



**UNIVERSIDAD DON BOSCO
VICERRECTORIA DE ESTUDIOS DE POSTGRADO**

**TRABAJO DE GRADUACION
DISEÑO DE UN SISTEMA DE ADMINISTRACIÓN, SUPERVISIÓN Y
CONTROL DE PROCESOS AUTOMÁTICOS BAJO UNA ARQUITECTURA
DISTRIBUIDA**

**PARA OPTAR AL GRADO DE
MAESTRO EN ARQUITECTURA DE SOFTWARE**

**ASESOR:
LUIS E. GARCIA**

**PRESENTADO POR:
JORGE ALBERTO FLORES VILLACORTA**

Antiguo Cuscatlán, La Libertad, El Salvador, Centroamérica.

Septiembre 2015

Resumen Ejecutivo

El presente documento describe el diseño de una solución de administración centralizada, monitoreo y control de procesos automáticos en una conocida compañía de transporte aéreo en El Salvador.

Los procesos automáticos son aplicaciones construidas en lenguajes Java, C# o Visual Basic y que son instalados como servicios de Windows^[1] o demonios de Linux^[2], para que realicen tareas de forma automática sin intervención directa de los usuarios. Por lo general estas aplicaciones procesan grandes cantidades de datos y no se tienen los mecanismos adecuados para una completa supervisión, consulta del estado y avance de estos procesos.

Esta falta de herramientas adecuadas para la administración y supervisión de los procesos automáticos, provoca que el área de Soporte IT de la compañía no pueda identificar o resolver de forma oportuna los problemas que se presentan cuando un proceso automático falla.

También, para la modificación de parámetros de configuración de los procesos automáticos no se cuenta con un apropiado sistema que lo facilite, ya que se requiere modificar los archivos de configuración de la aplicación y en algunos casos cambiar su código. Estos cambios en las aplicaciones y el respectivo acceso al servidor, obligatoriamente tienen que pasar por una gestión de control de cambios, en la que intervienen varias áreas del departamento de IT, para asegurarse que no vayan a existir riesgos de afectación sobre otras aplicaciones o la infraestructura misma y para llevar los respectivos registros de auditoría.

Este documento expone el diseño de un sistema de administración de procesos automáticos, que dentro de sus funcionalidades permite la supervisión y control, manejo de configuraciones y consulta de logs.

Contenido

Definición del Problema	1
Procesos Automáticos	1
Áreas de TI involucradas.....	1
Procesos Internos de Control.....	3
Características de la Solución	8
Alternativas de solución.....	9
Sistema de administración y supervisión de procesos	9
Interfaz centralizada de parámetros.....	12
Sistema de manejo de logs	12
Comparación de las alternativas.....	15
Selección de alternativas	17
Solución Propuesta.....	18
Módulos del sistema.....	18
Arquitectura de la solución	22
Vista de Componentes.....	22
Vista de Despliegue.....	29
Vista de Datos.....	36
Vista Lógica.....	39
Estándares y lineamientos	54
Desarrollo de procesos automáticos.....	54
Logging.....	57
Conclusiones.....	59
Referencias.....	61
Anexos	63
Prototipos de Interfaz Gráfica	63
Instrucciones de Instalación y Configuración	70
Instalación de Aplicación Web y Servicio de Configuraciones.....	70
Instalación y Configuración del Stack ELK.....	73

Lista de tablas

Tabla 1. Comparación de alternativas para el sistema de administración y supervisión de procesos.....	15
Tabla 2. Comparación de alternativas para la interfaz centralizada de parámetros.....	15
Tabla 3. Comparación de alternativas del sistema de manejo de logs.....	16
Tabla 4. Detalle del componente Aplicación Web	24
Tabla 5. Detalle del componente Servicio de Configuraciones	25
Tabla 6. Detalle del componente Base de Datos.....	26
Tabla 7. Detalle del componente Servicio JMS.....	26
Tabla 8. Detalle del componente Proceso Supervisor	27
Tabla 9. Detalle del componente Stack ELK.....	28
Tabla 10. Detalle del componente Logstash Forwarder	28
Tabla 11. Detalle del nodo Servidor de Base de Datos.....	31
Tabla 12. Detalle del nodo Servidor de Aplicaciones 1.....	32
Tabla 13. Detalle del nodo Servidor de logs.....	33
Tabla 14. Detalle del nodo Servidor Windows.....	34
Tabla 15. Detalle del nodo Servidor Linux.....	35
Tabla 16. Definición de la tabla SAS_APPLICATIONS	37
Tabla 17. Definición de la tabla SAS_PARAMETERS	37
Tabla 18. Definición de la tabla SAS_CHANGES.....	38
Tabla 19. Definición de la tabla SAS_MOD_REQUESTS	38
Tabla 20. Definición de la tabla SAS_REQ_APPROVALS	38
Tabla 21. Estructura de paquetes y clases de Aplicación Web.....	39
Tabla 22. Estructura de paquetes y clases de Servicio de Configuraciones	40
Tabla 23. Estructura de paquetes y clases de Supervisor de Procesos.....	40
Tabla 24. Descripción de clases del paquete Modelo.Entity de Aplicación Web	41
Tabla 25. Descripción de clases del paquete Modelo.Repository de Aplicación Web.....	42
Tabla 26. Descripción de clases del paquete Logica de Aplicación Web	43

Tabla 27. Descripción de clases del paquete Presentacion.AdministracionProcesos de Aplicación Web	44
Tabla 28. Descripción de clases del paquete Presentacion.SupervisionControl de Aplicación Web	44
Tabla 29. Descripción de clases del paquete Presentacion.ConsultaLogs de Aplicación Web	45
Tabla 30. Descripción de clases del paquete Presentacion.Auditoria de Aplicación Web	45
Tabla 31. Descripción de clases del paquete Modelo.Entity del Servicio de Configuraciones	45
Tabla 32. Descripción de clases del paquete Modelo.Repository del Servicio de Configuraciones	46
Tabla 33. Descripción de clases del paquete Logica del Servicio de Configuraciones	46
Tabla 34. Descripción de clases del paquete Servicio del Servicio de Configuraciones..	46
Tabla 35. Descripción de clases del paquete Integracion del Proceso Supervisor	47
Tabla 36. Descripción de clases del paquete Logica del Proceso Supervisor	47
Tabla 37. Descripción de clases del paquete Aplicacion del Proceso Supervisor	47

Lista de figuras

Figura 1. Diagrama de componentes	23
Figura 2. Diagrama de despliegue	30
Figura 3. Diagrama Entidad-Relación	36
Figura 4. Aplicación Web: paquetes.....	48
Figura 5. Aplicación Web: paquete Modelo.....	48
Figura 6. Aplicación Web: clases en el sub-paquete Entity.....	49
Figura 7. Aplicación Web: clases en el sub-paquete Repository.....	50
Figura 8. Aplicación Web: clases en el paquete Lógica	50
Figura 9. Aplicación Web: paquete Presentación.....	51
Figura 10. Servicio de Configuraciones: paquetes	51
Figura 11. Servicio de Configuraciones: clases en el paquete Modelo	52
Figura 12. Supervisor de Procesos: paquetes.....	52
Figura 13. Supervisor de Procesos: clases	53
Figura 14. Administración de Procesos: Pantalla inicial	63
Figura 15. Administración de Procesos: Pantalla de detalle.....	64
Figura 16. Administración de Procesos: Cambio de parámetros.....	64
Figura 17. Administración de Procesos: Registro de nuevo proceso.....	65
Figura 18. Administración de Procesos: Nuevo parámetro	65
Figura 19. Administración de Procesos: Edición de proceso	66
Figura 20. Administración de Procesos: Edición de parámetro.....	66
Figura 21. Supervisión y Control: Pantalla inicial.....	67
Figura 22. Supervisión y Control: Monitor de proceso	67
Figura 23. Consulta de logs: Pantalla inicial	68
Figura 24. Consulta de Logs: Pantalla de visualización	68
Figura 25. Auditoria: Pantalla inicial.....	69
Figura 26. Auditoria: Reporte	69
Figura 27. Sección "Runtime" de JBoss EAP.....	71
Figura 28. Ventana de seleccion de archivo WAR.....	71

Figura 29. Ventana de confirmación del archivo WAR	72
Figura 30. Configuración de la aplicación web	72

Definición del Problema

Procesos Automáticos

Dentro de una conocida compañía de transporte aéreo en El Salvador, existen muchos procesos principalmente de extracción, transformación, sincronización y cálculo que debido al alto volumen de datos, se realizan con aplicaciones que funcionan de forma automática sin intervención directa de los usuarios; estas aplicaciones son conocidas dentro de esta compañía como procesos automáticos.

Los procesos automáticos son desarrolladas en los lenguajes Java ^[3], C# o Visual Basic y son instalados para funcionar como servicios (servicios de Windows o demonios de Linux), en servidores dedicados a este fin. En la configuración de estas aplicaciones existen parámetros predefinidos para realizar sus tareas de procesamiento, parámetros como direcciones de servidores de archivos y bases de datos donde se tienen que conectar, factores numéricos de cálculo, las horas en que deben iniciar sus tareas, etc. por lo general son incluidos en archivos de configuración o en algunos casos como variables en el código mismo.

Áreas de TI involucradas

En esta compañía, los procesos automáticos son desarrollados, ejecutados y administrados bajo la responsabilidad del departamento de TI, ya que las características tecnológicas de los mismos requieren de la experticia de este departamento para su adecuado funcionamiento.

Dentro del departamento de TI de esta compañía, existe una clara segregación de funciones repartida entre varias áreas y en lo que respecta a los procesos automáticos, las áreas que intervienen son:

- Arquitectura y Desarrollo
- Aseguramiento de la Calidad
- Implementación
- Soporte
- Operaciones

Las responsabilidades de cada área se detallan a continuación:

- Arquitectura y Desarrollo, es el área encargada de diseñar y desarrollar estas aplicaciones.
- Aseguramiento de la Calidad, se encarga de probar los procesos automáticos antes de su puesta en producción y asegurarse de que éstos cumplan con los atributos de calidad acordados durante la etapa de diseño.
- Implementación, gestiona el paso a producción de los procesos automáticos, cumpliendo con una serie de controles y validaciones existentes en el departamento de TI.
- Operaciones, dentro de esta área existen varios equipos encargados de la infraestructura de la compañía (telecomunicaciones y redes, bases de datos, servidores, etc) y son los únicos que tienen los permisos de acceso para instalar los procesos automáticos en los servidores.

- Soporte, tiene la responsabilidad de identificar problemas y/o resolver los incidentes relacionados a los procesos automáticos que reportan las áreas usuarias o de negocio.

Procesos Internos de Control

Al igual que todas las aplicaciones en la compañía, el paso a producción de nuevos procesos automáticos o las modificaciones de éstos, se someten a un control definido por el proceso de Gestión de Cambios del departamento de TI, el cual está basado en las buenas prácticas de ITIL^[4] (Information Technology Infrastructure Library) y el cual también se encuentra certificado bajo la norma ISO 20000^[5].

De acuerdo a ITIL, el objetivo primordial de la Gestión de Cambios es que se realicen e implementen adecuadamente todos los cambios necesarios en la infraestructura y servicios TI garantizando el seguimiento de procedimientos estándar^[6].

También, según ITIL, la Gestión de Cambios debe trabajar para asegurar que los cambios:

- Están justificados.
- Se llevan a cabo sin perjuicio de la calidad del servicio TI,
- Están convenientemente registrados, clasificados y documentados,
- Han sido cuidadosamente testeados en un entorno de prueba,
- Se ven reflejados en la CMDB,
- Pueden deshacerse mediante planes de "retirada del cambio" (back-outs) en caso de un incorrecto funcionamiento tras su implementación^[7].

Debido a estos procesos de control, los cambios en los procesos automáticos son revisados bajo los criterios anteriores por un comité conformado por varias áreas de TI antes de ser ejecutados.

Problemas

En el estudio de la situación se han identificado los siguientes problemas:

- a) Supervisión y control de los procesos.
- b) Dificultad de cambio en los parámetros.
- c) Dificultad de acceso a los logs.
- d) Poca agilidad para resolver incidentes.
- e) No existe un estándar de Procesos Automáticos.

a) Supervisión y control de los procesos

No existe un mecanismo que permita ver desde un único punto, todos los servicios de Windows o demonios de Linux que están ejecutando procesos automáticos, ni tampoco se cuenta con una interfaz para conocer el estado de estos servicios, esto es ver si están corriendo o detenidos, de una forma centralizada y sin necesidad de recurrir al área de Operaciones para que realice esta actividad desde las mismas herramientas del sistema operativo; mucho menos es posible conocer el estado o avance de los procesos que ejecutan estos servicios, ya que ni existen un sistema que lo permita ni las aplicaciones han sido desarrolladas con la funcionalidad de informar sobre su avance.

Otro inconveniente de no contar con una plataforma de supervisión de procesos automáticos se presenta cuando se requiere reiniciar un servicio o detenerlo en el caso de que un proceso automático esté fallando o simplemente cuando es necesario que un proceso detenga sus tareas, forzosamente hay que solicitar un cambio de emergencia y seguir el proceso de Gestión de Cambios y el respectivo flujo de aprobación, para que el área de Operaciones se encargue de esta actividad.

b) Cambios de parámetros

El cambio de parámetros de forma sencilla y desde la misma aplicación, en los procesos automáticos es una funcionalidad inexistente, ya que para el desarrollo de estas aplicaciones no se contempla la creación de interfaces de configuración que permita modificar estos parámetros directamente desde la aplicación y sin detener el servicio que la está ejecutando.

Tampoco existe una interfaz centralizada para administrar los parámetros de todos los procesos automáticos.

Cuando es necesario realizar un cambio en los parámetros del proceso automático, por ejemplo, cuando se requiere cambiar una hora, algún factor numérico, una IP, etc. por lo general se tiene que modificar el archivo de configuración y en algunos casos, incluso, el código de la aplicación. De cualquier forma, un cambio siempre implica detener el servicio, conocer cómo realizar el cambio, modificar el parámetro, subir la aplicación modificada al servidor y reiniciar el servicio.

