

**IMPLEMENTACIÓN DE UNA APLICACIÓN DE E-COMMERCE EN JOYERÍAS
UTILIZANDO J2EE (JAVA 2 ENTERPRISE EDITION).**

**OMAR ANDRÉS ÁLVAREZ CARDOZO
YOANA PAOLA MALDONADO NAVARRO**

**PROYECTO DE GRADO
PARA OPTAR POR EL TÍTULO DE INGENIERÍA DE SISTEMAS**

**ASESOR
OLGA LUCÍA ROA BOHÓRQUEZ. Msc**

**UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.**

2010

Tabla de contenido

INTRODUCCIÓN	0
1. PLANTEAMIENTO DEL PROBLEMA	8
1.1 ANTECEDENTES	8
1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA	8
1.3 JUSTIFICACIÓN	9
1.4 OBJETIVOS	10
1.4.1 Objetivo General	10
1.4.2 Objetivos Específicos.	10
1.5 ALCANCES Y LIMITACIONES	10
1.5.1 Alcances.	10
1.5.2 Limitaciones. Se tendrá en cuenta que el aplicativo no incluirá:	11
2 METODOLOGÍA	12
3 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA.	13
4. MARCO DE REFERENCIA	14
4.1 MARCO TEÓRICO – CONCEPTUAL	14
4.1.1 Aplicación Web.	14
4.1.2 Ingeniería Web.	14
4.1.3 Metodologías de desarrollo de software.	15
4.1.4 Comercio electrónico	17
4.1.5 Arquitectura de software.	19
4.1.6 Herramientas de desarrollo	20
4.2 MARCO LEGAL O NORMATIVO	24
5. DESARROLLO INGENIERÍL	26
5.1 METODOLOGÍA DE DESARROLLO	26
5.2 FASE DE INICIO	27
5.2.1 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR.	27
5.2.2 Aproximación Arquitectural.	29
5.2.3 Cubrimiento de Requerimientos.	29
5.2.4 Actores del sistema.	30

5.2.5	PUNTOS DE VISTA FUNCIONAL.	32
5.2.6	Atributos de Calidad.	35
5.3	FASE DE ELABORACIÓN	42
5.3.1	Punto de vista funcional fase de elaboración.	42
5.3.2	Punto de Vista de Información.	63
5.3.3	Punto de Vista de Despliegue.	98
5.3.4	Evaluación de la Arquitectura.	101
5.3.5	Evaluación ATAM (Architecture Tradeoff Analysis Method).	103
5.4	FASE DE IMPLEMENTACIÓN	105
5.5	FASE DE TRANSICIÓN	106
6	ANÁLISIS Y RESULTADOS.	107
6.1	ANÁLISIS DE RESULTADOS PREVIOS A LA IMPLEMENTACIÓN	107
6.2	ANÁLISIS DE RESULTADOS DEL PRODUCTO FINAL	107
7	CONCLUSIONES Y RECOMENDACIONES	109
7.1	CONCLUSIONES	109
7.2	RECOMENDACIONES	109
8	BIBLIOGRAFÍA	111

LISTADO DE FIGURAS

Figura 1: Esquema general para un sistema de correo electrónico	18
Figura 2: Componentes de JSF	22
Figura 3: Diagrama de casos de uso nivel 1	34
Figura 4: Casos de uso Gestionar usuario	42
Figura 5: Caso de uso Gestionar Producto	43
Figura 6: Diagrama de casos de uso Gestionar Pedido	44
Figura 7: Diagrama de flujo de datos para validación de usuario.	62
Figura 8: Diagrama de componentes.	63
Figura 9: Diagrama de clases para Entity beans	65
Figura 10: Clase Entity Abono	66
Figura 11: Clase Entity TipoProd	67
Figura 12: Clase Entity Servicio	68
Figura 13: Clase Entity Departamento	69
Figura 14: Clase Entity Ciudad	70
Figura 15: Clase Entity Factura	71
Figura 16: Clase Entity Producto.	72
Figura 17: Clase Entity Usuario	74
Figura 18: Clase Entity Venta	75
Figura 19: Diagrama de clase Session bean AbonoFacade	76
Figura 20: Diagrama de clase Session bean CiudadFacade	77
Figura 21: Diagrama de clase Session bean DepartamentoFacade	78
Figura 22: Diagrama de clase Session bean VentaFacade	79
Figura 23: Diagrama de clase Session bean FacturaFacade	80
Figura 24: Diagrama de clase Session bean ProductoFacade	81
Figura 25: Diagrama de clase Session bean ServicioFacade	82
Figura 26: Diagrama de clase Session bean TipoProdFacade	83
Figura 27: Diagrama de clase Session bean UsuarioFacade	84
Figura 28: Diagrama entidad relación.	86
Figura 29: Modelos de flujo de información.	96
Figura 30: Modelo de ciclo de vida de información, Registro cliente	97

Figura 31: Modelo de ciclo de vida de información, Realizar Pedido_____	97
Figura 32: Modelo de ciclo de vida de información, modificar Pedido _____	98
Figura 33: Diagrama de despliegue _____	99
Figura 34: Árbol de utilidades_____	102
Figura 35: Interfaz gráfica _____	106

LISTADO DE TABLAS

Tabla 1: RUP vs Extreme programming -----	16
Tabla 2: Listado de los actores del sistema SIJEC -----	30
Tabla 3: Actores y Requerimientos-----	31
Tabla 4: Actores vs Puntos de vista-----	32
Tabla 5: Atributo de calidad: Seguridad-----	36
Tabla 6: Atributo de calidad: usabilidad -----	37
Tabla 7: Usuario nuevo accediendo al sistema (Usabilidad)-----	38
Tabla 8: Escenario de calidad: Ingresar al sistema -----	39
Tabla 9: Consulta de información básica de un Producto (Desempeño)-----	40
Tabla 10: Escenario de calidad Consultar Ventas (Disponibilidad)-----	41
Tabla 11: Caso de uso Registrar usuario -----	45
Tabla 12: Curso normal de los eventos caso de uso registrar usuario -----	46
Tabla 13: Caso de uso actualizar usuario -----	46
Tabla 14: Curso normal de los eventos Caso de uso actualizar datos -----	47
Tabla 15: Caso de uso validar usuario -----	48
Tabla 16: Curso normal de los eventos Caso de uso validar Usuario -----	49
Tabla 17: Caso de uso consultar usuarios-----	49
Tabla 18: Curso normal de los eventos Caso de uso consultar Usuarios-----	50
Tabla 19: Caso de uso ingresar productos-----	50
Tabla 20: Curso normal de los eventos Caso de uso Ingresar productos -----	51
Tabla 21: Caso de uso actualizar productos -----	52
Tabla 22: Curso normal de los eventos Caso de uso actualizar productos-----	52
Tabla 23: Caso de uso consultar productos-----	53
Tabla 24: Curso normal de los eventos Caso de uso Consultar productos-----	53
Tabla 25: Caso de uso seleccionar producto. -----	54
Tabla 26: Curso normal de los eventos Caso de uso seleccionar producto-----	54
Tabla 27: Caso de uso adicionar el producto al pedido -----	55
Tabla 28: Curso normal de los eventos Caso de uso adicionar producto.-----	55
Tabla 29: Caso de uso consultar el pedido de usuario. -----	56
Tabla 30: Curso normal de los eventos Caso de uso consultar pedido -----	56

Tabla 31: Caso de uso modificar el pedido actual del usuario-----	56
Tabla 32: Curso normal de los eventos Caso de uso modificar pedido. -----	57
Tabla 33: Caso de uso actualizar formulario del pedido actual del usuario.-----	57
Tabla 34: Curso normal de los eventos Caso de uso actualizar formulario del pedido actual. -----	58
Tabla 35: Caso de uso enviar formulario de pedido. -----	58
Tabla 36: Curso normal de los eventos Caso de uso enviar formulario de pedido. -----	59
Tabla 37: Caso de uso actualizar pago.-----	59
Tabla 38: Curso normal de los eventos Caso de uso actualizar pago-----	60
Tabla 39: Caso de uso salir -----	60
Tabla 40: Curso normal de los eventos Caso de uso salir-----	61
Tabla 41: Caso de uso Generar Reportes-----	61
Tabla 42: Curso normal de los eventos Caso de uso generar reportes -----	61
Tabla 43: Tabla ciudad-----	87
Tabla 44: Tabla departamento -----	88
Tabla 45: Tabla usuario -----	89
Tabla 46: Tabla factura-----	91
Tabla 47: Tabla servicio-----	92
Tabla 48: Tabla abono -----	93
Tabla 49: Tabla Tipo_prod -----	93
Tabla 50: Tabla producto -----	94
Tabla 51: Tabla ventas -----	95
Tabla 52: Evaluación ATAM del atributo sobre disponibilidad -----	103
Tabla 53: Evaluación ATAM del atributo sobre seguridad-----	104

INTRODUCCIÓN

El uso de Internet ha contribuido en el desarrollo de diferentes tipos de actividades comerciales. Gracias a la masiva utilización de éste medio como herramienta comercial se ha logrado una mayor acogida por parte de los clientes virtuales a través de aplicaciones Web.

Para hablar de los sistemas y las aplicaciones basados en Web (*WebApp*) se hace referencia al experto en diseño web Thomas Powell quien afirma que “las aplicaciones Web implican una mezcla de publicación impresa y desarrollo de software, de marketing e informática, de comunicaciones internas y relaciones externas, y de arte y tecnología”¹

Una *WebApp* permite establecer en Internet una comunicación abierta con todo el mundo. En la mayoría de los casos, la función primaria de una *WebApp* utiliza hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y vídeo.

Las aplicaciones web son cada vez más utilizadas a lo largo de diferentes tipos de negocios. “JSF (*Java Server Faces*), es una tecnología moderna diseñada para grandes aplicaciones que permite la división de tareas por capas en donde los componentes del sistema están claramente separados y la complejidad se reduce”.²

El presente proyecto comprende la aplicación de una arquitectura por capas con tecnología JAVA para las Joyerías y algunos tipos de negocios afines, haciendo eficiente parte de los procesos administrativos del negocio, brindándole a la empresa soluciones tecnológicas ágiles y seguras, ofreciendo posibilidades a los clientes de tener una mejor visualización del diseño y precio de una joya a la hora de comprarla, un manejo de un catálogo de productos o servicios, y un manejo del negocio de manera remota por parte del propietario.

¹ PRESSMAN S, Roger. Ingeniería de Software Un enfoque practico, McGraw Hill, Sexta Edición en Español, 2008.

² Tutoriales Cupi 2 – Motivación [documento en línea] disponible desde internet en: <http://cupi2.uniandes.edu.co/tutoriales/publicados/JSF_ColegioWeb/Motivacion.htm> [con acceso el 10 de mayo de 2009]

1. PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

En la actualidad existen páginas web publicitarias y sitios web para venta de joyas en Colombia, pero estas no ofrecen un servicio que ofrezca a los propietarios la administración del negocio mediante un seguimiento de sus productos y servicios. Como tampoco les brindan la posibilidad a los clientes interesados en argollas de hacerle posibles modificaciones en su diseño.

Uno de los trabajos realizados que anteceden este proyecto planteó la implementación de un sitio de mercadeo y comercio electrónico para las pymes, por medio de un portal para la comercialización entre personas, pequeñas y medianas empresas, dicho trabajo fue desarrollado bajo la web 2.0 y AJAX.

1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

La gran mayoría de las MiPymes (micro, pequeñas y medianas empresas)³ como son las joyerías en la actualidad no cuenta con un sistema de información que permita un registro y seguimiento organizado de sus productos. Por otro lado estos tipos de negocio tienen la necesidad de prestarles a los clientes una asistencia de manera remota para lograr comprar y acceder al catálogo de productos y servicios sin tener que asistir personalmente para obtener información o retirar sus pedidos.

¿Cómo implementar una aplicación de comercio electrónico (E-Commerce) para este tipo de MiPymes?

³ Según la Asociación Colombiana de las micro, pequeñas y medianas empresas existe la ley 905 de 2004 la cual define el tipo de empresa según el número de trabajadores y los activos totales [artículo en línea] disponible desde internet en: <http://www.acopi.org.co/index.php?option=com_content&task=view&id=19&Itemid=20> [con acceso el 10 de mayo de 2009].

1.3 JUSTIFICACIÓN

Según el Plan Nacional de Tecnologías de Información y de las Comunicaciones PNTIC se debe aumentar el uso de las TIC en las MiPymes e incrementar las actividades de comercio electrónico y de e-business debido a que las MiPymes representan el 98% de las empresas de Colombia⁴.

El PNTIC tiene como objetivo concientizar a las personas y empresas a utilizar las TIC en los procesos productivos de las MiPymes y así incrementar su competitividad. Fomentando así el comercio electrónico como un canal para que las empresas ofrezcan sus productos y servicios en línea.

El desarrollo de un aplicativo web para este tipo de MiPymes responderá a la necesidad de un sistema de información que permita un registro y seguimiento organizado de productos y servicios brindando la posibilidad de realizarlo remotamente.

Por otro lado resulta eficaz la comercialización de los productos (joyas) vía internet debido a que se proporciona información por medio de catálogos a los clientes interesados en los productos y servicios que ofrecen las joyerías sin la necesidad de asistir personalmente, logrando la atención a los clientes en zonas geográficas en donde no existen sucursales.

Los clientes quienes accederán a la información, lograrán hacer una compra de forma segura a cualquier hora del día con la opción de pagar con tarjeta de crédito.

El proyecto pretende obtener una aplicación flexible, segura y de fácil mantenimiento; características que se pueden lograr utilizando una arquitectura en capas con J2EE gracias a que es una herramienta de desarrollo orientada a objetos.

⁴ Plan Nacional de Tecnologías de la Información y las Comunicaciones [documento en línea] disponible desde internet en: <<http://www.colombiapntic.org/docs/080409-Plan%20Nacional%20de%20TIC.pdf>> [con acceso el 3 de mayo de 2009]

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un sistema de comercio electrónico (e-commerce) que permita el seguimiento de la actividad comercial en joyerías en tiempo real.

1.4.2 Objetivos Específicos.

1. Analizar los requerimientos funcionales y no funcionales para el aplicativo web de e-commerce.
2. Modelar una base de datos para garantizar el almacenamiento y control de la información de este tipo de negocio.
3. Diseñar la interface web que cumpla con los requerimientos funcionales y permita una interacción amigable del usuario con el sistema.
4. Implementar la aplicación de comercio electrónico para joyerías, utilizando la tecnología Java Server Faces.

1.5 ALCANCES Y LIMITACIONES

1.5.1 Alcances. Este proyecto culmina con el desarrollo de un aplicativo web basado en tecnología Java Server Faces que permita el seguimiento del negocio ofreciendo al usuario administrador el manejo de inventario, productos y servicios almacenados en una base de datos. El cual incluirá lo siguiente:

- El aplicativo web brindará a los clientes la oportunidad de realizar compras, se podrá acceder a catálogos de productos y servicios.
- El aplicativo web manejará diversas formas de navegación las cuales permitirán al usuario del sistema acceder de diferentes maneras para obtener la información del producto a consultar y brindará la posibilidad de realizar pedidos.

- El sistema permitirá visualizar las compras realizadas por los clientes, las facturas generadas para las diferentes ventas, los productos existentes en inventario, los clientes registrados y la actualización o modificación de los diferentes datos ingresados en el sistema.
- Para los usuarios internos de la empresa el aplicativo brindará la posibilidad de ingresar al sistema ventas que sean realizadas a los clientes de manera personal.
- Se simulará la compra con tarjeta de crédito e incluirá el manejo del carrito de compras.
- Se cumplirá con los dos requerimientos no funcionales más importantes que serán determinados dentro del análisis arquitectural del proyecto.
- Se realizarán pruebas unitarias a los componentes del software y pruebas de integración al mismo.

1.5.2 Limitaciones. Se tendrá en cuenta que el aplicativo no incluirá:

Módulos para generar balances de pérdidas y ganancias.

No se implementará un software de diseño gráfico para generar nuevas joyas, ni argollas en 3D.

La aplicación Web no tendrá un módulo real de pagos en línea con tarjeta de crédito pues para utilizar este servicio se debe implementar un *Web Service* dentro del proyecto, el cual debe tener acceso a este tipo de servicios a través de las diferentes redes bancarias de los clientes y a su vez debe manejar las transacciones y/o saldos del banco utilizado por la empresa beneficiada por el *Web Service*. Este trabajo puede ser más extenso que el mismo proyecto.

