



UNIVERSIDAD DON BOSCO

VICERRECTORÍA DE ESTUDIOS DE POSTGRADO

TRABAJO DE GRADUACIÓN

**PROPUESTA DE DISEÑO DE UNA PLATAFORMA DE INTEGRACIÓN DE UNA
APLICACIÓN MÓVIL DE CAPTURA DE PEDIDOS Y EL SISTEMA DE GESTIÓN DE
INVENTARIO BASADO EN TECNOLOGIA JAVA**

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

ASESOR

ING. JUAN OTHMARO MENJIVAR ROSALES

PRESENTADO POR:

ANTONIO HUMBERTO MORÁN NAJARRO

IVÁN ORLANDO ALVARADO NIÑO

ANTIGUO CUSCATLAN, LA LIBERTAD, EL SALVADOR

SEPTIEMBRE 2014

AGRADECIMIENTOS

Quiero agradecer a Dios todo poderoso por haberme concedido la oportunidad de iniciar y culminar la maestría, estoy convencido que sin su sabiduría nada hubiese sido posible, todo te lo debo a ti señor.

Agradezco a mi esposa e hijos por haberme comprendido en estos dos años en los cuales no pude dedicarles suficiente tiempo, sin su apoyo no hubiese terminada ésta carrera.

Agradezco a mi madre por haberme dado la vida y ser un ejemplo de superación a pesar de las adversidades, también te agradezco a ti papá que aunque no estás conmigo ahora, siempre fuiste para mí un ejemplo de humildad y sé que desde el cielo tú me guías siempre. .

Agradezco toda mi familia que de alguna u otra forma fueron un apoyo en estos dos años.

Agradezco a todos mis compañeros de maestría por haber compartido conmigo sus conocimientos en estos dos años y principalmente a mi compañero de tesis, creo que fueron dos años de viajes cansados pero que al final han valido la pena.

Agradezco también el tiempo y apoyo económico brindado por el consejo de directores y mis superiores de la Universidad de Sonsonate, definitivamente su apoyo ha sido valioso.

Gracias infinitas a todos, que el Dios de los cielos les recompense todo lo que hicieron por mí

AGRADECIMIENTOS

A Dios por darme la sabiduría y voluntad para seguir adelante a pesar de las adversidades. A mi esposa, por su confianza, paciencia, entrega y apoyo depositado en mí para cumplir este proceso.

A mi asesor de tesis, por su apoyo y colaboración en la ejecución de este trabajo.

A mi compañero de tesis quien fue un gran apoyo en todo el proceso.

A mis amigos y amigas, gracias por su apoyo y compañía en esta gran etapa de mi vida.

Agradezco también el apoyo económico brindado por el Consejo de Directores de la Universidad de Sonsonate y especialmente al Rector Ing. Jesús Adalberto Díaz y la decana de la Facultad de Ingeniería y Ciencias Naturales Arq. Mireya Pocasangre de Díaz, quienes han sido la piedra angular de este éxito, ya que sin su apoyo nada de esto sería posible.

Gracias infinitas a todos, que el Dios de los cielos les recompense todo lo que hicieron por mí

INDICE

1.	INTRODUCCIÓN.....	2
2.	MARCO TEÓRICO	3
2.1	DESCRIPCIÓN DE LA EMPRESA.	3
2.2	ESTRUCTURA JERÁRQUICA DE LA EMPRESA	3
2.3	DIAGNÓSTICO DE LA SITUACIÓN ACTUAL.....	4
2.4	ESTRATEGIAS DE INTEGRACIÓN DE SISTEMAS.	4
2.4.1	Soluciones Loosely Coupled.	5
2.4.1.1	<i>Arquitectura orientada a servicios.....</i>	5
2.4.1.2	<i>Web Services.</i>	6
2.5	JUSTIFICACIÓN DE LA ESTRATEGIA DE INTEGRACIÓN.	7
2.6	METODOLOGÍAS DE DESARROLLO ÁGIL.....	7
2.6.1	Metodologías ágiles.....	8
2.6.1.1	<i>SCRUM.....</i>	8
2.6.1.2	<i>Programación Extrema (Xp).....</i>	9
2.6.1.3	<i>AUP (RUP ÁGIL).....</i>	9
3	JUSTIFICACIÓN METODOLÓGICA	10
4	PROPUESTA DE SOLUCIÓN.....	12
4.1.1.	Fase de inicio.....	12
4.1.1.1.	<i>Visión/alcance del proyecto.....</i>	12

4.1.1.1.1 <i>Introducción.</i>	12
Alcance	12
4.1.1.2 Stakeholders.	13
4.1.1.3. Especificación de Requerimientos.	13
4.1.1.3.1 <i>Requerimientos de Usuarios.</i>	13
Requerimientos funcionales	13
Requerimientos no funcionales	14
4.1.1.3.2 <i>Requerimientos Operativos.</i>	16
Software de Servidores	16
Software de Clientes.	17
Harware de Servidores	17
<i>Hardware de Clientes.</i>	19
<i>Infraestructura De Red.</i>	19
<i>Recurso Humano</i>	20
4.2. FASE DE ELABORACION.	20
4.2.1. Mejora en los Procesos.	20
4.2.2. Diagramas se Casos de Uso.	22
4.3. FASE DE CONSTRUCCION.	23
4.3.1. Propuesta del diseño arquitectónico.	23
4.3.1.1. Diseño de Clases.	25

4.3.1.1.1	<i>Diseño de Clases del Aplicativo.....</i>	25
4.3.1.1.2	<i>Diseño de clases del Servicio Web de Integración</i>	26
4.3.1.2.	<i>Descripción de los componentes de la arquitectura.....</i>	27
4.3.1.3.	<i>Arquitectura de Vista Lógica.....</i>	35
4.3.1.3.1	<i>Flujos de trabajo del servicio web de integración.</i>	35
4.3.1.4.	<i>Vista de Implementación.....</i>	45
4.3.1.4.1	<i>Diagramas de Secuencia</i>	45
	<i>Diagrama de secuencia del aplicativo.....</i>	45
	<i>Diagrama de secuencia del servicio web</i>	48
4.3.1.4.2	<i>Diagrama lógico de la base de datos.....</i>	50
4.3.1.4.3	<i>Diagrama físico de la base de datos.</i>	51
5	CONCLUSIONES	52
6	REFERENCIAS	53
	ANEXO 1: DISEÑO DE INTERFACES GRÁFICAS.	54
	ANEXO 2: ESTUDIO COMPARATIVO SOBRE SCRUM, XP Y RUP.	62

RESUMEN

Este documento describe, a través de la exploración de la problemática identificada en el proceso de venta de productos de la Distribuidora B&D S.A de C.V., los elementos necesarios a tomar en cuenta para la solución, a través de una propuesta de diseño de una plataforma de integración del software de captura de pedidos y el sistema de gestión de inventario, basado en tecnología JAVA. Y recomienda además la forma en que los componentes de la arquitectura interactuarán entre sí, permitiendo que este constituya un modelo para la construcción de la aplicación.

Con la propuesta de diseño de la plataforma de integración del aplicativo de captura de pedidos y el sistema de gestión de inventario, será posible el acceso a los Web Services en telefonía móvil con plataforma Android, evaluando las características de la plataforma Android, las tecnologías involucradas en el desarrollo de Web Services y las APIs necesarias para llevar a cabo la implementación del sistema.

1. INTRODUCCIÓN

Distribuidora B&D S.A. de C.V., empresa dedicada a la distribución de productos comestibles por mayoreo de marcas como Nestlé y Kern's, está impulsando el modelo de preventa para organizar los pedidos y entrega de productos.

En el presente documento se explicará la problemática que existe en ejecutar el proceso de captura de pedidos y las necesidades de integrar la información al sistema de inventario y facturación.

La propuesta de integración de sistemas y modelo de captura de pedidos requiere de la aplicación metodológica de desarrollo de software, para lo cual se utilizará una metodología de desarrollo ágil. En el área de Ingeniería de Software, el término metodología (Pressman, 2005) se refiere a un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas computacionales.

Las Metodologías Agiles, presentan respuestas rápidas y efectivas al cambio; tienen un plan de proyecto flexible, y muestran simplicidad, de manera general, en el desarrollo.

Así, se espera que al utilizar una metodología para el desarrollo del software, ésta pueda proveer un conjunto de prácticas y herramientas que agilicen el proceso de desarrollo, ofreciendo un producto con alta calidad y que satisfaga las expectativas del cliente.

En la sección del Marco Teórico se hace un estudio sobre las diferentes estrategias de integración de sistemas y las principales metodologías de desarrollo ágil.

La sección de Justificación Metodológica, explica a través de un análisis comparativo cual es la metodología de desarrollo que más se adecua al proyecto.

La sección de la propuesta evalúa a fondo los requerimientos para establecer una propuesta arquitectónica que dé respuesta a las necesidades.

2. MARCO TEÓRICO

2.1 DESCRIPCIÓN DE LA EMPRESA.

Distribuidora B&D S.A. de C.V., empresa dedicada a la distribución de productos comestibles está utilizando el concepto de preventa para posicionar los productos de marcas como Nestlé y Kern's, al por mayor, a negocios de la zona occidental del país.

Distribuidora B&D S.A. de C.V. es una empresa que se dedica a la distribución de productos comestibles en la zona occidental del país, fue creada en enero de 2004.

Misión.

“Distribuir productos comestibles en todas las empresas de Sonsonate, y la zona sur de Ahuachapán”.

Visión:

“Consolidarse en la zona occidental como una empresa líder en la distribución de productos alimenticios de las mejores marcas”.

2.2 ESTRUCTURA JERÁRQUICA DE LA EMPRESA

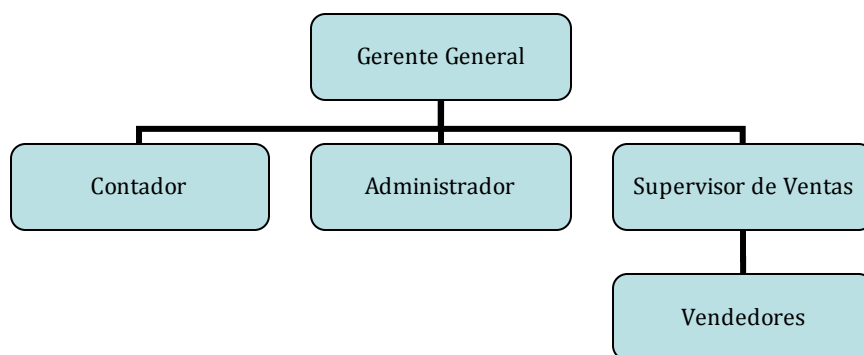


Figura 1: Estructura Jerárquica de B&D

2.3 DIAGNÓSTICO DE LA SITUACIÓN ACTUAL.

La estrategia de Distribuidora B&D S.A. de C.V. es enviar a los vendedores con rutas asignadas, y registrar los pedidos en formatos pre impresos y posteriormente cada vendedor digita los pedidos en un modulo del sistema de inventario y facturación, con el objetivo de calcular la cantidad de productos que en bodega se deben de preparar para el siguiente día para cada vendedor. Los vendedores rutereros no se limitan en los pedidos puesto que manejan un inventario alto y en el caso de que un producto tenga pocas existencias lo piden inmediatamente a la empresa a las empresas que ellos representan.¹

Utilizando el sistema de captura de pedidos, los vendedores tendrán la oportunidad de enviar el pedido vía internet al servidor sin necesidad de tener que llegar a la empresa, éste proceso integrará los pedidos capturados en ruta con la base de datos del sistema de inventario y facturación.

El propósito es que la aplicación utilice el plan de datos o el acceso a una red WIFI para el envío de la información al servidor. Sin embargo se ha considerado aquellas zonas geográficas en las que no hay cobertura de red o acceso a una red inalámbrica, en ese caso la aplicación guarda localmente el pedido y lo mantendrá en cola hasta que exista cobertura en la red.

2.4 ESTRATEGIAS DE INTEGRACIÓN DE SISTEMAS.

La integración de sistemas se refiere a la creación de uniones más firmes entre los diferentes sistemas de información basados en computadoras y bases de datos. La integración de sistemas se exige normalmente para lograr la integración de los negocios (SEI & O'Brien, 2014). Existen vínculos entre los sistemas y la integración del negocio, no obstante esa alineación representa un

¹ Hacer un diagnóstico de la situación actual sobre el proceso de captura de pedidos era el primer objetivo del anteproyecto.

reto. A continuación se mencionan algunas de las estrategias que algunas compañías han planteado para la integración de sistemas y del negocio. Las estrategias de integración de mayor importancia se citan a continuación:

- ERP (Planeación de recursos empresariales).
- Datawarehouse.
- Middleware. (RMI, JMS)
- EAI .
- eBussines.
- Soluciones loosely coupled.
 - Arquitectuta orientada a servicios.
 - WEBSERVICES.

2.4.1 Soluciones Loosely Coupled.

Las soluciones loosely coupled persiguen integrar sistemas de forma abierta y escalable. A continuación se describen los mecanismos de implementación.

2.4.1.1 *Arquitectura orientada a servicios.*

El término arquitectura orientada a servicios (SOA) representa la forma en que los Web Services son descritos y organizados de tal forma que servicios en la red se puedan poner a disposición de forma dinámica y con capacidad de ser descubiertos de forma automática.

SOA, es una forma lógica de diseñar sistemas de software que provean servicios para aplicaciones de usuario final o para otros servicios distribuidos en una red a través de interfaces publicadas con capacidad de ser descubiertas. (Marinelli & Kuna, 2007).

2.4.1.2 Web Services.

Los Web Services, permiten la formación de bloques de construcción para crear aplicaciones distribuidas que pueden ser publicadas y accedidas en el Internet. Son aplicaciones independiente de la plataforma y puede ser descrita, publicada, descubierta, orquestada y programada a través de artefactos XML, con el objetivo de desarrollar aplicaciones distribuidas interoperables. (SysperTec, 2013).

Los WEB SERVICES son aplicables a múltiples tecnologías, en Java para integrar los artefactos de XML es necesario implicar la librería JAX (Java API for XML) y JAXP (Java API for XML Processing), la cual permite hacer procesamiento y validación de XML.

Existe una gran variedad de servidores de aplicaciones para JAVA, uno de los mas extendidos es el GlassFish, el cual ofrece un servicio de vinculación (Binding) con aplicaciones cliente (JAXB, Java Architecture for XML Binding).

Arquitecturas involucradas:

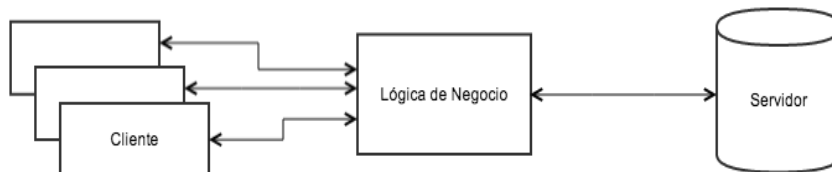


Figura 2: Arquitectura de tres capas de un Web Services

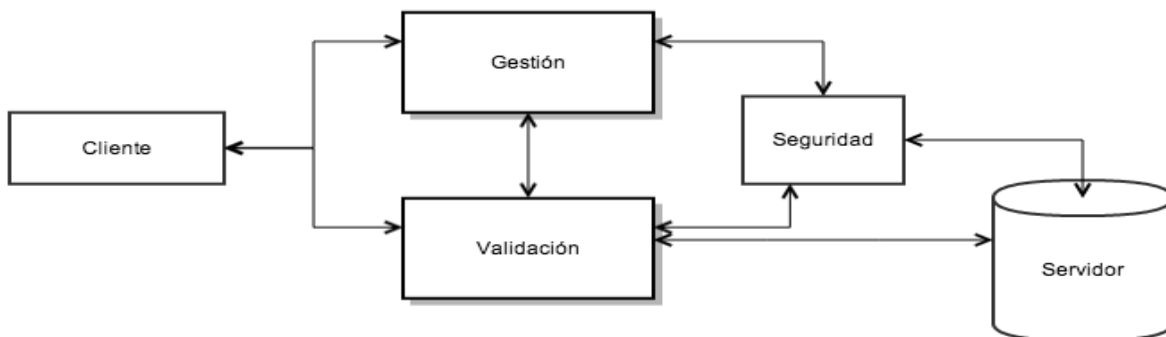


Figura 3: Arquitectura de N capas de un Web Service

2.5 JUSTIFICACIÓN DE LA ESTRATEGIA DE INTEGRACIÓN.

Una característica importante de la tecnología de los WEB SERVICES, es que permite que los sistemas puedan exponer sus servicios, utilizando la tecnología estándar de internet, para cualquier otro tipo de sistema que lo necesite sin que sea necesaria su modificación.

Dadas las características del problema en estudio, se requiere integrar un aplicativo diseñado para dispositivos móviles con un sistema legacy. En el que se usan tecnologías completamente diferentes, la tecnología de los WEB SERVICES son la mejor alternativa para compartir información. Ya que el acceso a internet a través del aplicativo móvil es respaldado por la dirección de la empresa.

Por otro lado los WEB SERVICES cuentan con la ventaja de que están soportados en forma nativa por cualquier plataforma, lo cual permite una solución de bajo costo y fácil de implementar.

2.6 METODOLOGÍAS DE DESARROLLO ÁGIL.

Las metodologías de desarrollo ágil según Alistair Cockburn son un conjunto de reglas ligeras pero suficientes para el comportamiento en el desarrollo de un proyecto orientadas a las personas y a la comunicación. (Mendoza Sanchez, 2004). Algunas de las principales metodologías se listan a continuación:

- Crystal clear
- Adaptative Software Development
- Feature Driven Development.
- Dynamic Systems Development Method
- SCRUM

- Extreme Programming
- AUP basado en RUP

Los objetivos que persiguen las metodologías ágiles, están enfocados en el desarrollo de software.

- Producir la primera entrega en semanas y disponer de retroalimentación rápida.
- Inventar soluciones simples, para que haya menos cambios y los que se hagan sean más fáciles.
- Hacer pruebas continuas para una detección de defectos.

2.6.1 Metodologías ágiles.

Las principales metodologías que se sometieron a evaluación en el presente trabajo son:

2.6.1.1 SCRUM.

La metodología SCRUM es simple, no se basa en seguimiento de planes, sino en la adaptación continua de las circunstancias de evolución del proyecto, es considerada como modelo ágil por la Agile Alliance.

Propone un enfoque en equipo por lo que se afirma que está orientado a las personas y enfatiza valores y práctica de gestión. Emplea la estructura de desarrollo ágil (incremental basada en iteraciones y revisiones). (Palacio, 2014)

Por estar enfocada en el trabajo en equipo requiere constantes reuniones que se clasifican de la siguiente manera:

- Daily SCRUM, son reuniones cortas que se celebran a diario en la que solo los involucrados participan.