Debido a que esto es una actividad donde se afecta directamente a la infraestructura, ya que se accede directamente al sistema operativo y al sistema de archivos del servidor, el área de Soporte debe hacer una solicitud de cambio y someter la misma al proceso de Gestión de Cambios, y solo si se aprueba esta solicitud, el área de Operaciones instala la aplicación modificada en el servidor.

c) Dificultad de acceso a los logs

Los logs son la primera fuente de información del área de Soporte para la identificar las causas de problemas o errores en los procesos automáticos^[8], sin embargo, no se tienen interfaces para acceder y consultar, de forma directa, los mensajes de log que generan los procesos automáticos.

Para acceder a los archivos de logs, se debe hacer una solicitud al área de Operaciones para que accedan al sistema de archivos del servidor y los extraigan. Esta actividad en ocasiones puede tomar mucho tiempo, ya que se depende de la disponibilidad de tiempo del área de Operaciones para atender estas solicitudes y del nivel de prioridad que se le dé a la solicitud.

d) Poca agilidad para resolver incidentes.

Debido a que consultar los archivos de log para determinar las causas de errores, cambiar parámetros modificando los archivos de configuración o código fuente y seguir todo el flujo de validaciones y aprobaciones de la Gestión de Cambios para desplegar nuevamente el proceso automático, es un proceso que puede tomar bastante tiempo, el

área de Soporte no puede brindar una respuesta o solución oportuna a las áreas de negocio afectadas. Debido a esta razón, los usuarios perciben que el servicio que presta el departamento de IT no es el mejor y esto se agudiza más cuando son procesos críticos donde puede haber pérdidas económicas o insatisfacción de clientes.

Sumado a esto, también la carga de trabajo que normalmente es grande en el área de Operaciones, influye también en que no se pueda dar una solución rápida. Esta carga de trabajo se ve aumentada cuando las aplicaciones como los procesos automáticos no cuentan con interfaces para el área de Soporte pueda supervisar el estado, acceder a los parámetros y consultar logs.

e) No existe un estándar de Procesos Automáticos.

Una de las causas principales de los problemas anteriormente descritos, es la falta de una arquitectura definida para los procesos automáticos que contemple los componentes necesarios para solucionar estos problemas; tampoco se cuenta con estándares y lineamientos que dictaminen como deben ser desarrollados los procesos automáticos y que funcionalidades deben incluirse para permitir una mejor administración y mantenimiento de los mismos.

Características de la Solución

El diseño de la solución a los problemas identificados con el actual manejo de los procesos automáticos, debe poseer las siguientes características:

- Debe contar con una interfaz que permita al área de Soporte supervisar los servicios de Windows o demonios de Linux que ejecutan procesos automáticos, en los distintos servidores destinados a este fin, de forma que sea posible conocer su estado.
- La interfaz de supervisión también debe permitir el control de los servicios de Windows y demonios de Linux, de tal manera que se puedan iniciar y detener sin tener que acceder a las herramientas de administración local del servidor.
- El diseño debe incluir también, una interfaz que permita cambiar parámetros de configuración en los procesos automáticos, sin tener que modificar su código o archivos de configuración e incluso sin tener que detenerlos.
- La solución diseñada debe integrar herramientas automatizadas para la recolección, almacenamiento e indexación de logs generados por los procesos automáticos, de tal forma que esto permita hacer consultas sobre un único repositorio central.

Alternativas de solución

Sistema de administración y supervisión de procesos

Para esta alternativa de solución se contemplan 2 opciones:

- a) Implementación de un sistema comercial existente
- b) Sistema desarrollado a la medida

Las ventajas y desventajas de estas opciones se analizan a continuación:

a) Implementación de un sistema comercial existente

Esta opción es la de adquirir e implementar un sistema de supervisión y administración de red, servicios y procesos del sistema operativo, con el fin de tener mayor control y visibilidad de los servicios Windows y demonios Linux que ejecutan procesos automáticos.

En el mercado existen sistemas propietarios muy conocidos como Server & Application Monitor de SolarWinds y sistemas de código abierto como Nagios XI, que permiten entre muchas otras cosas, monitorear servicios de Windows o demonios de Linux^[9].

Entre las funcionalidades que ofrecen estas herramientas, se pueden listar las siguientes:

- Interfaz web para acceder a varios servidores de forma remota.
- Visualización de rendimiento del servidor (uso de memoria, cpu, disco, etc).
- Auditoría de eventos del sistema operativo.
- Monitoreo de hardware de red.

- Monitoreo de servicios y procesos.
- Programar tareas.
- Alertas y notificaciones.
- Reportes.

A continuación se detallan las ventajas y desventajas de adquirir e implementar un sistema de este tipo:

Ventajas:

- Tiempo de implementación más cortos.
- Soporte y entrenamiento por parte de un proveedor.

Desventajas:

- No es posible supervisar el estado y avance de los procesos automáticos, ya que el monitoreo es a nivel del servicio o demonio.
- Estos sistemas brindan otras funcionalidades de administración del sistema operativo que no son necesitadas por el área de Soporte, lo que significa una inversión injustificada para la compañía. También, de contar con estas características, existiría el riesgo de que en algún momento hubiera intervención con las responsabilidades del área de Operaciones.

b) Sistema desarrollado a la medida

Otra alternativa para solucionar el problema de supervisión de procesos automáticos, es la de diseñar y desarrollar un sistema a la medida para supervisar únicamente aquellos servicios que ejecutan procesos automáticos.

El sistema debe ser diseñado para contar principalmente con las siguientes funcionalidades:

- **Administración.** Debe permitir registrar nuevos procesos automáticos, dar de baja procesos automáticos que ya no se ejecutan y mantener un inventario actualizado de todas estas aplicaciones.
- **Supervisión.** Mostrar el estado únicamente de los servicios de Windows o demonios de Linux que ejecutan procesos automáticos. También, debe permitir visualizar el estado y avance de las tareas que estas aplicaciones realizan; para esto se debe contar con una interfaz que habilite a los procesos automáticos para reportar este detalle de ejecución.
- **Control.** El sistema debe ser capaz de iniciar y detener los servicios de Windows o demonios de Linux que ejecutan los procesos automáticos registrados.

Desarrollar e implementar un sistema a la medida presenta las ventajas y desventajas siguientes:

Ventajas:

- Se contaría con todas las funcionalidades específicas que son requeridas por el área de Soporte.
- Se respetaría la actual segregación de funciones y responsabilidades, ya que el sistema únicamente daría acceso y control sobre procesos automáticos.

Desventajas:

- Mayor tiempo de espera antes de poder implementar y contar con las funcionalidades esperadas.

Interfaz centralizada de parámetros

No existe una solución comercial que cumpla con esta función específica, ya que esto requiere definir una interfaz única para todas las aplicaciones y que éstas sean desarrolladas para utilizar dicha interfaz, de manera que la única opción para contar con esta característica es construir una solución a la medida.

El diseño de la interfaz centralizada para cambio de parámetros debe contemplar lo siguiente:

- Interfaz de acceso a un único repositorio de parámetros, para que los procesos automáticos carguen sus configuraciones.
- Módulo de administración centralizada de parámetros, que permita realizar cambios en éstos, sin necesidad de detener los procesos automáticos.

Sistema de manejo de logs

Para esta alternativa también se analizan las opciones de adquirir e implementar un sistema comercial existente y la opción de desarrollar un sistema a la medida.

a) Implementación de sistema comercial existente

En el mercado existen varias soluciones, tanto propietarias como de código abierto, para el manejo de logs. Algunos de estos sistemas funcionan únicamente bajo el

estándar de facto, syslog y otros cuentan con más opciones como la lectura de logs desde archivos de texto^[10].

Algunos de los sistemas de manejo de log, que se pueden encontrar en el mercado se tienen:

- SolarWinds Log & Event Manager.
- NetIQ Sentinel Log Manager.
- Nagios Log Server.
- Stack ElasticSearch, Logstash y Kibana.

SolarWinds LEM y NetIQ SLM brindan herramientas más orientadas a log de eventos del sistema operativo, servicios, hardware de red, etc. utilizando syslog principalmente. El stack ELK (ElasticSearch, Logstash y Kibana)^[11] brinda más flexibilidad en cuanto a las fuentes de log, ya que permite configurar archivos de texto como origen y cuenta con una interfaz de servicios RESTful^[12] para acceder al repositorio indexado de logs. Por último, Nagios Log Server que usa como base el stack ELK, añade principalmente funcionalidades de alerta y respaldo automatizado.

Las ventajas de adquirir e implementar un sistema para el manejo de logs se detallan a continuación:

Ventajas:

- Rápida implementación y puesta en marcha del sistema.
- Sistemas optimizados para la gestión de logs que cuentan con una base de código bastante madura.

- Soporte especializado por parte de un proveedor.

Desventajas

- Dependiendo del tipo de licenciamiento ofrecido, los costos pueden llegar a ser altos con algunas soluciones propietarias.

b) Sistema desarrollado a la medida

Existe siempre la alternativa de invertir recursos en desarrollar una solución en casa para el manejo de logs, sin embargo esta opción no tendría mucho sentido, ya que se pueden adquirir muchos sistemas especializados, optimizados y con mayor capacidad para el manejo de logs y que incluso son de código abierto y que se pueden usar sin costos de licenciamiento como el stack ELK.

En el contexto de esta situación, las ventajas de esta alternativa son casi inexistentes y se identifican las siguientes desventajas:

- Mayor tiempo para la implementación en producción.
- El tiempo y costo de desarrollo podrían ser muy altos en comparación a implementar un sistema con bajo o ningún costo de licencia.
- Lograr el grado de optimización con la que cuentan soluciones ampliamente usadas a nivel mundial, podría requerir la inversión de muchos recursos.

Comparación de las alternativas

A continuación se presenta un cuadro comparativo entre las alternativas disponibles para cada uno de los componentes que son requeridos por la solución.

Tabla 1. Comparación de alternativas para el sistema de administración y supervisión de procesos

Sistema de administración y supervisión de procesos		
Criterio	Sistema comercial existente	Sistema desarrollado a la medida
Funcionalidad	Evaluación	
Supervisión de estado del servicio/demonio	Si	Si
Supervisión de estado de las tareas que ejecutan los procesos automáticos.	No	Si
Iniciar y detener servicios/demonios	Si	Si
Implementación	Evaluación	
Tiempo necesario para la puesta en marcha	Menor	Mayor
Capacitación/Entrenamiento	Si	Si
Costos	Evaluación	
Licenciamiento	Si	No
Soporte	Si	Si (Menor)
Desarrollo	No	Si

Tabla 2. Comparación de alternativas para la interfaz centralizada de parámetros

Interfaz centralizada de parámetros		
Criterio	Sistema comercial existente	Sistema desarrollado a la medida
Funcionalidad	Evaluación	
Repositorio central de parámetros de procesos automáticos	N/A	Si
Interfaz de administración de parámetros		Si
Interfaz de acceso a los parámetros, que pueda ser utilizada por los procesos automáticos		Si
Implementación	Evaluación	
Tiempo necesario para la puesta en marcha	N/A	Mayor
Costos	Evaluación	
Licenciamiento	N/A	No
Soporte		Si
Desarrollo		Si

Tabla 3. Comparación de alternativas del sistema de manejo de logs

Sistema de manejo de logs		
Criterio	Sistema comercial existente	Sistema desarrollado a la medida
Funcionalidad	Evaluación	
Recopilación de archivos de log desde distintos servidores.	Si	Si
Almacenamiento e indexado de logs	Si	Si
Consulta de logs desde una interfaz central.	Si	Si
Atributos de calidad	Evaluación	
Rendimiento óptimo	Si	Depende del diseño, desarrollo y pruebas.
Desarrollo	Evaluación	
Alta complejidad	N/A	Si
Implementación	Evaluación	
Tiempo necesario para la puesta en marcha	Menor	Mayor
Capacitación/Entrenamiento	Si	Si
Costos	Evaluación	
Licenciamiento	Si (propietario) No (open source)	No
Soporte	Si	Si (Menor)
Desarrollo	No	Si

Selección de alternativas

De acuerdo la evaluación de criterios y análisis de ventajas y desventajas de cada sub-sistema o componente planteado como alternativa de solución, se seleccionaron las siguientes opciones:

Sistema de administración y supervisión de procesos.

Opción seleccionada: Desarrollo a la medida.

Justificación: Es la opción más conveniente, ya que se asegura que se cuente con las funcionalidades específicas para la supervisión y control de procesos automáticos, por lo que la inversión de recursos se hace únicamente sobre estas características.

Interfaz centralizada de parámetros.

Opción seleccionada: Desarrollo a la medida.

Justificación: Debido a que no existe en el mercado una solución que solvente esta necesidad específica que tiene la compañía, la única alternativa es un desarrollo a la medida que incluya todas las funcionalidades buscadas por la solución.

Sistema de manejo de logs.

Opción seleccionada: Implementación de sistema comercial existente.

Justificación: En el mercado existen varias soluciones para el manejo de logs, algunas de las cuales se caracterizan por hacer un buen manejo de los recursos tecnológicos y de tener un rendimiento óptimo. Muchos de estos sistemas son desarrollados bajo un esquema de código abierto, por lo que no se tiene que invertir en adquisición de licencias, únicamente en el soporte y entrenamiento respectivo.

Solución Propuesta

Se propone a nivel de diseño, un sistema que brinde al área de Soporte IT de la compañía de transporte aéreo, las interfaces necesarias para supervisar y controlar los procesos automáticos; permitiéndole a esta área contar con los recursos tecnológicos para resolver los problemas que surgen con estas aplicaciones, y así brindar un mejor servicio a sus usuarios. Este sistema propuesto se denomina: Sistema de Administración, Supervisión y Control de Procesos Automáticos.

Aparte de la solución tecnológica, se propone la definición de un estándar para el desarrollo de procesos automáticos y los lineamientos que se deberán seguir para que las aplicaciones de este tipo puedan ser integradas al nuevo esquema de funcionamiento.

Módulos del sistema

La solución contará con un sistema compuesto por los siguientes módulos:

- a) Administración de Procesos.
- b) Supervisión y Control.
- c) Consulta de logs.
- d) Auditoría.

