expresar sus ideas con claridad	Borrar
COMUNICACIÓN Y PARTICIPACION	Permite el acercamiento con sus compañeros	Borrar
COMUNICACIÓN Y PARTICIPACION	Sabe escuchar a los demás	Borrar

Listo

1.6.2 Procesos

1.6.2.1 Configurar evaluación

Descripción: El supervisor utiliza esta etapa de Configuración de la evaluación para seleccionar a los empleados que contestarán los cuestionarios de evaluación de un empleado en evaluación. El supervisor solo podrá configurar evaluaciones para sus subalternos. Una vez la evaluación ha sido configurada puede dar inicio la evaluación.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.
3. Los cuestionarios de la evaluación debes estar creados.

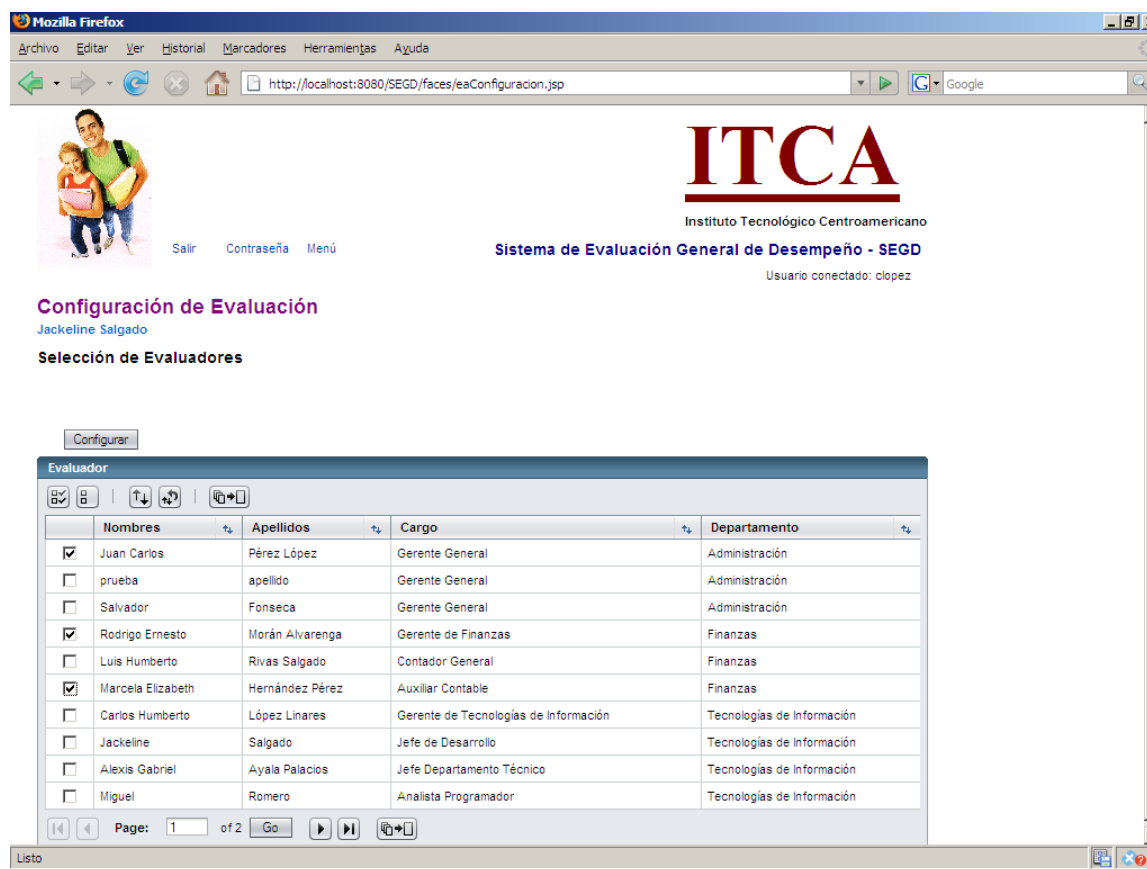
Pasos

Se desarrollo en esta parte se detalla como configurar las evaluaciones en el sistema, con lo que se define en los pasos siguientes para llevarla a cabo:

1. Primero el supervisor ve una lista de empleados a su cargo, los empleados están organizados por departamento.
2. El listado muestra los campos de **Nombre, Apellidos, Cargo y departamento**
3. El supervisor selecciona un empleado.
4. El sistema muestra una lista de cuestionarios.
5. El supervisor selecciona los cuestionarios que deberán ser contestados para la evaluación, selecciona a los empleados que contestarán los cuestionarios y define una fecha de inicio de la evaluación.