- Scrum de SCRUM, permite poner en común las actividades a ejecutar, sobre todo en las áreas de integración.
- Sprint planning meeting, se celebra cada 15 o 30 días, se utiliza para conformar nuevos equipos de trabajo que atenderán requerimientos específicos.
- Sprint review meeting, reunión para revisar el trabajo completado y el pendiente.
- Sprint retrospective, reunión para evaluar la entrega de cada sprint (tiempo en el que se lleva a cabo el trabajo), sobre el cual todos emiten su opinión.

2.6.1.2 Programación Extrema (Xp).

Es una metodología que surgió como respuesta a los problemas derivados de los cambios en los requerimientos. Se basa en la simplicidad, la comunicación y la reutilización de código desarrollado. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. (Letelier & Penadés)

Existen fases bien marcadas que se listan a continuación:

- Planificación
- Diseño
- Desarrollo
- Pruebas.

2.6.1.3 AUP (RUP ÁGIL).

El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Rational Unified Process, tiene sus orígenes en el modelo espiral de Barry Boehm (Ambler, 2014)

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

3 JUSTIFICACIÓN METODOLÓGICA

Tradicionalmente, los procesos de desarrollo de software llevan asociado un marcado acento en el control del proceso. Definen actividades, artefactos y documentación a producir, herramientas y notaciones a ser utilizadas, orden de ejecución de las actividades, entre otras definiciones. Si bien existen varios procesos de desarrollo. La mayoría se derivan del modelo de cascada propuesto por Boehm.

Estos procesos, denominados tradicionales, han demostrado ser efectivos. No obstante en proyectos en los que se tienen tiempos cortos de desarrollo y requerimientos cambiantes resultan ineficientes. Como alternativa a los métodos tradicionales de desarrollo, surgen las metodologías ágiles. Manteniendo prácticas esenciales de las metodologías tradicionales, las metodologías ágiles se centran en otras dimensiones del proyecto, como por ejemplo: la colaboración con los usuarios durante todas las etapas del proceso de desarrollo, y el desarrollo incremental de

software con iteraciones muy cortas que entregan una solución a la medida. Las prácticas ágiles están especialmente indicadas para productos cuya definición detallada es difícil de obtener desde el comienzo.

Sobre las metodologías ágiles hay un estudio desarrollado por Karla Medes Calo, Elsa Estevez y Pablo Fillottrani, en el que a través de un framework de evaluación miden en que forma las metodologías ágiles cumplen con los postulados del Manifiesto Ágil. A este efecto, el marco de trabajo define medidas que satisfacen la Teoría Representacional de la Medición a través de intervalos. (Mendez Calo, Estevez, & Fillottrani)

En el anexo 2 se presenta un estudio comparativo sobre SCRUM, XP y RUP. Todos los criterios analizados dan ventaja a la metodología RUP. Los principales están referidos al modelado con casos de uso y UML, ya que RUP los integra de forma nativa. Luego de revisar los criterios, y de revisar aspectos estratégicos que proveen funcionalidad incremental, se decide utilizar la metodología RUP. Se ha tomado en cuenta además la de limitación de compromisos por parte del equipo desarrollador, ya que el Manifiesto Ágil abre las posibilidades en la modificación de requerimientos e impacta el cronograma de presentación de la propuesta. RUP, limita la modificación de requerimientos en relación a SCRUM y XP y por ello resulta conveniente utilizar una metodología que permita cumplir el avance del proyecto.

Otro aspecto importante en la determinación de la metodología es el uso de UML, ya que, en el país se encuentra fuertemente difundido y permite modelar el comportamiento y la estructura de un sistema. En RUP se encuentra integrado de forma nativa.

4 PROPUESTA DE SOLUCIÓN.

En razón de las valoraciones especificadas en la justificación metodológica, la propuesta de integración se desarrollará con la metodología RUP/AUP. Las secciones que se describen a continuación corresponden a los componentes de dicho método aplicables a fase de análisis y diseño

4.1.1. Fase de inicio.

4.1.1.1. *Visión/alcance del proyecto.*

4.1.1.1.1 *Introducción.*

El propósito de desarrollar una propuesta de integración del aplicativo de inventario/facturación y el aplicativo de gestión de pedidos, es analizar y definir las características necesarias para diseñar una propuesta de software que permita el control del inventario en la captura de pedidos y posterior facturación, está centrado en ofrecer a los proveedores la posibilidad de capturar el pedido de forma digital en dispositivos móviles y que este sea enviado al sistema de inventario/facturación.

Alcance

Para el desarrollo del proyecto es necesario tener en cuenta que lo que se desea es agilizar el proceso de gestión de pedidos a través de una propuesta de integración de sistemas.

- Desarrollar un mecanismo que permita la comunicación entre el aplicativo del dispositivo móvil y el sistema de inventario y facturación.
- Desarrollar el aplicativo para la gestión de pedidos en el dispositivo móvil.
- Implementar medidas de seguridad que garanticen la fidelidad de los datos almacenados.

4.1.1.2 Stakeholders.

Stakeholders: Los representantes de los usuarios y portavoces de las necesidades de la empresa son los stakeholders. Para el caso de aplicación solamente se ha tratado con un stakeholder como representante de los usuarios y necesidades de la empresa. El cuál es el Ingeniero Ricardo Humberto Bermúdez López, quién es el gerente general de la empresa y es el que maneja las necesidades del sistema.

4.1.1.3.Especificación de Requerimientos.

4.1.1.3.1 Requerimientos de Usuarios.

Requerimientos funcionales

Todo proyecto de desarrollo de software debe contemplar la identificación de los requerimientos funcionales y como tal éste era uno de los objetivos del anteproyecto, por tanto para el Sistema para Gestión de pedidos preventiva para Distribuidora B&D, a través de dispositivos móviles basados en Android los requerimientos funcionales son:

ANÁLISIS DE REQUERIMIENTOS DE USUARIO	
1. Pedidos	
1.1	Crear Pedidos
1.1.1	Seleccionar el Cliente
1.1.2	Seleccionar los productos
1.1.3	Almacenar los datos del pedido en el dispositivo móvil
1.2	Visualizar Pedidos
1.2.1	Mostrar lista de pedidos en pantalla del móvil
2. Clientes	
2.1	Mostrar lista de clientes en pantalla del móvil
2.2	Mostrar clientes en pantalla

Tabla 1a: Análisis de Requerimientos funcionales

ANÁLISIS DE REQUERIMIENTOS DE USUARIO

3. Productos	
3.1	Mostrar Productos en pantalla del móvil
3.2	Consultar Precio y Existencia de productos en pantalla del móvil
4. Sincronizar	
4.1	Enviar Pedidos al servidor
4.2	Recibir datos de productos y clientes del servidor el móvil
5. Configurar	
5.1	Establecer los mecanismos de conexión con el servidor
5.2	Indicar el usuario y contraseña del servidor

Tabla 1b: Análisis de Requerimientos funcionales.

Requerimientos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo, estándares, etc. (Sommerville, 2005) Como su nombre lo indica son aquellos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en las interfaces del sistema.

ANÁLISIS DE REQUERIMIENTOS DE USUARIO NO FUNCIONALES	
1. Interfaz de uso fácil	
1.1.	El dispositivo donde se implementará el sistema debe ser táctil
1.2.	La interfaz debe ser sencilla
1.3.	La resolución debe adaptarse para cualquier dispositivo móvil basado en Android
1.4.	Diseñado para la versión 2.3.5 de Android
1.5.	Emitir mensajes de alerta, advertencia y de error; con el fin de orientar al usuario en el uso adecuado del sistema.
2. Seguridad	
2.1.	Validar el acceso al sistema por medio de usuario y contraseña.
2.2.	Permitir el acceso desde internet
2.3.	La información será serializada
2.4.	El protocolo de comunicación a utilizar será HTTPS
3. Rendimiento de la Aplicación	
3.1.	Proveer tiempo de respuesta ágil a cualquier proceso.
3.2.	El envío y recepción de información hacia y desde el servidor debe ser ágil.
4. Fiabilidad de la Aplicación	
4.1.	Mantener su funcionamiento aun cuando se presente cualquier error no crítico.
4.2.	Reducir la aparición de errores.
5. Interfaces de Software	
5.1.	La base de datos se guardará en el móvil en sqLite
5.2.	La información de productos y clientes se recibirá en el móvil en formato JSON
5.3.	Los pedidos serán enviados directamente a la tabla pedidos del sistema legacy
6. Calidad del Software	
6.1.	El programa podrá sincronizarse con el servidor siempre que haya una conexión a internet
6.2.	Será eficiente en el manejo de los recursos del móvil
6.3.	La aplicación debe ser fácil de aprender.

Tabla 2: Análisis de Requerimientos no funcionales

4.1.1.3.2 *Requerimientos Operativos.*

El correcto funcionamiento del software de gestión de pedidos requiere de consideraciones tecnológicas de software tanto para los servidores, como para los clientes, las cuales se detallan a continuación:

Software de Servidores

Los servidores involucrados en el proceso de registro del sistema de gestión de pedidos son: *el servidor web, de aplicaciones y de bases de datos.* Y las características de cada uno se muestran a continuación:

SERVIDOR WEB	
Característica	Software
Sistema Operativo	CentOs 5.0
Servidor web	GlassFish 3.1.2
Runtime	JDK 7 Update 3

Tabla 3: Descripción del Servidor Web

SERVIDOR DE APLICACIONES	
Característica	Software
Sistema Operativo	CentOs 5.0
Servidor de servicios web	GlassFish 3.1.2
Runtime	JDK 7 Update 3

Tabla 4: Descripción del Servidor de Aplicaciones

SERVIDOR DE DATOS	
Característica	Software
Sistema Operativo	CentOs 5.0
Sistema gestor de bases de datos	MySQL Server 6.0

Tabla 5: Descripción del Servidor de Datos

Software de Clientes

Los dispositivos que funcionarán como clientes son los teléfonos celulares (Smartphone), del personal responsable del proceso de preventa, para lo cual se recomienda el siguiente software

SOFTWARE DEL CLIENTE	
Característica	Software
Sistema Operativo	Android 2.2 (Froyo) o superior
Sistema gestor de bases de datos	SQLite 3.8

Tabla 6: Descripción del software del cliente

Hardware de Servidores

La distribuidora B&D, ya cuenta con 3 servidores para sus aplicaciones que están basadas en tecnología JAVA y MySQL. Dichos servidores disponen de las siguientes características, las cuales son suficientes para la propuesta del sistema de gestión de pedidos

Características	Servidor de aplicaciones	Servidor Web	Servidor de datos
Marca/Modelo	HP Proliant ML150 G6	HP Proliant ML150 G6	HP Proliant ML350 G7 E5620
Procesador	1x Xeon E5504 Quad Core 2.0 GHz	1x Xeon E5504 2.0 GHz	Intel Xeon E5620 Quad Core 2.4 GHz
Memoria RAM	2 GB	4 GB	8 GB
Disco Duro	500GB	500GB 6G SAS 7.2K 2.5 in DP MDL HDD	500GB 6G SAS 7.2K 2.5 in DP MDL HDD
Sistema Operativo	Windows Server 2008 R2	CentOs 5.0	CentOs 5.0
Periféricos	Teclado Mouse Puerto de red Gigabit Ethernet	Teclado Mouse Puerto de red Gigabit Ethernet	Teclado Mouse Puerto de red Gigabit Ethernet

Tabla 7: Descripción de los servidores actuales

Espacio de Disco Duro

La distribuidora B&D, ofrece productos de primera necesidad para la zona occidental del país, cuenta con 15 prevendedores que gestionan los pedidos de productos y de acuerdo a las solicitudes se establecen las rutas de entrega de producto. Se han aproximado la visita de 40 clientes diarios por ruta de cada prevendedor y cada cliente solicita un aproximado de 35 productos por pedido. A continuación se presentan las proyecciones de crecimiento a lo largo de 5 años que es el tiempo que se estima la vida útil del sistema. Tomando en cuenta que el crecimiento de clientes por preventa es del 10% anual y el de venta de productos es del 15% anual. Se presentan las siguientes proyecciones.

PROYECCIÓN DE CLIENTES Y PRODUCTOS PARA VENTA							
Periodo	Cantidad prevendedores	Visitas/Pedidos	KB por Transacción	Total KB	Cantidad de producto anual	KB	Total
Junio 2012-Mayo 2013	15	3300	2.5	8,250	5,544,000	1.8	9,979,200
Junio 2013-Mayo 2014	15	3630	2.5	9,075	6,375,600	1.8	11,476,080
Junio 2014-Mayo 2015	17	4433	2.5	11,082	7,331,940	1.8	13,197,492
Junio 2015-Mayo 2016	20	5536	2.5	13,840	8,431,731	1.8	15,177,115
Junio 2016-Mayo 2017	24	6969	2.5	17,422	9,696,491	1.8	17,453,683
Sub Totales				59,669			67,283,570
Total							67,343,239 KB 65,765MB

Tabla 8: Cálculo de la base de datos del servidor en 3 años

El espacio requerido según la proyección, es de 65,575MB es decir 12.84% de la capacidad de almacenamiento, por tanto la tecnología disponible tiene la capacidad de soportar los requerimientos planteados

Hardware de Clientes

Los equipos necesarios para la implementación del sistema básicamente son los teléfonos celulares cuyas características se describen a continuación:

TELÉFONOS	
Característica	Descripción
Red	GSM 850/900/1800/1900 – HSDPA 900/2100
Tamaño	320x480 pixels, 3.5 pulgadas o superiores
Acelerómetro	Si
Procesador	800 Mhz o superior
Navegador	Si
Wifi	Si
Plan de datos	Activo

Tabla 9: Descripción hardware(móvil) del cliente

Infraestructura De Red

La infraestructura de red, incluye redundancia en los dispositivos de conexión como enrutadores (capa 3), firewall y los switch core (capa 3), ya que la comunicación propuesta requiere confiabilidad en la conexión a internet y en el enlace de conexión a internet.

La topología de red es estrella y el cableado estructurado cuya acometida principal y los enlaces del Core implementan cable de fibra óptica multimodo y el cableado de los equipos de oficina con cable UTP cat 5e.

Recurso Humano

La distribuida B&D posee, una persona responsable de la Unidad de Informática capaz de administrar y darle seguimiento al software de gestión de pedidos. Algunas de las responsabilidades que deberá asumir en relación a la gestión del software se detallan a continuación.

- Administrar la base de datos de pedidos
- Dar seguimiento al monitor de carga de información
- Administrar la aplicación
- Brindar soporte a los usuarios del sistema
- Hacer los respaldos de datos para garantizar la disponibilidad de la información

4.2.FASE DE ELABORACION.

4.2.1. Mejora en los Procesos.

Con la propuesta de integración de sistemas, se logrará innovar el principal proceso de negocio de la empresa, ya que la implementación del aplicativo para la captura de pedidos en dispositivos móviles permitirá la captura de los pedidos en formato digital y al final del día se contará con el registro de los mismos en el sistema legacy.

SOLUCIONES A PROBLEMAS ENCONTRADOS

Problemas detectados	Propuesta de solución
Procedimientos	
Retrasos en el registro de los pedidos	Desarrollo de una estrategia de integración del sistema de pedidos en una plataforma Android con el sistema(legacy) de inventario y facturación Desarrollo de un WEB SERVICE, que permita la sincronización de clientes, productos y pedidos desde el sistema de pedidos en la plataforma Android con el sistema legacy

Tabla 10a: Problemas detectados y solución planteada

Problemas detectados	Propuesta de solución
Procedimientos	
Equivocaciones en el proceso de facturación.	Con la implementación de los WEB SERVICES y el consumo de estos en el aplicativo del dispositivo móvil. Se corrigen las equivocaciones en la captura del pedidos y como consecuencia los problemas que se generan en el proceso de Facturación.
Generación de pedidos a proveedores.	La integración permite disponer a través del Web Service, información actualizada, para que la gerencia haga los pedidos pertinentes a los proveedores.
Determinación de responsabilidades en equivocaciones de los pedidos.	Desarrollo de un log de actividades en el Web Service, que permita registrar las operaciones que los usuarios (proveedores), ejecutan en el aplicativo.
Innovación tecnológica	El desarrollo de Web Services, permite exponer la funcionalidad del sistema legacy en un mecanismo estándar y escalable.