Estos módulos serán las interfaces con las que el área de Soporte IT interactuará con todos los componentes de la solución.

a) Administración de Procesos.

El módulo de Administración de procesos permitirá consultar el estado de todos los procesos automáticos registrados; en este módulo se podrá visualizar si están corriendo o están detenidos y supervisar el avance de sus tareas mediante los reportes que los mismos procesos automáticos envíen al sistema.

Este módulo también facilitará la modificación de parámetros de configuración de los procesos automáticos, de tal forma que no sea necesario cambiar el código de las aplicaciones o sus archivos de configuración y tampoco se tenga que detener el servicio de Windows o demonio de Linux para aplicar estos cambios. Para esto, los procesos automáticos accederán a un servicio web donde consultaran sus parámetros de configuración antes de iniciar sus tareas.

Todos estos parámetros existirán en un repositorio central, al cual accederán los procesos automáticos a través del servicio de configuraciones para que carguen sus configuraciones; esto evitará que datos sensibles como credenciales de acceso a bases de datos, servidores FTP, etc. sean escritos en archivos de configuración y dispersos en distintas ubicaciones.

Administrar los parámetros de esta forma ayudará al área de Soporte IT a resolver de forma más rápida y oportuna las solicitudes de estos cambios y reducirá la intervención del área de Operaciones para estas modificaciones menores.

b) Supervisión y Control.

Mediante los componentes e interfaces del módulo de Supervisión y control, será posible consultar el estado de los servicios de Windows y demonios de Linux que ejecutan procesos automáticos, así como también, detener o iniciar los servicios o demonios.

La interacción de este módulo con los procesos automáticos será posible con la implementación de un proceso supervisor en cada servidor, que sirva como intermediario para recibir comandos y ejecutarlos localmente.

La comunicación entre este módulo de Supervisión y Control y el Proceso Supervisor, se realizará utilizando un servicio de cola de mensajes.

c) Consulta de logs

Para el módulo de Consulta de logs, se integrará el sistema ELK para el manejo de logs; este sistema está conformado por los servicios Elasticsearch, Logstash y Kibana.

Este sistema recolectará los archivos de log que generan los procesos automáticos y los enviará a un servidor central que actuará como indexador, en donde todos estos logs estarán almacenados y desde donde podrán ser consultados por el Sistema de Administración, Supervisión y Control de Procesos Automáticos.

d) Auditoría

Este módulo permitirá consultar y generar reportes de los cambios que se hagan en los procesos automáticos, pudiendo visualizar los usuarios responsables, los valores modificados y las fechas en que se han efectuado estos cambios.

La funcionalidad brindada por este módulo permitirá apegarse a los lineamientos de control y trazabilidad del departamento de TI de la compañía.

Arquitectura de la solución

Vista de Componentes.

Esta vista muestra los componentes que conforman la solución, el detalle de sus características y las relaciones existentes entre cada uno.

Los componentes del Sistema de Administración, Supervisión y Control de Procesos Automáticos son los siguientes:

1. Aplicación Web
2. Servicio de Configuraciones
3. Base de datos del sistema
4. Servicio JMS
5. Proceso Supervisor
6. Stack ELK
7. Logstash Forwarder

La tecnología de desarrollo estará basada principalmente en el lenguaje Java; esto permitirá que la solución sea multiplataforma^[13]. Esta característica es necesaria debido a que la compañía cuenta con una infraestructura heterogénea, conformada principalmente por servidores de aplicaciones en Windows Server y Red Hat Enterprise Linux.

Diagrama de componentes.

El siguiente diagrama muestra todos los componentes que conforman la solución completa y la relación que se da entre cada uno de ellos.

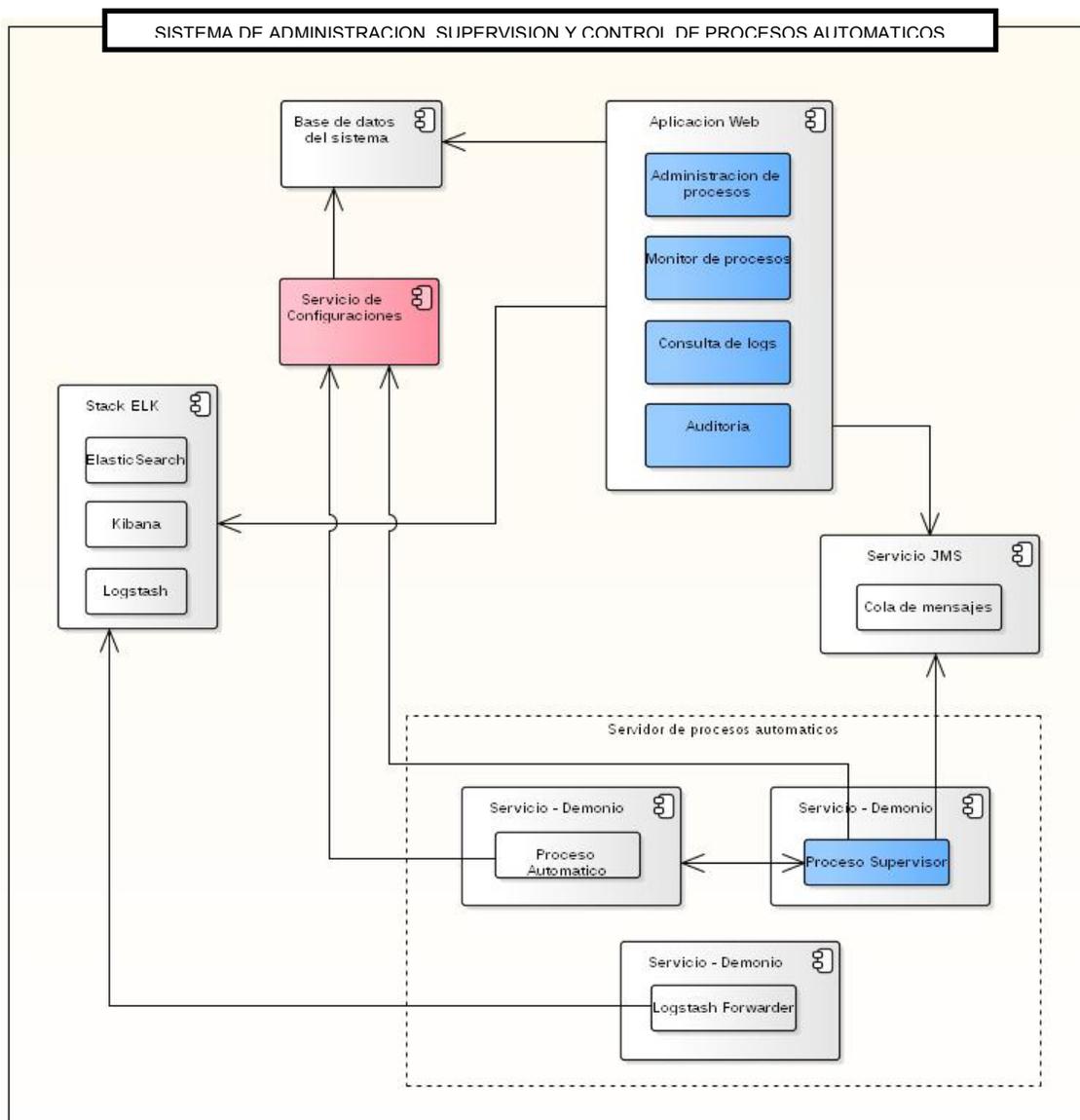


Figura 1. Diagrama de componentes

Detalle de componentes.

A continuación se presenta un resumen con el detalle de cada uno de los componentes y/o sub-sistemas de la solución.

Tabla 4. Detalle del componente Aplicación Web

Componente	Aplicación Web	
Descripción	<p>Este componente es la interfaz que permite a los usuarios interactuar con los demás componentes de todo el sistema.</p> <p>Los mensajes del módulo Monitor de Procesos que este componente distribuye por medio de <i>Servicio JMS</i> a <i>Proceso Supervisor</i> son:</p> <ul style="list-style-type: none"> • Brindar estado servicio/demonio. • Iniciar servicio/demonio. • Detener servicio/demonio. • Brindar estado proceso automático. <p>Los mensajes de respuesta que recibe de <i>Proceso Supervisor</i> son:</p> <ul style="list-style-type: none"> • Estado del servicio/demonio. • Estado del proceso automático. 	
Modelo arquitectónico		
Tipo	Aplicación web	
Patrón	MVC (Modelo-Vista-Controlador)	
Frameworks	Java SE 7, Spring MVC, Hibernate	
Lenguaje	Java	
Interfaces	Componente	Modo de comunicación
	Base de datos	DBMS Driver
	Servicio JMS	JMS ^[18]
	ElasticSearch	RESTful
Sub-componentes	<ul style="list-style-type: none"> • Administración de procesos • Monitor de procesos • Consulta de logs • Auditoría 	

Tabla 5. Detalle del componente Servicio de Configuraciones

Componente	Servicio de Configuraciones	
Descripción	<p>A través de este servicio web los procesos automáticos acceden a sus parámetros de configuración, los cuales cargan en memoria antes de ejecutar sus tareas.</p> <p>Utilizar este servicio web evita que los procesos automáticos guarden sus parámetros de configuración de forma local en archivos o en el código fuente; esto permite poder realizar cambios de forma remota y sin tener que manipular directamente ni detener los procesos automáticos.</p> <p>Debido a que este servicio web utiliza el protocolo estándar SOAP, puede ser fácilmente consumido o integrado en otras aplicaciones de la compañía que requieran un repositorio de parámetros de configuración, ya sean éstas de tipo web o de escritorio y bajo cualquier plataforma como Java o .NET Framework.</p>	
Modelo arquitectónico		
Tipo	Servicio web (protocolo SOAP)	
Patrón	SOA	
Frameworks	Spring Web Services, Java SE 7	
Lenguaje	Java	
Interfaces	Componente	Modo de comunicación
	Base de datos	DBMS Driver

Tabla 6. Detalle del componente Base de Datos

Componente	Base de datos del sistema
Descripción	<p>Es el repositorio de datos utilizado por la aplicación web principalmente para guardar el inventario de procesos automáticos y el historial de cambios.</p> <p>Esta misma base de datos es utilizada por <i>Servicio de Configuraciones</i> para acceder a los parámetros de los procesos automáticos que aquí se encuentran guardados.</p>

Tabla 7. Detalle del componente Servicio JMS

Componente	Servicio JMS
Descripción	<p>Este servicio se encargará de distribuir los mensajes que produce el módulo Monitor de Procesos del componente <i>Aplicación Web</i> y de recibir los mensajes de respuesta que envían los procesos automáticos a través del componente <i>Proceso Supervisor</i>.</p> <p>Los frameworks y protocolos sobre los que funciona este servicio de cola de mensajes permiten la reusabilidad del componente, ya que éste puede ser usado en otras soluciones o sistemas de la compañía.</p>
Modelo arquitectónico	
Tipo	Cola de Mensajes
Patrón	Request-Response
Frameworks	JBoss Messaging, Spring JMS
Lenguaje	Java

Tabla 8. Detalle del componente Proceso Supervisor

Componente	Proceso Supervisor	
Descripción	<p>Este componente para que funcione debe ser instalado como servicio de Windows o demonio de Linux en cada servidor de procesos automáticos.</p> <p>Cuando inicia su ejecución, consulta a <i>Servicio de Configuraciones</i> para obtener sus parámetros de configuración y cargarlos en memoria.</p> <p>Se encarga de recibir los mensajes provenientes de <i>Aplicación Web</i> a través de <i>Servicio JMS</i> y traducirlos al comando correspondiente de Windows o Linux. También, es el encargado de recibir los mensajes de estado que le envían los procesos automáticos y reenviarlo a <i>Servicio JMS</i> cuando le es solicitado.</p> <p>En resumen, este componente es la interfaz del sistema encargada de interactuar directamente con los procesos automáticos.</p>	
Modelo arquitectónico		
Tipo	Servicio/Demonio	
Patrón	Batch	
Frameworks	Spring Batch, JSW (Java Service Wrapper)	
Lenguaje	Java	
Interfaces	Componente	Modo de comunicación
	Servicio de Configuraciones	SOAP
	Servicio JMS	JMS

Tabla 9. Detalle del componente Stack ELK

Componente	Stack ELK
Descripción	<p>Este sistema se encarga del manejo de los logs que producen los procesos automáticos, recibidos del componente <i>Logstash Forwarder</i> que los envía desde cada servidor.</p> <p>Este componente se encarga de almacenar los logs de forma centralizada, permitiendo hacer búsquedas sobre estos datos a través de una interfaz web o través de una interfaz de servicios RESTful; siendo esta interfaz la utilizada para el Sistema de Administración, Supervisión y Control de Procesos Automáticos.</p> <p>Debido a que este componente es un completo sistema de manejo de logs independiente, la compañía tendrá la posibilidad de sacarle el máximo provecho, utilizándolo junto con otras aplicaciones que no son procesos automáticos pero que necesitan almacenar y procesar sus logs.</p>
Sub-componentes	<ul style="list-style-type: none"> • ElasticSearch • Logstash • Kibana

Tabla 10. Detalle del componente Logstash Forwarder

Componente	Logstash Forwarder	
Descripción	<p>Este componente tiene como función acceder a los archivos de log que generan los procesos automáticos y reenviar el contenido hacia el repositorio central del Stack ELK.</p> <p>Para que funcione, el componente debe ser instalado como servicio de Windows o demonio de Linux en cada servidor de procesos automáticos.</p>	
Interfaces	Componente	Modo de comunicación
	Stack ELK	Protocolo Lumberjack

Vista de Despliegue.

La infraestructura que soportará al Sistema de Administración, Supervisión y Control de Procesos Automáticos estará ubicada en los centros de datos de la compañía en El Salvador y contará con los siguientes nodos:

1. Servidor de Base de datos
2. Servidor de Aplicaciones 1
3. Servidor de Aplicaciones 2
4. Servidor de logs
5. Servidor Windows
6. Servidor Linux

Estos servidores deberán tener bien definida sus respectivas funciones, por ejemplo: de aplicaciones o de base de datos. La nomenclatura para la asignación de nombres será de acuerdo al estándar utilizado en la compañía.