Otra forma de implementar este tipo de pagos es contratar un proveedor que ofrezca este servicio, el cual suministra un código fuente que se debe integrar a la aplicación para acceder al *Web service* que sirve de intermediario con los bancos, pero no se implementará por el costo que genera este servicio para el proyecto.

2 METODOLOGÍA

El enfoque a emplear en la investigación es el empírico-analítico cuyo interés es el técnico porque el caso de las joyerías pretende interpretar y transformar la realidad existente a una aplicación técnica.

La investigación que se va a emplear es la aplicada porque su propósito es incrementar el conocimiento con fines prácticos o productivos y su acción está encaminada a los procesos de conocimientos relacionados con el desarrollo tecnológico y con la productividad.

3 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA.

El presente proyecto seguirá la línea de investigación institucional de la Universidad de San Buenaventura sede Bogotá; Tecnologías actuales y sociedad. En la Facultad de Ingeniería se concibe su sub-línea como sistemas de información y comunicación, por último, este proyecto se suscribe al campo temático desarrollo de software del programa de Ingeniería de Sistemas.

4. MARCO DE REFERENCIA

4.1 MARCO TEÓRICO – CONCEPTUAL

4.1.1 Aplicación Web. Una página web es un documento HTML (*Hyper Text Markup Language*) /XHTML (*Extended Hyper Text Markup Language*) al cual se accede en la mayoría de los casos mediante el protocolo HTTP(*HyperText Transfer Protocol*) de Internet.

Según el centro de desarrolladores de Mozilla⁵ para acceder a las páginas de un sitio web se hace desde un URL(*Universal Resource Locator*), ó raíz común llamado portada, que normalmente reside en un servidor físico. Dichos URLs permiten organizar las páginas en diferentes jerarquías.

Conallen, define una *aplicación Web* “como un *sistema Web* (servidor Web, red, protocolo HTTP, y browser) en el cual el usuario, a través de navegación y entrada de datos, afecta el estado del negocio. Mientras que un *sitio Web* es un sistema donde la navegación del usuario no permite modificar datos, o un sistema Web donde no hay lógica del negocio”⁶.

4.1.2 Ingeniería Web. Según Roger Pressman ⁷ la ingeniería Web aplica un enfoque genérico que se suaviza con estrategias, tácticas y métodos especializados. El proceso de ingeniería Web comienza con una formulación del problema planteado en el proyecto que pasa a resolverse con las WebApps. Se planifica el proyecto y se analizan los requisitos de la WebApp. Una de las metodologías más utilizadas para modelar la solución del problema planteado en el proyecto es la Rational Unified Process (RUP); este es un proceso de desarrollo de software que utiliza el lenguaje unificado de modelado UML.

⁵ The data URL scheme, [documento en línea] disponible desde internet en:

<https://developer.mozilla.org/en/The_data_URL_scheme> [con acceso el 1 de mayo de 2009].

⁶ CONALLEN, J. Modeling Web Application Architectures with UML. *Communications of the ACM*, 42(10), October, 1999, pp. 63-70

⁷ PRESSMAN, Op. cit.,220

No obstante, el Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*). “Es un estándar gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables”⁸.

Es importante resaltar que UML es una forma de modelamiento para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, entre otros.) y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

“UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. En UML 2.0 hay 13 tipos diferentes de diagramas. Entre los que están: Diagrama de clases, Diagrama de paquetes, Diagrama de actividades, Diagrama de casos de uso, Diagrama de estados, Diagrama de secuencia”⁸ que serán utilizados para modelar los procesos de la aplicación web.

4.1.3 Metodologías de desarrollo de software. A continuación se presenta una visión general de dos metodologías diferentes.

El propósito es mostrar las características de algunas. En la tabla 1 se hace una comparación entre la metodología RUP y XP. Los aspectos a comparar son: fases, tiempo, ciclo de vida y propiedades.

⁸ OMG Unified Modeling Language TM (OMG UML), Infrastructure, Version 2.2 [Libro en línea] disponible desde internet en: <<http://www.omg.org/docs/formal/09-02-04.pdf>> [con acceso el 19 marzo de 2009]

Tabla 1: RUP vs Extreme programming

	Rational Unified Process RUP	XP(Extreme Programming)
Fases	<ul style="list-style-type: none"> • Inicio • Elaboración • Construcción • Transición 	<ul style="list-style-type: none"> • Planeación • Diseño • Codificación • Pruebas
Tiempo	Largo plazo	Corto plazo.
Modelo o ciclo de vida	Ciclo de vida en cascada a menor escala	Modelo basado en prototipos
Propiedades	<ul style="list-style-type: none"> -Asigna tareas y responsabilidades de una forma disciplinada. -Utiliza la arquitectura basada en componentes. -Control de cambios (modelado visual). -Verificación de la calidad del software. 	<ul style="list-style-type: none"> -Pruebas unitarias: pruebas que se realizan a los principales procesos. -Re fabricación: se basa en la reutilización de código. -Programación en pares: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Este proyecto utiliza la metodología RUP pues está dirigida por casos de uso y centrada en la arquitectura. Incluye también artefactos como el diagrama de clases, modelo de componentes, entre otros, en donde se pretende utilizar diferentes conceptos de arquitectura de software. Mientras que XP es usada para desarrollos a corto plazo con practicas adaptativas en vez de predictivas.

4.1.4 Comercio electrónico “Es el intercambio electrónico de datos que se utiliza para dar soporte a transacciones comerciales, es decir, un intercambio de valores entre un vendedor o comerciante (ofertante de bienes y servicios) y un comprador (demandante de bienes y servicios)”⁹.

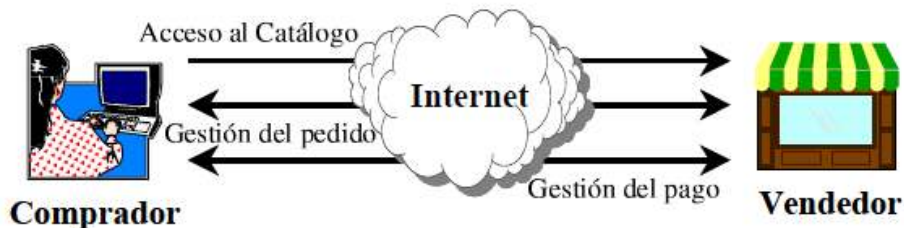
La mayor parte de los sistemas actuales de comercio electrónico consiste básicamente en un vendedor que oferta sus productos mediante un catálogo electrónico y unos compradores de dichos productos que pueden consultar dichas ofertas y comprar los productos. El ciclo de compra se resume en las siguientes etapas:

- La entidad vendedora (Ofertante) oferta sus productos mediante un catálogo actualizado.
- La entidad compradora (demandante) consulta los productos ofertados.
- La entidad compradora (demandante) realiza el pedido.
- La entidad vendedora (ofertante) acepta el pedido y lo confirma.
- Finalmente es necesario realizar la entrega del producto y el pago (ambos procesos se pueden realizar indistintamente de forma electrónica o no).

En la Figura 1 se muestra el esquema básico para un sistema general de comercio electrónico.

⁹ Modelos para comercio electrónico basados en sistemas intermediarios. Gallego Fernández, M. Isabel. [documento en línea] disponible desde internet en: < <http://www.tesisenred.net/TDX-0613102-140302> > [con acceso el 20 de noviembre de 2010]

Figura 1: Esquema general para un sistema de correo electrónico



Fuente: Modelos para comercio electrónico basados en sistemas intermediarios. Gallego Fernández, M. Isabel. [documento en línea] disponible desde internet en: <
<http://www.tesisenred.net/TDX-0613102-140302> > [con acceso el 20 de noviembre de 2010]

El concepto de comercio electrónico no es solo el hecho de la compra o venta de bienes por medios electrónicos, existen una serie de actividades comerciales antes y después de la venta que también deben introducirse al escenario digital. Dentro de este grupo de actividades comerciales se debe mencionar: publicidad, búsqueda de información sobre productos y proveedores de dichos productos, negociación sobre precios, condiciones y plazos de entrega, entre otros.

Todos los intercambios de datos entre compradores y vendedores en el contexto electrónico o virtual se realizan de forma remota mediante la utilización de redes de transmisión como es el caso de Internet.

Internet es una red abierta que presenta una serie de limitaciones para garantizar que dichas transacciones se realicen de forma segura. Hay que mencionar que los métodos de pago electrónico es el aspecto del comercio electrónico más desarrollado en la actualidad y ha contribuido de una forma decisiva en su desarrollo.

A continuación se enumeran algunas de las ventajas que representa el comercio electrónico.

- a) Acceso a mercados globales, importante para las pequeñas y medianas empresas, ya que se abren nuevas oportunidades de negocio que permiten mejorar los beneficios.

- b) Permite a los comerciantes mejorar la atención al cliente mediante sistemas de personalización y fidelización de clientes.
- c) Para los compradores, permite ampliar su capacidad de acceso a un gran número de bienes y conseguir unas mejores condiciones de compra. Los mercados globales permiten generar una oferta más variada de bienes y servicios.

4.1.5 Arquitectura de software.

En la última década cambió la visión que los desarrolladores tienen de los sistemas de software. Esta nueva visión se llamó "arquitectura". Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". Este nuevo aspecto hace posible el estudio de sistemas ya implementados así como el desarrollo de nuevos, según la categoría del problema que resuelven y el tipo de "arquitectura". Los distintos niveles de abstracción de la funcionalidad de los sistemas, están asociados con diferentes aspectos y componentes de su "arquitectura". Esta asociación se manifiesta en forma clara en las distintas etapas del proceso de desarrollo de software¹⁰.

Una arquitectura de un sistema está dada por un conjunto de estructuras indispensables para razonar sobre el sistema, que comprende elementos de software, las relaciones entre ellos y las propiedades de ambos¹¹.

- a) Arquitectura candidata: "Es un conjunto de estructuras estáticas y dinámicas que tienen el potencial para exhibir los requerimientos de comportamiento y calidad del sistema"¹².
- b) Elementos arquitecturales: Son piezas fundamentales desde las cuales un sistema puede ser analizado y construido, estas deben definir claramente un conjunto de

¹⁰ Arquitectura de Software [documento en línea] disponible desde internet en: <<http://materias.fi.uba.ar/7572/#contenidos>> [con acceso el 30 de noviembre de 2010]

¹¹ ROZANSKI, Nick y WOODS, Eoin. Software Systems Architectures – Working with *Stakeholders* Using Viewpoints and Perspectives. Addison-Wesley. 2005.

¹² Documenting Component and Connector Views with UML 2.0, Technical Report. ESC-TR-2004-08

responsabilidades, fronteras y las respectivas interfaces que definen los servicios ofrecidos a otros elementos arquitecturales. Entre estas piezas están:

- *Stakeholders*: es una persona, grupo, entidad o actor con unos intereses y/o requerimientos dentro de la realización de una arquitectura¹⁰.
- *Concerns*: “es un requerimiento, un objetivo, una intención o una aspiración que un Stakeholder tiene para la arquitectura”¹⁰.
- Descripción arquitectural: “es un conjunto de productos que documentan una arquitectura de manera que los stakeholders pueden entender y demostrar que sus concerns están presentes dentro de la arquitectura”¹⁰.
- Puntos de vista: “es una colección de patrones, estándares y convenciones para construir un tipo de vista”¹⁰.
- Vistas: es una representación de uno o más aspectos estructurales de una arquitectura que ilustra como ésta contempla uno o varios concerns definidos por los stakeholders¹⁰.
- Perspectivas “son una colección de actividades, tácticas y guías que son usadas para asegurar que los sistemas exhiban un conjunto de propiedades de calidad que requieren que sean consideradas en varias vistas arquitecturales.”¹⁰

4.1.6 Herramientas de desarrollo

- ***Java Server Faces***

Las aplicaciones web son cada vez más utilizadas a lo largo de diferentes tipos de negocios. JSF (*Java Server Faces*), es un estándar moderno diseñado para grandes aplicaciones que permite la división de tareas por capas en donde los componentes del sistema están claramente separados y la complejidad se reduce¹³.

JSF es el único que tiene una especificación creada por el *Java Community Process* (JCP); esto lo transforma en un estándar y como principal consecuencia todas las

¹³ Tutoriales Cupi 2 Op. cit.,

implementaciones, tanto las de código fuente abierto como las comerciales deben respetarla. Por otro lado, JSF 2.0¹⁴, la versión actual del *framework*, forma parte de la especificación J2EE (*Java 2 Platform, Enterprise Edition*) 6.0 con lo cual todas las implementaciones de servidores J2EE 6.0 lo deben soportar¹⁵.

Es un *framework* Web J2EE de la familia de código fuente abierto, Server-Side. Este es básicamente una estructura de software formada de componentes personalizables e intercambiables que permiten desarrollar una aplicación. Los *framework* se pueden considerar como una aplicación genérica incompleta y configurable a la que podemos agregarle algunas piezas para concretar una aplicación.

Entre los objetivos principales de los *frameworks* están: reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un *framework web* J2EE, es un conjunto de componentes de software, por ejemplo clases JAVA, descriptores y archivos en XML(*Extensible Markup Language*), basados en la plataforma J2EE, que se ejecutarán en servidores J2EE.

El *frameworks web de JSF*, responde al patrón de diseño MVC (Modelo Vista Controlador) para web. “Este patrón es una guía para el diseño de arquitecturas de aplicaciones que ofrecen una fuerte interactividad con usuarios; el cual organiza la aplicación en tres subsistemas separados: el Modelo que representa los datos de la aplicación y sus reglas de negocio, la Vista formada por un conjunto de pantallas que representa la interfaz de usuario de la aplicación (en web primordialmente son formularios HTML de entrada y salida), y el Controlador que es el encargado de procesar las peticiones de los usuarios y controlar el flujo de ejecución del sistema”¹⁶

JSF está construido sobre la API de *Servlets* y provee un conjunto de librerías de JSP (*JavaServer Pages*) que permiten incluir componentes de interfaz de usuario (IU) en

¹⁴ CONALLEN, J. “Modeling Web Application Architectures with UML”. *Communications of the ACM*, 42(10), October, 1999, pp. 63-70.

¹⁵ Struts y JavaServer Faces, cara a cara [documento en línea] disponible desde internet en: <<http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/109.pdf>> [con acceso el 11 de mayo de 2009]

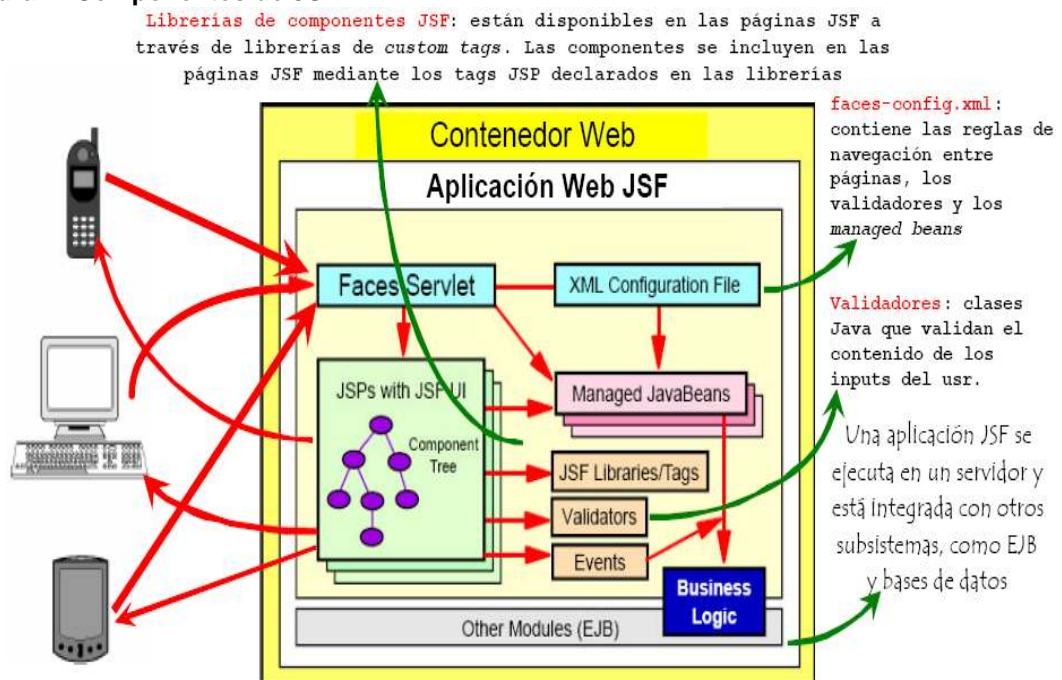
¹⁶ Struts y JavaServer Faces Op. cit.,

páginas dinámicas. Sin embargo, es posible utilizar una tecnología de presentación diferente a la del cliente web.¹⁷

El uso de un aplicativo web desarrollado bajo el *framework* JSF como herramienta sistemática de información para las Joyerías y algunos tipos de negocios afines, pretende hacer eficiente parte de los procesos administrativos del negocio, brindándole a la empresa soluciones tecnológicas ágiles y seguras, ofreciéndole la posibilidad a los clientes de tener una mejor visualización del diseño y precio de una joya a la hora de comprarla, un manejo de un catalogo de productos o servicios, y un manejo del negocio de manera remota por parte del propietario.