Tabla 10b: Problemas detectados y solución planteada

4.2.2. Diagramas se Casos de Uso.

Casos de Uso: Son derivados de los requerimientos del sistema, son el resultado del análisis de las necesidades de los usuarios, para éste proyectos los casos de uso son:

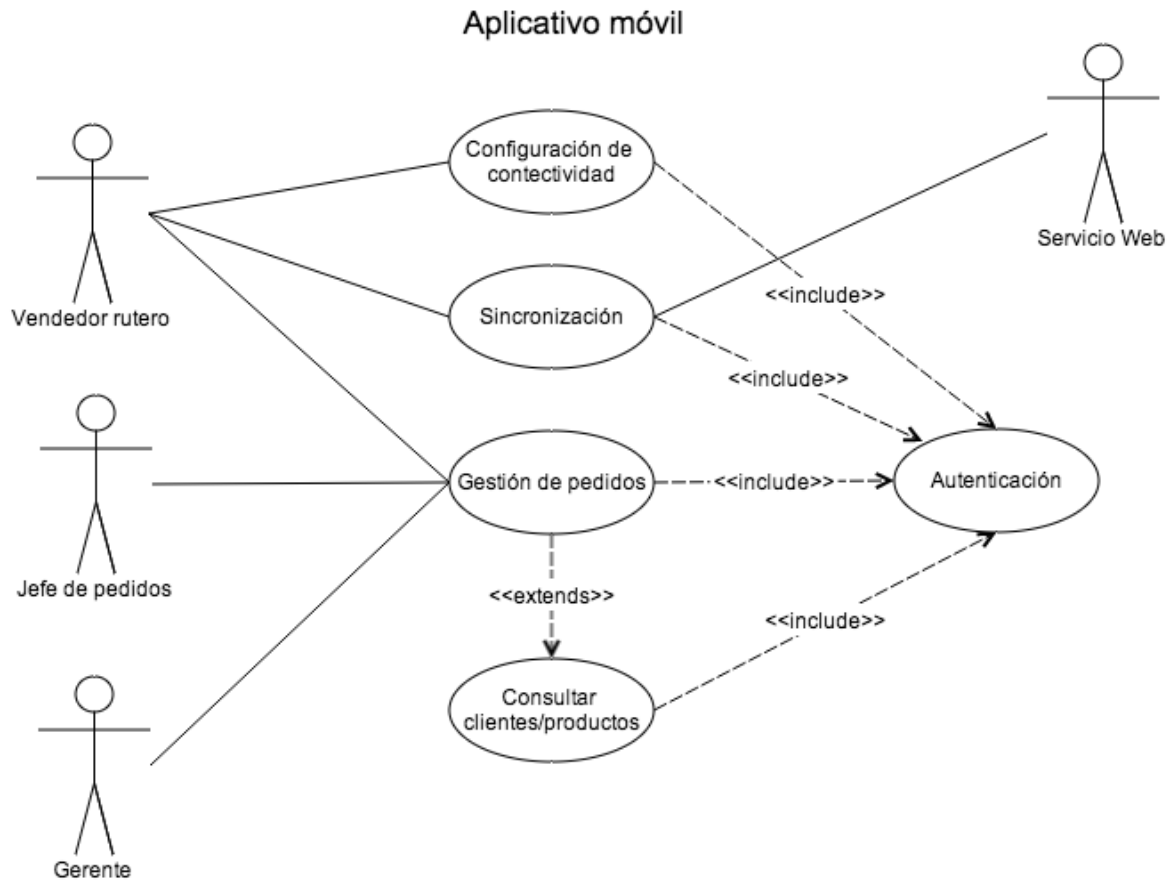


Figura 4: Diagrama de Casos de uso del aplicativo móvil

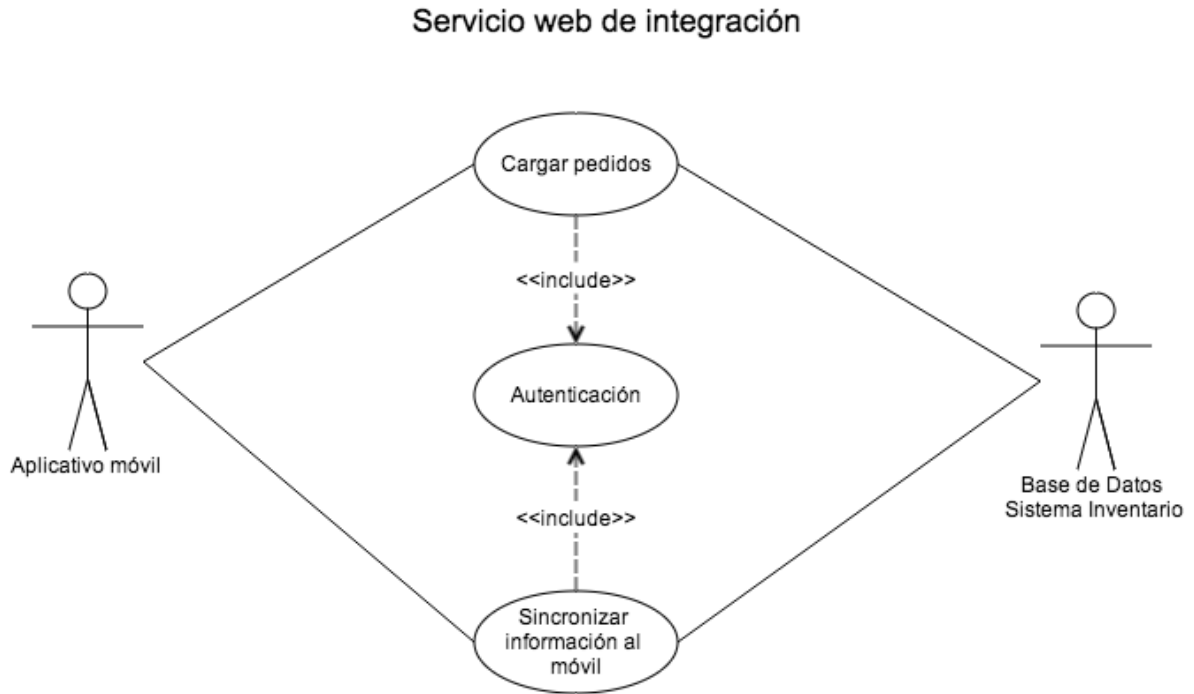


Figura 5: Diagrama de casos de uso del WEB SERVICE

4.3.FASE DE CONSTRUCCION.

4.3.1. Propuesta del diseño arquitectónico.

El modelo arquitectónico de un software consiste en diseñar un marco de trabajo básico, en donde se definen aspectos generales del software y el modo en que la estructura ofrece una integridad conceptual al sistema.

Un modelo arquitectónico simple se puede considerar que está compuesto por la estructura jerárquica de los componentes y la manera en que dichos componentes interactúan entre sí. El modelo de una solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí. En el anexo 1 se presenta el diseño de las GUI de la aplicación móvil, sin embargo para resolver el problema de integración y respondiendo a uno de los objetivos (objetivo 4) del anteproyecto, se presenta el siguiente diseño arquitectónico:

PROPUESTA DE DISEÑO ARQUITECTONICO

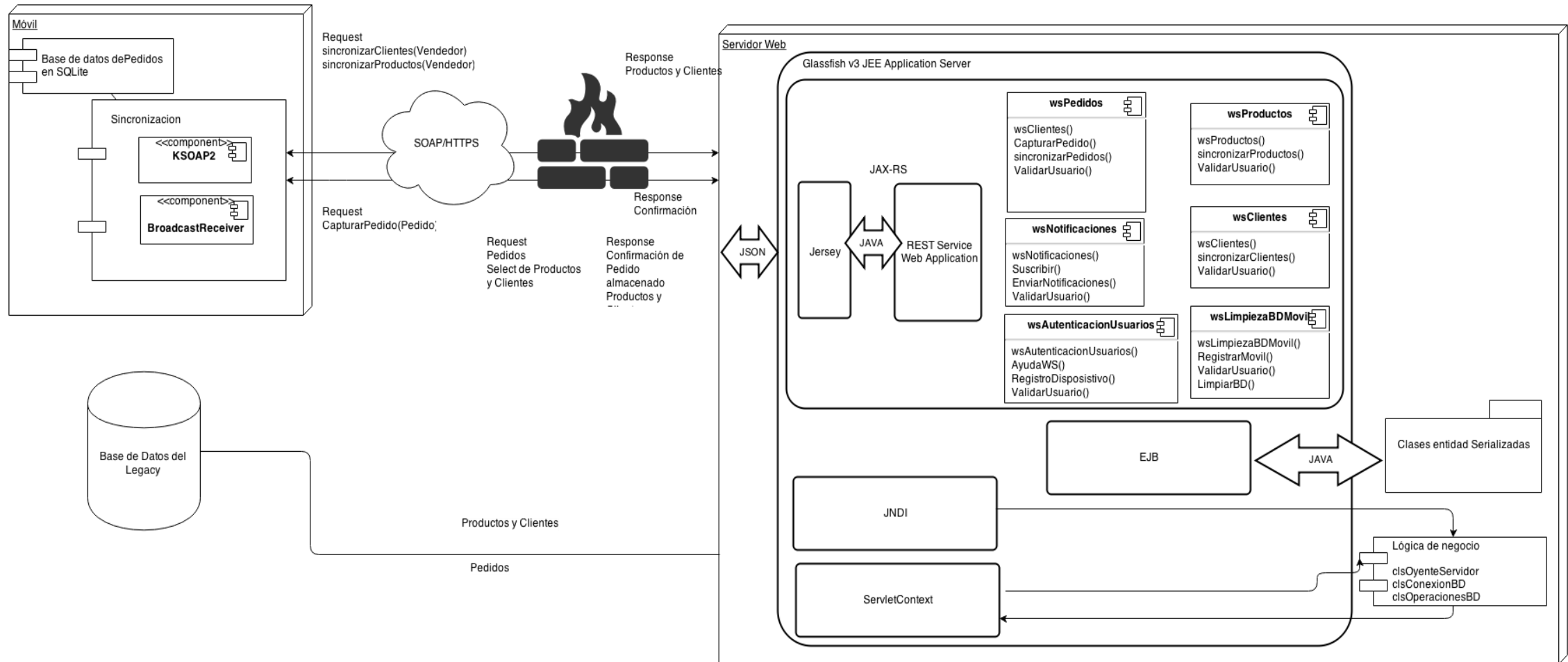


Figura 6: Diagrama Propuesta de diseño arquitectónico²

² Con la presentación de éste diseño arquitectónico y que en las siguientes páginas se describe a totalidad, se está cumpliendo los últimos tres objetivos planteados en el anteproyecto. Es decir la definición de los protocolos, los elementos de seguridad y la arquitectura de la propuesta con sus interfaces.

4.3.1.1. Diseño de Clases.

4.3.1.1.1 Diseño de Clases del Aplicativo

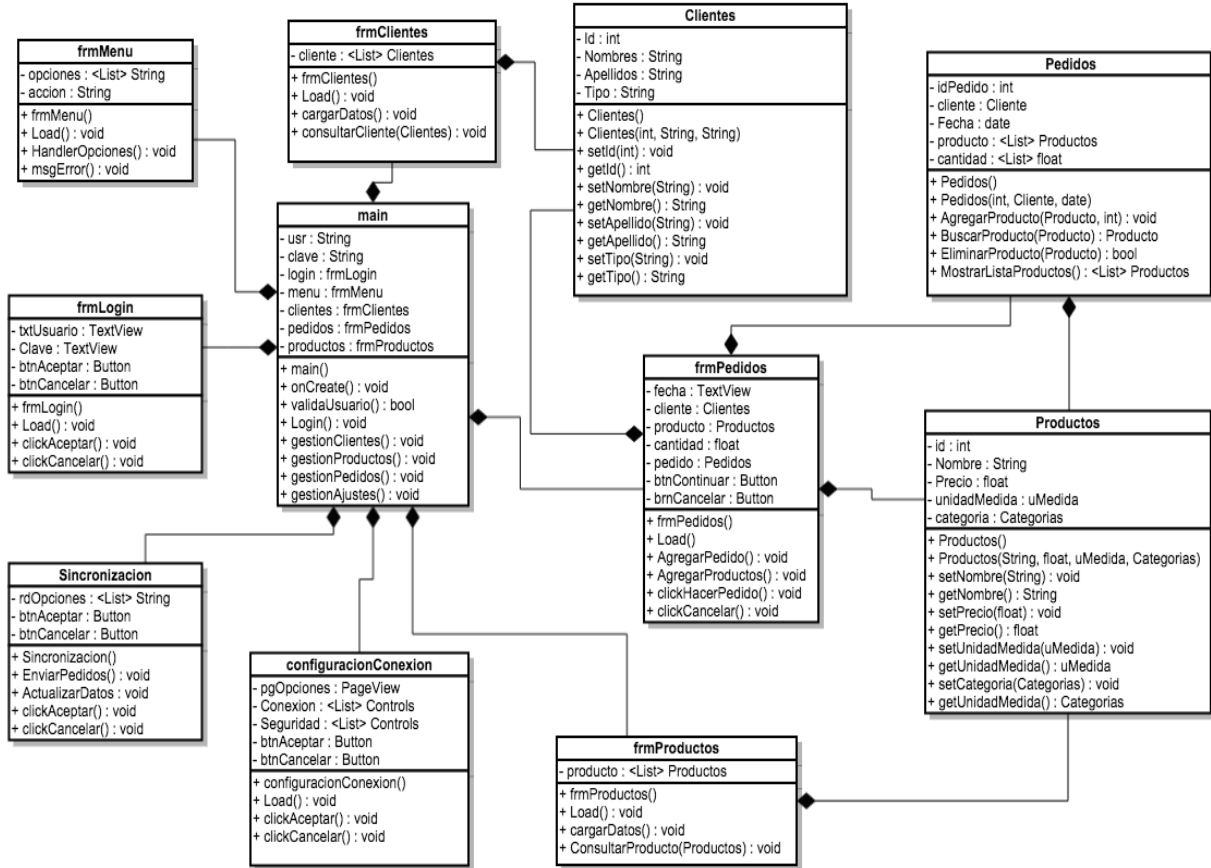


Figura 7: Diseño de Clases del aplicativo

4.3.1.1.2 Diseño de clases del Servicio Web de Integración

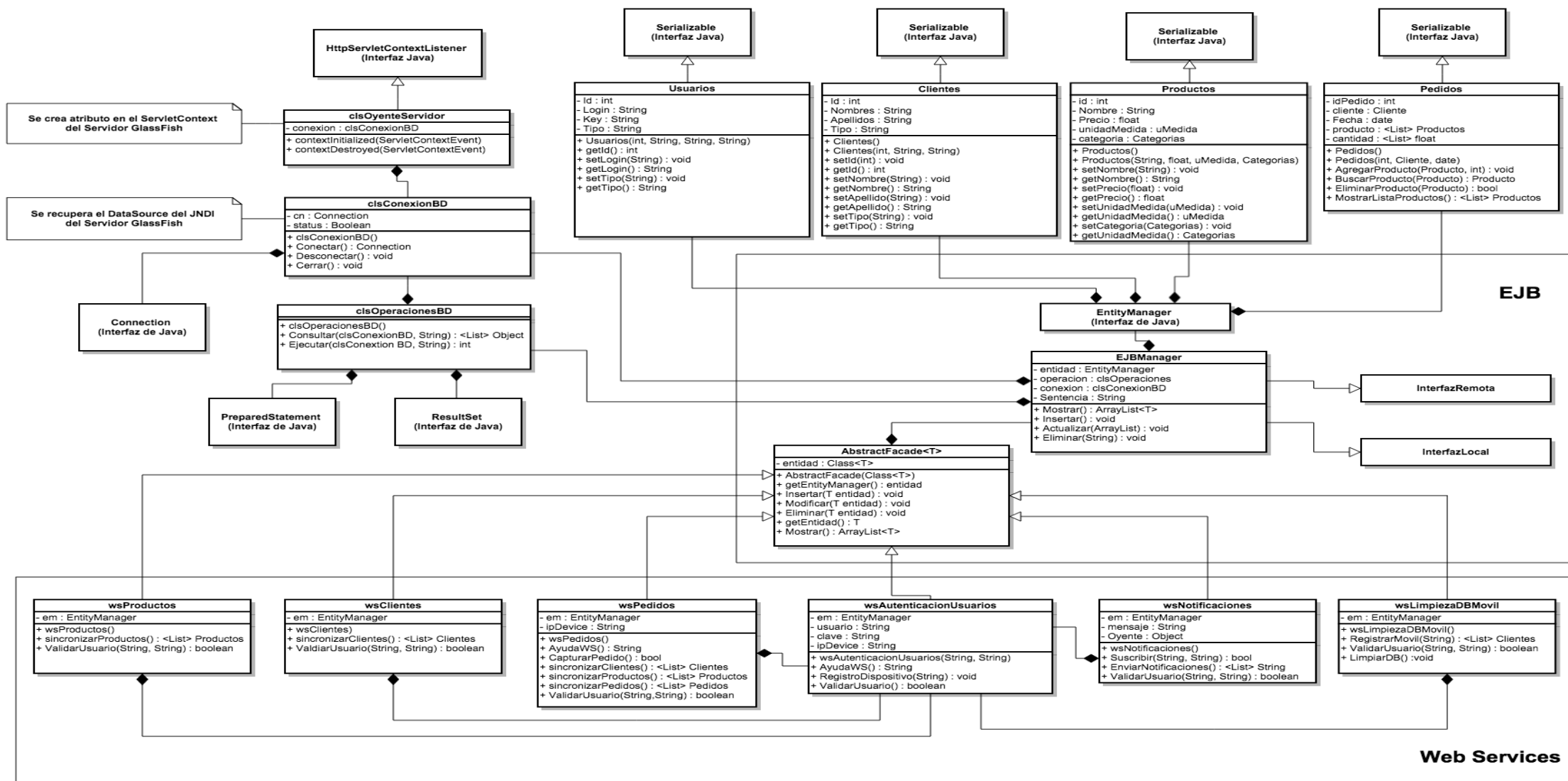


Figura 8: Diseño de Clases del servicio web de integración

4.3.1.2.Descripción de los componentes de la arquitectura.

APLICATIVO DEL MÓVIL

Nombre	Base de datos de pedidos SQLite
Tipo	Sistema gestor de base de datos.
Objetivo	Almacenar la información de productos, clientes y pedidos obtenidos a través del servicio web de integración.
Entradas	Listas de productos, clientes y pedidos.
Salidas	Pedidos
Tecnología	Librería nativa de Android
Descripción	La librería SQLite de Android permite almacenar la información solicitada al servicio web de integración .

Tabla 11: Descripción del componente Base de Datos de SQLite

Nombre	KSOAP2
Tipo	Protocolo de comunicación de datos
Objetivo	Consumir servicios web basados en SOAP.
Entradas	Cadena de strings en formato JSON serializadas.
Salidas	Cadena de string de pedidos en formato JSON serializados
Tecnología	Adaptación de W3C adaptado para Android
Descripción	La personalización de la librería para Android permite recoger los datos enviados por el servicio web de integración e interpretar el formato de JSON.

Tabla 12: Descripción del componente KSOAP2

Nombre	BroadcastReceiver
Tipo	Interfaz oyente de Android
Objetivo	Escuchar eventos originados por la conectividad
Entradas	Estado de conectividad (WIFI, Red de datos)
Salidas	Notificación del estado de la conectividad
Tecnología	Interfaz nativa de Android
Descripción	El interfaz, permite construir una clase de Java que lo implementa y así poder escuchar los eventos generados por el estado de la conectividad del móvil (Sea WIFI o Red de datos) entre otros eventos. Con ello se le da tratamiento a las pedidos y sincronizaciones que se habían solicitado en un momento en el que no se disponía de conectividad.

Tabla 13: Descripción del componente BroadcastReceiver

Nombre	JSON
Tipo	Formato de texto. JavaScript Object Notation
Objetivo	Intercambiar datos entre el aplicativo móvil y el servicio web de integración de forma ligera.
Entradas	Cadena de texto a convertir.
Salidas	Cadena de texto convertida al formato JSON.
Tecnología	JavaScript.
Descripción	Formato de texto, que permite enviar/recibir información del servicio web de integración de forma comprimida.

Tabla 14: Descripción del componente JSON

Servicio Web de Integración

Nombre	Glassfish Server
Tipo	Contenedor o servidor de aplicaciones Web para tecnología JAVA.
Objetivo	Ofrecer el servicio web de integración.
Entradas	Peticiones de tipo request.
Salidas	Respuestas de tipo response
Tecnología	Servidor de aplicaciones JAVA. J2EE
Descripción	Contenedor web, que permite desarrollar aplicaciones J2EE. Integra de forma nativa JAX-RS, Enterprise JavaBeans, JNDI

Tabla 15: Descripción del componente GlassFish Server

Nombre	Jersey
Tipo	Implementación de referencia para JAX-RS
Objetivo	Construir el servicio web de integración a partir de la implementación de JAX-RS.
Entradas	Clases implicadas en el servicio web.
Salidas	Servicio web.
Tecnología	Servidor de aplicaciones JAVA. J2EE
Descripción	Implementación de calidad de producción de Sun, la cual facilita la creación de servicios web RESTful.

Tabla 16: Descripción del componente Jersey

Nombre	JAX-RS
Tipo	API Java
Objetivo	Proporcionar el soporte para la creación del servicio web de integración.
Entradas	Acceso a la clase relativa, Tipos de peticiones HTTP de un recursos, anotaciones.
Salidas	Recursos JAX-RS y descriptores.
Tecnología	API de JAVA J2EE que responde al estilo arquitectónico Representational State Transfer (REST).
Descripción	Interfaz de programación de aplicación que permite la creación de servicios web.