Diagrama de despliegue.

El siguiente diagrama muestra la configuración de despliegue de cada uno de los componentes del sistema en los distintos elementos de infraestructura y la comunicación que se da entre éstos.

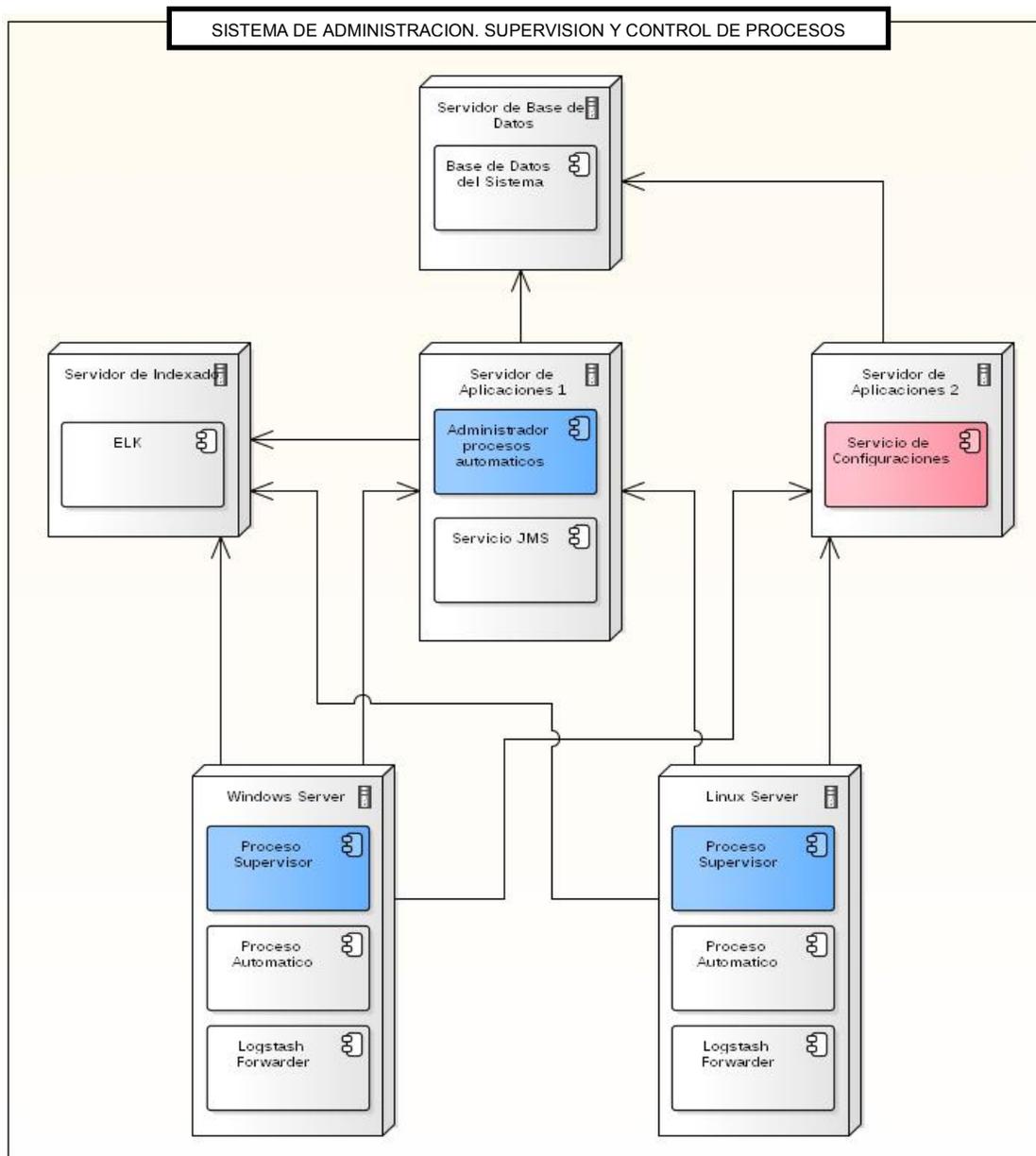


Figura 2. Diagrama de despliegue

Detalle de nodos.

A continuación se presenta el detalle de cada uno de los elementos de infraestructura que soportan la solución.

Tabla 11. Detalle del nodo Servidor de Base de Datos

Servidor	Servidor de Base de Datos
Descripción y objetivo del servidor	
Este servidor contendrá la base de datos que usará el Sistema de Administración, Supervisión y Control de Procesos Automáticos para guardar el inventario de aplicaciones y los parámetros de configuración de estas. Esta base de datos será accedida por la aplicación web y el servicio web de la solución.	
Servicios	
Nombre	Tipo
Oracle	Motor de base de datos
Configuración de software	
Sistema operativo	HP-UX 11i v3
Contenedor de aplicaciones	
Otros	Oracle Database 11g R2

Tabla 12. Detalle del nodo Servidor de Aplicaciones 1

Servidor	Servidor de Aplicaciones 1
Descripción y objetivo del servidor	
Servidor de aplicaciones que contendrá la aplicación web del Sistema de Administración, Supervisión y Control de Procesos Automáticos. También, en este servidor residirá el servicio de cola de mensajes JMS.	
Servicios	
Nombre	Tipo
Sistema de Administración, Supervisión y Control de Procesos Automáticos	Aplicación web
Servicio JMS	Servicio de cola de mensajes JMS
Configuración de software	
Sistema operativo	Red Hat Enterprise Linux 6 Update 6
Contenedor de aplicaciones	JBoss Application Server 7.1
Otros	Java SE Runtime Environment 8

Servidor	Servidor de Aplicaciones 2
Descripción y objetivo del servidor	
Este servidor de aplicaciones contendrá el servicio web de configuraciones, al cual accederán los procesos automáticos para cargar sus configuraciones.	
Servicios	
Nombre	Tipo
Servicio de Configuraciones	Servicio Web SOAP
Configuración de software	
Sistema operativo	Red Hat Enterprise Linux 6 Update 6
Contenedor de aplicaciones	JBoss Application Server 7.1
Otros	Java SE Runtime Environment 8

Tabla 13. Detalle del nodo Servidor de logs

Servidor	Servidor de logs
Descripción y objetivo del servidor	
Servidor donde estará desplegado el sistema para el procesamiento, almacenamiento e indexado de logs.	
Servicios	
Nombre	Tipo
ElasticSearch	Servicio de indexación y búsqueda.
Logstash	Servicio de recolección, procesamiento y almacenamiento de logs
Kibana	Página web
Configuración de software	
Sistema operativo	Red Hat Enterprise Linux 6 Update 6
Contenedor de aplicaciones	Apache Tomcat 7
Otros	<ul style="list-style-type: none"> • ElasticSearch 1.6.0 • Logstash 1.5.2 • Kibana 4.1.1 • Java SE Runtime Enviroment 8

Tabla 14. Detalle del nodo Servidor Windows

Servidor	Servidor Windows
Descripción y objetivo del servidor	
<p>Este nodo representa a los servidores Windows en los cuales residen servicios que ejecutan procesos automáticos.</p> <p>En estos servidores tendrá que instalarse el Proceso Supervisor como servicio para que los procesos automáticos que alojan sean integrados al Sistema de Administración, Supervisión y Control de Procesos Automáticos.</p>	
Servicios	
Nombre	Tipo
Proceso Supervisor	Servicio local del Sistema de Administración, Supervisión y Control de Procesos Automáticos.
Configuración de software	
Sistema operativo	Microsoft Windows Server 2008 R2
Contenedor de aplicaciones	
Otros	Java SE Runtime Enviroment 8

Tabla 15. Detalle del nodo Servidor Linux

Servidor	Servidor Linux
Descripción y objetivo del servidor	
<p>Este nodo representa a los servidores Linux en los cuales residen servicios que ejecutan procesos automáticos.</p> <p>En estos servidores tendrá que instalarse el Proceso Supervisor como demonio para que los procesos automáticos que alojan sean integrados al Sistema de Administración, Supervisión y Control de Procesos Automáticos.</p>	
Servicios	
Nombre	Tipo
Proceso Supervisor	Demonio del Sistema de Administración, Supervisión y Control de Procesos Automáticos.
Configuración de software	
Sistema operativo	Red Hat Enterprise Linux 6 Update 6
Contenedor de aplicaciones	
Otros	Java SE Runtime Enviroment 8

Vista de Datos.

Los sub-sistemas *Aplicación Web* y *Servicio de Configuraciones*, que forman parte de la solución, utilizarán una base de datos Oracle 11g, ya que este SGDB (Sistema Gestor de Base de Datos) es el estándar actual de la compañía.

Diagrama entidad-relación.

Diagrama que muestra la estructura y la relación de las tablas que registrarán los datos de los procesos automáticos, sus parámetros y los cambios que se realicen en éstos.

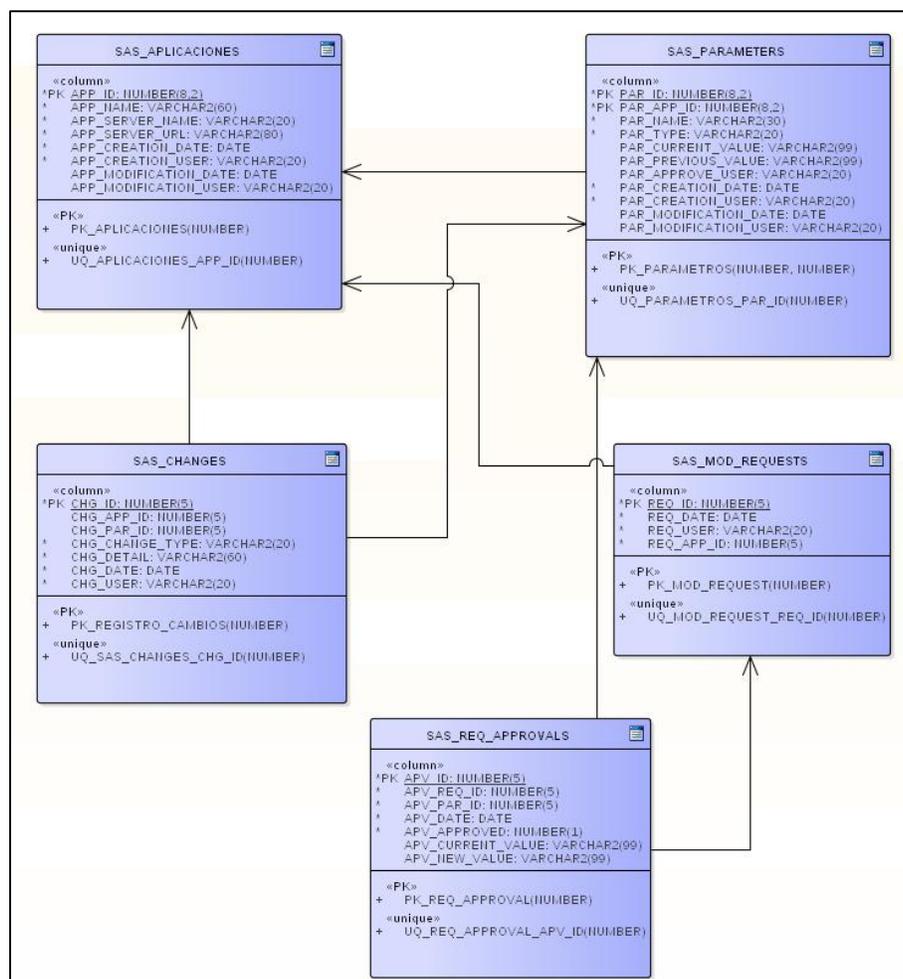


Figura 3. Diagrama Entidad-Relación

Diccionario de datos.

A continuación se presenta el detalle de las definiciones de cada una de las tablas que componen el modelo de datos de la solución.

Tabla 16. Definición de la tabla SAS_APPLICATIONS

Tabla	SAS_APPLICATIONS			
Columna	Tipo	PK	FK	Nulo
APP_ID	NUMBER(5,0)	Si		No
APP_NAME	VARCHAR2(60)			No
APP_SERVER_NAME	VARCHAR2(20)			No
APP_SERVER_URL	VARCHAR2(80)			No
APP_CREATION_DATE	DATE			No
APP_CREATION_USER	VARCHAR2(20)			No
APP_MODIFICATION_DATE	DATE			Si
APP_MODIFICATION_USER	VARCHAR2(20)			Si

Tabla 17. Definición de la tabla SAS_PARAMETERS

Tabla	SAS_PARAMETERS			
Columna	Tipo	PK	FK	Nulo
PAR_ID	NUMBER(5,0)	Si		No
PAR_APP_ID	NUMBER(5,0)		Si	No
PAR_NAME	VARCHAR2(30)			No
PAR_TYPE	VARCHAR2(20)			No
PAR_CURRENT_VALUE	VARCHAR2(99)			Si
PAR_PREVIOUS_VALUE	VARCHAR2(99)			Si
PAR_APPROVE_USER	VARCHAR2(20)			Si
PAR_CREATION_DATE	DATE			No
PAR_CREATION_USER	VARCHAR2(20)			No
PAR_MODIFICATION_DATE	DATE			No
PAR_MODIFICATION_USER	VARCHAR2(20)			No

Tabla 18. Definición de la tabla SAS_CHANGES

Tabla	SAS_CHANGES			
Columna	Tipo	PK	FK	Nulo
CHG_ID	NUMBER(5)	Si		No
CHG_APP_ID	NUMBER(5)		Si	No
CHG_PAR_ID	NUMBER(5)		Si	No
CHG_CHANGE_TYPE	VARCHAR2(20)			No
CHG_DETAIL	VARCHAR2(60)			No
CHG_DATE	DATE			No
CHG_USER	VARCHAR2(20)			No

Tabla 19. Definición de la tabla SAS_MOD_REQUESTS

Tabla	SAS_MOD_REQUESTS			
Columna	Tipo	PK	FK	Nulo
REQ_ID	NUMBER(5)	Si		No
REQ_DATE	DATE			No
REQ_USER	VARCHAR2(20)			No
REQ_APP_ID	NUMBER(5)		Si	No

Tabla 20. Definición de la tabla SAS_REQ_APPROVALS

Tabla	SAS_REQ_APPROVALS			
Columna	Tipo	PK	FK	Nulo
APV_ID	NUMBER(5)	Si		No
APV_REQ_ID	NUMBER(5)		Si	No
APV_PAR_ID	NUMBER(5)		Si	No
APV_DATE	DATE			Si
APV_APPROVED	NUMBER(1)			Si
APV_CURRENT_VALUE	VARCHAR2(99)			Si
APV_NEW_VALUE	VARCHAR2(99)			Si

Vista Lógica

En esta vista se detalla la organización de paquetes y las respectivas clases que conforman los sub-sistemas *Aplicación Web*, *Servicio de Configuraciones* y *Supervisor de Procesos*.