La Figura 2, muestra los componentes principales del *framework* JSF que intervienen en la construcción de una aplicación web J2EE, como también su flexibilidad para aceptar peticiones provenientes de diferentes clientes.

Figura 2: Componentes de JSF



Fuente: Struts y JavaServer Faces, cara a cara [documento en línea] disponible desde internet en: < <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/109.pdf> > [con acceso el 11 de mayo de 2009]

¹⁷ Struts y JavaServer Faces Op. cit.,

Análogo a los *Struts*, JSF implementa el patrón *Front-Controller*, el cual centraliza el manejo de peticiones provenientes de los clientes. En JSF, este controlador central, es un objeto *servlet*, llamado *FacesServlet*.

- **Enterprise JavaBeans**

Enterprise JavaBeans (EJB)¹⁸ es una arquitectura que permite la creación de componentes de aplicaciones distribuidas y orientadas a transacciones. Las aplicaciones escritas utilizando EJB son escalables, transaccionales y multiusuarios.

Se caracteriza por contener la lógica del negocio, por crear y manejar las instancias por medio del container EJB, y además puede ser configurado editando sus parámetros de entorno vía archivos XML.

Las características de seguridad y transacciones se encuentran separadas de las clases EJB, lo que permite la operación de aplicaciones externas.

- **Tipos de Enterprise Java Beans**

La arquitectura de EJB define tres tipos diferentes de objetos enterprise beans:

Session beans¹⁹: son objetos no persistentes que modelan la lógica de los procesos de negocio, es decir, modelan acciones y se ejecutan en el servidor.

Cada session bean mantiene una interacción con un cliente que se desarrolla a través de la ejecución de los distintos métodos que provee. Existen dos tipos de ellos que varían en la forma de modelar esta interacción: *stateless* y *stateful*.

Los *stateless* modelan los procesos de negocios que tienden naturalmente a una única interacción, por tanto no requieren de mantener un estado entre múltiples invocaciones. Mientras que los *stateful* están diseñados para servir procesos de negocio que abarcan

¹⁸ Rozanski N, Woods E. "Software Systems Architecture" Addison_Wesley. 2005.

¹⁹ Rozanski N, Woods E. "Software Systems Architecture" Addison_Wesley. 2005.

múltiples llamados a funciones o transacciones. Para lograr esto es necesario guardar el estado en que se encuentra el bean luego de cada ejecución del cliente, el cual se mantiene y actualiza para cada nueva invocación del mismo cliente.

Entity beans²⁰: contienen el modelo de datos del negocio y la lógica interna de los datos. Su tiempo de vida es tan largo como los datos en el sistema de almacenamiento que representan. Otorgan una vista de objeto Java a los datos del negocio guardados en una unidad de almacenamiento. Permiten un acceso compartido de múltiples usuarios y tienen un tiempo de vida independiente de la duración de las sesiones de los clientes.

Message-driven beans²¹: Modelan acciones, pero sólo se ejecutan luego de recibir un mensaje, contienen la lógica de procesar un mensaje en forma asíncrona.

4.2 MARCO LEGAL O NORMATIVO

Colombia cuenta con una serie de normas para el desarrollo de software y la comercialización de este y es por esto que este proyecto debe regirse por ellas.

En la primera parte hay que referenciar al decreto 1360 de 1989 el cual establece políticas con respecto al uso y empleo del software libre en sus sistemas de información. El decreto, en el artículo primero define como software libre al programa licenciado por su autor para ofrecer a sus usuarios la libertad de ejecutar su programa para cualquier propósito, estudiar la manera de operar el programa y mejorar el programa al igual que las distribuciones de las mismas.

Por otra parte, hay que tener en cuenta la Ley 633, específicamente el artículo 91 que dice lo siguiente: "Todas las páginas Web y sitios de Internet de origen colombiano que operan en el Internet y cuya actividad económica sea de carácter comercial, financiera o de prestación de servicios, deberán inscribirse en el Registro Mercantil y suministrar a la

²⁰ Rozanski N, Woods E. "Software Systems Architecture" Addison_Wesley. 2005.

²¹

Dirección de Impuestos y Aduanas Nacionales DIAN, la información de transacciones económicas en los términos que esta entidad lo requiera”²²

Con respecto a los estándares para la realización del proyecto se tiene que tener en cuenta las recomendaciones de la W3C²³ (World Wide Web Consortium), el cual desarrolla pautas y normativas para la web y el código que esta contenga pueda ser entendible por cualquier navegador web.

²² Reforma Tributaria 2000 Ley 633. [Artículo en línea] disponible desde Internet en: <<http://www.cijuf.org.co/leyes/l633.html>> 03/02/2009

²³ About the World Wide Web Consortium (W3C) [Artículo en línea] disponible desde internet en: <<http://www.w3.org/Consortium/>>03/05/2009

5. DESARROLLO INGENIERÍL

5.1 METODOLOGÍA DE DESARROLLO

En el desarrollo del proyecto se adoptó la metodología RUP²⁴, siguiendo así las fases de inicio, elaboración, construcción y transición descritas a continuación:

En la fase de inicio se identifican los principales actores que van a hacer uso del sistema de Información. Se realiza el levantamiento de requerimientos, y se elabora un modelo de casos de uso general. Aquí se realiza un esbozo provisional de la posible arquitectura que se implementará para el desarrollo del sistema.

Tanto en la fase de inicio como en la de elaboración se utiliza el punto de vista funcional, pues es el punto central de un análisis de arquitectura, permite visualizar a través de los requerimientos funcionales las necesidades y la relación directa con el análisis y diseño de la aplicación. Es importante en este proyecto para cumplir todos los requerimientos y tener una visión más precisa del modelo a implementar, permite explicar claramente a los *stakeholders* sus requerimientos y cómo estos se van a cumplir con la solución propuesta.

En la fase de elaboración se diseñan los diferentes puntos de vista arquitectónicos en los cuales se realizan diagramas de casos de uso detallados, de clases, de despliegue y componentes y el modelo de entidad relación.

El punto de vista de despliegue es necesario para identificar la infraestructura a nivel de red, ambiente de ejecución y tecnologías requeridas para el desarrollo de la aplicación.

Con el punto de vista de Información se puede identificar el manejo de la persistencia de los datos, el posible volumen de datos que se administraría con el sistema, la relación que debe existir entre los componentes y los repositorios de datos. Adicionalmente es importante identificar las características de la información como son sus flujos y ciclo de vida.

²⁴ JACOBSON, Ivar ; BOOCH, Grady y RUMBAUGH, James. El Proceso unificado de desarrollo de software, Addison-Wesley , 2000.

Con la fase de implementación se prepara el entorno para garantizar el buen desarrollo del proyecto y la disponibilidad de todos los medios requeridos; entre ellos: el servidor, el gestor de bases de datos y las herramientas de generación de código.

Por último, en la fase de transición se realiza el montaje y configuración del sistema de información web y se socializa el manual de usuario. Además, se realiza pruebas de aceptación por parte del usuario y se generan las mejoras requeridas, las cuales se aplican a la versión final del software.

5.2 FASE DE INICIO

5.2.1 Descripción General del Sistema a Desarrollar. El objetivo es desarrollar un sistema de ventas en línea para una PYME dedicada a la venta de joyería. En ella se requiere registrar los pedidos y ventas. Donde un pedido es una petición de productos y una venta es registrada en una factura.

El proceso de negocio que describe este escenario es el siguiente:

a. Cuando una empresa o cliente requiera algún tipo de bien, deberá diligenciar un formato de pedido. Este formulario debe contener el nombre del bien a comprar, la cantidad deseada, la información del comprador.

El mismo formulario puede contener múltiples ítems de compra por lo que se requiere un total aproximado de toda la solicitud, del cual puede sacar productos antes de proceder a realizar el formulario de pedido y enviarlo.

b. Cuando todos los ítems sean seleccionados, el pedido deberá ser cerrado y se procederá a la ejecución del pago, el cual se podrá realizar a través de pago electrónico con tarjeta de crédito o con bonos electrónicos vendidos por el almacén, consignación bancaria.

c. Efectuado el pago, el almacén debe proceder a realizar el envío a las empresas o clientes, sin embargo en caso de que algún o algunos de los productos no se encuentren en stock, se procederá a enviar parcialmente el pedido y se establecerá contacto directo con el cliente para acordar cambios en su orden de compra y devoluciones.

El sistema también deberá generar los siguientes reportes:

- El total de ventas por mes.
- El producto que más se vendió por mes.
- El cliente que más ha comprado en el mes.

El sistema debe soportar el crecimiento exponencial de pedidos.

El usuario o cliente, debe registrarse en la página cuando ingrese por primera vez a realizar un pedido y compra.

Objetivos de la documentación del sistema

- Realizar levantamiento de información a través de entrevistas (ver modelo de entrevista en anexo A) y análisis de documentos existentes que puedan aportar información al desarrollo del proyecto.
- Identificar todos los actores para reconocer los requerimientos del sistema.
- Identificar el modelo funcional y de información requerido que contenga los actores y los requerimientos del sistema.
- Identificar un estilo arquitectural que permita de manera flexible implementar el sistema, cubriendo los requerimientos de los diferentes actores.

Fundamento de la Solución

La solución propone la implementación de una arquitectura N-tier (o de n-capas), ya que permite que la información se procese de manera segura a través de las capas. Esta solución es proyectada a través del análisis de los requerimientos planteados por los actores del sistema en la sección 5.2.4.

La utilización de esta arquitectura simplificará el desarrollo y manejo de una aplicación con una seguridad en los datos más confiable y un fácil manejo por parte de los *stakeholders*.

5.2.2 Aproximación Arquitectural. Esta sección ofrece una explicación de la arquitectura aplicada en el diseño y formula los puntos de vista utilizados en el desarrollo del proyecto.

Para este proyecto es indicado el uso de la arquitectura por niveles (N-Tier), pues permite administrar correctamente el negocio y las interfaces del sistema, así mismo, maneja la interfaz gráfica independientemente de la lógica del negocio y a su vez éste tampoco depende del nivel de datos. Así se obtiene bajo acoplamiento y alta cohesión en los componentes del software desarrollados en cada capa del sistema.

Como no hay dependencia directa se podrá manipular en forma correcta un ambiente web sin discriminar dispositivos o componentes que requieran conectarse a este, permitiendo el manejo de componentes independientes y la adición de nuevos si es necesario, reduciendo el impacto que esto puede generar en la aplicación además sin limitar el uso de un motor de bases de datos particular sino al contrario admitiría el uso de varios de ellos si la aplicación así lo requiere.

Los puntos de vista esbozados para el proyecto como las vistas de despliegue, funcional y de información brindan diferentes perspectivas a lo largo del proceso de análisis y diseño del sistema.

5.2.3 Cubrimiento de Requerimientos. La arquitectura de este sistema facilitará el cumplimiento de los requerimientos como la seguridad de los datos y una alta usabilidad para todos los actores.

El manejo del requerimiento de seguridad se tiene presente en los puntos de vista funcional y de información, se valida la identificación de los usuarios que van a ingresar al

sistema, adicionalmente a través de los componentes de autenticación se identifican los roles de cada usuario, para evitar ingresos no requeridos y evitar pérdida de datos a través de manipulación incorrecta de la aplicación; y el acceso a la capa de persistencia se configura a través de la arquitectura implementada.

5.2.4 Actores del sistema. Esta sección presenta una lista de los actores involucrados en el Sistema de Información para joyerías con e-commerce (SIJEC).

Para cada uno de ellos, se deben listar las necesidades que van a ser tenidas en cuenta en este documento. Esta información se presenta en forma de matriz. En la tabla 2, se describen cada uno de los actores del sistema, en la tabla 3, las filas representan cada uno de los actores y las columnas el tipo de actor y las necesidades o requerimientos.

Tabla 2: Listado de los actores del sistema SIJEC

Actores	Descripción
Administrador	Persona principal en el manejo del negocio que debe tener acceso a reportes correspondientes a las ventas.
Director de Sistemas	Persona encargada de manejar los requerimientos de hardware y software del sistema.
Cliente	Persona que a través del sistema realizarán compras en la empresa.
Representante de ventas	Persona ubicada en la joyería encargada de vender.

Tabla 3: Actores y Requerimientos

Actores	Requerimientos
Administrador	<ul style="list-style-type: none"> • El total de ventas por mes. • El producto que más se vendió por mes. • El cliente que más ha comprado por mes. • Conocer los pedidos de compra después de haber sido ingresados al sistema. • Registrar, modificar ciudades, departamentos y productos.
Cliente	<ul style="list-style-type: none"> • Realización de pedidos a través de internet. • Registrarse en el sistema. • El proceso de solicitud debe ser claro y conciso. • Manejar protocolos de identificación, autenticación y autorización. • Realizar abonos.
Representantes de ventas	<ul style="list-style-type: none"> • Registrar clientes en el sistema. • Registrar abonos de clientes. • Hacer pedidos de clientes. • Consultar productos.
Desarrollador del sistemas	<ul style="list-style-type: none"> • Conocer los requerimientos de hardware y software.
Sistema	<ul style="list-style-type: none"> • Soportar un crecimiento anual del 5%.

De acuerdo con los requerimientos definidos por cada uno de los actores se determinan los puntos de vista más importantes para la arquitectura a desarrollar. En la tabla 4 se determinan los puntos de vista que se presentan para cada *actor*.

Tabla 4: Actores vs Puntos de vista

Actores	Puntos de vista
Gerente	De información y Funcional.
Representante de ventas	De información y Funcional.
Director de sistemas	Funcional, De información y despliegue
Clientes	Funcional.

5.2.5 Puntos de Vista Funcional. A través de este punto de vista se identifican los elementos funcionales del sistema y sus principales responsabilidades. A partir de ellos se generan diagramas que permiten identificar los servicios que se deben prestar.

A continuación, se listan los requerimientos funcionales del sistema.

a. Gestionar usuario

- Crear usuario (Nombre, apellido, celular, dirección, teléfono, Código, ciudad, email).
- Modificar información del usuario
- Consultar información del usuario.
- Validar usuario

b. Gestionar producto

- Capturar información del producto (código Producto, clave, valor, cantidad, descripción, fecha ingreso, tipo).

- Modificar información del producto.
- Consultar información del producto.

c. Gestionar Pedido

- Ingresar cantidad de productos a pedir.
- Adicionar el producto al pedido.
- Consultar el pedido actual del usuario.
- Modificar el pedido actual del usuario.
- Actualizar formulario del pedido actual del usuario.
- Enviar formulario de pedido.

d. Gestionar Pagos.