6. Por último el supervisor selecciona el botón de **Configurar**.

1.6.2.2 Selección de evaluadores.



1.6.2.3 Iniciar evaluación

Descripción: El supervisor utiliza esta etapa de Iniciar evaluación para permitir a los empleados evaluadores iniciar la evaluación del empleado en evaluación, al finalizar este proceso notifica por correo a los evaluadores que la evaluación ha iniciado.

Condiciones previas:

1. El supervisor debe estar registrado en el sistema.
2. El supervisor debe registrar su ingreso al sistema.
3. Los cuestionarios debe estar creados.
4. La configuración de la evaluación debe estar terminada.

Pasos

Se desarrollo en esta parte se detalla cómo dar inicio a contestar un cuestionario nuevo, con lo que se define en los pasos siguientes para llevarla a cabo:

1. Primero el supervisor activa la evaluación.
2. Luego se definirá sus respuestas con el botón de opción **Valoración**.
3. Por último para cerrar esta fase el usuario presiona el botón Actualizar.
4. El empleado al final utiliza esta etapa de Contestar evaluación para calcular el desempeño anual de otro empleado. Una vez el cuestionario ha sido llenado, es visible para los superiores del empleado en evaluación.

ITCA
Instituto Tecnológico Centroamericano

Sistema de Evaluación General de Desempeño - SEGD

Usuario conectado: clopez

Actualizar

Pregunta	Valoración
Domina su área trabajo	MB
Incorpora nuevos conocimientos en su trabajo	MB
Es una fuente de nuevas ideas	B
Resuelve adecuadamente los problemas	E
Logra el óptimo desempeño de sus funciones	MB
Revisa y mejora la efectividad de su trabajo	R
Se esmera porque sus clientes internos y externos estén satisfechos con sus servicios	R
Realiza su trabajo con calidad	MB
Planifica adecuadamente su trabajo	B
Demuestra capacidad para expresar sus ideas con claridad	E
Permite el acercamiento con sus compañeros	E
Sabe escuchar a los demás	MB
Participa en actividades institucionales	MB
Participa activamente en iniciativas de mejora en su área	MB
Participa activamente en las reuniones	NA
Tiene habilidad para comunicarse efectivamente con los demás	NA