Detalle de Paquetes y Clases.

Tabla 21. Estructura de paquetes y clases de Aplicación Web

Sistema	Aplicación Web	
Paquetes	Sub-paquetes	Clases
Modelo	Entity	Application Change ModRequest Parameter ReqApproval
	Repository	GenericRepository ApplicationRepository ChangeRepository ModRequestRepository ParameterRepository ReqApprovalRepository
Lógica		AdministracionProcesosController AuditoriaController ConsultaLogsController SupervisionControlController
Presentación	AdministracionProcesos	CambioValorParametro EdicionParametro EdicionProceso Index DetalleProceso NuevoParametro NuevoProceso
	Auditoria	Index Reporte
	ConsultaLogs	Index Reporte
	SupervisionControl	MonitorProceso Index

Tabla 22. Estructura de paquetes y clases de Servicio de Configuraciones

Sistema	Servicio de Configuraciones	
Paquetes	Sub-paquetes	Clases
Modelo	Entity	Application Parameter
	Repository	GenericRepository ApplicationRepository ParameterRepository
Lógica		DespachadorConfiguracion
Servicio		Configuracion

Tabla 23. Estructura de paquetes y clases de Supervisor de Procesos

Sistema	Supervisor de Procesos	
Paquetes	Sub-paquetes	Clases
Integración		ConfigServiceConsumer MessageQueueConsumer
Lógica		ApplicationHelper ProcessHandler SupervisionHandler ControlHandler
Aplicación		Supervisor

Descripción de Clases.

A continuación se presenta una descripción general de cada una de las clases que forman parte de los paquetes existentes en los componentes de la solución.

Tabla 24. Descripción de clases del paquete Modelo.Entity de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Modelo.Entity
Clase	Descripción
Application	Clase básica para modelar un objeto único con base en un registro de la tabla SAS_APPLICATIONS, en la configuración del ORM se debe mapear a esta tabla.
Change	Clase básica para modelar un objeto único de la tabla SAS_CHANGES.
ModRequest	Clase básica para modelar un objeto único con base en un registro de la tabla SAS_MOD_REQUESTS.
Parameter	Clase básica para modelar un objeto único con base en un registro de la tabla SAS_PARAMETERS.
ReqApproval	Clase básica para modelar un objeto único con base en un registro de la tabla SAS_REQ_APPROVALS.

Tabla 25. Descripción de clases del paquete Modelo.Repository de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Modelo.Repository
Clase	Descripción
GenericRepository	<p>Clase base que contiene las operaciones más comunes (fetch, find, add, attach, delete) para acceder a los registros de un tipo genérico, haciendo uso de las clases de acceso a datos que provee el ORM.</p> <p>Las clases repositorio de cada entidad deben heredar de esta clase para poder implementar las operaciones que reciben y devuelven objetos del tipo respectivo.</p>
ApplicationRepository	Esta clase implementa las operaciones de acceso y persistencia de objetos del tipo Application.
ChangeRepository	Implementa las operaciones de acceso y persistencia de objetos del tipo Change.
ModRequestRepository	Implementa las operaciones de acceso y persistencia de objetos del tipo ModRequest.
ParameterRepository	Implementa las operaciones de acceso y persistencia de objetos del tipo Parameter.
ReqApprovalRepository	Implementa las operaciones de acceso y persistencia de objetos del tipo ReqApproval.

Tabla 26. Descripción de clases del paquete Logica de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Logica
Clase	Descripción
AdministracionProcesosController	Clase del módulo de Administración de Procesos encargada de procesar las peticiones generadas desde las páginas de presentación y consultar a la capa del modelo para mostrar los datos necesarios de vuelta en la capa de presentación.
SupervisionControlController	Clase controlador que se encarga de procesar las peticiones y mostrar las páginas del módulo de Supervisión y Control.
ConsultaLogsController	Clase controlador que se encarga de procesar las peticiones y mostrar las páginas del módulo de Consulta de Logs.
AuditoriaController	Clase controlador que se encarga de procesar las peticiones y mostrar las páginas del módulo de Auditoria.

Tabla 27. Descripción de clases del paquete Presentacion.AdministracionProcesos de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Presentacion.AdministracionProcesos
Clase	Descripción
Index	Página JSP inicial del módulo de Administración de Procesos que muestra el listado de procesos automáticos existentes.
DetalleProceso	Página JSP que muestra el detalle o datos de cada proceso automático de forma individual.
CambioValorParametro	Página JSP con el formulario que permite cambiar los valores de los parámetros de cada proceso automático.
NuevoProceso	Página JSP que muestra el formulario de creación o registro de un nuevo proceso automático.
EdicionProceso	Página JSP con el formulario para editar los datos de un proceso automático.
NuevoParametro	Página JSP que contiene el formulario para crear un parámetro de configuración de un proceso automático.
EdicionParametro	Página JSP con el formulario de edición de datos de un parámetro.

Tabla 28. Descripción de clases del paquete Presentacion.SupervisionControl de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Presentacion.SupervisionControl
Clase	Descripción
Index	Página JSP inicial del módulo de Supervisión y Control que muestra el listado de procesos automáticos y las opciones de monitoreo.
MonitorProceso	Página JSP para mostrar el detalle de estado de un proceso automático y las opciones de control sobre el mismo.

Tabla 29. Descripción de clases del paquete Presentacion.ConsultaLogs de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Presentacion.ConsultaLogs
Clase	Descripción
Index	Página JSP inicial del módulo de Consulta de Logs que muestra un formulario para la búsqueda de logs.
Reporte	Página JSP que muestra los mensajes de log de acuerdo a la búsqueda realizada en el formulario de búsqueda.

Tabla 30. Descripción de clases del paquete Presentacion.Auditoria de Aplicación Web

Sistema	Aplicación Web
Estructura de paquete	Presentacion.Auditoria
Clase	Descripción
Index	Página JSP inicial del módulo de Auditoria que muestra un formulario para la consulta de registros de auditoria.
Reporte	Página JSP que muestra el reporte de registros de auditoria.

Tabla 31. Descripción de clases del paquete Modelo.Entity del Servicio de Configuraciones

Sistema	Servicio de Configuraciones
Estructura de paquete	Modelo.Entity
Clase	Descripción
Application	Clase básica para modelar un objeto único con base en un registro de la tabla SAS_APPLICATIONS, en la configuración del ORM se debe mapear a esta tabla.
Parameter	Clase básica para modelar un objeto único con base en un registro de la tabla SAS_PARAMETERS.

Tabla 32. Descripción de clases del paquete Modelo.Repository del Servicio de Configuraciones

Sistema	Servicio de Configuraciones
Estructura de paquete	Modelo.Repository
Clase	Descripción
GenericRepository	<p>Clase base que contiene las operaciones más comunes (fetch, find, add, attach, delete) para acceder a los registros de un tipo genérico, haciendo uso de las clases de acceso a datos que provee el ORM.</p> <p>Las clases repositorio de cada entidad deben heredar de esta clase para poder implementar las operaciones que reciben y devuelven objetos del tipo respectivo.</p>
ApplicationRepository	Esta clase implementa las operaciones de acceso y persistencia de objetos del tipo Application.
ParameterRepository	Implementa las operaciones de acceso y persistencia de objetos del tipo Parameter.

Tabla 33. Descripción de clases del paquete Logica del Servicio de Configuraciones

Sistema	Servicio de Configuraciones
Estructura de paquete	Logica
Clase	Descripción
DespachadorConfiguracion	Esta clase se encarga de consultar la capa de modelo de acuerdo a los parámetros que se reciben en la capa de servicio, asimismo se encargará de devolver los respectivos resultados a esta capa de servicio.

Tabla 34. Descripción de clases del paquete Servicio del Servicio de Configuraciones

Sistema	Servicio de Configuraciones
Estructura de paquete	Servicio
Clase	Descripción
Configuracion	Esta clase expone como servicios los métodos necesarios para que los procesos automáticos accedan a sus parámetros de configuración.

Tabla 35. Descripción de clases del paquete Integracion del Proceso Supervisor

Sistema	Proceso Supervisor
Estructura de paquete	Integracion
Clase	Descripción
ConfigServiceConsumer	Esta clase expone métodos que consumen directamente el Servicio de Configuraciones para que el Proceso Supervisor a través de su capa de lógica acceda a sus parámetros de configuración.
MessageQueueConsumer	Esta clase contiene métodos para consumir (consultar y enviar mensajes) la cola de mensajes del sistema.

Tabla 36. Descripción de clases del paquete Logica del Proceso Supervisor

Sistema	Proceso Supervisor
Estructura de paquete	Logica
Clase	Descripción
ApplicationHelper	Clase que contiene métodos para acceder a los parámetros de configuración.
ProcessHandler	Esta clase expone métodos que permiten al Proceso Supervisor recibir y devolver los mensajes de estado que generan los procesos automáticos.
SupervisionHandler	A través de los métodos de esta clase el Proceso Supervisor puede resolver las consultas de estado de los servicios/demonios y de los procesos automáticos.
ControlHandler	Esta clase está compuesta por métodos que permiten traducir los mensajes que se envían desde el módulo de Supervisión y Control, en los comandos respectivos del sistema operativo.

Tabla 37. Descripción de clases del paquete Aplicacion del Proceso Supervisor

Sistema	Proceso Supervisor
Estructura de paquete	Aplicación
Clase	Descripción
Supervisor	Esta contiene los métodos que se ejecutan cuando se arranca el servicio o demonio y que se encargan de hacer los llamados a la capa Logica.

Diagramas de Clases.

Aplicación Web.

El siguiente diagrama muestra la organización lógica de capas en que se divide el componente Aplicación Web de la solución, también el diagrama permite observar la estructura de paquetes que conforma cada capa.



Figura 4. Aplicación Web: paquetes

Los siguientes diagramas detallan la organización de los paquetes Entity y Repository de la capa Modelo, las clases que forman parte de éstos y las relaciones entre cada una.