Tabla 17: Descripción del componente JAX-RS

Nombre	AbstractFacade<T>
Tipo	Clase abstracta que se define para crear una capa de abstracción de los servicios web. (Correspondiente al patrón de desarrollo Facade)
Objetivo	Definir una capa de abstracción sobre los Web Services concretos.
Entradas	Objeto abstracto de tipo Class, que puede representar a cualquier clase entidad.
Salidas	Métodos que concretizan de acuerdo a la capa de persistencia las operaciones a ejecutar en la clase EJBManager.
Tecnología	Clase de Java J2EE.
Descripción	Permite crear una capa de abstracción para establecer mecanismos concretos en el Web Service. Las operaciones ejecutadas por el Web Service concreto son llamadas a través de la clase EJBManager. El patrón Facade permite proporcionar una interfaz simple para un sistema complejo.

Tabla 18: Descripción del componente AbstractFacade<T>

Nombre	REST Service
Tipo	Servicio web concreto construido a partir del API JAX-RS.
Objetivo	Ofrecer los mecanismos de sincronización de productos, usuarios, clientes, notificaciones del servidor, utilerías para limpiar la base de datos del móvil y envío de pedidos hacia el aplicativo del dispositivo móvil.
Entradas	Solicitudes de tipo GET para clientes, productos y pedidos. Y autenticación de usuarios.
Salidas	Listas de productos, clientes y pedidos serializados en formato JSON.
Tecnología	Clases de Java generadas por el API JAX-RS J2EE.
Descripción	Servicio web que permite la sincronización de pedidos, productos y clientes.

Tabla 19: Descripción del componente Servidor REST Service

Nombre	EntityManager
Tipo	Interfaz de Java perteneciente al paquete de persistencia
Objetivo	Mapeo entre una tabla relacional y su objeto Java, proporciona métodos para manejar la persistencia de un Bean de entidad.
Entradas	Clases de tipo entidad
Salidas	Capa de abstracción en el manejo de las clases entidad.
Tecnología	Interfaz de Java del paquete de persistencia.
Descripción	A través del interfaz EntityManager se puede modelar de forma abstracta las entidades de negocio del aplicativo.

Tabla 20: Descripción del componente EntityManager

Nombre	Clases Entidad Serializadas (Productos, Clientes, Pedidos, Usuarios)
Tipo	Clases de Java de tipo bean que implementan el interfaz Serializable.
Objetivo	Persistir información de la base de datos del sistema legacy. Información que se envía de forma serializada.
Entradas	Registros individuales consultados a la base de datos.
Salidas	Objetos serializados.
Tecnología	Clases bean de Java.
Descripción	Clases entidad de Java serializadas que permiten persistencia a las tablas de la base de datos del sistema legacy.

Tabla 21: Descripción del componente Clases Entidad Serializadas

Nombre	JNDI
Tipo	API para servicios de directorio (Interfaz de Nombrado y Directorio Java).
Objetivo	Permite a los clientes descubrir y buscar objetos y datos a través de un nombre.
Entradas	Nombre del objeto que almacena el pool de conexiones.
Salidas	Objeto que representa el pool de conexiones.
Tecnología	API. J2EE.
Descripción	Recurso que almacena el pool de conexiones.

Tabla 22: Descripción del componente JNDI

Nombre	ServletContext
Tipo	Interfaz de Java, que representa el contexto de la aplicación
Objetivo	Depositar atributos de tipo DataSource en el servidor cuya disponibilidad estará a nivel del contexto del servidor.
Entradas	Información para el objeto de tipo DataSource.
Salidas	Objeto de tipo DataSource
Tecnología	Clases de Java. J2EE.
Descripción	Recurso utilizado para almacenar el objeto de tipo DataSource, en el que se especifican la información de conectividad.

Tabla 23: Descripción del componente ServletContext

Nombre	HttpServletContextListener
Tipo	Interfaz de Java, que representa un oyente del servidor.
Objetivo	Captura los eventos cuando el servidor arranca o cuando se detiene.
Entradas	Un evento desencadenado por el interfaz HttpServletRequest Instancia de la clase clsConexionBD.
Salidas	Ejecución del método contextInitialized() o contextDestroyed(). Objeto de tipo conexion creado o destruido en el contexto.
Tecnología	Interfaz de Java. J2EE.
Descripción	Se crea la clase clsOyente que hereda del interfaz HttpServletContextListener, con el objetivo de crear un objeto de tipo conexion en el contexto del servidor (ServletContext). Y aprovechar la seguridad que el contenedor ofrece para administrar las conexiones a través del pool (Cola de conexiones). Cuando se ejecuta el evento contextDestroyed(), se recupera el objeto de tipo conexion y se destruye para que este no este disponible.

Tabla 24: Descripción del componente HttpServletContextListener.

Nombre	LÓGICA DE NEGOCIO. clsOyenteServidor
Tipo	Clases de Java de tipo oyente y controladores que implementan el interfaz ServletContextListener, para escuchar los eventos del contenedor web (GlassFish), y dar tratamiento especial al pool de conexiones cuando el servidor está detenido o iniciado.
Objetivo	Ofrecer mecanismos de conexión a la base de datos de forma segura. Evitando la creación de múltiples conexiones.
Entradas	Solicitud de conexión.
Salidas	Conexión satisfactoria.
Tecnología	Clases de Java. J2EE.
Descripción	Clases de Java que manejan la lógica de negocio.

Tabla 25: Descripción del componente lógica de negocio (clsOyenteServidor)

Nombre	LÓGICA DE NEGOCIO. clsConexionBD.
Tipo	Clases de Java de tipo Connection que representa la conexión al sistema gestor de base de datos.
Objetivo	Establecer conectividad a la base de datos.
Entradas	El DataSource recuperado del contexto del servidor.
Salidas	Objeto de tipo conexión.
Tecnología	Clases de Java. J2EE.
Descripción	La clase permite en el constructor recepcionar el DataSource del contexto del servidor e inicializar el atributo de tipo Connection (ubicado en el paquete javax.sql). Ofrece los mecanismos para conectar y desconectar el objeto de tipo conexión.

Tabla 26: Descripción del componente lógica de negocio (clsConexionBD)

Nombre	LÓGICA DE NEGOCIO. clsOperacionesBD.
Tipo	Clases de Java pertenecientes al paquete javax.sql.
Objetivo	Ofrecer mecanismos de ejecución de sentencias SQL sobre el objeto instancia de la clase clsConexionBD.
Entradas	Instancia de la clase clsConexionBD. Sentencia SQL.
Salidas	Ejecución de la sentencia SQL. Si es una sentencia SELECT devuelve una variable de tipo List cuyos nodos son objetos definidos en tiempo de ejecución por el EJB. Si son sentencias INSERT, UPDATE o DELETE, devuelve en una variable de tipo entero la cantidad de registro afectados por la instrucción.
Tecnología	Clases de Java. J2EE.
Descripción	La clase clsOperaciones permite ejecutar sentencias SQL. Implementando el interfaz preparedStatement (Envío de sentencias SQL precompiladas) y el ResultSet (Que representa la colección de registro resultado de una sentencia SELECT).

Tabla 27: Descripción del componente lógica de negocio (clsOperacionesBD)

Nombre	EJB. Clase EJBManager
Tipo	API que forma parte del estándar de construcción de aplicaciones empresariales.
Objetivo	Proporcionar un modelo de componentes distribuido estándar del lado del servidor para centrarse en el desarrollo de la lógica de negocio
Entradas	Clases entidad serializadas a través del EntityManager. Instancia de la clase clsOperacionesBD Instancia de la clase clsConexionBD.
Salidas	Clases Enterprise JavaBean de sesión que encapsulan las operaciones (consulta, inserción, actualización y eliminación) de cada entidad.
Tecnología	Clases de Java. J2EE.
Descripción	Permite recuperar el objeto de conexión del contexto del servidor e invoca el método conectar sobre el objeto recuperado. Con ello implementa los métodos de la clase clsOperacionesBD (Consultar(), Ejecutar()), para lanzar las sentencias SQL sobre el objeto conexión activo. El Interfaz EntityManager permite abstraer la clase entidad sobre la cual se ejecutan las sentencias SQL.

Tabla 28: Descripción del componente EJB (Clase EJBManager)

Nombre	EJB. InterfazLocal
Tipo	Interfaz del API EJB.
Objetivo	Definir los métodos del ciclo de vida del Enterprise Java Bean.
Entradas	Clase EJB de sesión, que implementará las acciones a la base de datos
Salidas	Interfaz que contiene los métodos del ciclo de vida el EJB.
Tecnología	Interfaz de Java para el manejo de persistencia J2EE.
Descripción	El interfaz define los métodos del ciclo de vida del Bean de sesión sin estado, que se usa en el aplicativo. Extiende del paquete javax.ejb.EJBHome

Tabla 29: Descripción del componente EJB (InterfazLocal)

Nombre	EJB. InterfazRemota
Tipo	Interfaz del API EJB.
Objetivo	Defini los métodos de negocio del EJB.
Entradas	Clase EJB de sesión, que implementará las acciones a la base de datos.
Salidas	Clases Enterprise JavaBean de sesión que encapsulan las operaciones (consulta, inserción, actualización y eliminación) de cada entidad.
Tecnología	Interfaz de Java para el manejo de persistencia J2EE.
Descripción	El interfaz define los métodos de negocio del Bean de sesión sin estado. Extiende del paquete javax.ejb.EJBObject.

Tabla 30: Descripción del componente EJB (InterfazRemota)

4.3.1.3.Arquitectura de Vista Lógica

4.3.1.3.1 Flujos de trabajo del servicio web de integración.

Flujo de trabajo para creación/eliminación del objeto conexión.

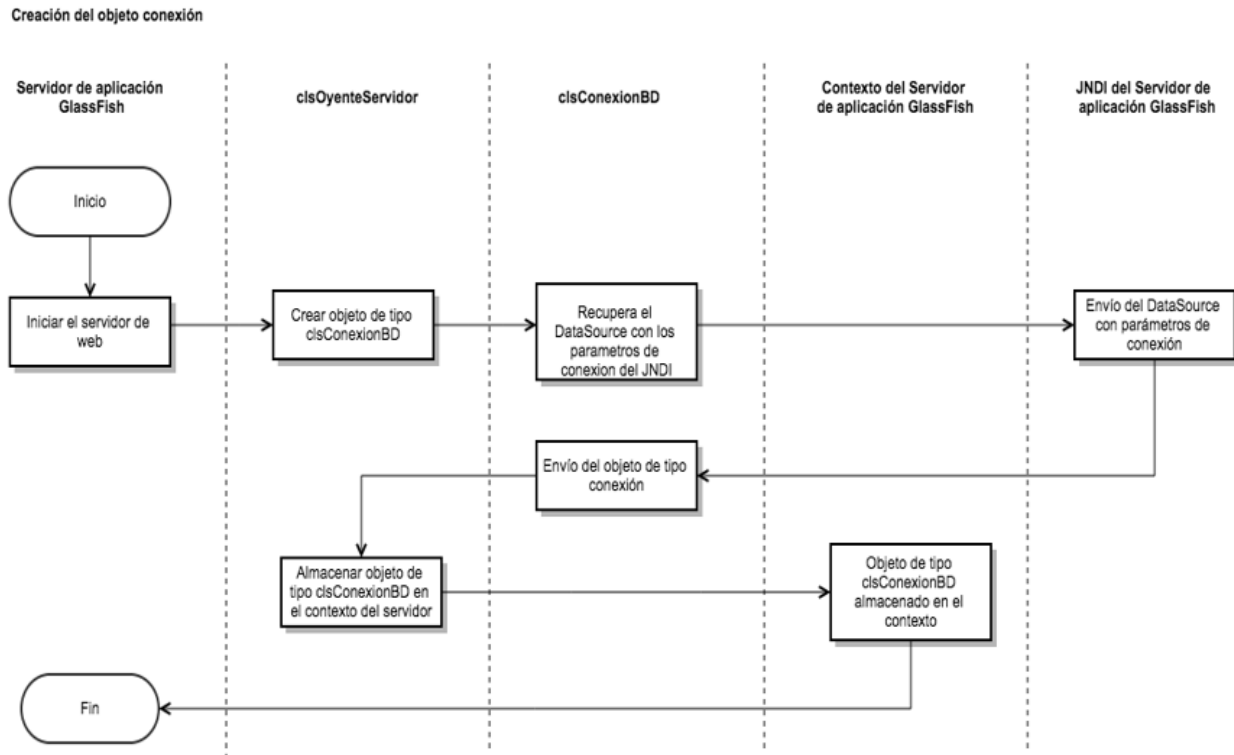


Figura 9: Flujo de trabajo para la creación del objeto conexión

Eliminación del objeto conexión

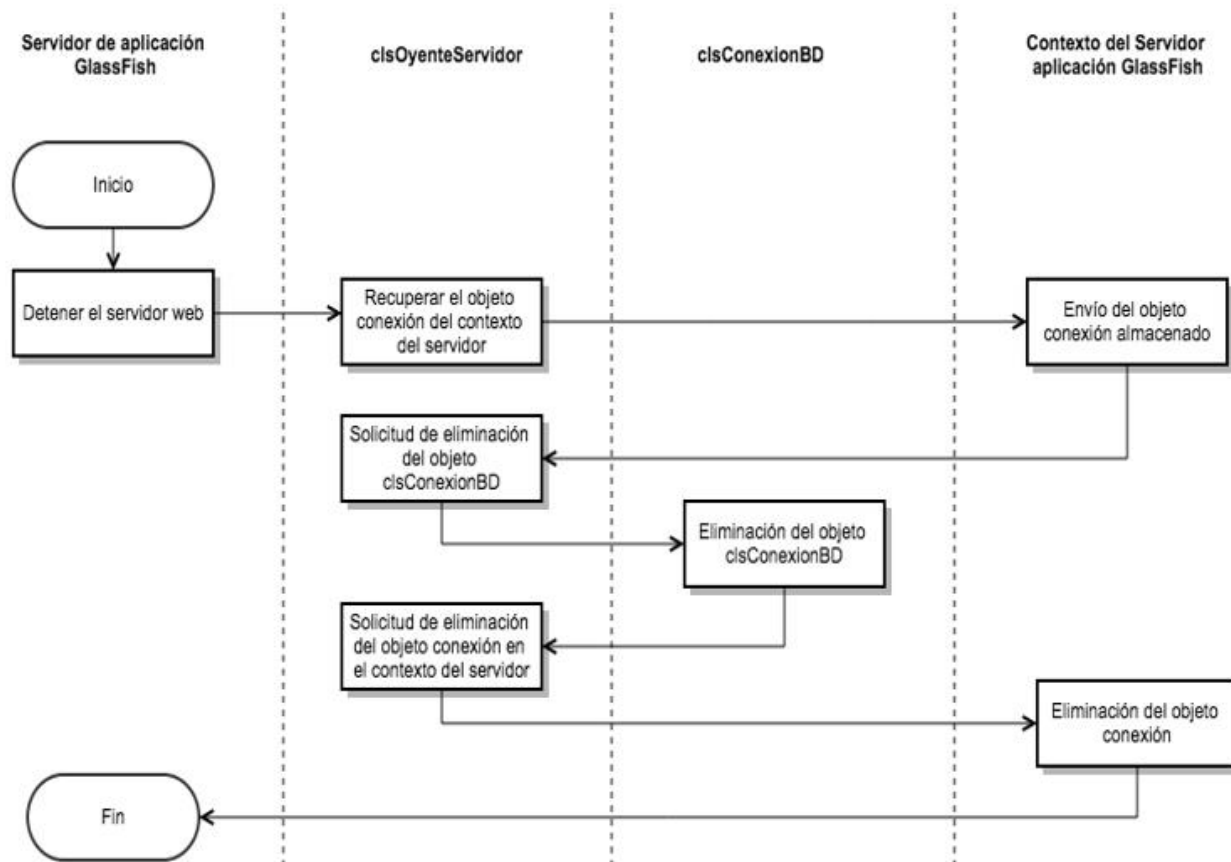


Figura 10: Flujo de trabajo para la eliminación del objeto conexión

Flujo de trabajo para invocación de los Web Services.