Listo

1.6.2.4 Evaluaciones Pendientes.

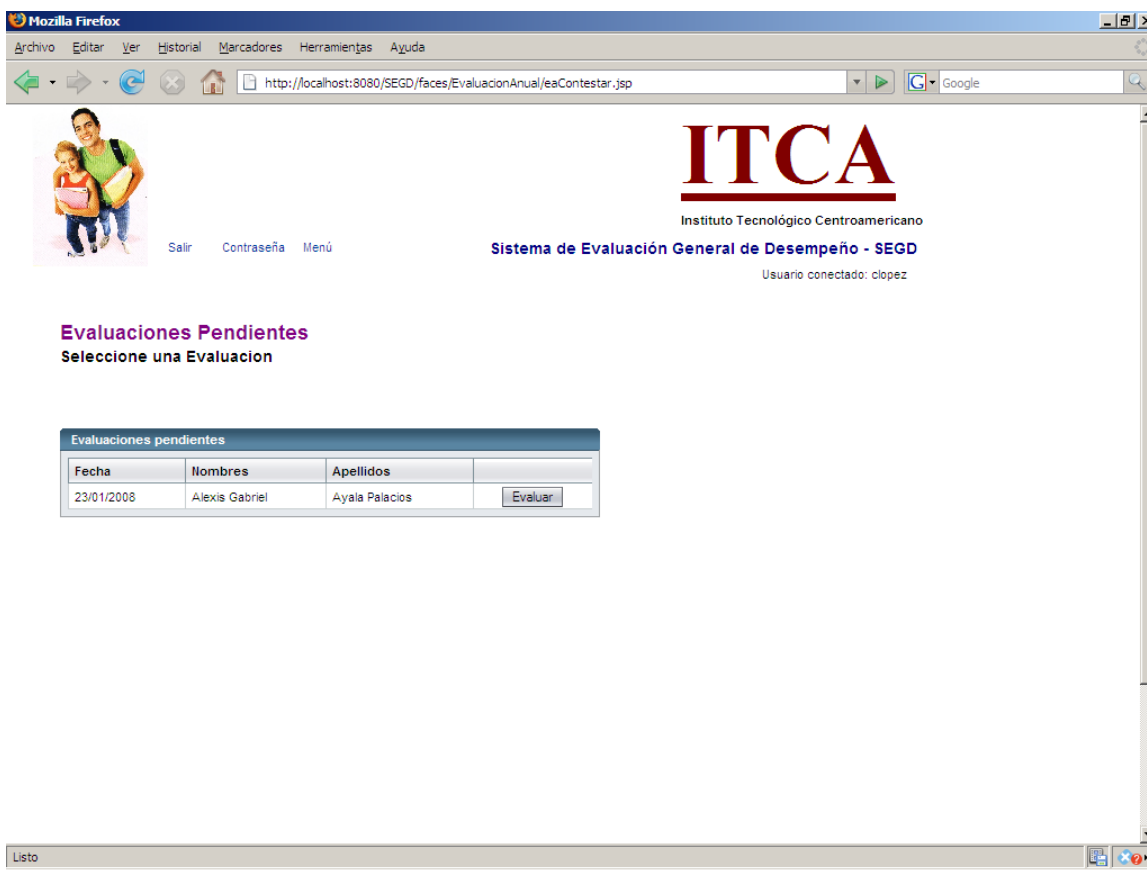
Descripción: El evaluador utiliza esta etapa de Evaluaciones Pendientes para notificar a los empleados de cuales evaluaciones no han sido iniciadas o completadas.

Condiciones previas:

1. El empleado debe estar registrado en el sistema.
2. El empleado debe registrar su ingreso al sistema.
3. El cuestionario debe estar creado.
4. La configuración de la evaluación debe estar terminada.
5. La evaluación debe estar iniciada.

Pasos

1. El supervisor ve la lista de empleados pendientes de completar las evaluaciones. Se presiona el botón de **Enviar** para notificar que están pendientes de ser completadas.



1.6.3 Reportes

1.6.3.1 Resultados

ITCA
Instituto Tecnológico Centroamericano
Sistema de Evaluación General de Desempeño - SEG D
Usuario conectado: clopez

Resultado de Evaluaciones
Evaluación Anual

Año * 2008

Fecha Evaluacion	Cuestionario	Nombres	Apellidos	
12/01/2008	FRHCO-19	Alexis Gabriel	Ayala Palacios	Reporte
23/01/2008	FRHCO-19	Alexis Gabriel	Ayala Palacios	Reporte
31/01/2008	FRHCO-19	Jackeline	Salgado	Reporte
20/02/2008	FRHCO-19	Alexis Gabriel	Ayala Palacios	Reporte
14/03/2008	FRHCO-12	Jackeline	Salgado	Reporte
14/03/2008	FRHCO-19	Alexis Gabriel	Ayala Palacios	Reporte

Listo

Report Viewer - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost/ReportServer/7/ITCA/rptEvaluacionDesempenio&rs:Command=Render&rc:Parameters=fal