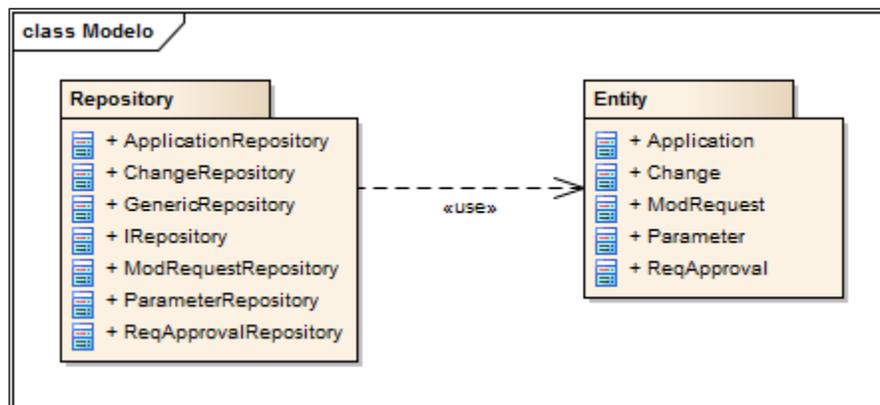


Figura 5. Aplicación Web: paquete Modelo

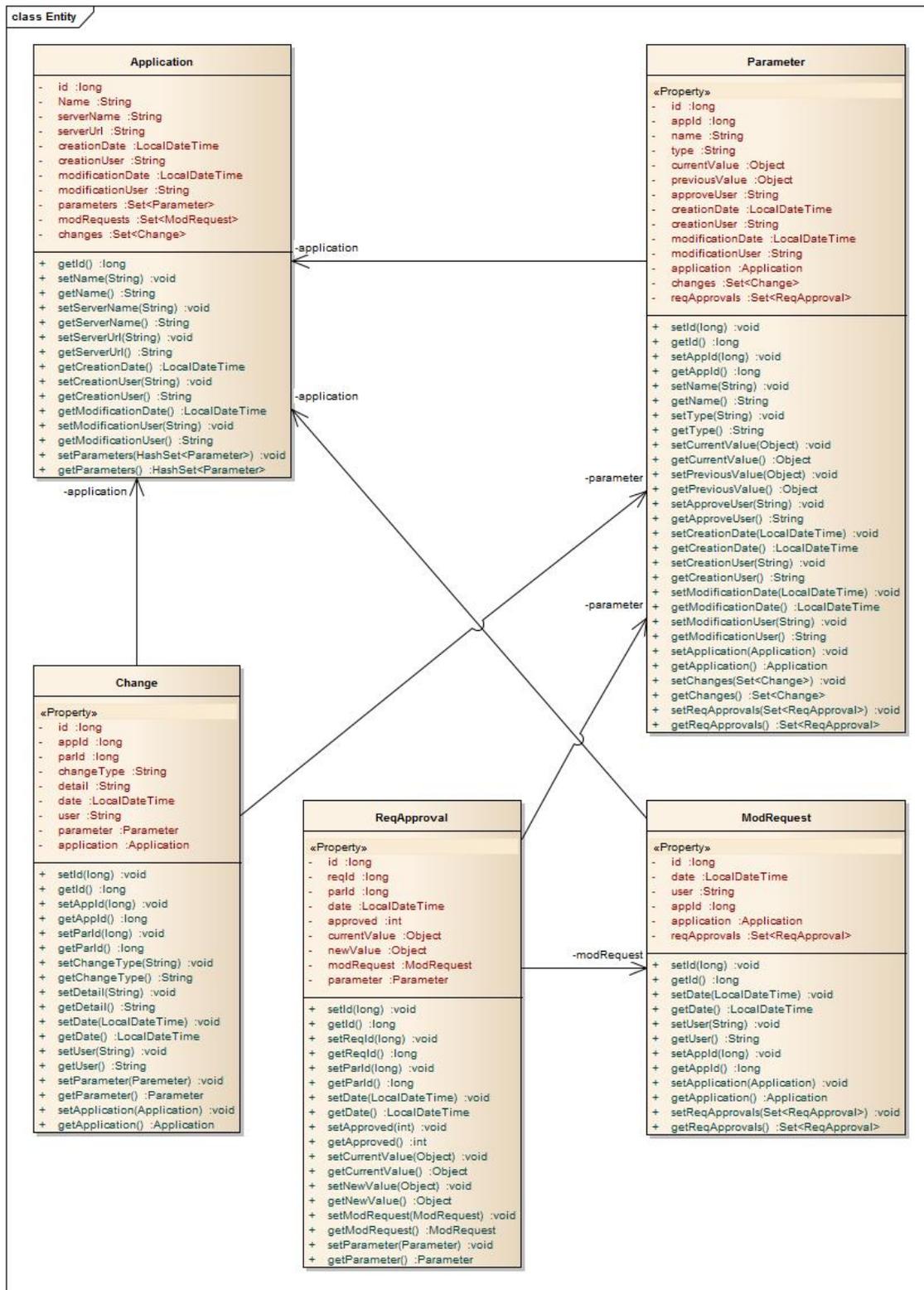


Figura 6. Aplicación Web: clases en el sub-paquete Entity

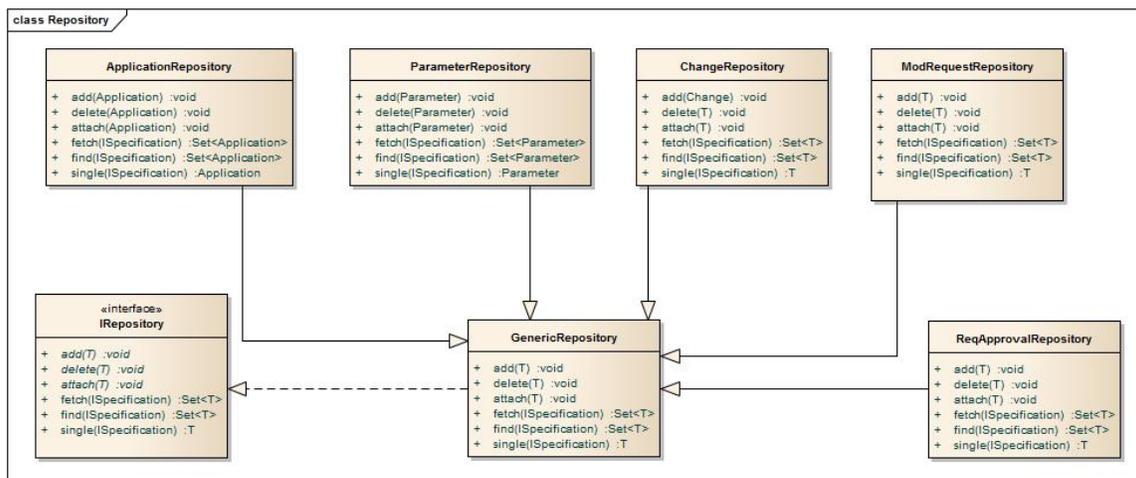


Figura 7. Aplicación Web: clases en el sub-paquete Repository

El diagrama a continuación muestra las clases o controladores que componen la capa lógica; como se puede observar, existe una clase correspondiente para cada uno de los módulos del sistema.

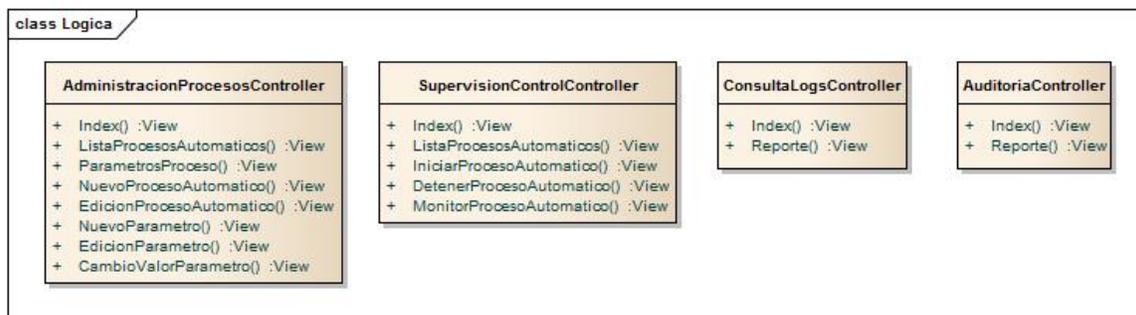


Figura 8. Aplicación Web: clases en el paquete Lógica

Organización de la capa de Presentación, donde se especifican los paquetes de cada módulo del sistema y las paginas JSP en cada uno de éstos.

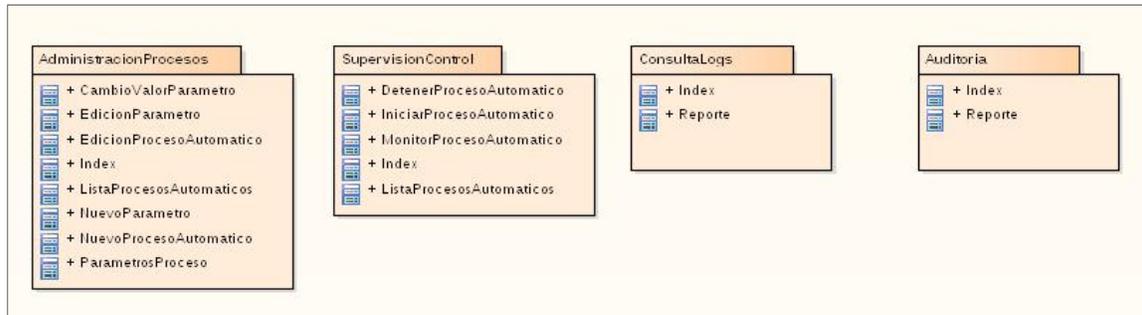


Figura 9. Aplicación Web: paquete Presentación

Servicio de Configuraciones.

El diagrama a continuación muestra las capas lógicas en que se divide el Servicio de Configuraciones y su correspondiente relación de clases. Por último, el segundo diagrama muestra las clases correspondientes a los paquetes Entity y Repository de la capa Modelo.

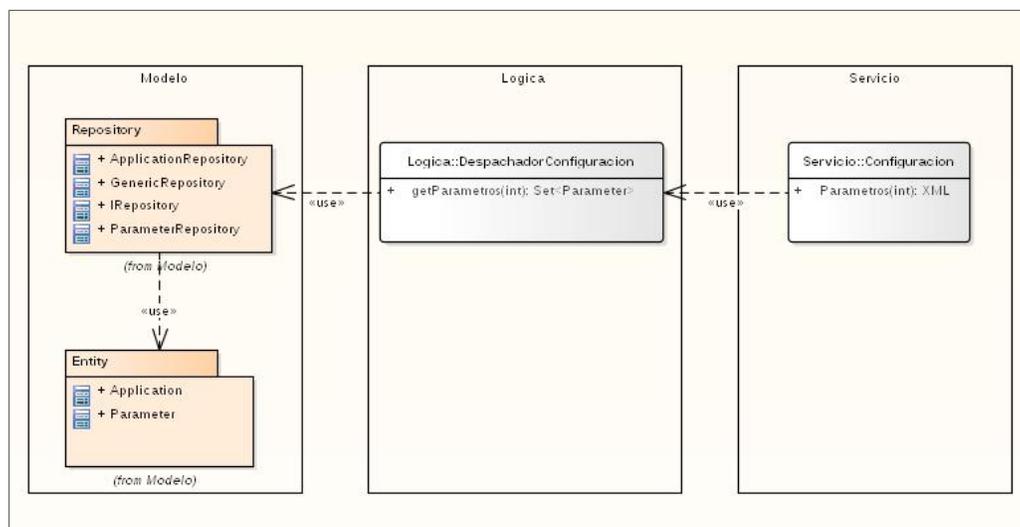


Figura 10. Servicio de Configuraciones: paquetes

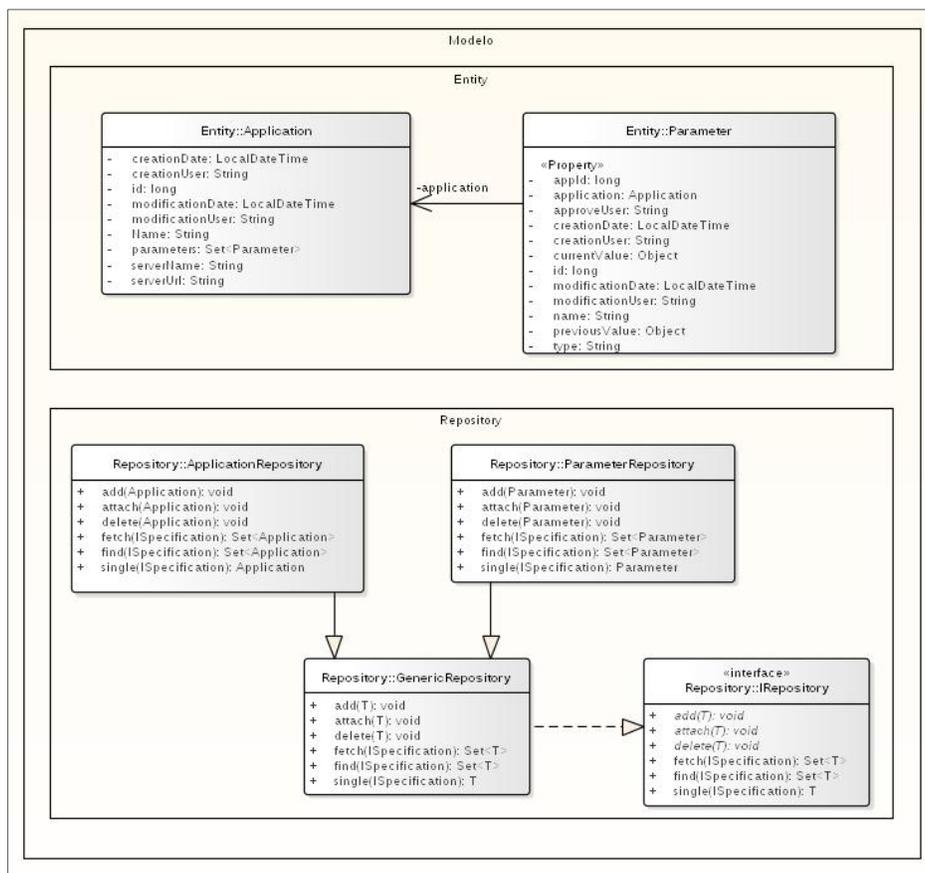


Figura 11. Servicio de Configuraciones: clases en el paquete Modelo

Supervisor de Procesos.

Los siguientes diagramas permiten observar la organización lógica de capas en que se divide el Proceso Supervisor y como se relacionan las clases que contiene cada una de estas capas.