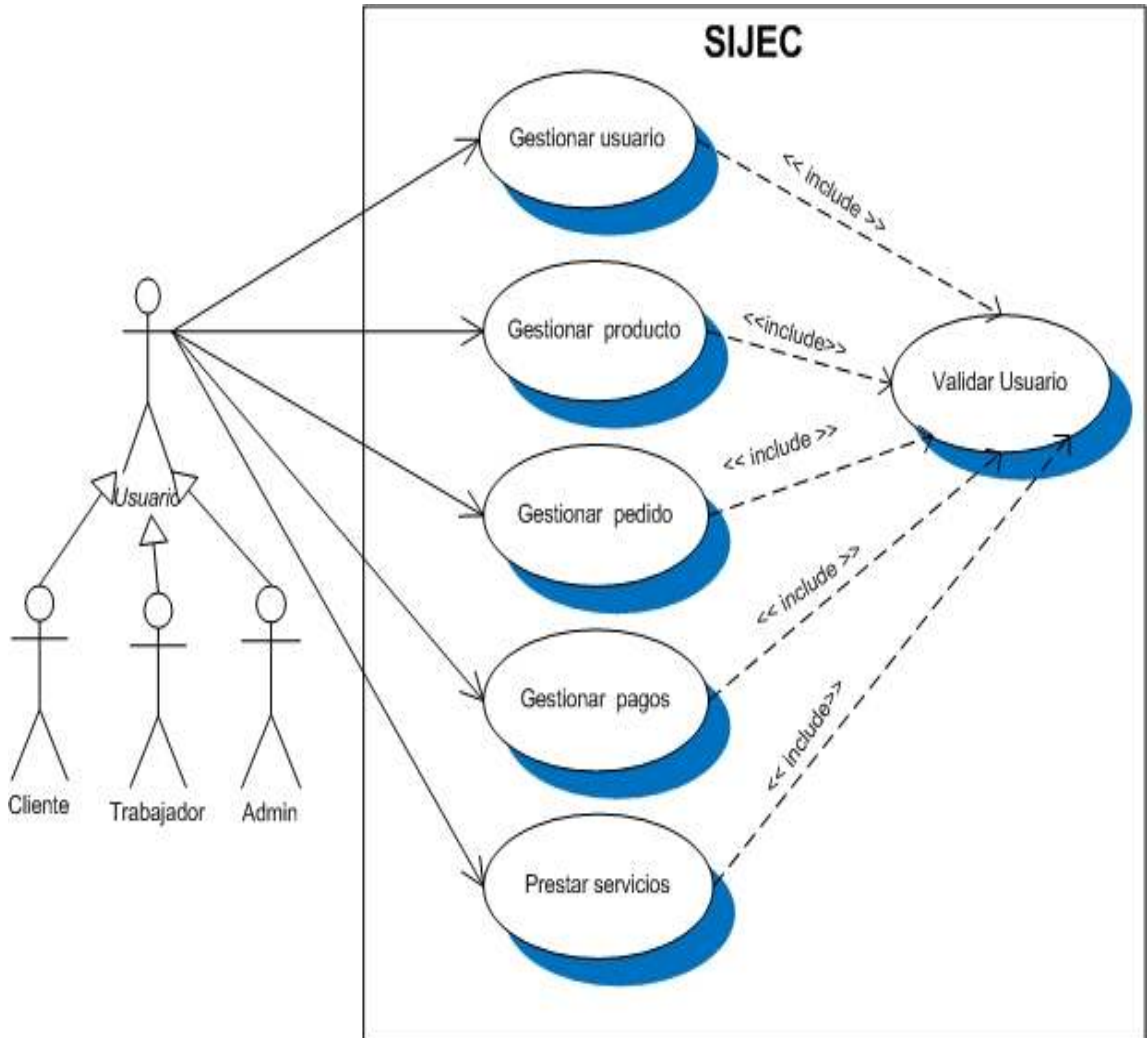
- Actualizar pagos.
- Pagar Pedido.

e. Prestar Servicios.

- Elegir el servicio requerido.
- Ingresar información.

• **Diagrama de Casos de uso general.** Presentado en la figura 3 muestra los casos de uso globales para el sistema de información web con *e-commerce* (SIJEC). En donde un usuario representa a un cliente, un trabajador o un administrador.

Figura 3: Diagrama de casos de uso nivel 1



En este nivel de caso de uso se presenta la generalización de un usuario en donde el puede ser un cliente un administrador o un trabajador. El cliente puede acceder a la gestión de usuario, pedidos y pagos, el trabajador puede acceder a la gestión de usuario, pedidos, pagos y servicios y el Administrador puede acceder a todas las funcionalidades del sistema.

5.2.6 Atributos de Calidad. En un sistema los atributos de calidad son aspectos a los cuales se les puede asignar una métrica que definen la calidad y las características que el sistema debe soportar. Dichos atributos se pueden analizar de una forma más sencilla ubicándolos dentro de un árbol de utilidades (ver sección 5.3.4, figura 34) al final de la fase de elaboración, el cual permitirá visualizar fácilmente las jerarquías de los atributos con sus respectivos escenarios para poder determinar los puntos más sensibles del sistema.

a) Perspectivas. En esta sección se describe el comportamiento y los atributos de calidad de los requerimientos que afectan la arquitectura de software. Aquí se especificarán los atributos de seguridad y usabilidad de forma más detallada, considerados como los más importantes en el proyecto debido a la importancia requerida en la seguridad de los datos y el nivel de manejo de sistemas de información por parte de los usuarios.

En la tabla 5 se desarrollan algunos puntos para el atributo de calidad de seguridad. Entre esos puntos se encuentran la calidad deseada, su aplicabilidad, los requerimientos, las actividades y tácticas para llevarlas a cabo.

Tabla 5: Atributo de calidad: Seguridad

<Seguridad>	
Calidad Deseada	Alta desempeño en la protección de la información
Aplicabilidad	Aplica para la vista Funcional, de concurrencia y desarrollo.
Requerimientos	Políticas, amenazas y disponibilidad
Actividades	<ul style="list-style-type: none"> • Definir políticas de seguridad. • Identificar los riesgos de seguridad en los datos. • Evaluar las amenazas del sistema. • Diseñar la implementación del sistema.
Tácticas	<ul style="list-style-type: none"> • Usar mecanismos de registro y autenticación. • Proteger la integridad de la información y asegurar la disponibilidad.

En la tabla 6 se desarrollan los puntos para el atributo de calidad de usabilidad. Entre esos puntos se encuentran la calidad deseada, su aplicabilidad, los requerimientos, las actividades y tácticas para llevarlas a cabo.

Tabla 6: Atributo de calidad: usabilidad

<Usabilidad>	
Calidad Deseada	Facilidad de las personas para interactuar con el sistema
Aplicabilidad	Aplica para la vista Funcional y de información.
Requerimientos	<ul style="list-style-type: none"> • Los clientes deben realizar compras con facilidad. • Los usuarios de la empresa no se deben demorar más de una hora en aprender a manejar todo el sistema. • El proceso de navegación por el sistema debe ser simple entendible y consistente.
Actividades	<ul style="list-style-type: none"> • Identificar los tipos de usuarios que van a utilizar el sistema y sus capacidades. • Entender el contexto en el que el sistema va a ser usado. • Filtros, reportes y análisis de información.
Tácticas	<ul style="list-style-type: none"> • Separar la implementación de la interface de usuario de la lógica del negocio. • Categorizar los datos que se van a presentar. • Definir las reglas de navegación para que el usuario pueda desplazarse a través del aplicativo de manera sencilla y práctica.

b) Escenarios de Calidad. Un escenario de calidad es un caso concreto, más detallado que se puede predecir sin haber implementado el software, en donde se puede visualizar si el sistema cumple o no con un atributo de calidad. Esta sección presenta los diferentes escenarios de calidad utilizados para modelar los estímulos y las respuestas esperadas por el sistema una vez en ejecución y que han sido tenidos en cuenta durante la definición de la arquitectura.

La tabla 7 presenta un escenario de calidad para el caso en que un nuevo usuario accede al sistema, expresando las respuestas que tiene este para manipular el sistema y cuanto tarda en manejarlo.

Tabla 7: Usuario nuevo accediendo al sistema (Usabilidad)

Usuario nuevo accediendo al sistema (Usabilidad)	
Descripción	Un usuario que nunca ha ingresado al sistema, y no sabe usarlo, desea registrarse para luego poder hacer una compra de un producto.
Fuente	Usuario no registrado.
Estímulo	Un usuario que desea comprar un artículo.
Artefacto	El sistema
Ambiente	Bajo condiciones normales.
Respuesta	El usuario aprende a usar el sistema y puede acceder a él y realizar una compra de un artículo deseado.
Medida de la Respuesta	Máximo en 6 minutos para que un usuario pueda registrarse y aprender a hacer una compra.

Para ingresar al sistema un usuario debe haberse registrado previamente. La tabla 8 describe el comportamiento de este escenario de calidad para determinar si el sistema cumple o no con el atributo de calidad de seguridad bajo condiciones normales o extremas y cómo responde.

Tabla 8: Escenario de calidad: Ingresar al sistema

Ingresar al sistema (Seguridad)	
Descripción	Un usuario del sistema desea entrar al sistema usando su usuario y contraseña ingresados al momento de hacer su registro previamente.
Fuente	Usuario registrado previamente.
Estímulo	Dar clic en login
Artefacto	Módulo de control de acceso.
Ambiente	Bajo condiciones normales y extremas.
Respuesta	El usuario accede a su cuenta luego de ingresar los datos requeridos.
Medida de la Respuesta	El 100% de los usuarios deben identificarse para acceder al sistema.

El siguiente escenario, consulta de información básica de un Producto, determina bajo qué condiciones se debe llevar a cabo el cumplimiento del atributo de calidad de desempeño. En la tabla 9, se describe el comportamiento de este, la acción que hace que se evidencie el atributo de desempeño y la respuesta del sistema.

Tabla 9: Consulta de información básica de un Producto (Desempeño)

Consulta de información básica de un Producto (Desempeño)	
Descripción	Un usuario del sistema con privilegios de cliente, ingresa al sistema y en la pantalla de consultar producto, ingresa un parámetro de búsqueda y obtiene toda la información almacenada sobre el producto y su cantidad en stock. Al dar clic en consultar, el tiempo de despliegue en pantalla de la información debe ser menor a 10 seg.
Fuente	Usuario con permisos de consultar información de productos.
Estímulo	Dar clic para desplegar la información de un producto determinado.
Artefacto	Módulo de consulta de productos.
Ambiente	Operaciones Normales
Respuesta	Se despliega la información básica del cliente deseado.
Medida de la Respuesta	Menor a 2 segundos.

El escenario Consultar Ventas muestra bajo qué condiciones se debe llevar a cabo el cumplimiento del atributo de calidad de disponibilidad. En la tabla 10, se hace una descripción general del escenario, quienes intervienen, los estímulos y las respuestas del sistema.

Tabla 10: Escenario de calidad Consultar Ventas (Disponibilidad)

Consultar Ventas (Disponibilidad)	
Descripción	Un usuario del sistema con privilegios de administrador, ingresa al sistema y en el menú de administrador intenta conocer el reporte de ventas realizadas.
Fuente	Un usuario con permisos de generar reportes de ventas
Estímulo	Dar clic en generar reporte de ventas.
Artefacto	Módulo de generación de reportes.
Ambiente	Operaciones Normales.
Respuesta	Se despliega el reporte de las ventas.
Medida de la Respuesta	El 99.9% de los reportes son generados correctamente.

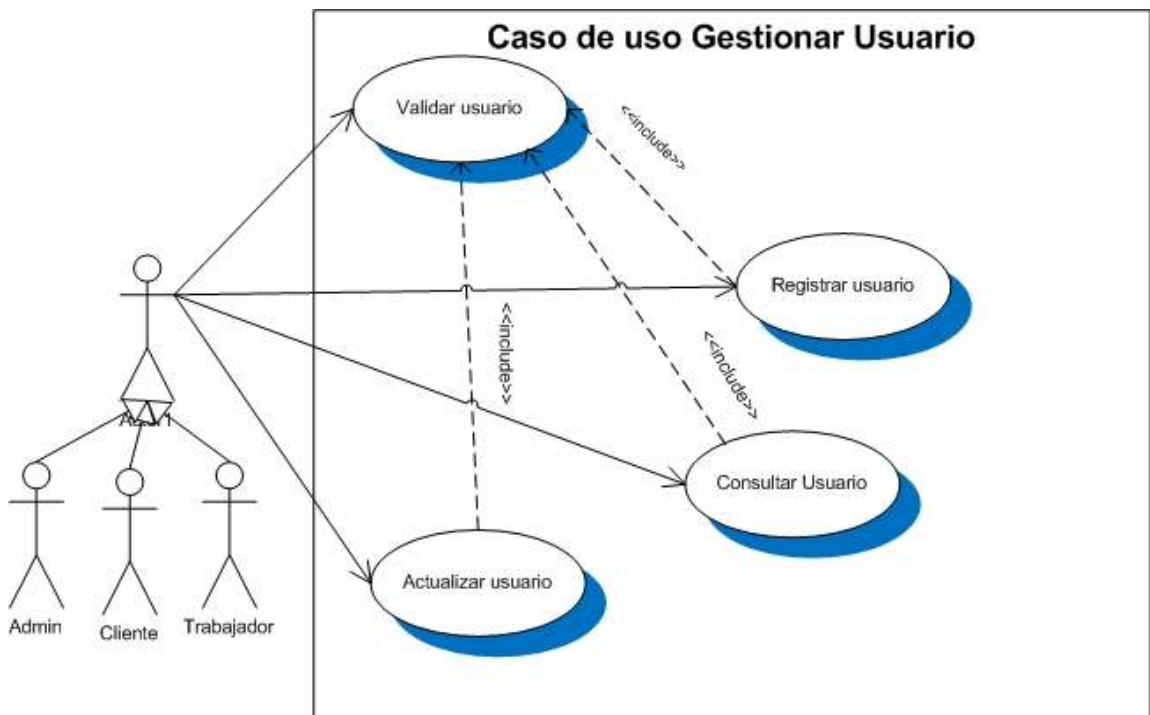
5.3 FASE DE ELABORACIÓN

En esta fase se diseñan diagramas de casos de uso, de actividades y de componentes desarrollados a través de los puntos de vista de despliegue, funcional y de información los cuales se describen a continuación.

5.3.1 Punto de vista funcional fase de elaboración. Aquí se modelan los diagramas de casos de uso de forma detallada. A partir de ellos se generan diagramas que permiten identificar los servicios que se deben prestar y se modela el diagrama de componentes que debe ser implementado para que el sistema cumpla con los requerimientos y los atributos de calidad de los actores.

A continuación se describirán los diferentes niveles de casos de uso para cada actor.