Google

Seleccionar un formato Exportar

ITCA-FEPADE

FRHCO-19 REVISION DEL DESEMPEÑO CARGOS INTERRELACIONADOS

NOMBRE: Alexis Gabriel Ayala Palacios

CARGO: Jefe Departamento Técnico

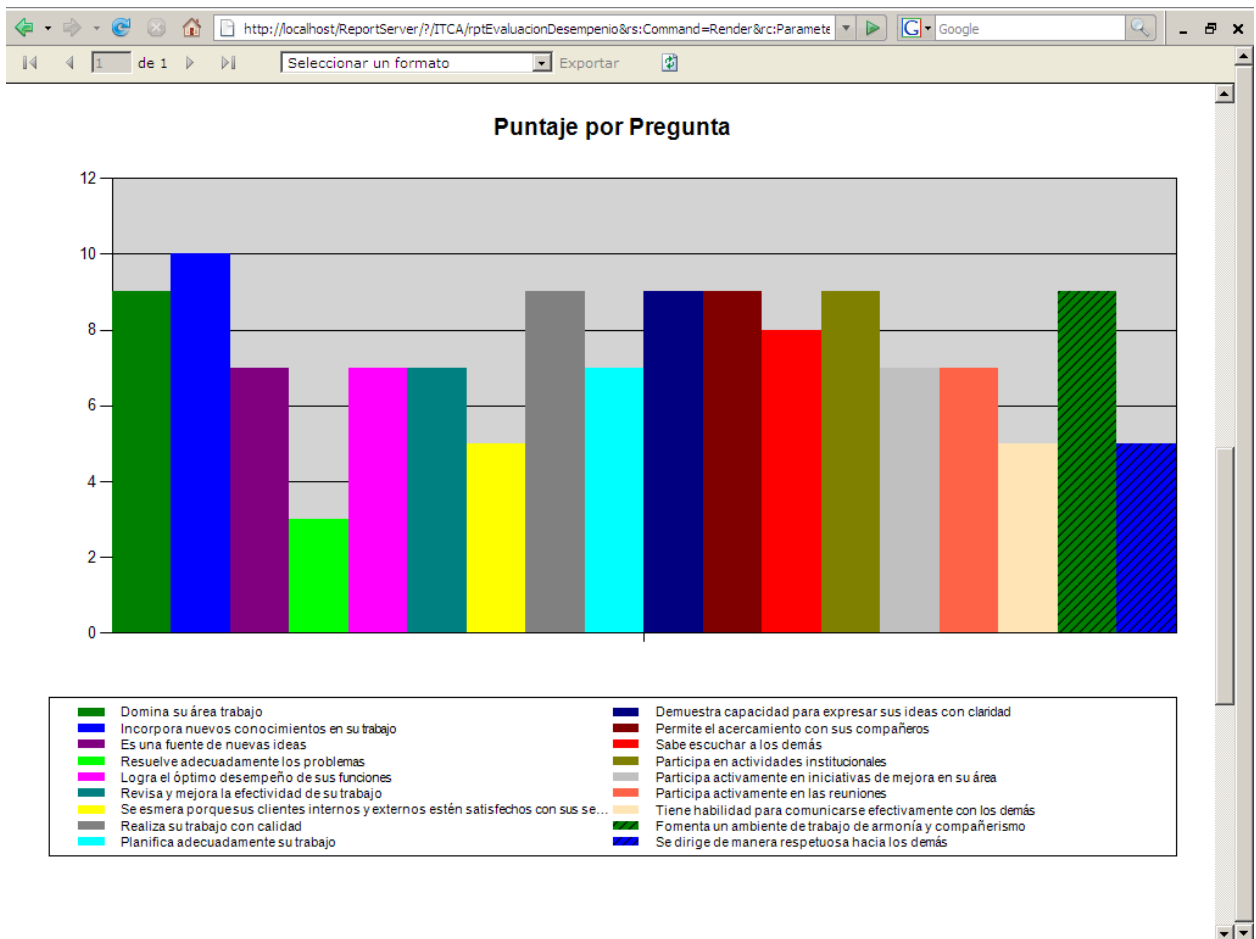
ÁREA: Tecnologías de Información

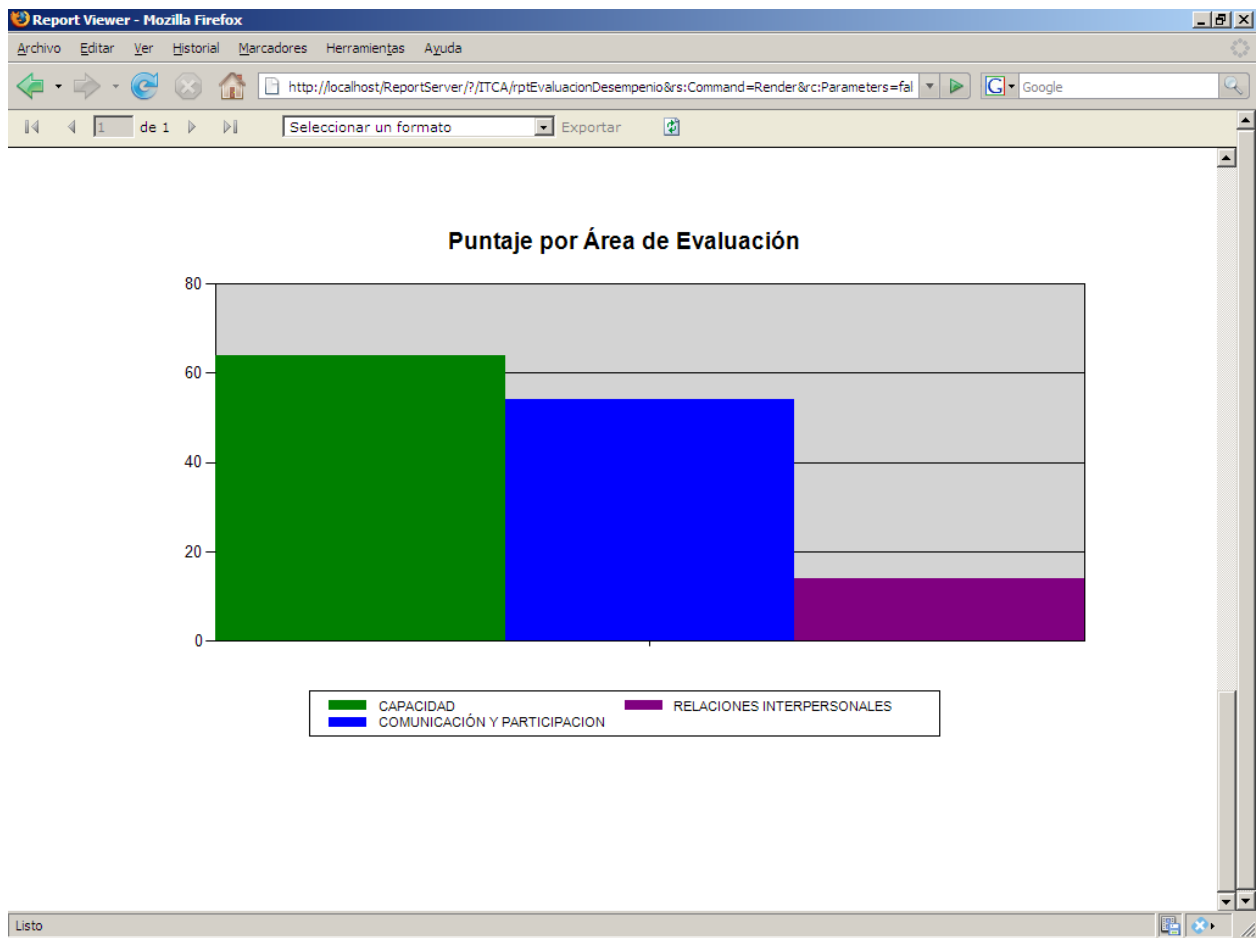
JEFE: Carlos Humberto López Linares

FECHA: 14/03/2008

PREGUNTA	VALORACION					
	0	1	2	3	4	5
CAPACIDAD	NA	NM	R	B	MB	E
Domina su área trabajo	4	0	0	0	0	0
Incorpora nuevos conocimientos en su trabajo	4	0	0	0	0	0
Es una fuente de nuevas ideas	3	0	0	0	0	0
Resuelve adecuadamente los problemas	5	0	0	0	0	0
Logra el óptimo desempeño de sus funciones	4	0	0	0	0	0
Revisa y mejora la efectividad de su trabajo	2	0	0	0	0	0
Se esmera porque sus clientes internos y externos estén satisfechos con sus servicios	2	0	0	0	0	0
Realiza su trabajo con calidad	4	0	0	0	0	0
Planifica adecuadamente su trabajo	3	0	0	0	0	0
Total CAPACIDAD	31	0	0	0	0	0
COMUNICACIÓN Y PARTICIPACION	NA	NM	R	B	MB	E
Demuestra capacidad para expresar sus ideas con claridad	5	0	0	0	0	0
Permite el acercamiento con sus compañeros	5	0	0	0	0	0
Sabe escuchar a los demás	4	0	0	0	0	0

Listo





1.7 Módulo de Seguridad

1.7.1 Agregar departamentos

Esta pantalla se encarga de administrar el catálogo de departamentos y áreas de la institución. Para entrar a esta pantalla ingresamos al menú Seguridad y seleccionamos la opción Departamentos. Una vez dentro de la pantalla el sistema nos mostrará un listado de los departamentos existentes en el sistema.