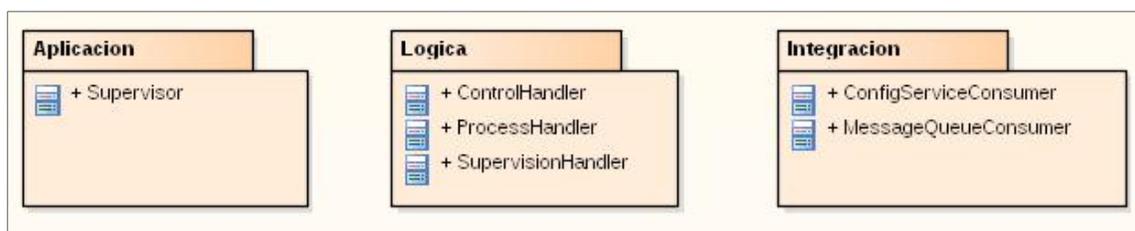


Figura 12. Supervisor de Procesos: paquetes

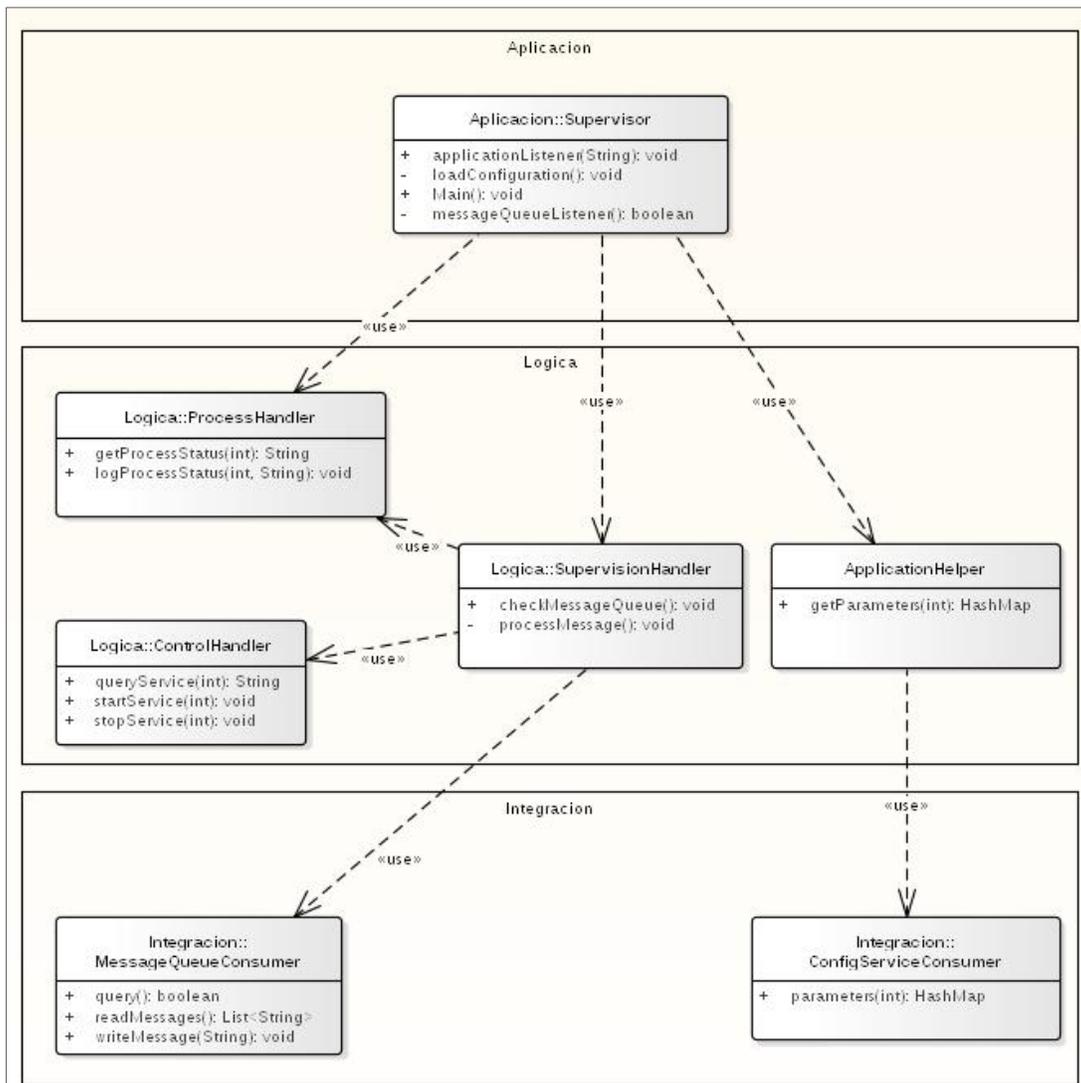


Figura 13. Supervisor de Procesos: clases

Estándares y lineamientos

Desarrollo de procesos automáticos.

Lenguajes de programación.

Los lenguajes de programación que deberán usarse exclusivamente son Java y C#, ya que éstos son los permitidos por el estándar de desarrollo actual de la compañía.

Las versiones mínimas de los entornos de desarrollo de software para Java y C# son:

- Java SE JDK 7.
- .NET Framework 4.5.

Tipo de aplicación.

Para que la aplicación funcione correctamente como un proceso automático, deberá compilarse de manera que funcione como servicio de Windows o demonio de Linux.

Si se trabaja con C# y .NET Framework se deberá compilar como un proyecto “Servicio de Windows” en Visual Studio y si se trabaja con Java se deberá hacer uso de la herramienta Java Service Wrapper y configurarse para ejecutar el archivo jar de la aplicación.

La aplicación no deberá depender ni hacer uso de componentes que requieran de algún tipo de interacción con el usuario o que puedan ocasionar algún conflicto o interrupción en la ejecución del servicio o demonio, como por ejemplo: interfaz gráfica de usuario, ventanas de diálogo, entrada/salida de consola, etc.

Parámetros.

Para que el proceso automático se integre correctamente a la interfaz de cambio de parámetros del Sistema de Administración, Supervisión y Control de Procesos Automáticos, se deberán seguir los siguientes lineamientos:

- Los valores de los parámetros que sean utilizados para que el proceso automático pueda ejecutar sus tareas o que son utilizados dentro de la lógica de su algoritmo, por ejemplo: IP de servidor FTP, Hora de ejecución, Tasa de cambio, Mes de cálculo, etc., no deben ser escritos en archivos de configuración ni en el código fuente.
- El proceso automático al iniciar su ejecución o iniciar sus tareas, deberá consultar al servicio web “Servicio de Configuraciones” del Sistema de Administración, Supervisión y Control de Procesos Automáticos, para obtener los valores de los parámetros.
- La URL del servicio web Servicio de Configuraciones, será el único parámetro que se guardará localmente en un archivo de configuración.

Componentes.

El proceso automático deberá incluir los siguientes componentes:

- Notificador de estado. Este componente debe enviar mensajes de estado o avance de las tareas que ejecuta la aplicación, a la interfaz expuesta por el Proceso Supervisor. El detalle de integración con la interfaz del Proceso Supervisor deberá ser consultado en la respectiva guía técnica de este componente del sistema.

- Componente de logging. Este módulo debe ser el encargado de registrar los mensajes de información de eventos, excepciones, errores, advertencias, etc. que sean generados por el proceso automático. Estos mensajes deben ser escritos en archivos de texto plano y siguiendo el patrón definido más adelante en la sección Logging.

Bibliotecas.

Para la generación de mensajes de log se recomienda el uso de las siguientes bibliotecas de acuerdo a la plataforma de desarrollo utilizada:

- Procesos automáticos desarrollados en .NET Framework: Enterprise Library o Log4net.
- Procesos automáticos desarrollados en Java: Log4j 2.

El uso de estas bibliotecas no es obligatorio, sin embargo, si se utiliza otra diferente o se desarrolla un generador de log propio de la aplicación, debe tomarse en cuenta el cumplimiento de los lineamientos de configuración y patrón de mensajes definidos en la sección Logging.

Logging

Formato de log.

Los mensajes de log deberán ser registrados en archivos de texto plano y guardados con la extensión txt.

Configuración.

Los archivos de log deberán ser depositados por el proceso automático en la siguiente estructura de directorio:

- para Windows: “..\directorio_aplicación\log\”,
- y para Linux: “../directorio_aplicación/log”.

El tamaño de cada archivo de log no deberá sobrepasar los 5 MB de espacio en disco y dado el caso que sea necesario registrar más mensajes cuando se ha alcanzado este tamaño, se debe generar otro archivo y así sucesivamente hasta un máximo de 10 archivos; y cuando se alcance el límite de 10 archivos se deberá limpiar y reutilizar el archivo más antiguo.

Esta configuración de archivos es soportada por las bibliotecas Enterprise Library, Log4net y Log4j, ya recomendadas en este documento.

Patrón de mensajes.

Para que los mensajes de log puedan ser reconocidos por el sistema de manejo de logs a través del componente Logstash Forwarder, éstos deben ser escritos utilizando el siguiente patrón:

[Nombre aplicación][Fecha hora: YYYYMMDD HH:MI:SS][Tipo
mensaje][Módulo o clase][Tarea u operación][Mensaje]

Los tipos de mensaje pueden ser:

- INFO.
- WARN.
- ERROR.

Ejemplo:

[GeneradorTiquetes][20150715 17:35:20][INFO][Tiquete][Emisión][Obteniendo
información de reserva]

Conclusiones

El diseño del Sistema de Administración, Supervisión y Control de Procesos Automáticos, le permitirá a la compañía de transporte aéreo, y más específicamente a su área de Soporte de TI, contar con una alternativa de solución para sus planes de solventar la problemática analizada en este documento.

La propuesta cuenta con los componentes tecnológicos necesarios que permitirán al área de Soporte realizar una mejor supervisión y control de los procesos automáticos, también, el sistema les brindará las interfaces con las que podrán acceder directamente a las configuraciones de estas aplicaciones y los registros de log que éstas generan; todo esto les dará la capacidad de identificar problemas y realizar cambios de forma oportuna.

Antes de implementar un sistema informático como el que se propone en este documento, se deben evaluar las distintas alternativas que pudieran dar solución a los problemas planteados. Ya sea que estas alternativas sean sistemas que se tengan que adquirir o se tengan que desarrollar, se debe hacer una evaluación con base a criterios definidos de acuerdo a las características y funcionalidades buscadas. Este proceso permitirá seleccionar las opciones de solución más adecuadas al contexto de la problemática.

Cuando se diseña una solución que busca resolver problemas tecnológicamente complejos, ésta no siempre resultará ser una aplicación única con todas las funcionalidades necesarias, sino un sistema formado por distintos componentes o subsistemas que resuelven un problema en específico, pero integrados de tal forma que el conjunto sea percibido como un solo componente.

Es importante tomar en cuenta que cuando una solución implica la integración de distintas tecnologías y sistemas, ya sean estos existentes o actuales, como de futuras implementaciones, también es importante definir estándares y lineamientos como parte de la misma solución, ya que esto permitirá que el funcionamiento e integración sea de acuerdo a lo esperado.

Referencias

- [1] Microsoft Corporation (s.f.). Introducción a las aplicaciones de servicios de Windows. Obtenido de Microsoft Developer Network. URL: [https://msdn.microsoft.com/es-es/library/d56de412\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/d56de412(v=vs.110).aspx)
- [2] Canonical Ltd (s.f.). Daemon. Obtenido de doc.ubuntu-es. URL: <http://doc.ubuntu-es.org/Daemon>
- [3] The Apache Software Foundation (s.f.). Daemon: Java based daemons or services. Obtenido de Apache Commons. URL: <http://commons.apache.org/proper/commons-daemon/>
- [4] Axelos Ltd (s.f.). Fundamentos de la Gestión TI. Obtenido de ITIL. URL: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/vision_general_gestion_servicios_TI/vision_general_gestion_servicios_TI.php
- [5] International Organization for Standardization (s.f.). ISO/IEC 20000-1:2011 Information technology - Service management - Part 1: Service management system requirements. Obtenido de ISO. URL: <https://www.iso.org/obp/ui/#iso:std:iso-iec:20000:-1:ed-2:v1:en>
- [6] Axelos Ltd (s.f.). Gestión de Cambios. Obtenido de ITIL. URL: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_cambios/vision_general_gestion_de_cambios/vision_general_gestion_de_cambios.php
- [7] Axelos Ltd (s.f.). Gestión de Cambios: Introducción y Objetivos. Obtenido de ITIL. URL: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_cambios/introduccion_objetivos_gestion_de_cambios/introduccion_objetivos_gestion_de_cambios.php
- [8] Historial. (s.f.). En Wikipedia. Recuperado el 20 de junio de 2015 de URL: <https://es.wikipedia.org/wiki/Historial>
- [9] JAMES, Hayden. 20 Top Server Monitoring & Application Performance Monitoring (APM) Solutions. Hayden James [en línea]. 12 de noviembre de 2014. Disponible en URL: <http://haydenjames.io/20-top-server-monitoring-application-performance-monitoring-apm-solutions>
- [10] LURIE, Andy. Top 47 Log Management Tools. ProfitBricks Blog [en línea]. 19 de mayo de 2014. Disponible en URL: <https://blog.profitbricks.com/top-47-log-management-tools/>

- [11] Vanderzyden, John. Welcome to the ELK Stack: Elasticsearch, Logstash, and Kibana. Qbox Blog [en línea]. 17 de julio de 2015. Disponible en URL: <http://blog.qbox.io/welcome-to-the-elk-stack-elasticsearch-logstash-kibana>
- [12] IBM Corporation (s.f.). RESTful Web services: The basics. Obtenido de IBM Developer Network. URL: <http://www.ibm.com/developerworks/library/ws-restful>
- [13] Plataforma Java. (s.f.). En Wikipedia. Recuperado el 6 de julio de 2015 de URL: https://es.wikipedia.org/wiki/Plataforma_Java

Anexos

Prototipos de Interfaz Gráfica

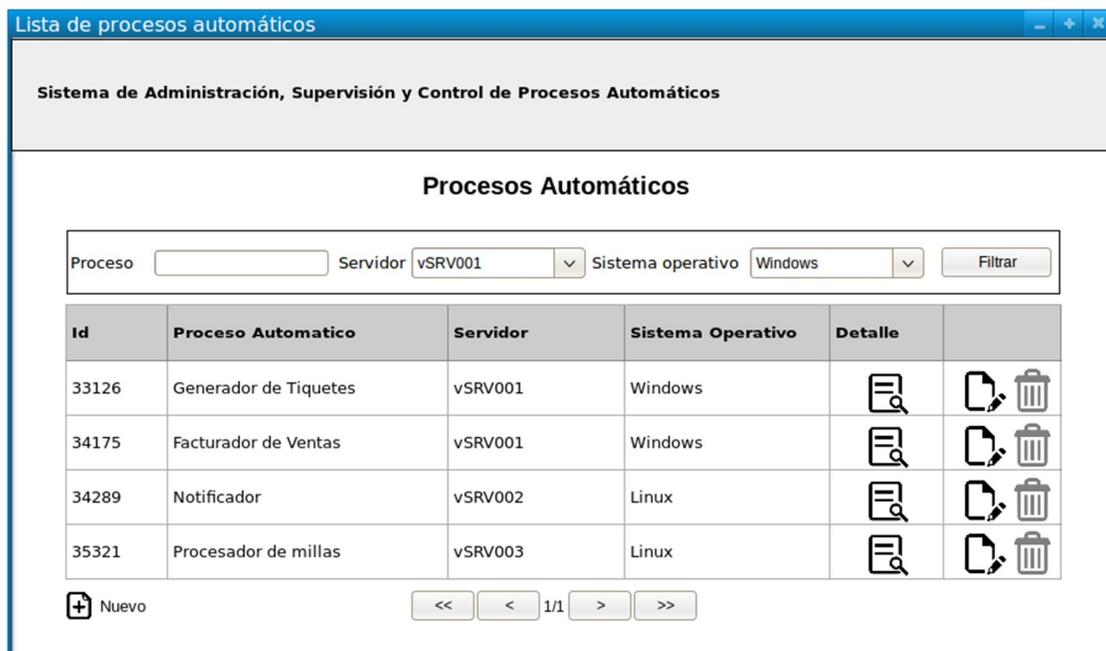


Figura 14. Administración de Procesos: Pantalla inicial

Detalle de proceso automatico - /AdministracionProcesos/DetalleProceso.jsp

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Detalle de Proceso Automatico

Aplicación

ID Aplicación 33126 **Nombre** Generador de Tiquetes

Descripción Proceso automatico que se encarga de la emisión de reservas

Servidor

Servidor vSRV001 **Sistema Operativo** Windows **Dirección de red** 176.230.12.162