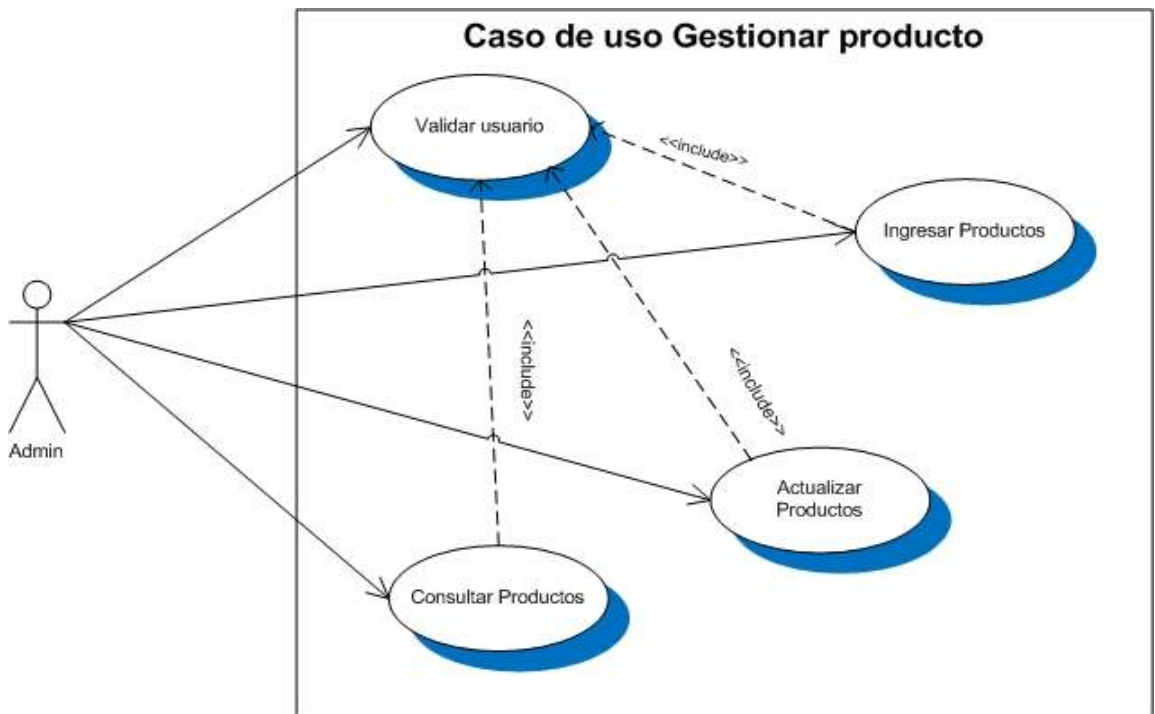
Figura 4: Casos de uso Gestionar usuario



En la figura 4 se muestra el caso de uso Gestionar Usuario el cual es ejecutado por los clientes, trabajadores o el administrador, de él se desprende otros casos de usos para

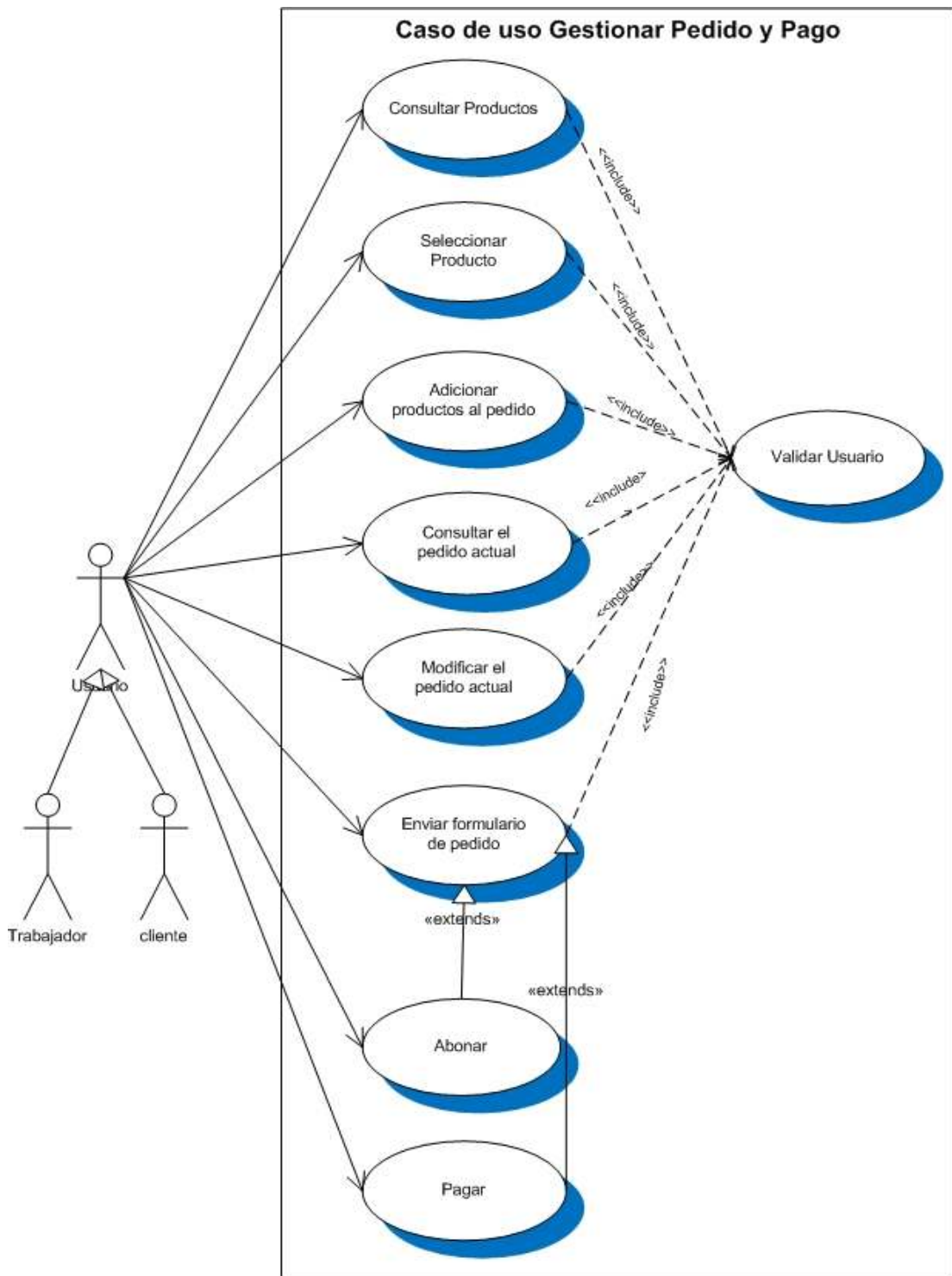
registrar, consultar y actualizar Usuarios. Para que esto se lleve a cabo todos los actores deben previamente validarse ante el sistema.

Figura 5: Caso de uso Gestionar Producto



En la figura 5 se muestra el caso de uso Gestionar Producto el cual es ejecutado por el administrador, de él se desprenden otros casos de usos como registrar, consultar y actualizar la información de los productos. Para que esto se lleve a cabo el administrador debe validarse previamente en el sistema.

Figura 6: Diagrama de casos de uso Gestionar Pedido



En la figura 6 se representa de forma detallada el casos de uso que se ejecutan cuando un cliente quiere acceder y comprar un producto. Los casos de uso consultar, seleccionar, adicionar productos al pedido, consultar, modificar y enviar pedido, constituyen el carrito de compras de la aplicación.

Sólo los usuarios trabajador y cliente pueden acceder a estas funcionalidades con previa autenticación de usuario.

- **Especificación de requerimientos**

En las siguientes tablas se describen los casos de uso de cada uno de los diagramas anteriores. Se muestra el nombre, los actores que intervienen, el tipo el propósito y el resumen para cada caso de uso, además, que hace el actor y cómo el sistema responde a sus acciones.

La tabla 11 describe el caso de uso mostrado en la figura 5.

Tabla 11: Caso de uso Registrar usuario

Caso de uso	Registrar Usuario.
Actores	Usuario
Tipo	Primario y esencial.
Propósito	Permitir al usuario hacer la inscripción en el sistema de la joyería.
Resumen	El usuario inicia este caso de uso.

La tabla 12 muestra que debe hacer un usuario para registrar por primera vez sus datos y como el sistema responde ante estas acciones.

Tabla 12: Curso normal de los eventos caso de uso registrar usuario

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario ingresa al formulario de registro de datos.	2. Se conecta a la base de datos.
3. El actor digita los datos en los campos.	4. Verifica si es correcta la información ingresada en los diferentes campos de formulario.
5. El actor realiza el almacenamiento de los datos ingresados.	6. Se despliega en pantalla un mensaje donde le pregunta al usuario si está seguro de registrarse.
7. El actor da respuesta a la pregunta visualizada.	8. Se despliega en pantalla una ventana de confirmación de registro completado.

La tabla 13 describe el caso de uso mostrado en la figura 4.

Tabla 13: Caso de uso actualizar usuario

Caso de uso:	Actualizar Usuario.
Actores:	Usuario
Tipo:	Primario y esencial.
Propósito:	Permitir al usuario actualizar los datos de inscripción.
Resumen:	El usuario inicia ingresando.

La tabla 14 muestra que debe hacer un usuario para actualizar sus datos y cómo el sistema responde ante estas acciones.

Tabla 14: Curso normal de los eventos Caso de uso actualizar datos

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario ingresa al formulario de registro de datos.	2. Se despliega en pantalla el formulario de actualización de datos, con los datos anteriores del usuario.
3. El actor digita los datos en los campos que desee modificar.	4. Verifica si es correcta la información ingresada en los diferentes campos de formulario.
5. El actor realiza el almacenamiento de los datos ingresados.	6. Se despliega en pantalla un mensaje donde le pregunta al usuario si está seguro de actualizar o no su formulario.
7. El actor da respuesta a la pregunta visualizada.	8. Se despliega en pantalla una ventana de confirmación de actualización realizada.
Cursos Alternos	
Línea 7: El actor puede cancelar la actualización de su formulario.	

La tabla 15 describe el caso de uso mostrado en la figura 4.

Tabla 15: Caso de uso validar usuario

Caso de uso	Validar Usuario.
Actores:	Usuario
Tipo:	Inclusión
Propósito:	Permitir al usuario ingresar en el sistema de la joyería.
Resumen	El usuario inicia este caso de uso. Valida al usuario mediante un login y password a verificarse con su respectivo registro de usuario para que pueda utilizar el sistema.
Precondición	Se requiere haber ejecutado antes el caso de uso de registro Usuario.
Excepciones	No hubo validación: el <i>login/password</i> no se valido correctamente. Se solicita al usuario volver a registrarse.

La tabla 16 muestra que debe hacer un usuario para validarse ante el sistema y cómo el sistema responde ante estas acciones.

Tabla 16: Curso normal de los eventos Caso de uso validar Usuario

Acciones del Actor	Respuesta del sistema
1. El usuario visualiza las opciones del sistema.	2. El sistema muestra las opciones de registrarse por primera vez, "OK" y "Salir".
3. Si el actor elige "registrarse por primera vez"	4. El sistema ejecuta el caso de uso Registrar usuario. Muestra el formulario de inscripción.
5. Si el actor elige "OK"	6. Valida el registro de usuario mediante un login y un password. Una vez validado el usuario, muestra los servicios del sistema.
7. El actor da respuesta a la pregunta visualizada.	8. Se despliega en pantalla una ventana de confirmación de registro completado.

La tabla 17 describe el caso de uso mostrado en la figura 4.

Tabla 17: Caso de uso consultar usuarios

Caso de uso:	Consultar usuarios.
Actores:	Administrador.
Tipo:	Primario y esencial.
Propósito:	Consultar los usuarios registrados en la base de datos.
Resumen	El administrador ingresa a la opción de consulta para verificar registros.

La tabla 18 muestra que debe hacer un usuario para consultar sus datos y cómo el sistema responde ante estas acciones.

Tabla 18: Curso normal de los eventos Caso de uso consultar Usuarios

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el administrador escoge la opción de consultar usuarios.	2. Se despliega en pantalla una tabla de los usuarios de acuerdo al criterio de consulta.
3. El actor selecciona los criterios de consulta.	4. Muestra en pantalla el resultado de la consulta. Si el usuario a consultar no existe se despliega un mensaje de error
5. Visualiza la información.	.

La tabla 19 describe el caso de uso ingresar productos mostrado en la figura 5.

Tabla 19: Caso de uso ingresar productos

Caso de uso:	Ingresar Productos
Actores:	Administrador.
Tipo:	Primario y esencial.
Propósito:	Brindar al cliente mayor surtido de productos para su disposición.
Resumen	El administrativo digita los datos de los nuevos productos para almacenarlos en la base de datos del sistema.

La tabla 20 muestra que debe hacer un administrador para ingresar un producto y cómo el sistema responde ante estas acciones.

Tabla 20: Curso normal de los eventos Caso de uso Ingresar productos

Acciones del Actor	Respuesta del sistema
<p>1. Este caso de uso comienza cuando el usuario selecciona la opción adicionar productos.</p>	<p>2. Se despliega en pantalla el formulario de adición de productos.</p>
<p>3. El actor adiciona los nuevos productos ingresando los datos de código, clave, costo, precio, cantidad, descripción fecha ingreso.</p>	<p>4. Verifica sí es correcta la información ingresada en los diferentes campos de formulario.</p>
<p>5. Guarda las acciones realizadas.</p>	<p>6. Almacena los nuevos productos en la base de datos.</p>
	<p>7. Se despliega en pantalla una ventana de confirmación de registro de productos.</p>
<p>Cursos alternos</p>	
<p>Línea 4: si no se logra registrar aparece el siguiente mensaje. No se ha adicionado el producto porque falta llenar algunos campos obligatorios para el formulario.</p>	<p>Línea 6: Ocurrió un error el producto que desea adicionar ya está registrado en la base de datos.</p>

La tabla 21 describe el caso de uso actualizar productos mostrado en la figura 5.

Tabla 21: Caso de uso actualizar productos

Caso de uso:	Actualizar productos.
Actores:	Administrador.
Tipo:	Primario y esencial.
Propósito:	Modificar cantidades, precios, y costos de los productos registrados en la base de datos.
Resumen	El administrativo tiene el privilegio de realizar las modificaciones de los productos y posteriormente actualizarlo en la base de datos.

La tabla 22 muestra que debe hacer un administrador para actualizar los productos y como el sistema responde ante estas acciones.

Tabla 22: Curso normal de los eventos Caso de uso actualizar productos

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario selecciona la opción actualizar productos.	2. Se despliega en pantalla el formulario de actualizar de productos.
3. El actor modifica los campos que desea actualizar del producto como la clave, costo, precio, cantidad, descripción o de fecha ingreso.	4. Verifica sí es correcta la información ingresada en los diferentes campos de formulario.
5. Guarda las acciones realizadas.	6. Almacena la nueva información del producto en la base de datos
	7. Se despliega en pantalla una ventana de confirmación de actualización de productos.
Cursos alternos	
Línea 4: si no se logra registrar aparece el siguiente mensaje. No se ha adicionado el producto porque falta llenar algunos campos obligatorios para el formulario.	

La tabla 23 describe el caso de uso Consultar Producto mostrado en la figura 5.

Tabla 23: Caso de uso consultar productos

Caso de uso:	Consultar productos
Actores:	Administrador.
Tipo:	Primario y esencial.
Propósito:	Consultar los productos registrados en la base de datos.
Resumen	El administrador ingresa a la opción de consulta para verificar registros.

La tabla 24 muestra que debe hacer un administrador para consultar un producto y como el sistema responde ante estas acciones.

Tabla 24: Curso normal de los eventos Caso de uso Consultar productos

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el administrador escoge la opción de consultar productos.	2. Se despliega en pantalla una tabla de los productos de acuerdo al criterio de consulta.
3. El actor selecciona los criterios de consulta.	4. Muestra en pantalla el resultado de la consulta
5. Visualiza la información	

La tabla 25 describe el caso de uso seleccionar producto mostrado en la figura 6.

Tabla 25: Caso de uso seleccionar producto.

Caso de uso	Seleccionar el producto.
Actores:	Usuario
Tipo:	Primario y esencial.
Propósito:	Seleccionar el producto que se desea pedir.
Resumen	El usuario comienza seleccionando los productos que se encuentran listados.

La tabla 26 muestra que debe hacer un cliente para seleccionar un producto y cómo el sistema responde ante estas acciones.

Tabla 26: Curso normal de los eventos Caso de uso seleccionar producto

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario está en el formulario de consulta de productos.	2. El sistema muestra el formulario de consulta al usuario.
3. El actor selecciona el producto en el formulario de consulta.	4. El sistema muestra el producto al usuario.

La tabla 27 describe el caso de uso Adicionar productos al pedido mostrado en la figura 6.

Tabla 27: Caso de uso adicionar el producto al pedido

Caso de uso:	Adicionar El Producto Al Pedido
Actores:	Usuario.
Tipo:	Primario y esencial.
Propósito:	Anexar los productos al formulario de pedido
Resumen:	El usuario adiciona el producto al formulario de pedido.

La tabla 28 muestra que debe hacer un administrador para adicionar un producto al pedido y cómo el sistema responde ante estas acciones.

Tabla 28: Curso normal de los eventos Caso de uso adicionar producto.

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario escoge la opción de pedidos.	
2. El actor después de haber seleccionado el producto e ingresado la cantidad, se adiciona al formulario de pedido.	3. Se enlaza con su respectivo formulario de pedido para la adición del producto.

La tabla 29 describe el caso de uso consultar el pedido de usuario mostrado en la figura 6.

Tabla 29: Caso de uso consultar el pedido de usuario.

Caso de uso:	Consultar el pedido de usuario
Actores:	Usuario.
Tipo:	Primario y esencial.
Propósito	Verificar los productos adicionales al pedido.
Resumen	El usuario ingresa a la sección de pedidos en proceso.

La tabla 30 muestra que debe hacer un cliente para consultar un pedido y cómo el sistema responde ante estas acciones.

Tabla 30: Curso normal de los eventos Caso de uso consultar pedido

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario ingresa a la sección de consulta de pedido que el usuario actualmente está realizando y comienza su carga	2. Se despliega en pantalla el formulario de pedido que está realizando el usuario.

La tabla 31 describe el caso de uso modificar el pedido de usuario mostrado en la figura 6.

Tabla 31: Caso de uso modificar el pedido actual del usuario

Caso de uso:	Modificar el pedido actual del usuario
Actores:	Usuario.
Tipo:	Primario y esencial.
Propósito:	Permitir corregir y verificar el pedido actual del usuario.
Resumen	El usuario visualiza la consulta general del pedido permitiendo realizar modificaciones tanto de cantidades como la eliminación de un determinado producto antes seleccionado en el pedido.