Web Service wsAutenticacionUsuarios

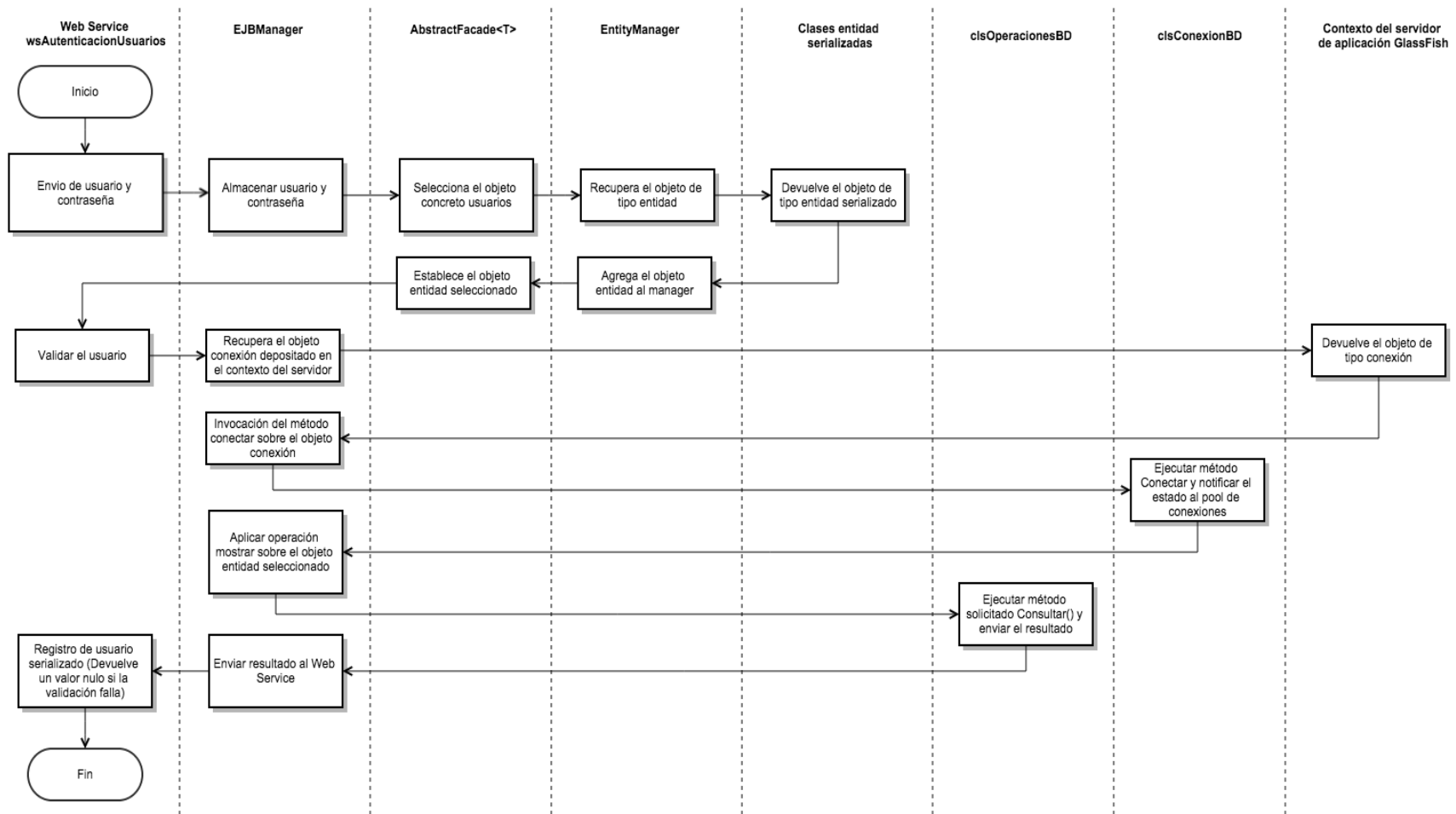


Figura 11: Flujo de trabajo para el Web Service wsAutenticacionUsuarios

Web Service wsProductos

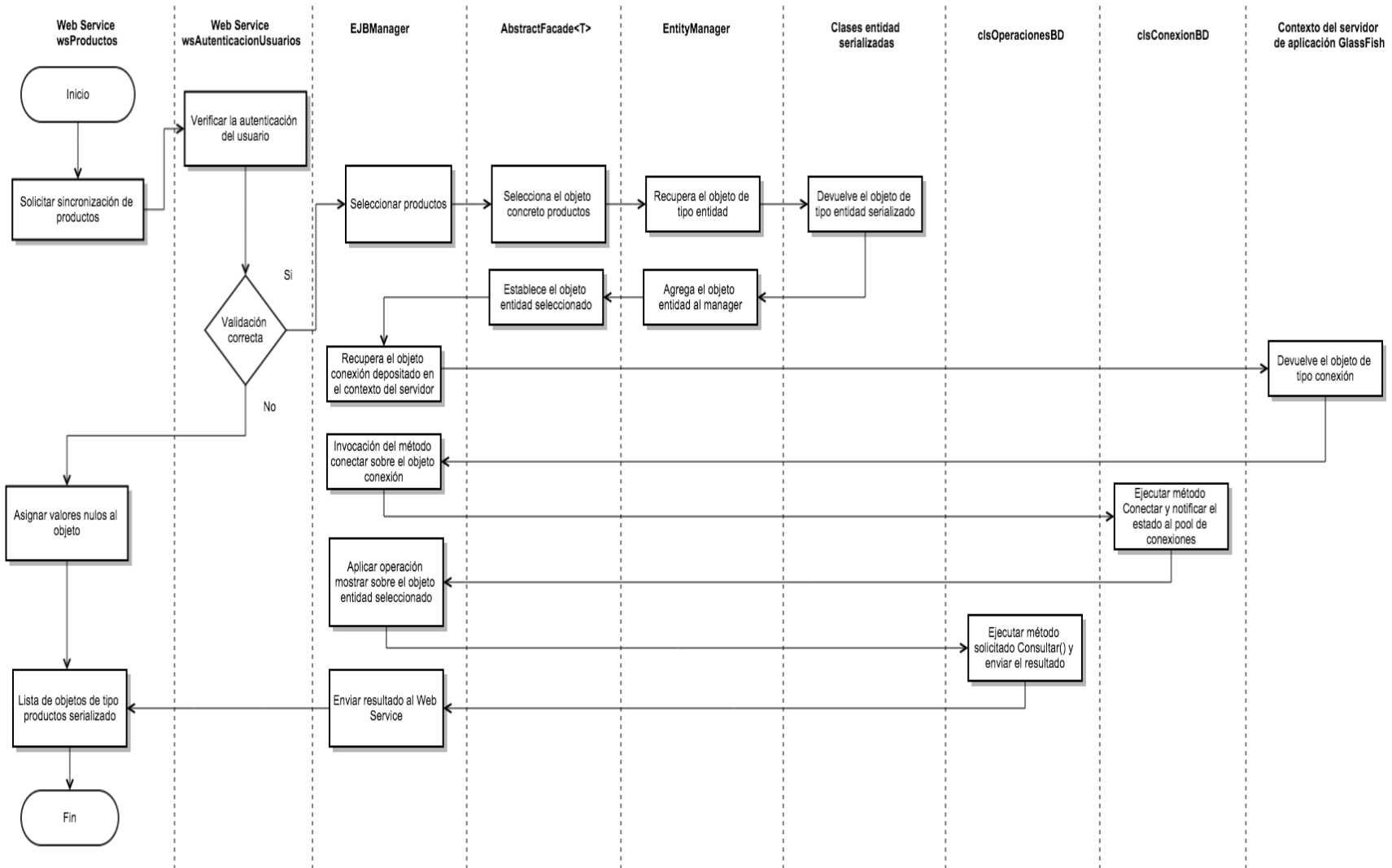


Figura 12: Flujo de trabajo para el Web Service wsProductos

Web Service wsClientes

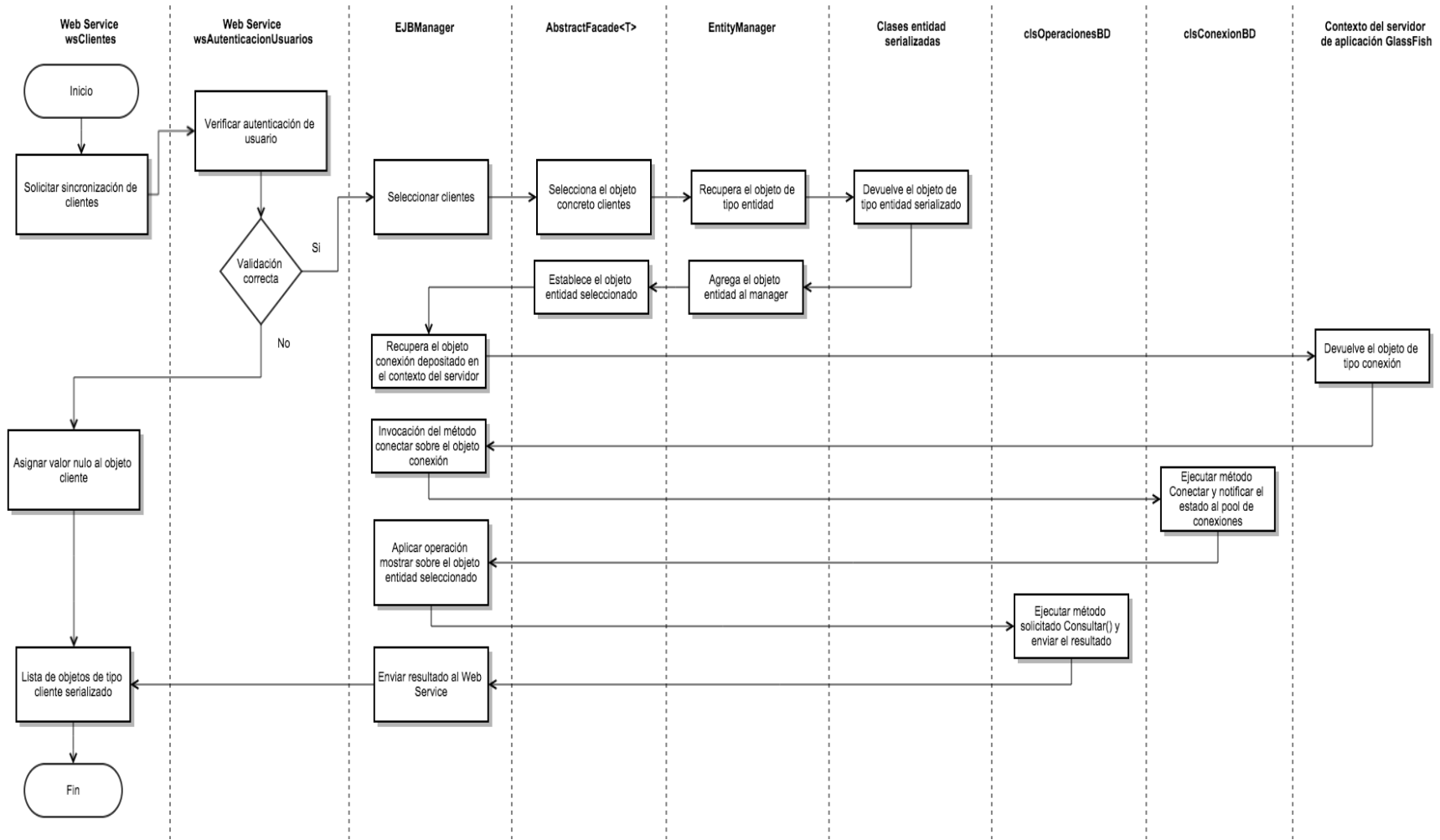


Figura 13: Flujo de trabajo para el Web Service wsClientes

Web Service wsPedidos proceso sincronización de pedidos

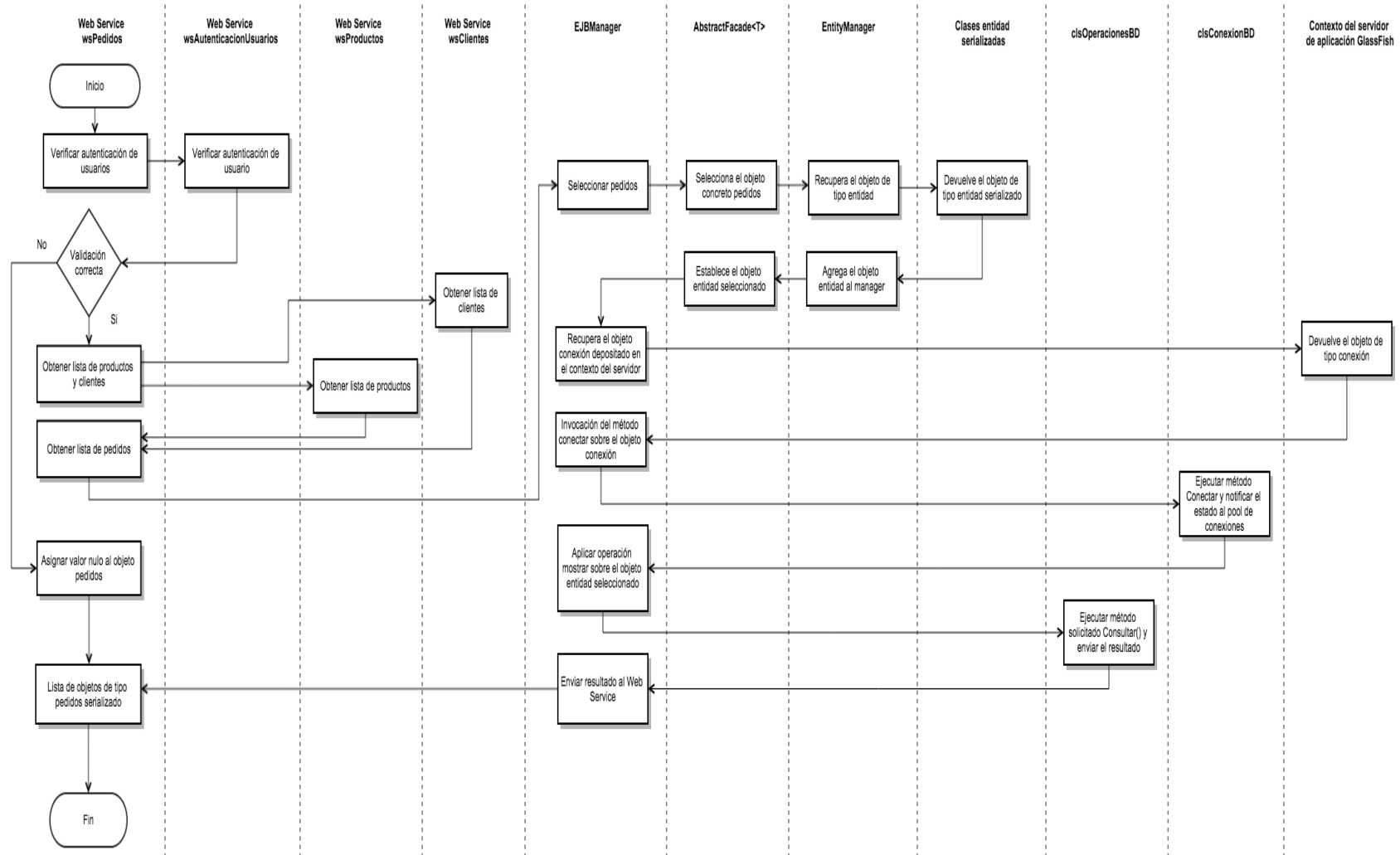


Figura 14: Flujo de trabajo para el Web Service wsPedidos(sincronización de pedidos)

Web Service wsPedidos proceso captura de pedidos

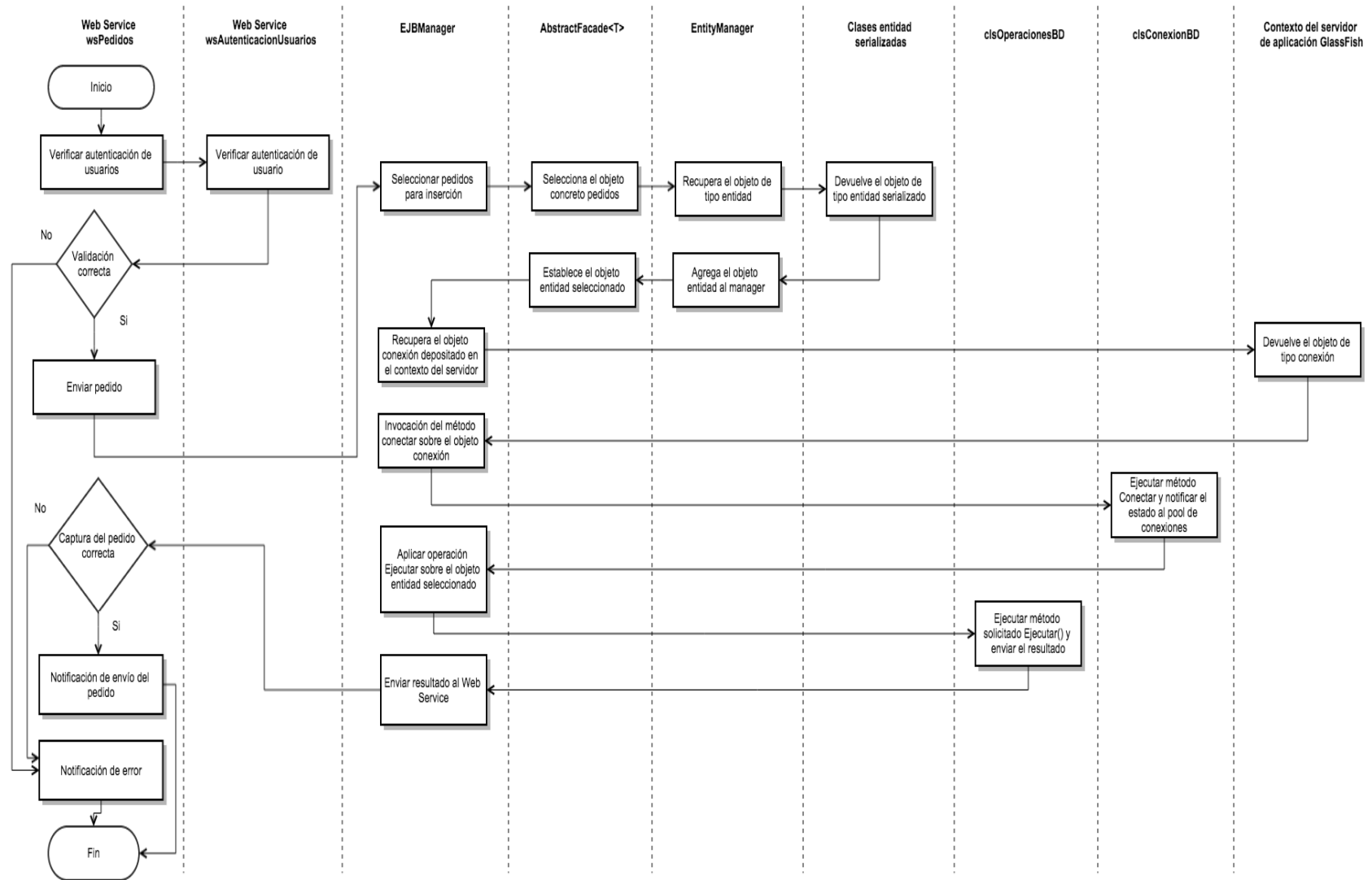


Figura 15: Flujo de trabajo para el Web Service wsPedidos(proceso captura de pedidos)