Parámetros

Parametro	Valor	
FTP	192.228.117.97	
dbserver	SQLSRV02	
dbUser	autoproc1	
dbPassw	trsc12wx	

Figura 15. Administración de Procesos: Pantalla de detalle

Cambio de parámetro

Cambio de Parámetro

Nombre: Tasa de Cambio

Valor:

Figura 16. Administración de Procesos: Cambio de parámetros

Nuevo proceso automático - /AdministracionProcesos/NuevoProceso.jsp

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Registro de Nuevo Proceso Automatico

Nombre Descripción

Servidor Sistema Operativo Dirección de red

Parametros + Agregar

Parametro	Tipo dato	Valor	
FTP	String	192.228.117.97	 
dbserver	String	SQLSRV02	 
Intentos	int	3	 

Figura 17. Administración de Procesos: Registro de nuevo proceso

Nuevo parámetro

Nuevo Parámetro

Nombre

Tipo

Valor

Cambio necesita aprobación

Usuario aprueba

Correo usuario

Figura 18. Administración de Procesos: Nuevo parámetro

Edición de proceso automático - /AdministracionProcesos/EdicionProceso.jsp

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Edición de Proceso Automatico

ID:

Nombre: Descripción:

Servidor: Sistema Operativo: Dirección de red:

Parametros + Agregar

Parametro	Tipo dato	Valor	
FTP	String	192.228.117.97	 
dbserver	String	SQLSRV02	 
Intentos	int	3	 

Figura 19. Administración de Procesos: Edición de proceso

Edición de parámetro

Edición de Parámetro

Nombre:

Tipo:

Valor:

Cambio necesita aprobación

Usuario aprueba:

Correo usuario:

Figura 20. Administración de Procesos: Edición de parámetro

Supervisión de procesos automáticos

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Supervisión de Procesos Automáticos

Proceso Servidor vSRV001 Sistema operativo Windows

Id	Proceso Automatico	Servidor	Sistema Operativo	Estado Servicio	Supervisar
33126	Generador de Tiquetes	vSRV001	Windows	Corriendo	
34175	Facturador de Ventas	vSRV001	Windows	Detenido	
34289	Notificador	vSRV002	Linux	Corriendo	
35321	Procesador de millas	vSRV003	Linux	Corriendo	

<< < 1/1 > >>

Figura 21. Supervisión y Control: Pantalla inicial

Monitor de proceso automático

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Monitor de Proceso Automático

Servidor

Servidor vSRV001 **Sistema Operativo** Windows **Dirección de red** 176.230.12.162

Aplicación

ID Aplicación 33126 **Nombre** Generador de Tiquetes

Descripción Proceso automatico que se encarga de la emisión de reservas

Servicio

Estado del servicio Corriendo

Proceso

Estado del proceso Paso #1 Inicio: 23-07-2015 01:00:00 Búsqueda de reservas

Figura 22. Supervisión y Control: Monitor de proceso

Consulta de logs

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Consulta de Logs

Proceso automatico

Fecha de log

Tipo de Mensaje

Figura 23. Consulta de logs: Pantalla inicial

Consulta de logs

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Consulta de Logs

Proceso automático: Generador de Tiquetes **Fecha de log:** 23/07/2015

Tipo de Mensaje Hora Modulo

Vista de texto Vista de tabla

```

INFO stream destroy 2011-04-12 00:00:23 1709446838 108.7.233.33 - 8300 246290695 3324.0
INFO session disconnect 2011-04-12 00:00:23 1709446838 108.7.233.33 - 8300 246290695 33
INFO stream stop 2011-04-12 00:00:23 2064587339 173.179.251.44 - 5377 275734425 2952.7
INFO stream destroy 2011-04-12 00:00:23 2064587339 173.179.251.44 - 5377 275734425 2952
INFO session disconnect 2011-04-12 00:00:23 2064587339 173.179.251.44 - 5377 275734425
INFO session connect-pending 2011-04-12 00:00:29 1691358208 67.87.121.38 - 3466 3073 0.
INFO session connect 2011-04-12 00:00:29 1691358208 67.87.121.38 - 3466 3073 0.125 - -
INFO stream create 2011-04-12 00:00:29 1691358208 67.87.121.38 - 3537 3444 0.0 - 1 0

```

Figura 24. Consulta de Logs: Pantalla de visualización

Reporte de auditoria

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Reporte de Auditoria

Proceso automatico

Fecha de log

Tipo de Cambio

Figura 25. Auditoria: Pantalla inicial

Reporte de auditoria

Sistema de Administración, Supervisión y Control de Procesos Automáticos

Reporte de Auditoria

Proceso Automatico: Generador de Tiquetes

Tipo de Cambio Hora Usuario

Fecha	Tipo de cambio	Parametro	Detalle cambio	Usuario

Figura 26. Auditoria: Reporte

Instrucciones de Instalación y Configuración

Instalación de Aplicación Web y Servicio de Configuraciones.

Para el despliegue de los componentes web de la solución, Aplicación Web y Servicio de Configuraciones, se recomienda utilizar el contenedor de aplicaciones JBoss EAP 6.

Los pasos necesarios para realizar el despliegue de una aplicación web y de un servicio web son los mismos, por lo que las siguientes instrucciones se tienen que llevar a cabo tanto para el componente Aplicación Web y Servicio de Configuraciones.

Pasos para desplegar un proyecto web en JBoss EAP:

1. En la consola de administración de JBoss EAP se debe dirigir a la opción del menú superior llamada “Runtime”.
2. Estando en la sección “Runtime” se debe hacer clic sobre la opción “Manage Deployments” que se encuentra en la sección “Server” del menú izquierdo. En la imagen siguiente se pueden observar las opciones mencionadas.

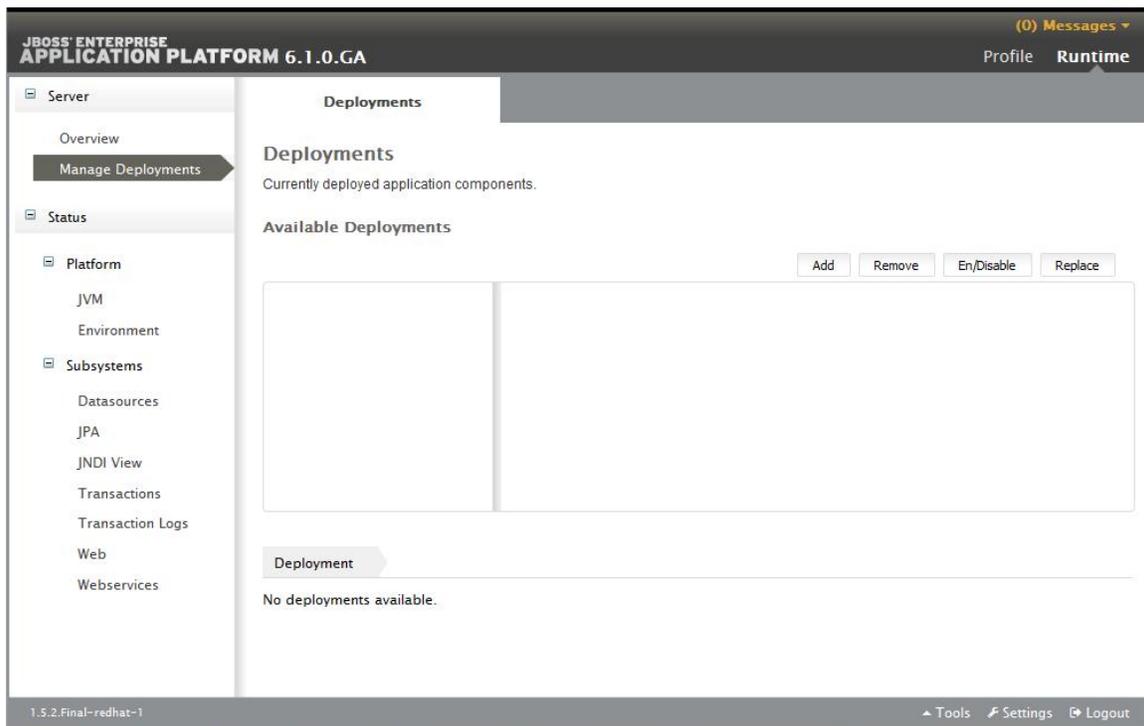


Figura 27. Sección "Runtime" de JBoss EAP

3. Dentro de la página "Deployments" se debe hacer clic sobre el botón "Add" para que aparezca una ventana que permitirá seleccionar el archivo WAR de la aplicación.

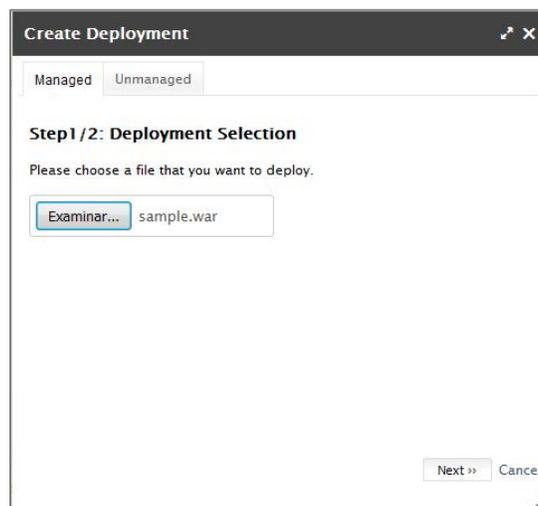


Figura 28. Ventana de selección de archivo WAR

4. En el siguiente paso se tendrá que verificar los datos del archivo WAR cargado y se debe hacer clic en el botón “Save”.



Figura 29. Ventana de confirmación del archivo WAR

5. Una vez cargado el archivo WAR, se debe hacer clic sobre la opción web del mismo para configurar el campo “Context Root”, el cual es el nombre mediante el cual se accederá a la aplicación web.

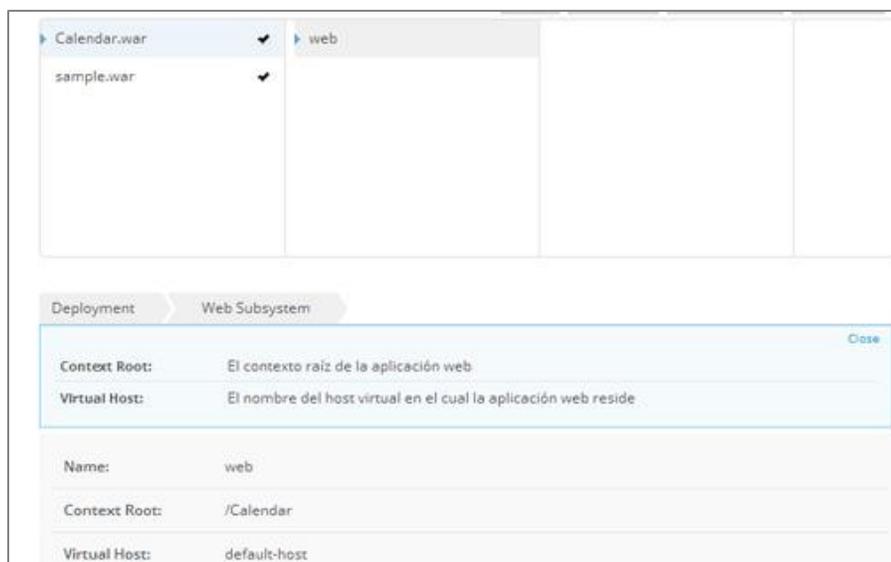


Figura 30. Configuración de la aplicación web

Instalación y Configuración del Stack ELK.

Se recomienda la instalación de Elasticsearch, Logstash y Kibana sobre un servidor que cuente con el sistema operativo Red Hat Enterprise Linux 6. Uno de los requisitos del sistema para que funcione el stack ELK, es que se cuente con una instalación de Java Runtime Environment.

A continuación se describen los pasos de instalación de Elasticsearch:

1. Instalar primero Elasticsearch. Para esto se debe ejecutar el comando siguiente, el cual descargará el archivo rpm y realizará la instalación:

```
sudo rpm -Uvh
```

```
https://download.elasticsearch.org/elasticsearch/elasticsearch/elasticsearch-1.3.2.noarch.rpm.
```

2. Configurar Elasticsearch para restringir el acceso a la instancia sobre el puerto 9200, indicándolo en el archivo “/etc/elasticsearch/elasticsearch.yml”.
3. Habilitar el servicio de Elasticsearch para que sea iniciado en el arranque del servidor:

```
sudo systemctl enable elasticsearch.
```

4. Iniciar el servicio ejecutando el siguiente comando:

```
sudo systemctl start elasticsearch.
```

A continuación se detallan los pasos para instalar y configurar Logstash:

1. Instalar Logstash con el siguiente comando:

```
sudo yum localinstall
```

```
https://download.elasticsearch.org/logstash/logstash/logstash-1.4.1-
```

```
1_bd507eb.noarch.rpm
```

2. Configurar el puerto 5000 para que Logstash reciba mensajes de log bajo el protocolo Lumberjack en el archivo “/etc/logstash/conf.d/01-lumberjack-input.conf”.
3. Configurar los filtros de mensajes de log, siguiendo el patrón de mensajes estándar de la solución, en el archivo “/etc/logstash/conf.d/10-syslog.conf”.
4. Configurar Logstash para que almacene los logs en Elasticsearch, indicándolo en el archivo “/etc/logstash/conf.d/30-lumberjack-output.conf”.

Pasos de instalación de Kibana:

1. Descargar la versión 4 de Kibana con el siguiente comando:

```
wget https://download.elasticsearch.org/kibana/kibana/kibana-4.0.1-linux-
```

```
x64.tar.gz.
```

2. Descomprimir el archivo “kibana-4.0.1-linux-x64.tar.gz” dentro del directorio “/opt/kibana”.
3. Configurar Kibana para iniciar como servicio al arrancar el servidor, en el archivo “/etc/systemd/system/kibana4.service”
4. Iniciar el servicio de Kibana con los siguientes comandos: “sudo systemctl start kibana4” y luego “sudo systemctl enable kibana4”.