La tabla 32 muestra que debe hacer un cliente para modificar el pedido actual y cómo el sistema responde ante estas acciones.

Tabla 32: Curso normal de los eventos Caso de uso modificar pedido.

Acciones del Actor	Respuesta del sistema
<p>1. Este caso de uso comienza cuando el usuario visualiza los productos adicionales a su pedido.</p>	<p>2. El sistema le da la posibilidad al usuario de realizar modificaciones tanto de cantidades como la eliminación total del producto adicionado en el formulario de pedido.</p>
<p>3. El actor tiene el permiso de realizar las modificaciones tanto de cantidades como la eliminación total del producto adicionado en el formulario de pedido.</p>	<p>4. El sistema actúa de acuerdo a la decisión del cliente.</p>

La tabla 33 describe el caso de uso actualizar formulario del pedido actual del usuario mostrado en la figura 6.

Tabla 33: Caso de uso actualizar formulario del pedido actual del usuario.

Caso de uso:	Actualizar Formulario Del Pedido Del Usuario
Actores:	Usuario.
Tipo:	Primario y esencial.
Propósito:	Verificar las correcciones realizadas en el formulario de pedido actual del usuario.
Resumen:	El usuario una vez realizadas las correcciones respectivas de su pedido ingresa a la sección de actualización.

La tabla 34 muestra que debe hacer un cliente para actualizar el formulario del pedido actual y cómo el sistema responde ante estas acciones.

Tabla 34: Curso normal de los eventos Caso de uso actualizar formulario del pedido actual.

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario visualiza los productos adicionales a su pedido.	
2. El usuario después de haber realizados las respectivas modificaciones de su pedido, ingresa a la sección de actualización.	3. Se despliega en pantalla un mensaje, su pedido ha sido actualizado.
4. El actor acepta el mensaje.	

La tabla 35 describe el caso de uso enviar formulario de pedido mostrado en la figura 6.

Tabla 35: Caso de uso enviar formulario de pedido.

Caso de uso:	Enviar formulario de pedido.
Actores:	Usuario.
Tipo:	Primario y esencial.
Propósito:	Permitir al usuario llevar a cabo la realización de su pedido.
Resumen:	El usuario confirma la realización de su pedido ingresando a la sección.

La tabla 36 muestra que debe hacer un cliente para enviar el formulario del pedido y cómo el sistema responde ante estas acciones.

Tabla 36: Curso normal de los eventos Caso de uso enviar formulario de pedido.

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario finaliza el llenado del formulario de pedido y procede a enviarlo.	2. Se despliega en pantalla un mensaje donde le pregunta al usuario si está seguro de enviar o no su pedido.
3. El actor da respuesta a la pregunta visualizada.	4. Si la respuesta es afirmativa se despliega en pantalla un mensaje, su pedido se ha realizado exitosamente, de lo contrario volverá al formulario de consulta de pedido.
	5. Registrar la información en la base de datos.
Cursos alternos	
Línea 3: El actor puede cancelar el pedido.	

La tabla 37 describe el caso de uso actualizar pago mostrado en la figura 6.

Tabla 37: Caso de uso actualizar pago.

Caso de uso:	Actualizar pago
Actores:	Usuario.
Tipo:	Primario y esencial.
Propósito:	Realizar un abono a una factura que no haya sido cancelada en su totalidad.
Resumen:	El usuario ingresa la cantidad de dinero que desea abonar y efectúa el pago correspondiente.

La tabla 38 muestra que debe hacer un cliente para abonar a una factura y cómo el sistema responde ante estas acciones.

Tabla 38: Curso normal de los eventos Caso de uso actualizar pago

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario selecciona la opción de abonar.	2. Se despliega en pantalla un mensaje donde el sistema informa la cantidad de dinero que falta por pagar.
3. El actor digita la cantidad de dinero que desea abonar.	4. Se despliega en pantalla un mensaje donde el sistema informa la cantidad de dinero que falta por pagar o si será cancelado en su totalidad.
	5. Registrar la información en la base de datos.

La tabla 39 describe el caso de uso salir.

Tabla 39: Caso de uso salir

Caso de uso:	Salir
Actores:	Usuario, Bases de datos.
Tipo:	Primario y esencial.
Propósito:	Terminar la ejecución de la aplicación.
Resumen:	El usuario tiene la posibilidad de salir del sistema de pedidos sin realizar necesariamente un pedido.

La tabla 40 muestra que debe hacer un cliente para salir del sistema y cómo el sistema responde ante estas acciones.

Tabla 40: Curso normal de los eventos Caso de uso salir

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario ingresa al formulario de actualización de datos.	2. Se despliega en pantalla un mensaje de terminación de la sección del sistema de pedidos.

La tabla 41 describe el caso de uso generar reportes.

Tabla 41: Caso de uso Generar Reportes

Caso de uso:	Generar reportes
Actores:	Administrador.
Tipo:	Primario y esencial.
Propósito:	Consultar la información contable.
Resumen:	Administrador inicia este caso de uso. Ofrece funcionalidad de consultar la información de las ventas.

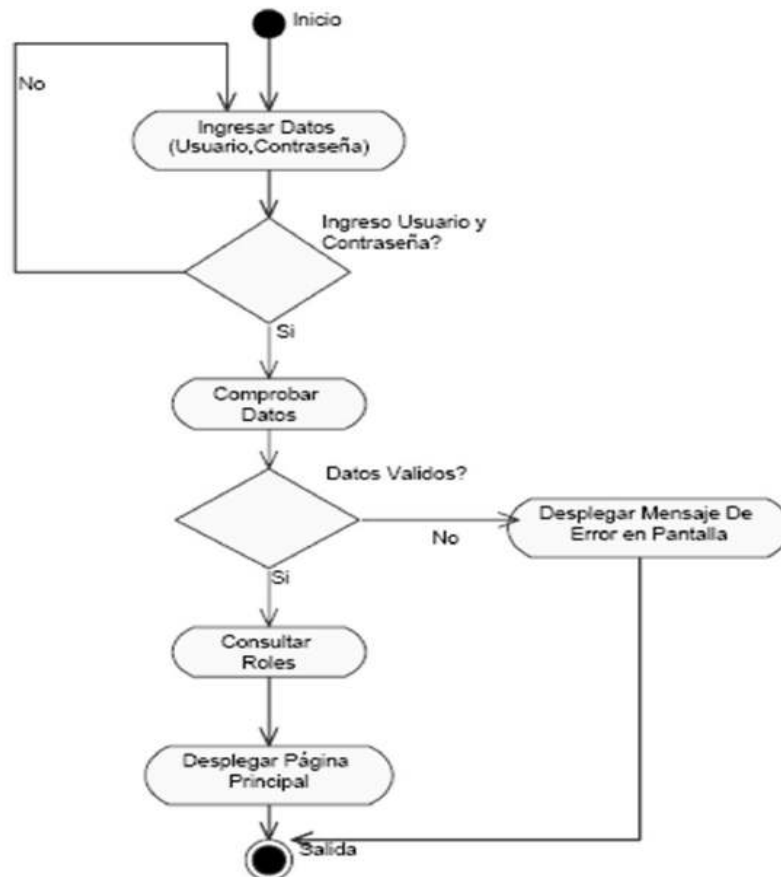
La tabla 42, muestra que debe hacer un administrador para generar reportes y cómo el sistema responde ante estas acciones.

Tabla 42: Curso normal de los eventos Caso de uso generar reportes

Acciones del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario ingresa a la opción de generar reportes.	2. El sistema muestra el tipo de reporte a generar.
3. El usuario elige la el tipo de reporte a consultar.	4. Según la elección del usuario el sistema muestra el reporte.

- **Diagrama de actividades.** A través de estos diagramas se identifican claramente los flujos de cada uno de los requerimientos funcionales, en este caso se tuvieron presentes los requerimientos más representativos.

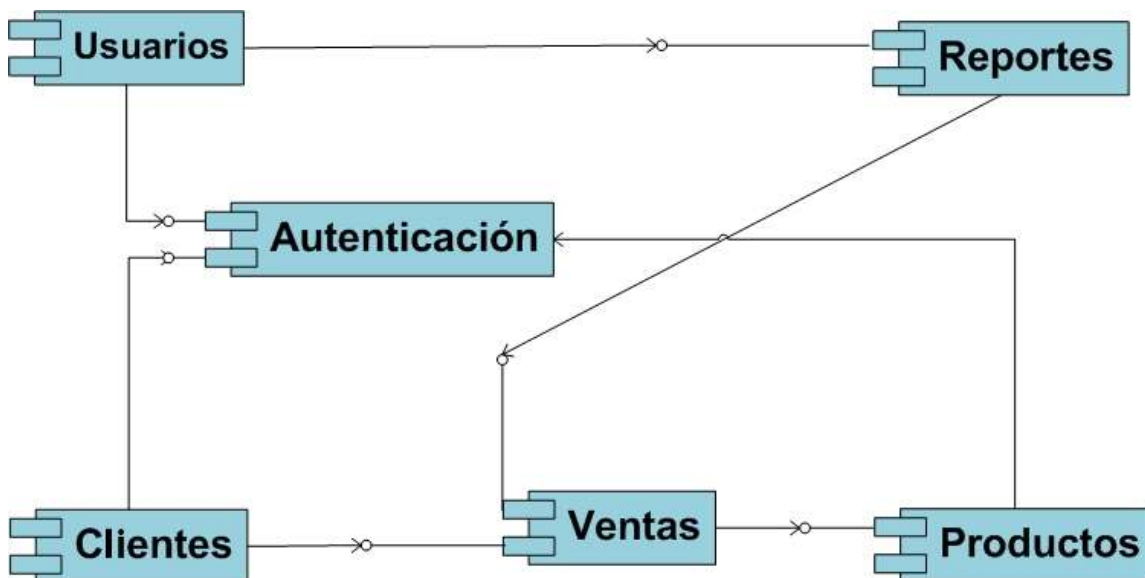
Figura 7: Diagrama de flujo de datos para validación de usuario.



La figura 7 representa el proceso de validación de un usuario para iniciar se debe ingresar el usuario y la contraseña, a continuación el sistema procede a comprobar los datos si estos son válidos despliega las páginas correspondientes a cada rol. Si no son válidos se despliega un mensaje de error en la página.

- **Diagrama de componentes.** Presentado en la figura 8, muestra los componentes que se consideran necesarios para cubrir los requerimientos presentados.

Figura 8: Diagrama de componentes.



Los componentes que describen principalmente el sistema son: “Usuarios” (Personas que pertenecen a la empresa con un papel específico) que solicitarán “Reportes” de “Ventas” y después de hacer su “Autenticación” harán el manejo de los “Productos”. Los “Clientes” por su parte, efectuarán compras y en algunos casos necesitaran “Autenticación”.

5.3.2 Punto de Vista de Información. A través de este punto de vista se modela la vista que identifica la forma en que se distribuye y persiste la información de la aplicación, teniendo presentes los flujos de datos, los dueños de la información, la calidad de los datos y la separación entre la aplicación por componentes y los repositorios que van a mantener los datos.

- **Modelos de Estructuras Estáticas de Datos.** Con estos diagramas de clases se tiene una aproximación de las entidades que se requieren para mantener los datos, la dirección de las relaciones entre estos componentes y la multiplicidad de estas relaciones. Es importante identificar si las relaciones son: uno-uno, uno-muchos, muchos-uno, muchos—muchos y como se van a direccionar.

- **Diagrama de clases.** Presentado en la figura 9 se utilizaron nueve entity bean: Usuario, TipoProd, Factura, Venta, Producto, ciudad, departamento, abono y servicio cada una con una correspondencia directa a las tablas existentes en la base de datos.

La mayoría de las relaciones entre cada una de las clases son de 1 a muchos, por ejemplo un usuario pertenece a una ciudad pero en una ciudad puede registrarse muchos usuarios.

Un usuario puede generar muchas facturas, pero una factura sólo la genera un cliente.

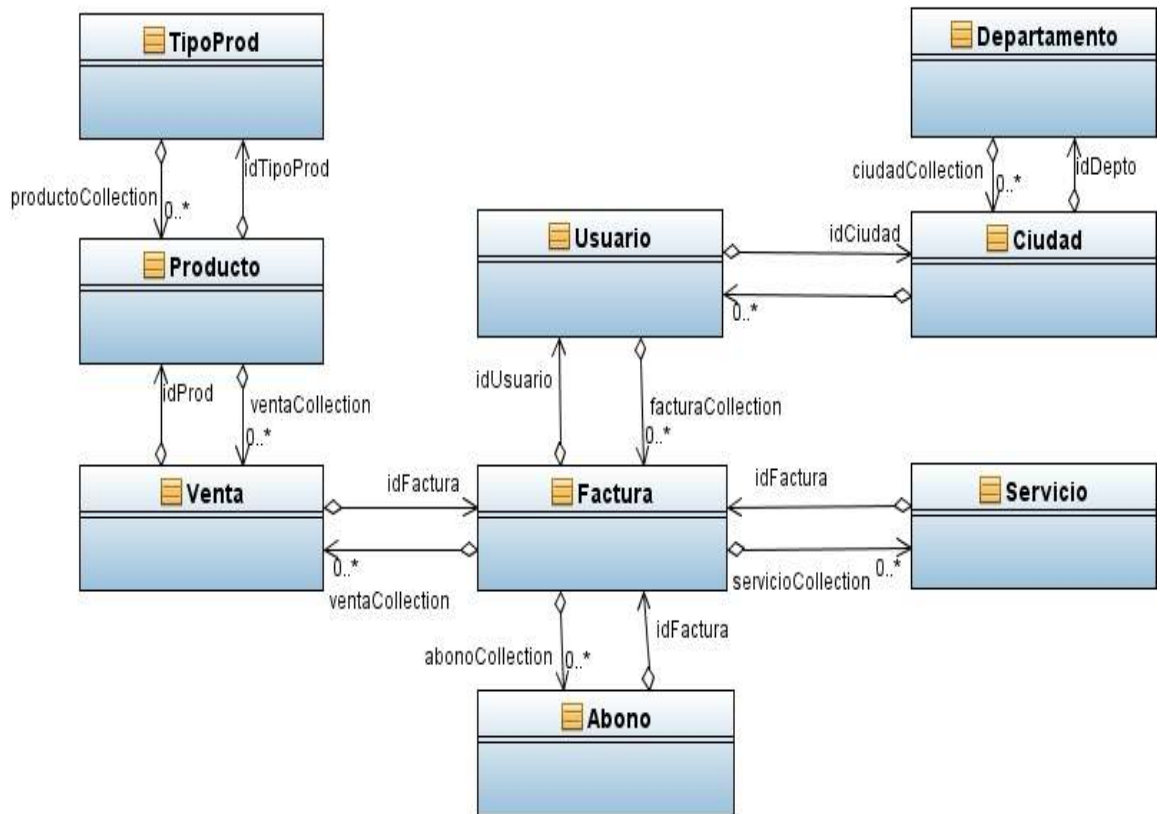
En una factura se pueden registrar muchas ventas y una venta solo pertenece a una factura.

Un producto sólo puede ser de a un tipo de producto, pero a un tipo de producto pueden pertenecer muchos productos.

Por otro lado un departamento posee muchas ciudades y municipios pero una ciudad solo puede pertenecer a un departamento.

Para finalizar una factura puede tener muchos abonos y muchos servicios.

Figura 9: Diagrama de clases para Entity beans




Cada una de las clases identificadas en el diagrama de clases de la figura 9 será descrita a continuación:

Figura 10: Clase Entity Abono



Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla abono, posee los siguientes atributos: idAbono llave primaria, fecha y valor. También posee métodos *get* y *set* para cada uno de los atributos que serán utilizados para obtener o cambiar los valores de sus atributos, un método *hashCode()* para el manejo de arreglos de objetos y un método *toString()* para devolver cadenas de caracteres que contengan información de este objeto.