Web Service wsNotificaciones proceso suscripción a las notificaciones

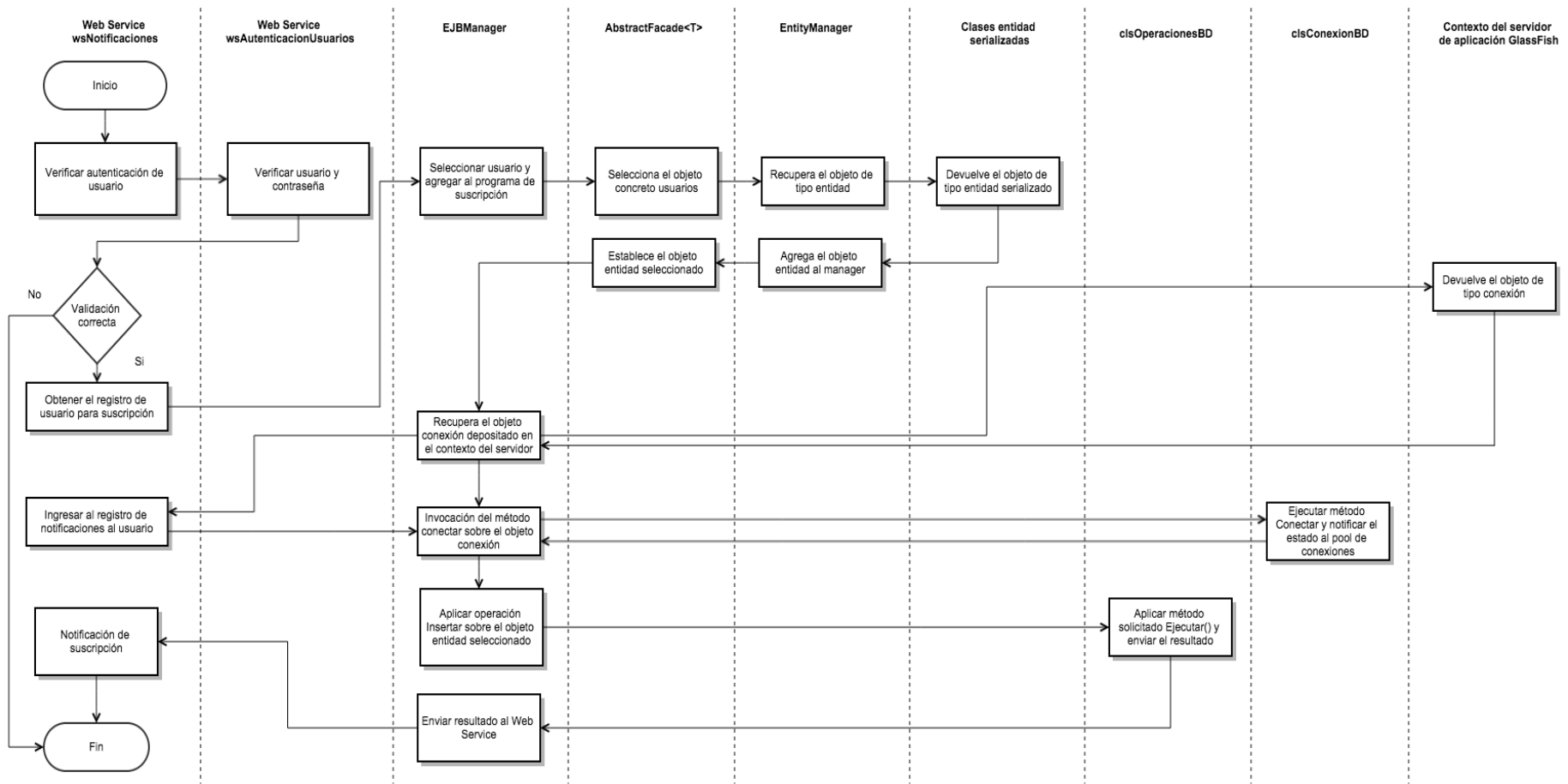


Figura 16: Flujo de trabajo para el Web Service wsNotificaciones(proceso suscripción a las notificaciones)

Web Service wsNotificaciones proceso envío de notificaciones

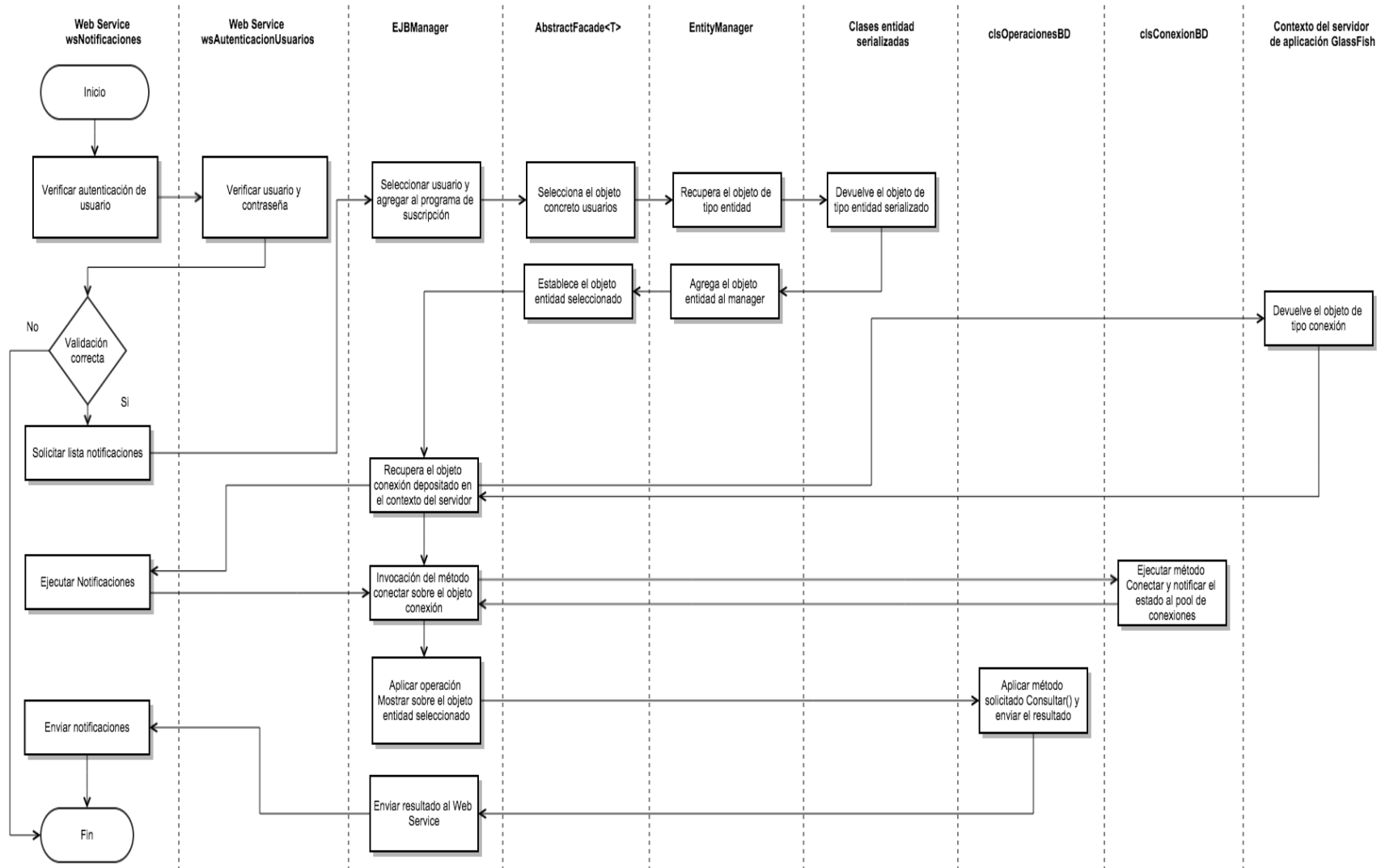


Figura 17: Flujo de trabajo para el Web Service wsNotificaciones(proceso envío de notificaciones)

Web Service wsLimpiezaDBMovil

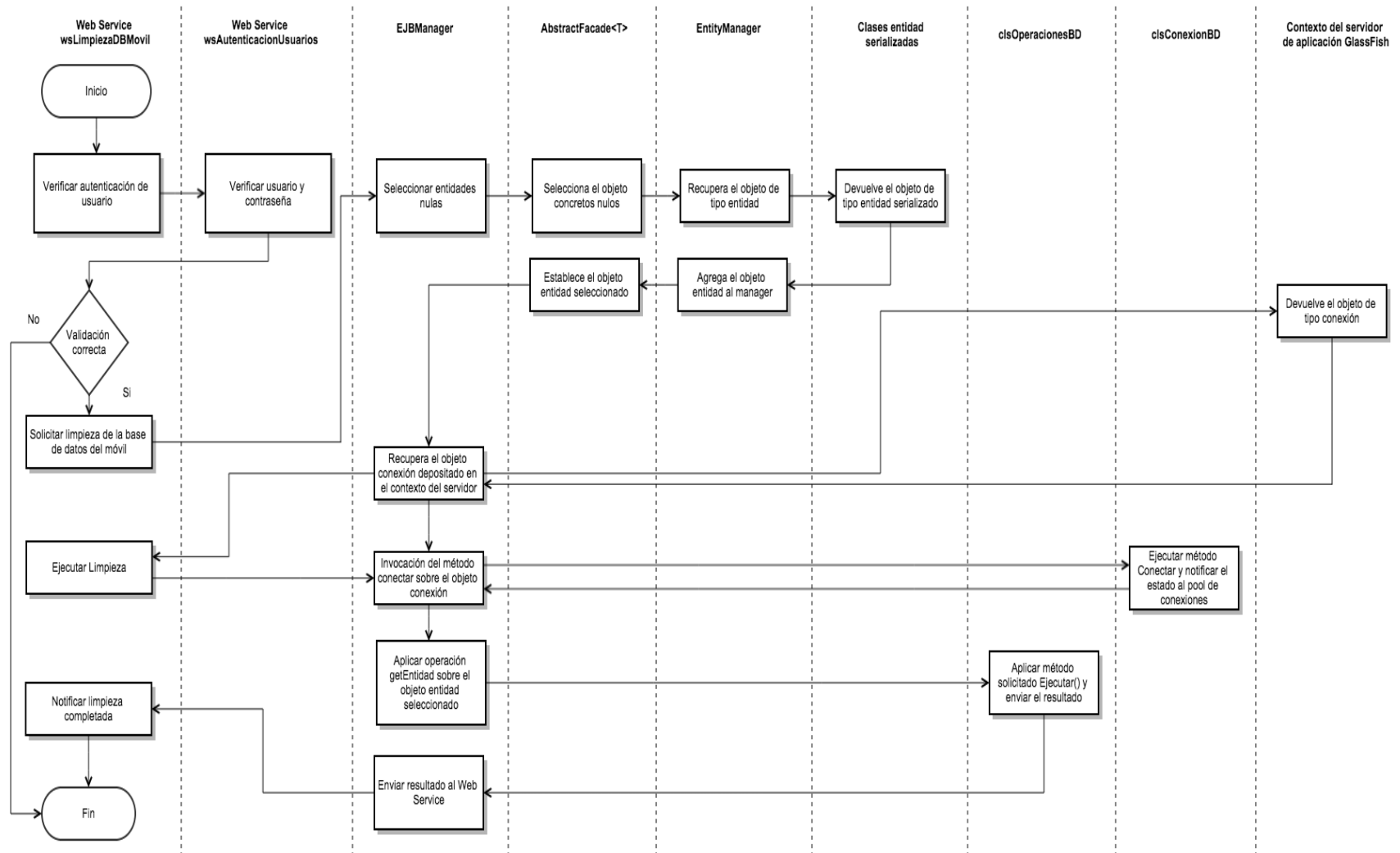


Figura 18: Flujo de trabajo para el Web Service wsLimpiezaDBMovil

4.3.1.4. Vista de Implementación.

4.3.1.4.1 Diagramas de Secuencia

Diagrama de secuencia del aplicativo.

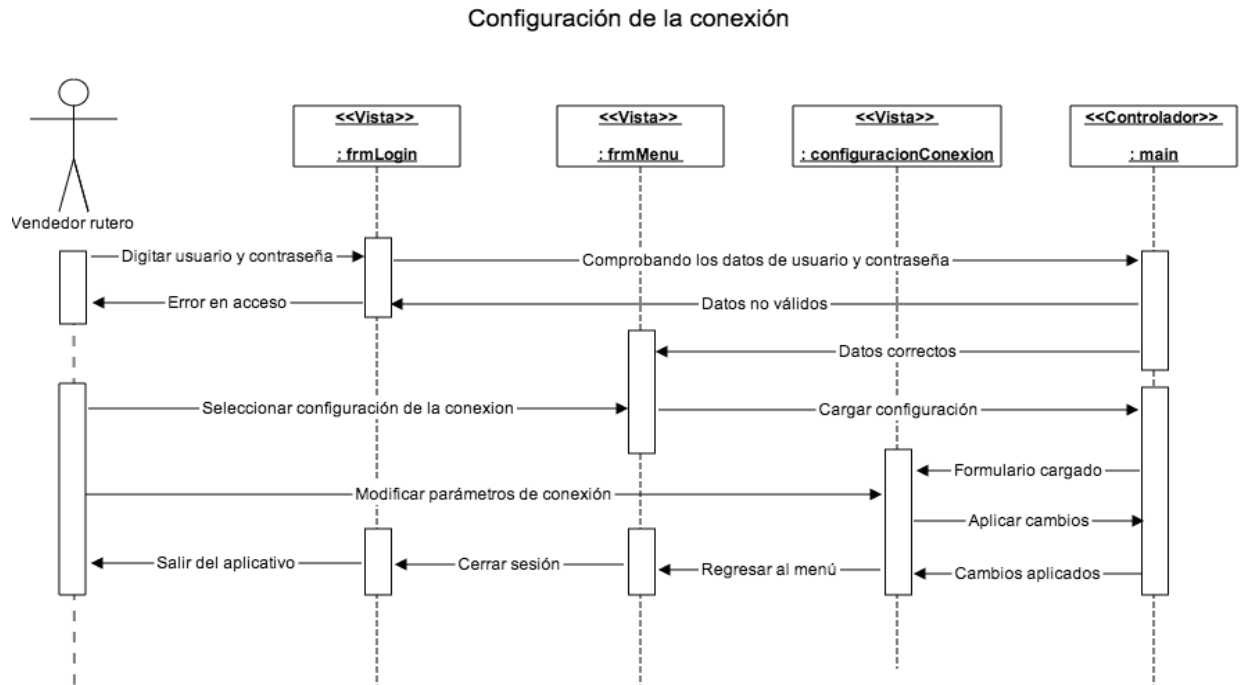


Figura 19: Diagrama de Secuencia del caso de uso “configuración de la conexión”

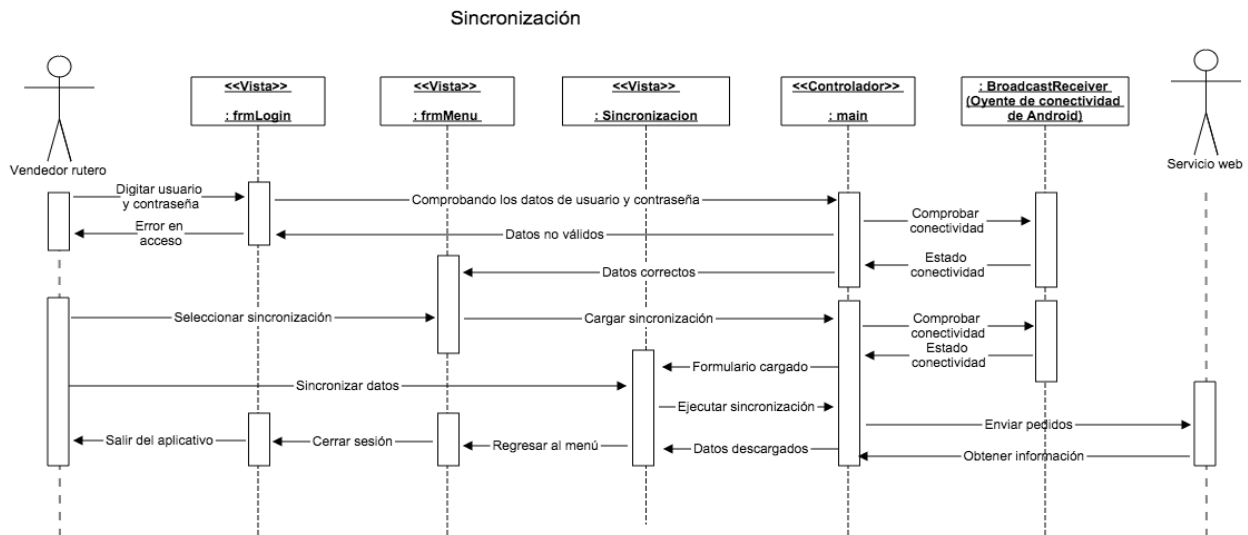


Figura 20: Diagrama de Secuencia del caso de uso “sincronización”

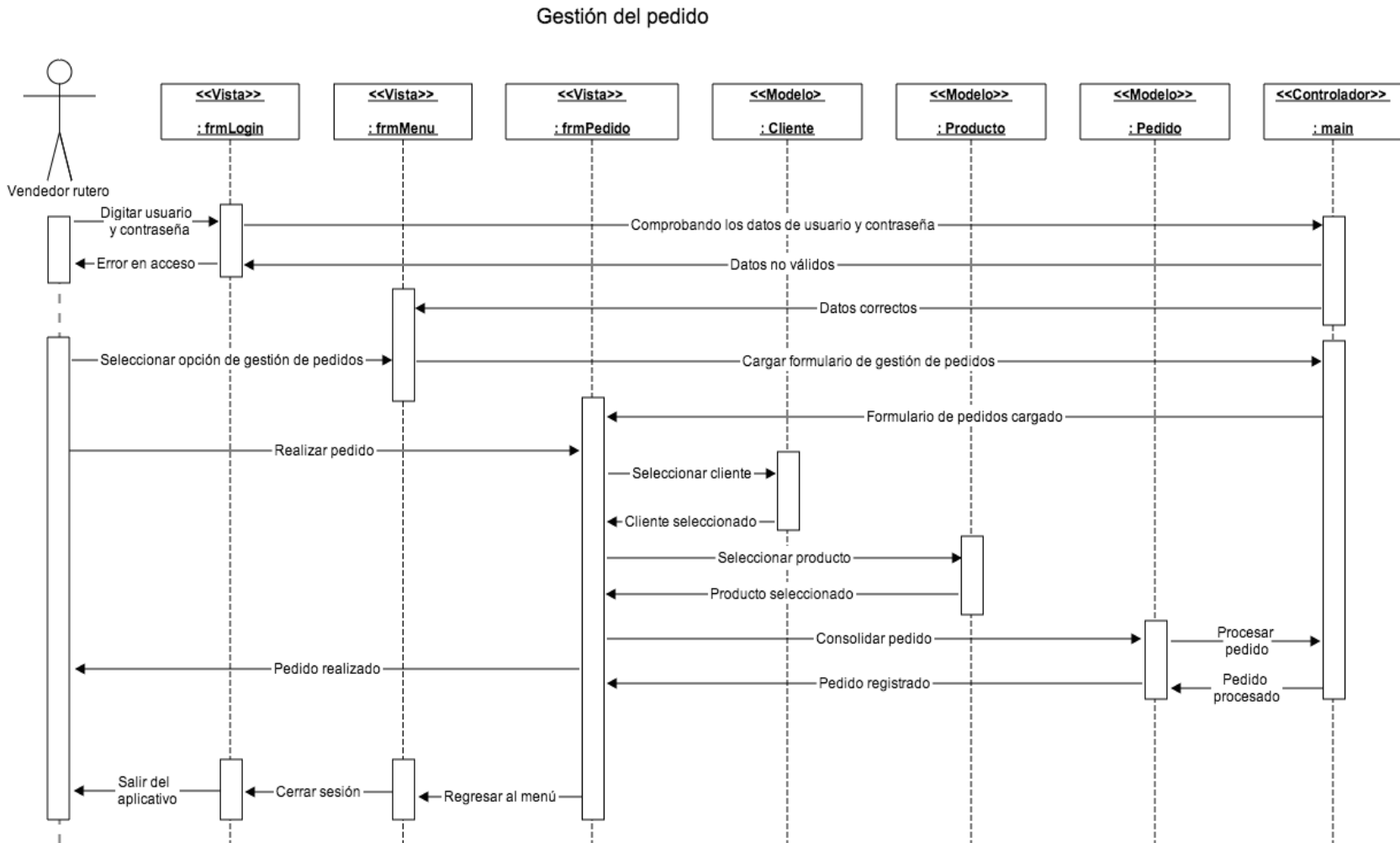


Figura 21: Diagrama de Secuencia del caso de uso “gestión del pedido”