Agregar un nuevo Departamento

si queremos agregar un nuevo departamento, basta con dar un clic en el botón "Agregar Departamento", el programa creará un registro en blanco al final del listado

de departamentos, una vez ha sido llenado con el nombre del departamento damos clic en el botón “Guardar Cambios”.

Modificar un Departamento existente

Buscamos el departamento que deseamos modificar, una vez encontrado, hacemos la modificación requerida y damos un clic al botón “Guardar Cambios”.

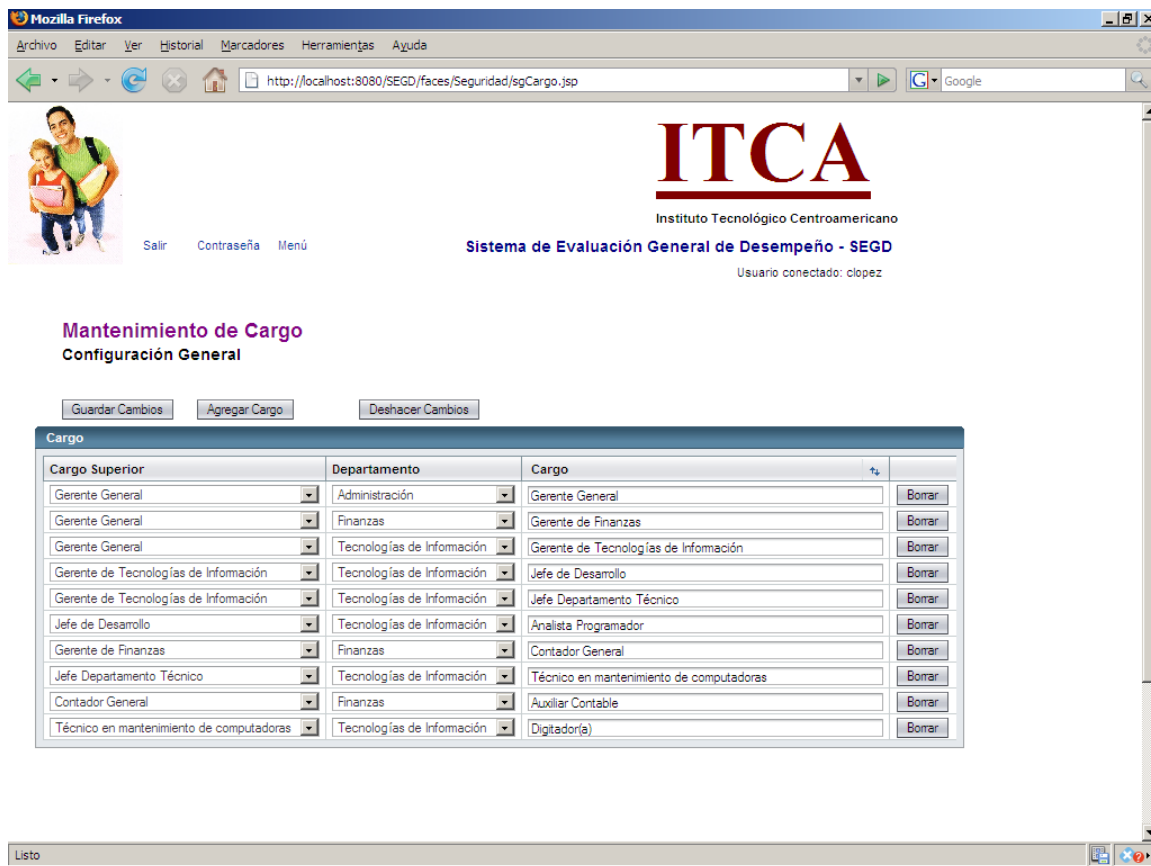
Eliminar un Departamento

Buscamos el departamento que deseamos eliminar, una vez encontrado, damos un clic al botón “Borrar” que se encuentra al lado derecho del registro.

Si en algún momento deseamos cancelar una operación damos un clic al botón “Deshacer Cambios”.