Figura 11: Clase Entity TipoProd

 TipoProd
<i>Attributes</i>
<pre>private long serialVersionUID = 1L private Integer idTipoProd private String nombre</pre>
<i>Operations</i>
<pre>public TipoProd() public TipoProd(Integer idTipoProd) public TipoProd(Integer idTipoProd, String nombre) public Integer getIdTipoProd() public void setIdTipoProd(Integer idTipoProd) public String getNombre() public void setNombre(String nombre) public Producto[0..*] getProductoCollection() public void setProductoCollection(Producto productoCollection[0..*]) public int hashCode() public boolean equals(Object object) public String toString()</pre>

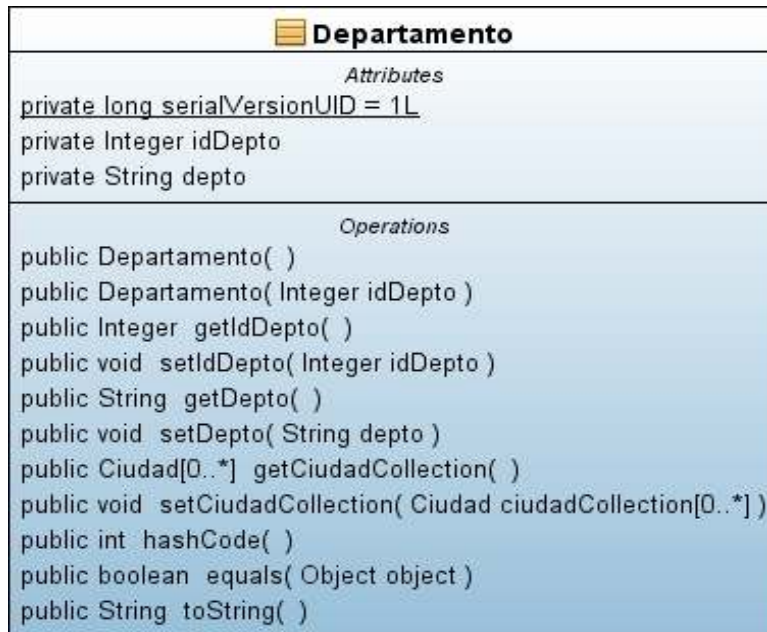
Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla tipo producto, posee los siguientes atributos: idTipoProd llave primaria de la tabla, y el nombre del tipo de producto. También muestra los métodos *get* y *set* para esta clase y los métodos *hashCode()* y *toString* explicados en el *entity* anterior.

Figura 12: Clase Entity Servicio



Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla servicio, posee los siguientes atributos: idServicio llave primaria de la tabla, valor, descripción y estado. También muestra los métodos para obtener o cambiar los valores de sus datos.

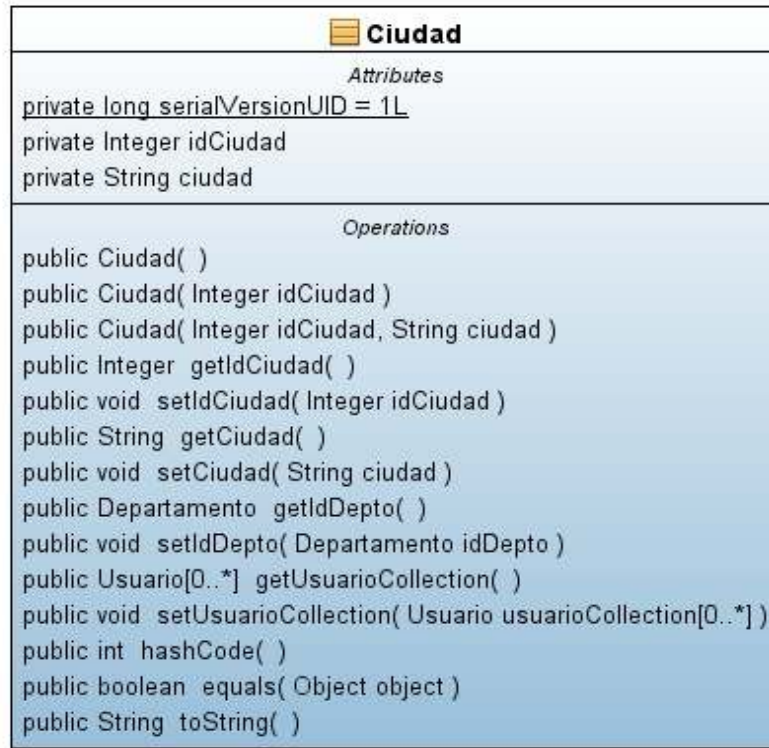
Figura 13: Clase Entity Departamento



Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla departamento, posee los siguientes atributos: idDepto llave primaria de la tabla, y el nombre del departamento.

Los métodos *get* y *set* se utilizan para obtener y cambiar la información respectivamente de los atributos de la clase.

Figura 14: Clase Entity Ciudad



Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla ciudad, posee los siguientes atributos: idCiudad llave primaria de la tabla, y el nombre la ciudad.

Los métodos get y set se utilizan para obtener y cambiar la información respectivamente de los atributos de la clase.

Figura 15: Clase Entity Factura



Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla factura, posee los siguientes atributos: idFactura llave primaria de la tabla, total, tipo y las fechas de Ingreso y de cierre.

Los métodos get y set se utilizan para obtener y cambiar la información respectivamente de los atributos de la clase.

Figura 16: Clase Entity Producto.



Esta clase se utiliza para dar una visión y acceso orientado a objetos de la tabla producto, posee los siguientes atributos: idProducto llave primaria de la tabla, clave, costo, valor, fecha de ingreso y la cantidad.

Los métodos get y set se utilizan para obtener y cambiar la información respectivamente de los atributos de la clase.

Figura 17: Clase Entity Usuario



Esta clase (ver figura 17) se utiliza para dar una visión y acceso orientado a objetos de la tabla Usuario, posee los siguientes atributos: idUsuario llave primaria de la tabla, cedula,

nombre, apellido, celular, dirección, teléfono, correo, usuario, contraseña y tipo usuario. Los métodos get y set se utilizan para obtener y cambiar la información respectivamente de los atributos de la clase.

Figura 18: Clase Entity Venta



La clase especificada en la figura 18 se utiliza para dar una visión y acceso orientado a objetos de la tabla venta, posee los siguientes atributos: idVenta llave primaria de la tabla, estado, valor y cantidad.

Los métodos get y set se utilizan para obtener y cambiar la información respectivamente de los atributos de la clase.

Diagrama de Clases para Session Beans

Los bean de sesión implementan la lógica del negocio, ofreciendo un conjunto de métodos para la interacción entre el cliente y la aplicación. Para la realización de cada bean de sesión se implementaron interfaces remotas y locales y se desarrollan los métodos para crear, editar, encontrar y eliminar un objeto Entity.

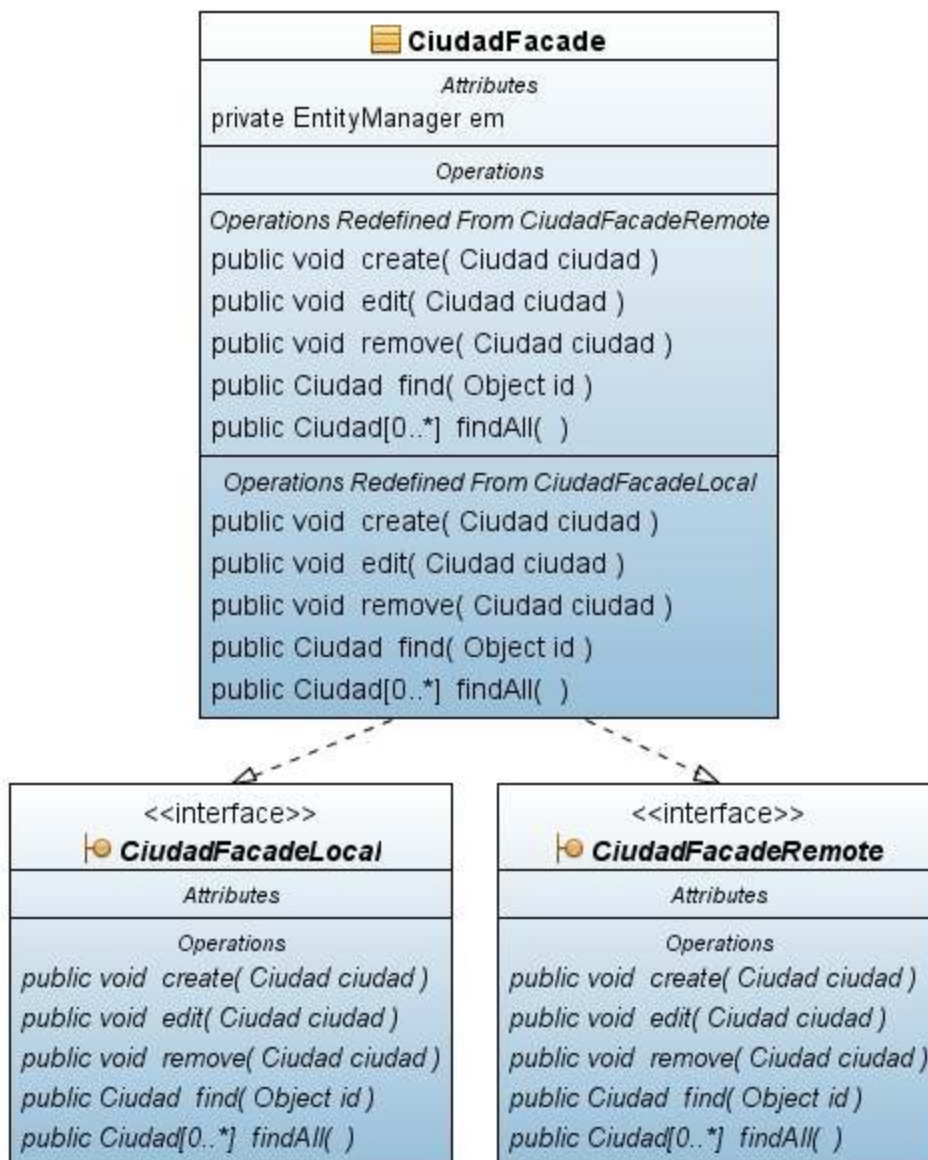
Figura 19: Diagrama de clase *Session bean* AbonoFacade



El bean de sesión AbonoFacade (ver figura 19) posee métodos que reciben como parámetro un entity Abono para crear, editar o eliminar un registro de la tabla Abono de la base de datos del sistema.

La clase AbonoFacade implementa a las interfaces AbonoFacadeRemote y AbonoFacadeLocal.

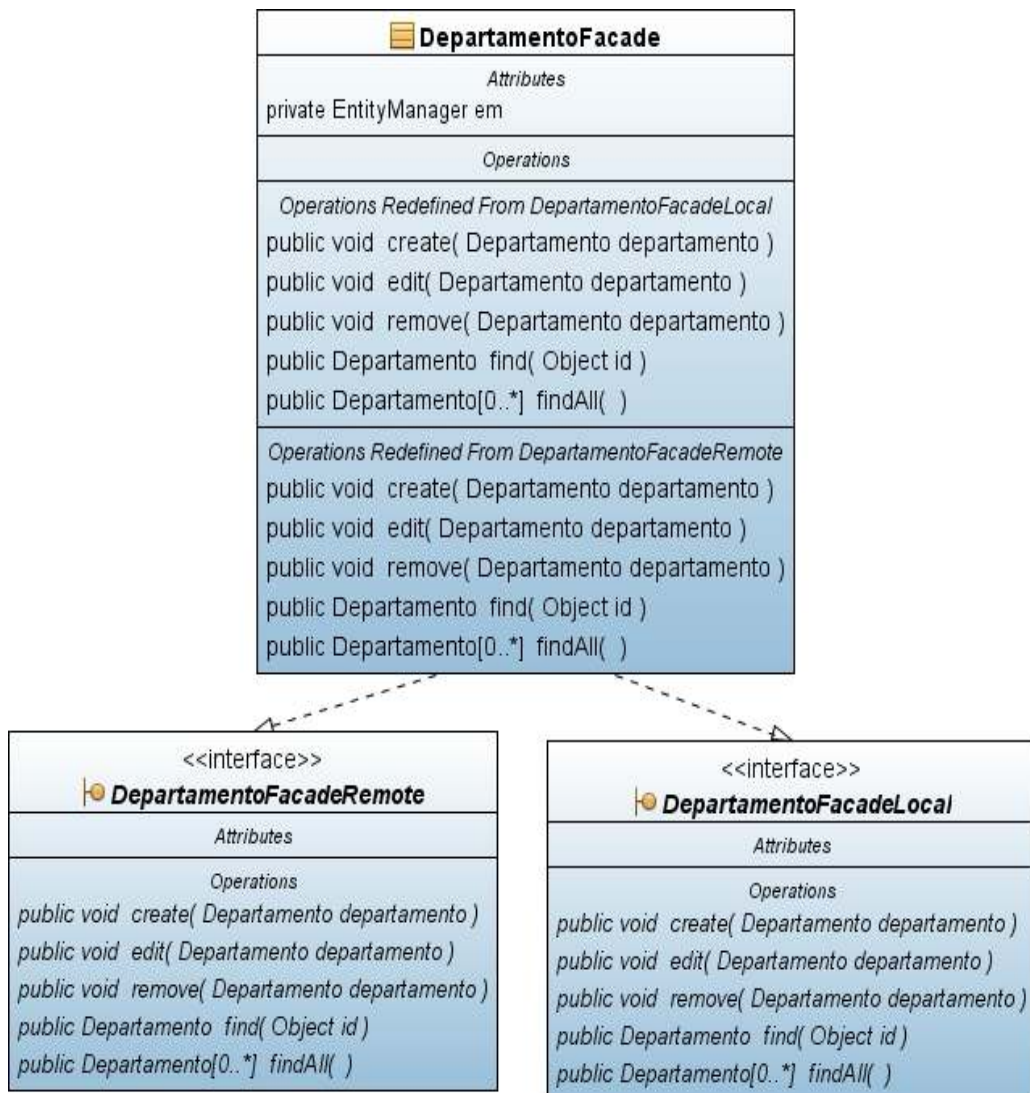
Figura 20: Diagrama de clase *Session bean* CiudadFacade



La clase CiudadFacade (ver figura 20) posee métodos que reciben como parámetro un entity ciudad para crear, editar o eliminar un registro de la tabla Ciudad de la base de datos del sistema.

La clase CiudadFacade implementa a las interfaces CiudadFacadeRemote y CiudadFacadeLocal.

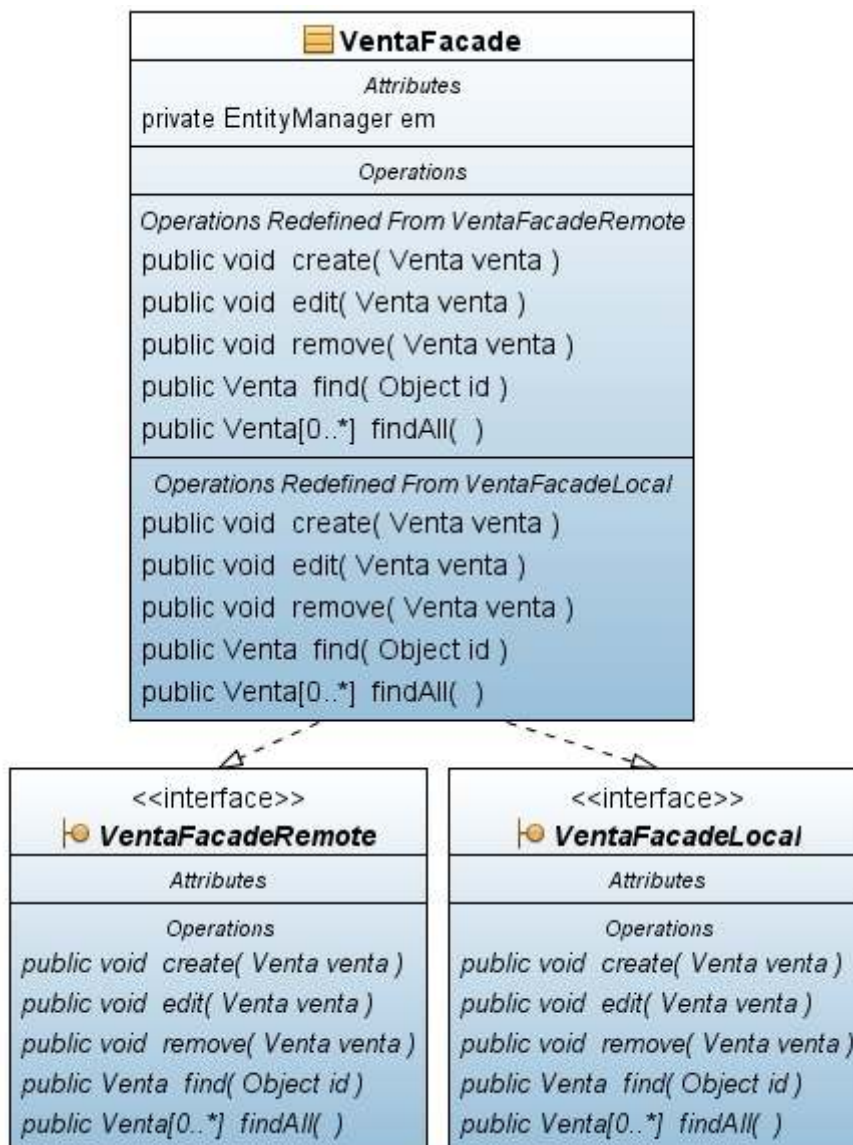
Figura 21: Diagrama de clase Session bean DepartamentoFacade



La clase DepartamentoFacade (ver figura 21) posee métodos que reciben como parámetro un entity Departamento para crear, editar o eliminar un registro de la tabla departamento de la base de datos del sistema.