Consulta de clientes/productos

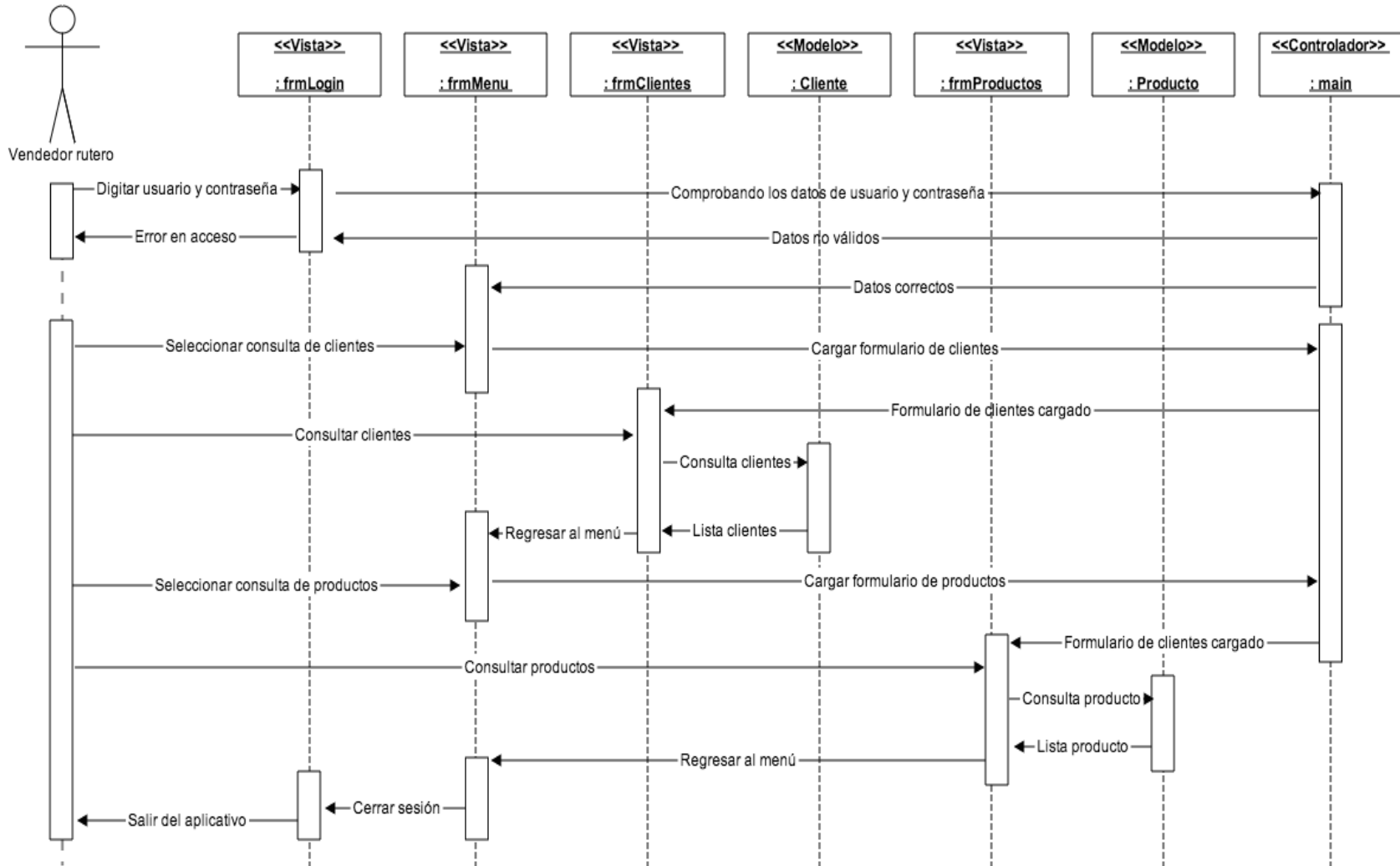


Figura 22: Diagrama de Secuencia del caso de uso “Consulta de clientes y productos”

Diagrama de secuencia del servicio web

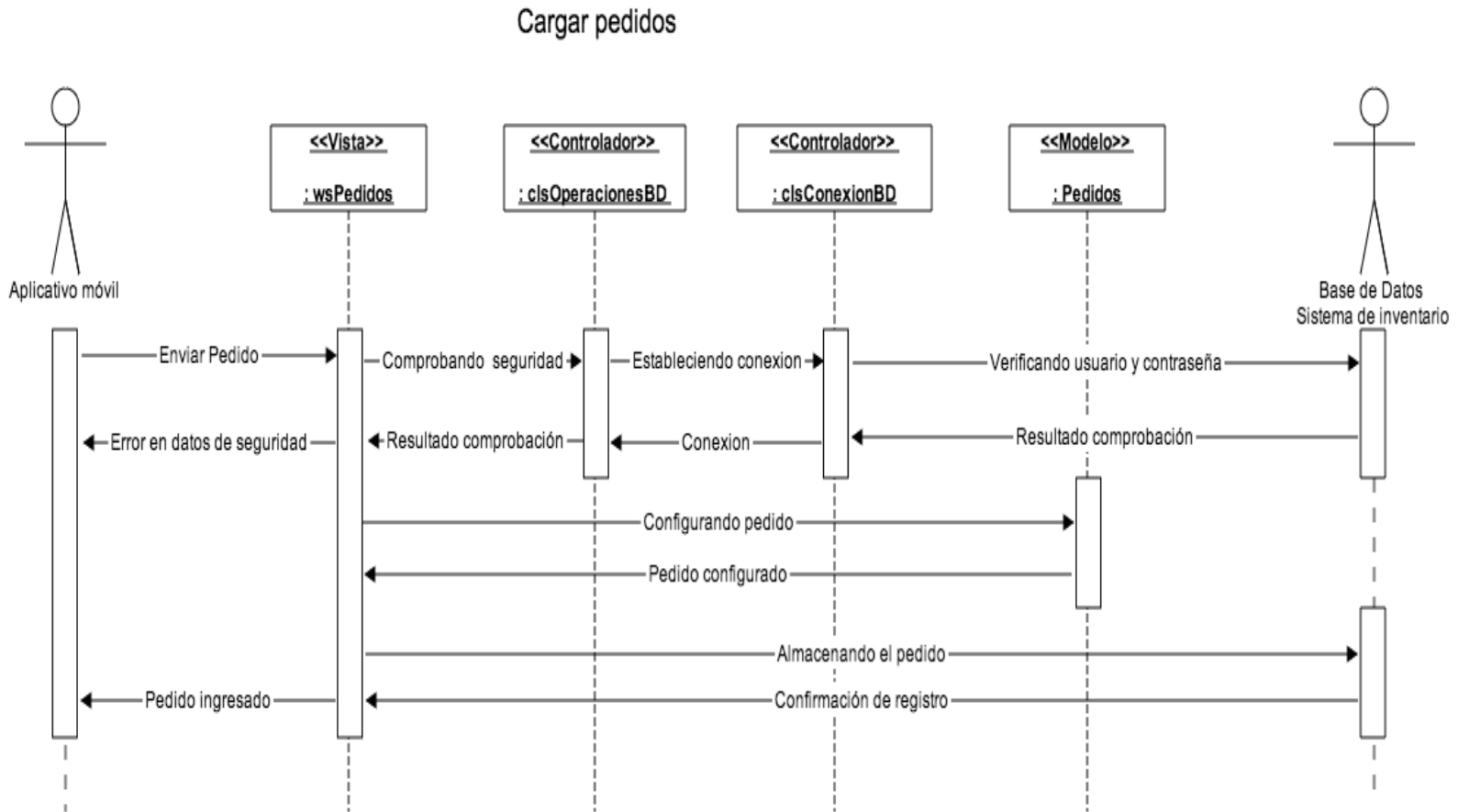


Figura 23: Diagrama de Secuencia del caso de uso “cargar pedidos”

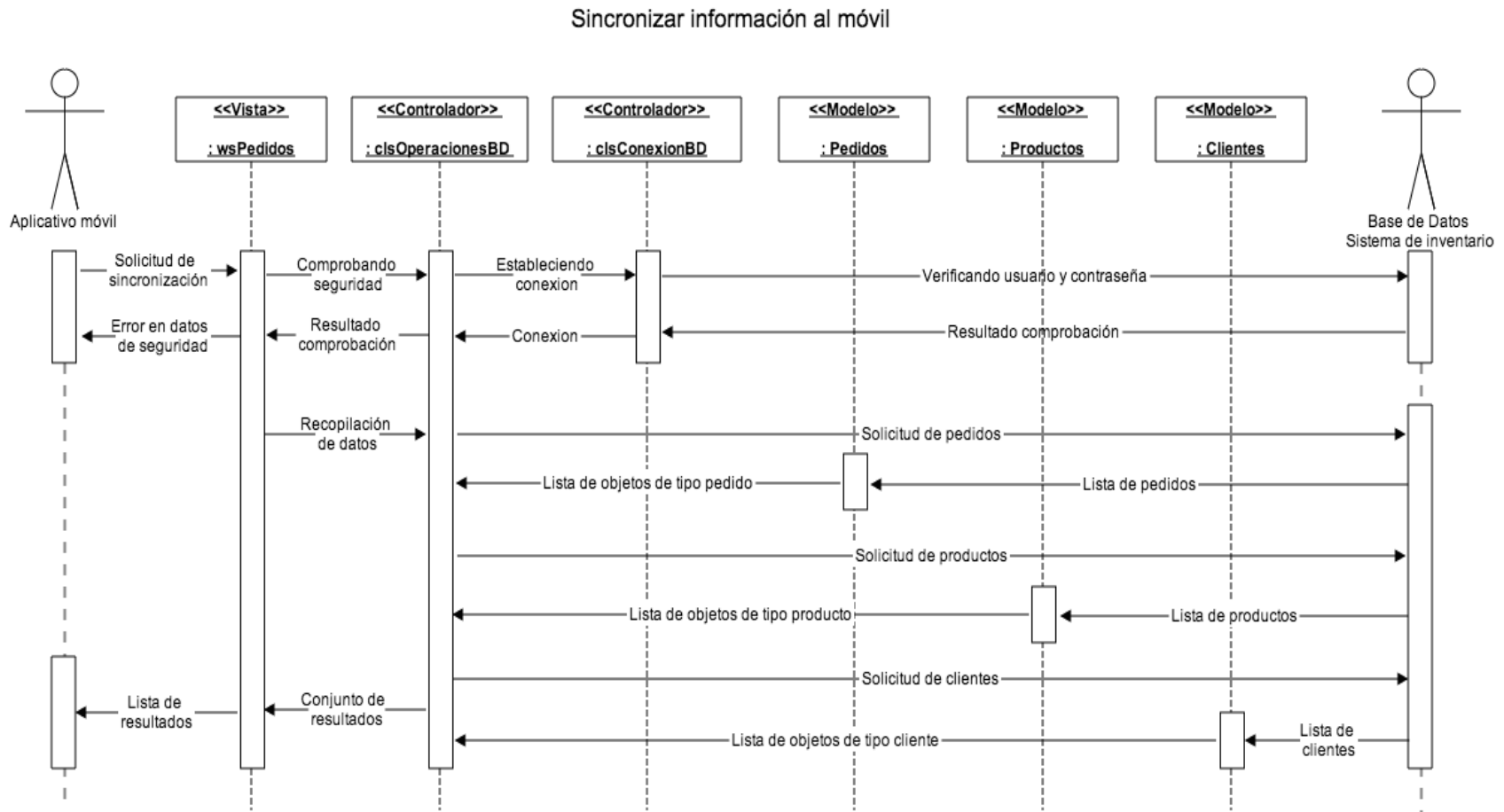


Figura 24: Diagrama de Secuencia del caso de uso “sincronizar información al móvil”

4.3.1.4.2 Diagrama lógico de la base de datos.

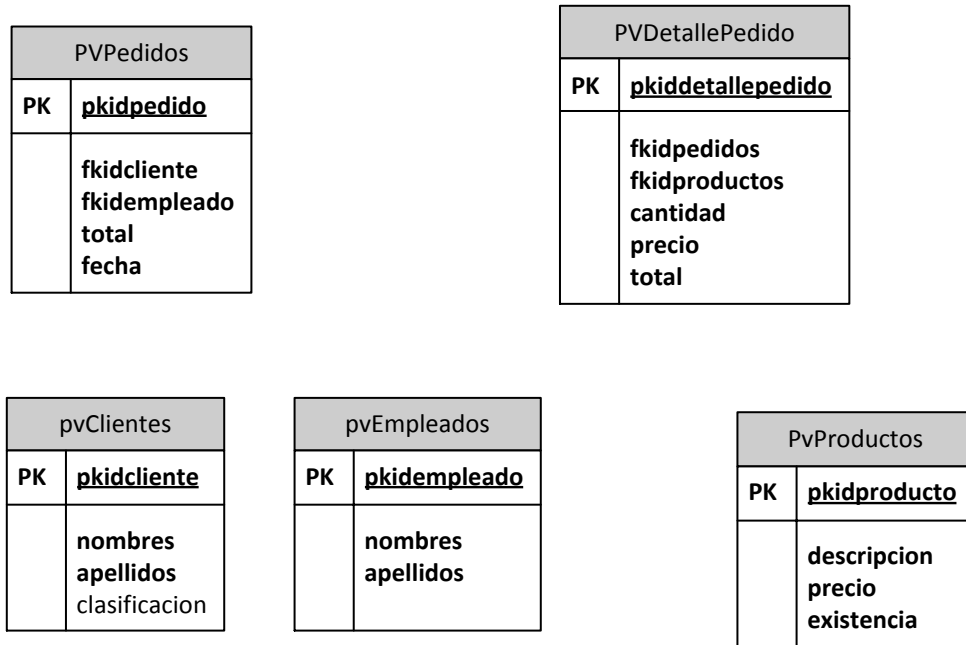


Figura 15: Diagrama Lógico de la base de datos

4.3.1.4.3 Diagrama físico de la base de datos.

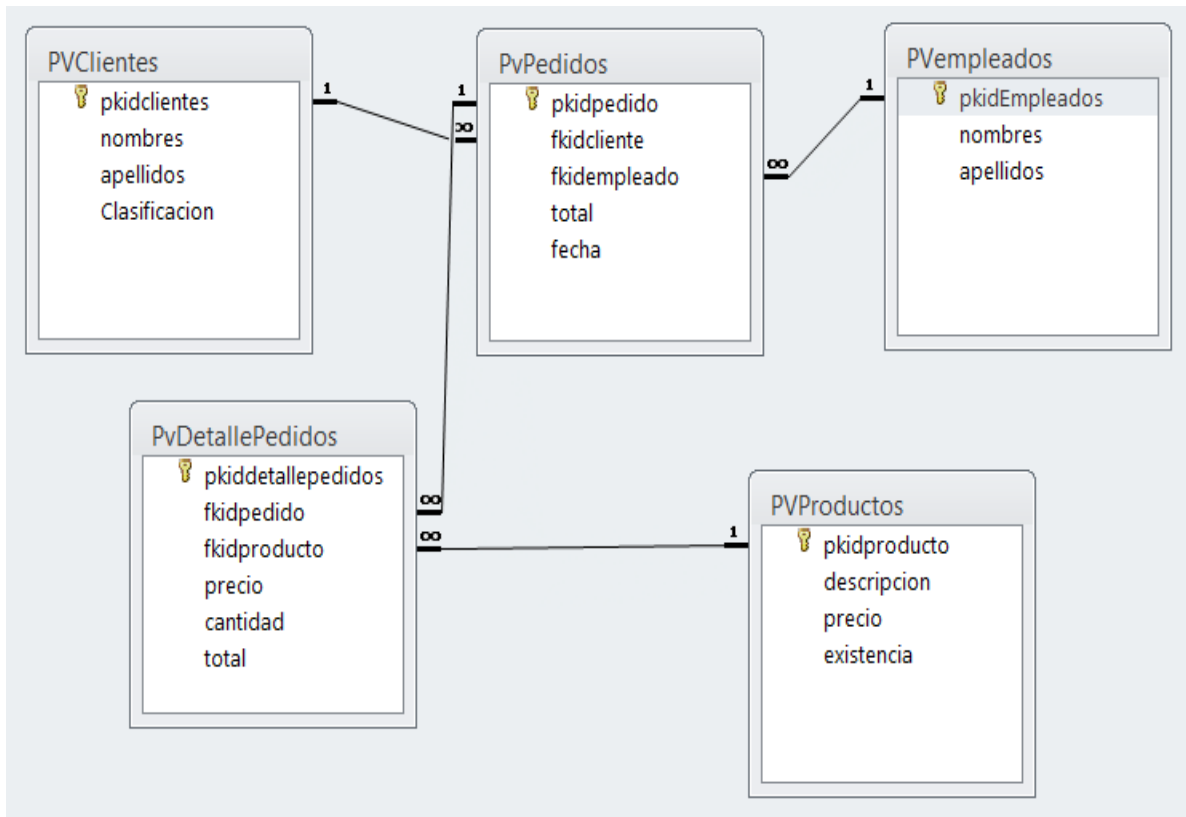


Figura 26: Diagrama Físico de la base de datos

5 CONCLUSIONES

Al finalizar el presente trabajo se concluye lo siguiente:

1. Las aplicaciones móviles están creciendo de forma exponencial y las empresas cada día necesitarán integrar aplicaciones basadas en plataformas móviles con sistemas legacy.
2. Los WEB SERVICE son una buena estrategia de integración debido a que son compatibles con la mayoría de los lenguajes de programación y que en la mayoría de los casos son desarrollos periféricos a los sistemas legacy, es decir no requieren que se modifiquen dichos sistemas, sino sólo acceso a la data.
3. Una de las formas de acceder a los WEB SERVICES de manera segura, es utilizar el protocolo HTTPS y un firewall en el lado del servidor de la empresa.
4. La utilización de patrones de diseño ayudó en cierta medida a la construcción de ésta propuesta de diseño debido a que son soluciones ya probadas para problemas específicos.
5. La metodología de desarrollo ágil, AUP basada en RUP, para éste proyecto es mejor debido a que el proyecto queda en una propuesta de diseño.