1.7.2 Agregar cargo de empleados

Esta pantalla se utiliza para configurar y administrar el catálogo de cargos de los empleados u organigrama institucional dentro del sistema. Para entrar a esta pantalla buscamos en el menú Seguridad la opción “Cargos” y la seleccionamos dando un clic. Inmediatamente el sistema nos mostrará un listado de los cargos configurados en el sistema, junto con el respectivo cargo del superior y el departamento al que pertenece.



Agregar un nuevo Cargo

Si queremos agregar un nuevo cargo, debemos dar un clic en el botón “Agregar Cargo”, el programa creará un registro en blanco al final del listado de Cargos, una vez ha sido llenado con el nombre del cargo, el departamento al que pertenece y el cargo del superior damos clic en el botón “Guardar Cambios”.

Modificar un Cargo existente

Buscamos el cargo que deseamos modificar, una vez encontrado, hacemos la modificación requerida y damos un clic al botón “Guardar Cambios”.

Eliminar un Cargo

Buscamos el cargo que deseamos eliminar, una vez encontrado, damos un clic al botón “Borrar” que se encuentra al lado derecho del registro.

Si en algún momento deseamos cancelar una operación damos un clic al botón “Deshacer Cambios”.

1.7.3 Agregar empleado.

Esta pantalla se utiliza para manejar el registro de los usuarios que tienen acceso al sistema. Para ingresar a esta pantalla lo hacemos a través del menú Seguridad seleccionado la opción empleados. En este momento el sistema nos mostrará una lista de los empleados que están registrados en el sistema.

ITCA
Instituto Tecnológico Centroamericano
Sistema de Evaluación General de Desempeño - SEGD
Usuario conectado: clopez

Mantenimiento de Empleados
Configuración General

Buscar por Login

Login	Cargo	Nombres	Apellidos	email	Estado	Tipo
ayala	Técnico en mantenimiento de computadoras	Alexis Gabriel	Ayala Palacios		Activo	E
clopez	Jefe de Desarrollo	Carlos Humberto	López Linares		Activo	S
jperez	Gerente General	Juan Carlos	Pérez López		Activo	E
jsalgado	Analista Programador	Jackeline	Salgado		Activo	E
lrivas	Auxiliar Contable	Luis Humberto	Rivas Salgado		Activo	E
mhernandez	Gerente General	Marcela Elizabeth	Hernández Pérez		Activo	E
mmorero	Gerente General	Miguel	Romero		Activo	E
ocontreras	Gerente General	Oscar Manuel	Contreras Pérez		Activo	E
prueba	Gerente General	Salvador	Fonseca	sfonseca@itca.edu.sv	Activo	E
prueba2	Gerente General	prueba	apellido		Activo	E

Page: 1 of 2 Go

Agregar un nuevo Empleado

Si queremos agregar un nuevo empleado, debemos dar un clic en el botón “Agregar Empleado”, el programa creará un registro en blanco al final del listado de Empleados, el registro en blanco debe ser llenado con los datos del empleado: nombre de usuario del sistema, cargo, nombres y apellidos, correo electrónico, estado dentro del sistema (activo o inactivo), tipo de empleado (jefe o subalterno). Una vez ha sido llenados todos los datos del empleado procedemos a dar un clic en el botón “Guardar Cambios”.

En este momento el sistema asigna una contraseña genérica al empleado (ITCA), esta contraseña deberá ser utilizada para ingresar al sistema la primera vez, ya que cuando el empleado ingrese, se detectará que es su primer ingreso y se le obligará a cambiar su contraseña por otra más segura. De todas formas al momento de crear un empleado o en cualquier otro momento se puede asignar una contraseña nueva a través del botón “Contraseña” al lado derecho de registro.

Modificar los datos de un empleado existente

Buscamos al empleado que deseamos modificar, una vez encontrado, hacemos la modificación requerida y damos un clic al botón “Guardar Cambios”.

Eliminar un empleado

Los empleados no deben eliminarse, a no ser que sea un registro creado por equivocación. Si el empleado ya no trabaja dentro de la institución, o simplemente ya no debe tener acceso al sistema, solamente debe cambiarse el estado de activo a inactivo. Esto se debe a que este empleado ya tiene un registro histórico que debe conservarse.

Si en algún momento deseamos cancelar una operación damos un clic al botón “Deshacer Cambios”.