6 REFERENCIAS

- [1] Ambler, S. W. (10 de JULIO de 2014).
<http://www.ambysoft.com/unifiedprocess/agileUP.html>. Obtenido de
<http://www.ambysoft.com/unifiedprocess/agileUP.html>:
<http://www.ambysoft.com/unifiedprocess/agileUP.html>
- [2] Berka, K. (1982). *Measurement: Its Concepts, Theories and Problems (Boston Studies in the Philosophy of Science)* (Vol. 72). Boston: Kluwer.
- [3] Letelier, P., & Penadés, M. C. (s.f.). *Metodologías ágiles para el desarrollo de software : eXtreme Programming (XP)*. Universidad Politécnica de Valencia, Departamento de sistemas informáticos y Computación.
- [4] Marinelli, M., & Kuna, H. (2007). *SOA y BPEL*. Universidad de Málaga, Doctorado.
- [5] Mendez Calo, K., Estevez, E., & Fillottrani, P. (s.f.). Un framework para la evaluación de metodologías ágiles. Argentina.
- [6] Mendoza Sanchez, M. A. (2004). *Metodologías de desarrollo de software*. Informatizate.net.
- [7] Palacio, J. (2014). *Scrum Manager las reglas de scrum*. (S. Creative, Ed.) Safe Creative.
- [8] SEI, & O'Brien, W. (2014). *Software Engineering Institute*. Recuperado el 2014, de Software Engineering Institute: <http://www.sei.cmu.edu/library/abstracts/news-at-sei/architect4q02.cfm>
- [9] Sommerville, I. (2005). *INGENIERIA DEL SOFTWARE*. MADRID: PEARSON.
- [10] SysperTec. (Diciembre de 2013). Create interactive bidirectional connections between host and server applications through web and MQ services. *SOA & WOA Web Services*.

ANEXO 1: DISEÑO DE INTERFACES GRÁFICAS.

1.1 GESTIÓN DE PEDIDOS

Figura 1: Diseño de pantalla para realizar el pedido

DESCRIPCIÓN DE LA PANTALLA DE GESTIÓN DE PEDIDOS					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytPedido1	Correlativo	001	Código	Pedidos1
Usuario	Todos los usuarios del sistema				
Objetivo:	Solicitar la fecha y cliente para el pedido				
Descripción	Permite seleccionar el cliente para el pedido y continuar con el proceso				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
txtfechaPedido		X			X
cmbCliente	X			X	X
cmdAceptar				X	
cmdCancelar				X	

Tabla 1: Descripción de la pantalla Gestión de Pedido

Encabezado ...

Realizar Pedido Proceso 2/3

Cliente :

Producto :

Cantidad :

Total \$: 9,999.99

Producto	Cantidad
Nombre del Producto1	99.99
Nombre del Producto2	99.99
Nombre del Producto3	99.99

Figura 2: Diseño de pantalla Gestión de Pedido (Paso 2)

DESCRIPCIÓN DE LA PANTALLA DE GESTIÓN DE PEDIDOS					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytPedido2	Correlativo	002	Código	Pedidos2
Usuario	Todos los usuarios del sistema				
Objetivo:	Seleccionar los productos y cantidad para el pedido				
Descripción	Permite Agregar los productos, calcular la cantidad total del pedido y continuar con el proceso				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
txtCliente		X			X
cmbProductos	X			X	X
txtCantidad	X			X	X
txtAgregar				X	
lblTotal			X		
cmdSiguiente				X	
cmdAnterior				X	
cmdAceptar				X	
cmdCancelar				X	
lstDetallePedido			X		

Tabla 2: Descripción de la pantalla gestión de pedidos (paso 2)

Figura 3: Diseño de pantalla Gestión de Pedidos (Paso 3)

DESCRIPCIÓN DE LA PANTALLA DE GESTIÓN DE PEDIDOS					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytPedido3	Correlativo	003	Código	Pedidos3
Usuario	Todos los usuarios del sistema				
Objetivo:	Finalizar el pedido				
Descripción	Permite finalizar el pedido o regresar a modificar el pedido				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
txtCliente		X			X
lbTotal			X		
cmdFinalizar				X	
cmdAnterior				X	

Tabla 3: Descripción de la Pantalla Gestión de Pedidos (Paso 3)

CONSULTA DE CLIENTES

Figura 4: Diseño de pantalla "Consulta de Clientes"

DESCRIPCIÓN DE LA PANTALLA DE CONSULTA DE CLIENTES					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytConsultaClientes	Correlativo	004	Código	ConsClientes
Usuario	Todos los usuarios del sistema				
Objetivo:	Visualizar los clientes de la ruta del vendedor.				
Descripción	Permite visualizar los clientes que pertenecen a la ruta del vendedor				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
lstClientes		X			X
cmdCerrar				X	

Tabla4: Descripción de Pantalla "Consulta de Clientes"

CONSULTA DE PRODUCTOS

Consulta de Productos		
Nombre del Producto	Precio	Existencia
Producto 1	999.99	999.99
Producto 2	999.99	999.99
Producto n	999.99	999.99

Cerrar

Figura 5: Diseño de Pantalla "Consulta de Productos"

DESCRIPCIÓN DE LA PANTALLA DE CONSULTA DE PRODUCTOS					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytConsultaProductos	Correlativo	005	Código	ConsProductos
Usuario	Todos los usuarios del sistema				
Objetivo:	Visualizar los productos existentes en la base de datos				
Descripción	Permite visualizar los productos que tiene la base de datos y que han sido actualizados				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
lstProductos		X			X
cmdCerrar				X	

Tabla 5: Descripción de la pantalla "Consulta de Productos"

SINCRONIZACIÓN

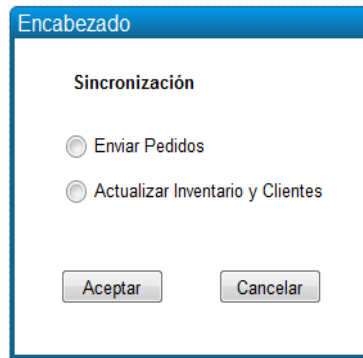


Figura 6: Diseño de Pantalla "Sincronización"

DESCRIPCIÓN DE LA PANTALLA SINCRONIZACIÓN					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytSincronizar	Correlativo	006	Código	Sincronizar
Usuario	Todos los usuarios del sistema				
Objetivo:	Sincronizar el servidor con el dispositivo móvil				
Descripción	Permite enviar los pedidos al servidor y actualizar la base de datos de los clientes e inventario en el móvil				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
rbtEnviarPedidos				X	
rbtActualizar				X	
cmdAceptar				X	
cmdCerrar				X	

Tabla 6: Descripción de pantalla "Sincronización"

CONFIGURACIÓN DE PARÁMETROS DE CONEXIÓN

Figura 7: Diseño de la Pantalla "Parámetros de Configuración"

DESCRIPCIÓN DE LA PANTALLAS DE CONFIGURACIÓN DE PARÁMETROS					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytConfiguracion	Correlativo	007	Código	Configurar
Usuario	Todos los usuarios del sistema				
Objetivo:	Configurar los parámetros para acceder al servidor remoto				
Descripción	Permite configurar los parámetros por medio de los cuales se conectará al servidor remoto				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado		
pgfConfigurar				X	
Rbthttp	X			X	X
Rbftfp	X			X	X
Txtipservidor	X			X	X
Txtpuerto	X			X	X
TxtubicacionArch	X			X	X
txtUsuario	X			X	X
txtPassword	X			X	X
cmdGuardar				X	
cmdCerrar				X	

Tabla 7: Descripción de la pantalla "Configuración de parámetros"

ACCESO AL SISTEMA

Figura 8: Diseño de Pantalla "Acceso al sistema"

DESCRIPCIÓN DE LA PANTALLA DE ACCESO					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	frmlogin	Correlativo	008	Código	Login
Usuario	Todos los usuarios del sistema				
Objetivo:	Ingresar al sistema				
Descripción	Ésta pantalla validará las credenciales del usuario en el sistema				
Elementos de la interfaz					
Nombre	Fuente			Editable	Requerido
	Ingresado	Recuperado	Calculado	X	X
txtUsuario	X			X	X
txtPassword	X			X	X
cmdAceptar				X	
cmdCancelar				X	

Tabla 8: Descripción de la pantalla "acceso al sistema"

MODIFICAR CONTRASEÑA DEL USUARIO

Figura 9: Diseño de Pantalla "Modificar contraseña del usuario"

DESCRIPCIÓN DE LA PANTALLA DE MODIFICAR CONTRASEÑA					
PROYECTO:	PROPUESTA DE DISEÑO				
Nombre:	lytModificarPassW ord	Correlativ o	009	Códig o	ModificarPasswo rd
Usuario	Todos los usuarios del sistema				
Objetivo:	Modificar la contraseña del usuario				
Descripción	Ingresar los datos que desea modificar y guardarlos en la base de datos.				
Elementos de la interfaz					
Nombre	Fuente			Edítabl e	Requerido
	Ingresado	Recuperad o	Calculad o	X	X
txtPasswordActual	X			X	X
txtPasswordNuevo	X			X	X
txtConfirmarPass	X			X	X
cmdAceptar				X	
cmdCancelar				X	

Tabla 9: Descripción de la pantalla "Modificar Contraseña"

ANEXO 2: ESTUDIO COMPARATIVO SOBRE SCRUM, XP Y RUP.

Para determinar cuál es la más idónea para aplicar al desarrollo de la propuesta de integración.

Se hará el siguiente análisis:

1. Se medirá de que manera las metodologías cumplen con los postulados del Manifiesto Ágil. A este efecto se define en qué medida satisfacen la Teoría Representacional de la Medición (Berka, 1982). Las medidas se definen usando escala de intervalos.

A continuación se definen los postulados:

- **Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas.**

Tres premisas sustentan este principio: a) los integrantes del equipo son el factor principal de éxito en un proyecto; b) es más importante construir el equipo de trabajo que construir el entorno; y c) es mejor crear el equipo y que éste configure el entorno en base a sus propias necesidades.

- **Valorar el desarrollo de software que funcione por sobre una documentación.**

El principio se basa en la premisa que los documentos no pueden sustituir ni ofrecer el valor agregado que se logra con la comunicación directa entre las personas a través de la interacción con los prototipos. Se debe reducir al mínimo indispensable el uso de la documentación que genera trabajo y que no aporta un valor directo al producto.

- **Valorar la colaboración con el cliente por sobre la negociación contractual.**

En el desarrollo ágil el cliente se integra y colabora con el equipo de trabajo como un integrante más. El contrato en sí no aporta valor al producto, es sólo un formalismo que establece líneas de responsabilidad entre las partes.

- **Valorar la respuesta al cambio por sobre el seguimiento de un plan.**

La evolución rápida y continua deben ser factores inherentes al proceso de desarrollo. Se debe valorar la capacidad de respuesta ante los cambios por sobre la capacidad de seguimiento y aseguramiento de planes pre-establecidos.

El mecanismo de evaluación mide de qué manera las metodologías cumplen con los postulados del Manifiesto Ágil. La evaluación provee mediciones para los 4 postulados. La medida de cada postulado se define como la suma de las medidas de los atributos relacionados.

Por ejemplo, el postulado 1 (P1). *Valorar al individuo y las interacciones del equipo de desarrollo por encima del proceso y las herramientas*, se mide sumando la medida de cómo la metodología valora al individuo y a las interacciones del equipo (P1.1) y la medida de cómo valora al proceso y a las herramientas (P1.2).

El atributo que los principios intentan enfatizar (atributo positivo) se mide en una escala de 0 a 5, y el otro atributo (atributo negativo) en una escala de -5 a 0. De este modo, cada principio podría obtener una medida entre -5 en el caso que ambos tomen el peor valor (-5 el atributo negativo y 0 el atributo positivo), y 5 en el caso que ambos atributos tomen el mejor valor (0 en el atributo negativo y 5 el atributo positivo). Si se obtiene un valor cero o cercano a cero, significa que la metodología no valora significativamente el atributo positivo por sobre el negativo, en cuyo caso, el principio del Manifiesto Ágil no se satisface plenamente.

A continuación se detalla la tabla de evaluación de cada postulado:

TABLA DE EVALUACIÓN DE CADA POSTULADO			
P1	Valorar al individuo y las interacciones del equipo de desarrollo por encima del proceso y las herramientas.		
P1.1	Valorar al individuo y las interacciones	P1.2	Valorar al proceso y las herramientas.
Valor	Descripción	Valor	Descripción
0	No define roles para individuos	-5	Define actividades, entregables, herramientas de desarrollo y de gestión.
1	Clara definición de roles para individuos	-3	Define actividades, entregables y herramientas de desarrollo.
2	Clara definición de roles y responsabilidades.	-2	Define actividades y entregables
3	Clara definición de roles, responsabilidades y conocimientos técnicos.	-1	Define actividades para cada iteración
5	Clara definición de roles, responsabilidades, conocimientos técnicos e interacciones entre miembros del equipo de trabajo	0	Define actividades para el proyecto pero no a nivel de cada iteración.
P2	Valorar el desarrollo de software que funcione por sobre una documentación.		
P2.1	Valorar el desarrollo de software que funcione	P2.2	Valorar la documentación exhaustiva.
Valor	Descripción	Valor	Descripción
0	Generar entregable al finalizar el proyecto	-5	Requiere documentación detallada al comienzo del proyecto.
3	Generar entregable con testing satisfactorio al finalizar cada iteración.	-2	Requiere solo documentación necesaria al comienzo de cada iteración.
5	Generar entregable con testing satisfactorio e integrado con el resto de las funciones al finalizar cada iteración.	0	No requiere documentación para comenzar a implementar la funcionalidad incluida en una iteración.

Tabla 1a: Tabla de Evaluación de Postulados

TABLA DE EVALUACIÓN DE CADA POSTULADO			
P3	Valorar la colaboración con el cliente por sobre la negociación contractual.		
P3.1	Valorar la colaboración con el cliente	P3.2	Valorar la negociación contractual
Valor	Descripción	Valor	Descripción
0	Cliente colabora a demanda del equipo	-5	Existe contratación detallada al inicio y no se aceptan cambios.
3	Cliente es parte del equipo, responde consultas y planifica las iteraciones.	-2	La contratación exige contemplar cambios durante el proyecto.
5	Cliente es parte del equipo, responde consultas, planifica iteraciones, y colabora en la escritura de requerimientos y pruebas.	0	El contrato por la construcción del producto no aporta valor al producto.
P4	Valorar la respuesta al cambio por sobre el seguimiento de un plan		
P4.1	Valorar la respuesta al cambio	P4.2	Valorar el seguimiento de un plan.
Valor	Descripción	Valor	Descripción
0	No prevé incorporar cambios durante la ejecución del proyecto.	-5	Define un plan detallado al inicio del proyecto.
1	Prevé introducir solo cambios de alta prioridad	-3	Define un plan detallado de iteraciones.
4	Permite la evolución y el cambio, pero no es recomendable en la iteración en curso.	-2	Define un plan para cada iteración.
5	Permite introducir cambios en la iteración en curso.	0	No define planificación alguna.

Tabla 1b: Tabla de Evaluación de Postulados

Análisis comparativo entre SCRUM, XP y RUP

Postulados Metodología	P1			P2			P3			P4		
	P1.1	P1.2	Tota	P2.1	P2.2	Tota	P3.1	P3.2	Tota	P4.1	P4.2	Total
SCRUM	5	-3	2	5	-2	3	5	0	5	4	-2	2
XP	5	-2	3	5	-2	3	5	0	5	5	-3	2
RUP	4	-1	3	5	-5	0	3	-2	1	1	-2	-1

Tabla 2: Análisis comparativo entre SCRUM, XP y RUP

La aplicación de la tabla de evaluación de los postulados permite explicar lo siguiente:

Postulado 1. SCRUM y XP obtienen 5 en el atributo P1.1 RUP obtiene una valoración de 4, ya que Scrum y XP valoran al individuo en mayor medida a RUP, estas definen roles y responsabilidades y reconocen la importancia y promueven la capacitación de los integrantes del equipo. Scrum obtiene -3 en el atributo P1.2 ya que define actividades, entregables y herramientas de desarrollo; mientras que XP obtiene -2 porque solo define actividades y entregables, RUP obtiene -1, porque únicamente define actividades para cada iteración.

Al establecer los totales, SCRUM obtiene 2 puntos y XP y RUP obtienen 3. De modo que XP y RUP. satisfacen mejor que Scrum el postulado 1.

Postulado 2. Tanto SCRUM, XP y RUP obtienen 5 puntos en el atributo P2.1. En el atributo P2.2 se evalúan con un valor de -2 Scrum y XP ya que ambas sólo requieren documentación para iteración planificada; RUP se evalúa con -5 debido a la relación contractual con el cliente. Al totalizar el postulado 2 Scrum y XP satisfacen de mejor manera que RUP.

Postulado 3. Tanto SCRUM como XP obtienen el mejor valor en ambos atributos -5 puntos para P3.1 y 0 puntos para al atributo P3.2. Ambas consideran al cliente como un miembro

del equipo, que colabora desde la planificación de las iteraciones hasta la escritura de los requerimientos y pruebas funcionales. En RUP el cliente es considerado parte del equipo desde la respuesta a las preguntas generadas, aportando una valoración de 3 puntos. La valoración del atributo P3.2 obtiene una valoración de -2, ya que los cambios deben registrarse en contrato. Al totalizar SCRUP y XP logran una ventaja considerable sobre RUP.

Postulado 4. En el atributo P4.1 Scrum obtiene un valor 4 ya que si bien permite introducir cambios, no se recomienda en el sprint en curso. De ser prioritario el cambio en el sprint en curso, se debe estimar nuevamente el esfuerzo requerido y si es necesario, quitar tareas del sprint ya planificado. XP obtiene el máximo valor debido a que los cambios se pueden incorporar durante la iteración. El estar pendiente de los cambios significa detallar en el plan las acciones correspondientes por lo cual XP obtiene -3 puntos, RUP es calificado con un punto debido a la incorporación de cambios únicamente prioritarios. En el atributo P4.2 Scrum es calificado con -2, XP con -3 y RUP con -2. Al totalizar el postulado 4 Scrum y XP mantienen la misma calificación, mientras RUP logra una diferencia de un punto.

Finalmente en tres de los cuatro postulados Scrum y XP obtienen la misma puntuación, excepto el postulado 1 donde XP supera a Scrum por 1 punto. RUP en tres de los postulados se mantiene con los menores valores. En términos de los postulados Scrum y XP son los que más se apegan, No obstante en razón de las características propias del modelo de integración que se propone, la tecnología que el cliente posee y los compromisos (Relación contractual) entre el equipo desarrollador de la propuesta y el cliente, la metodología adecuada es RUP.

2. Además de hacer una evaluación apegados a los postulados del Manifiesto Agile, se hace una evaluación basada en el criterio del equipo de trabajo que presenta la propuesta de integración.

Comparación de metodologías ágiles			
Metodología Criterios	SCRUM	XP	RUP
Basado en casos de uso	No	No	SI
Compuesta por un ciclo de vida	Si	Si	Si
Permite un desarrollo rápido	Si	Si	Si
Mantiene objetivos específicos y entregables para cada ciclo	Si	No	Si
Permite varias alternativas tecnológicas para su desarrollo	Si	Si	Si
Mantiene la calidad del sistema en cada iteración	Si	No, el sistema se depura incrementalmente	Si
Relación con UML	Integrable	Integrable	Nativa

Tabla 3: Comparación de metodologías ágiles.