La clase DepartamentoFacade implementa a las interfaces DepartamentoFacadeRemote y DepartamentoFacadeLocal.

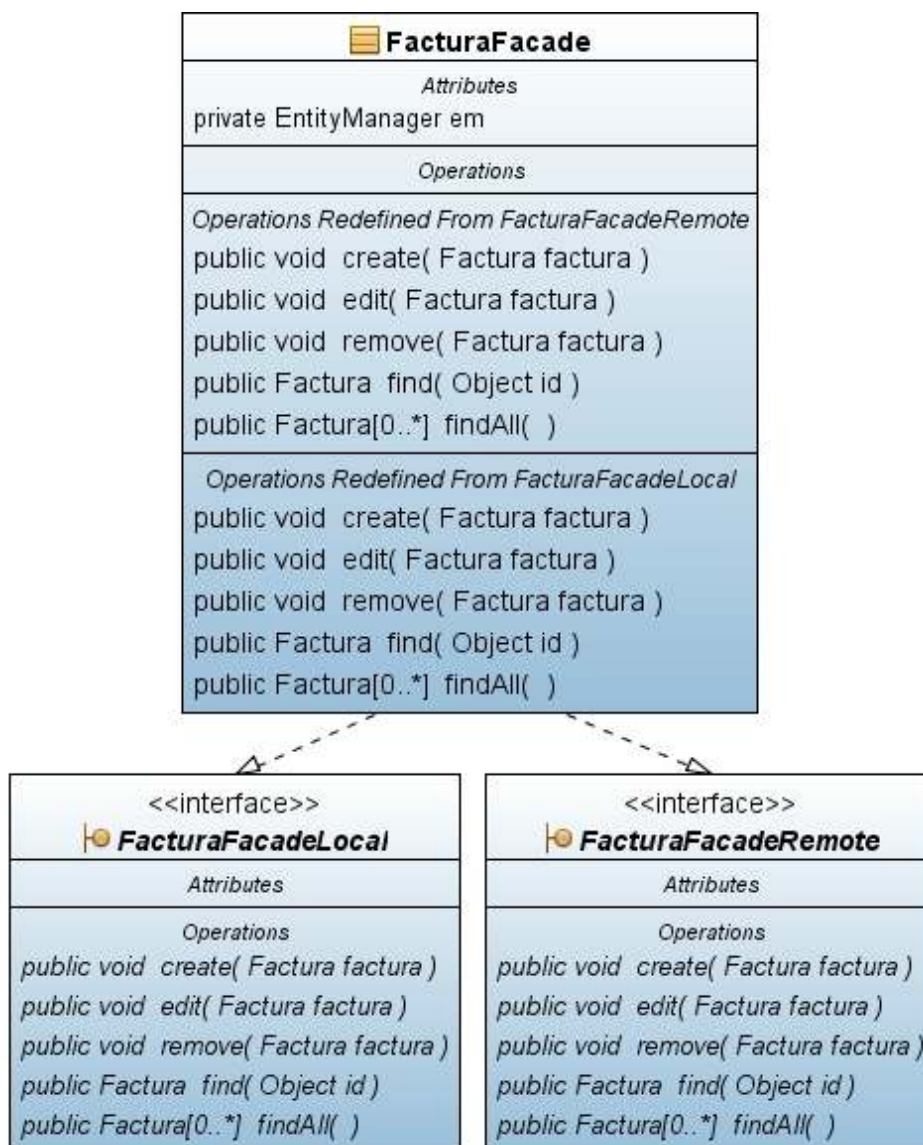
Figura 22: Diagrama de clase Session bean VentaFacade



La clase VentaFacade (ver figura 22) posee métodos que reciben como parámetro un *entity* venta para crear, editar o eliminar un registro de la tabla venta de la base de datos del sistema.

La clase VentaFacade implementa a las interfaces VentaFacadeRemote y VentaFacadeLocal.

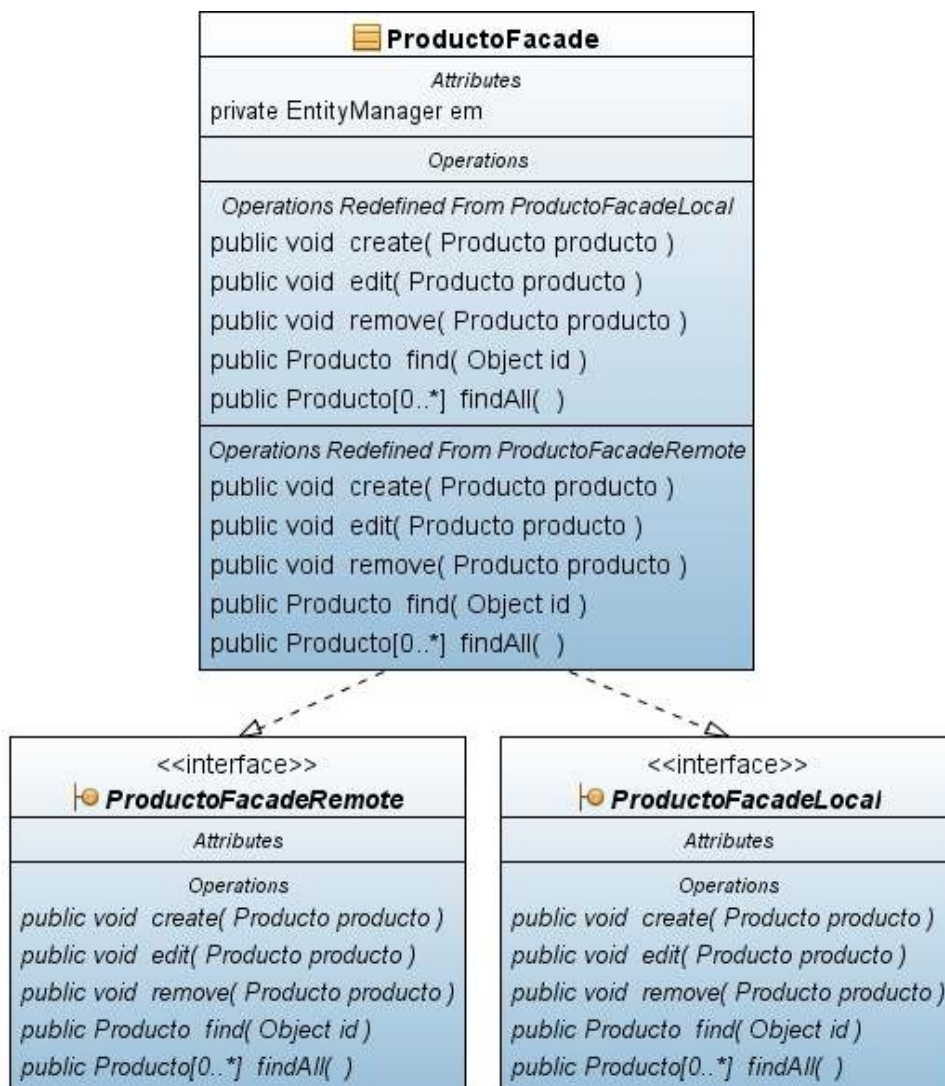
Figura 23: Diagrama de clase *Session bean* FacturaFacade



La clase FacturaFacade (ver figura 23) posee métodos que reciben como parámetro un *entity* Factura para crear, editar o eliminar un registro de la tabla factura de la base de datos del sistema.

La clase FacturaFacade implementa a las interfaces FacturaFacadeRemote y FacturaFacadeLocal.

Figura 24: Diagrama de clase *Session bean* ProductoFacade

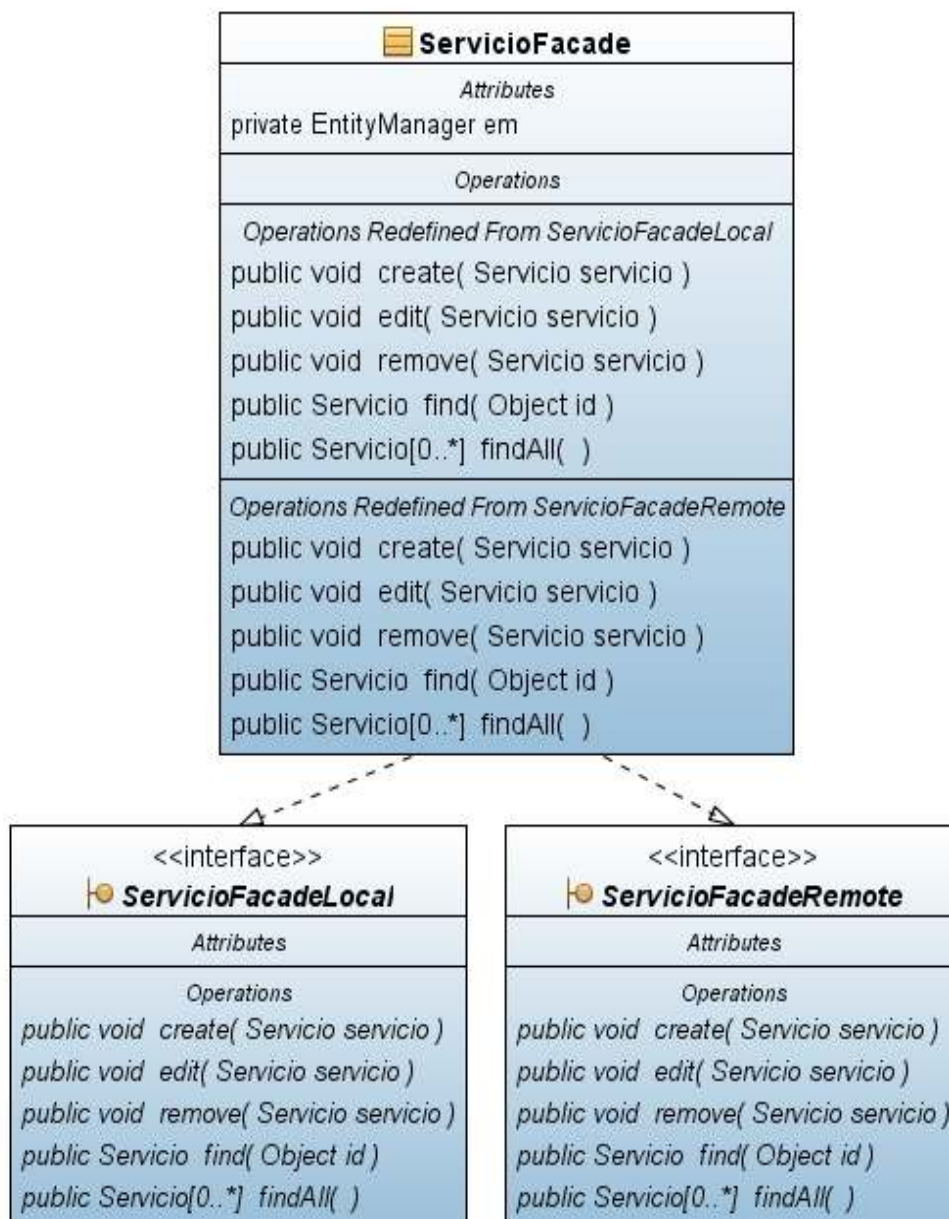


La clase ProductoFacade (ver figura 24) posee métodos que reciben como parámetro un

entity Producto para crear, editar o eliminar un registro de la tabla producto de la base de datos del sistema.

La clase ProductoFacade implementa a las interfaces ProductoFacadeRemote y ProductoFacadeLocal.

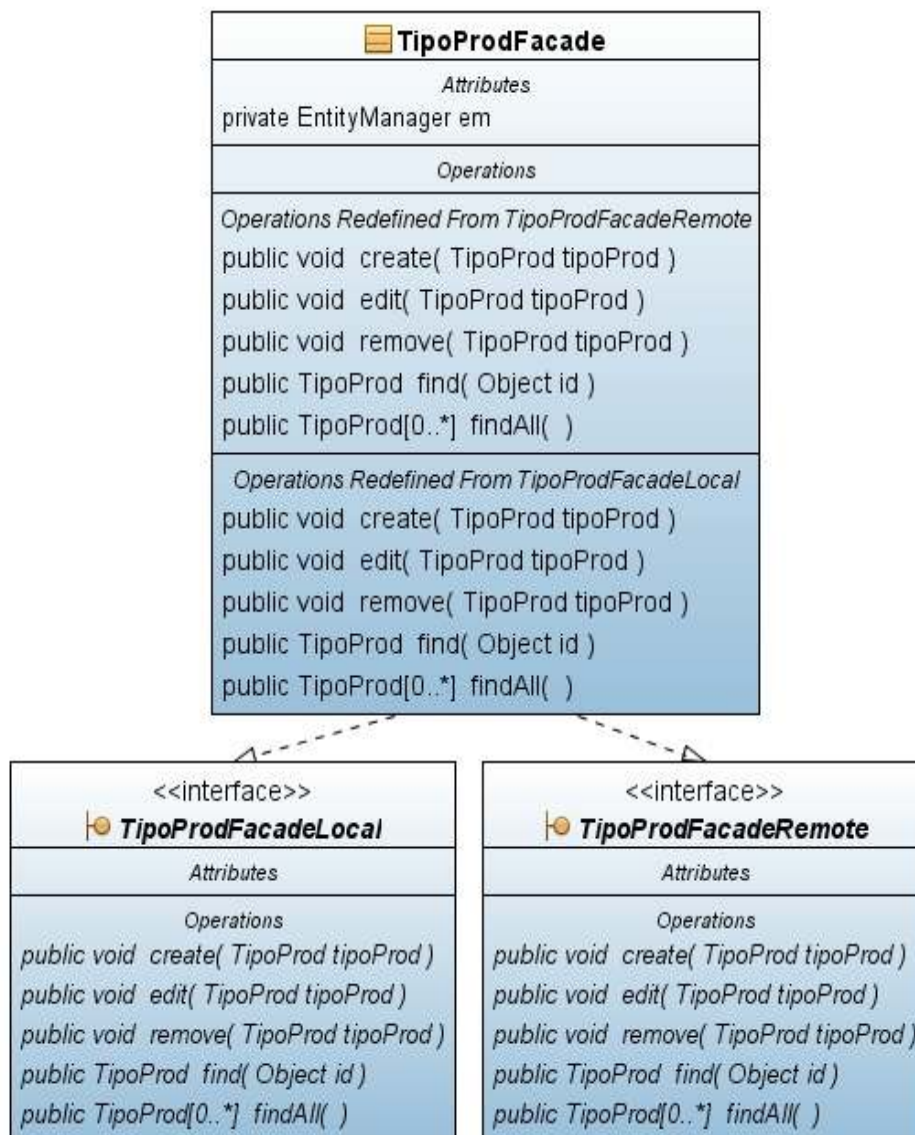
Figura 25: Diagrama de clase Session bean ServicioFacade



La clase ServicioFacade (ver figura 25) posee métodos que reciben como parámetro un entity Servicio para crear, editar o eliminar un registro de la tabla servicio de la base de datos del sistema.

La clase ServicioFacade implementa a las interfaces ServicioFacadeRemote y ServicioFacadeLocal.

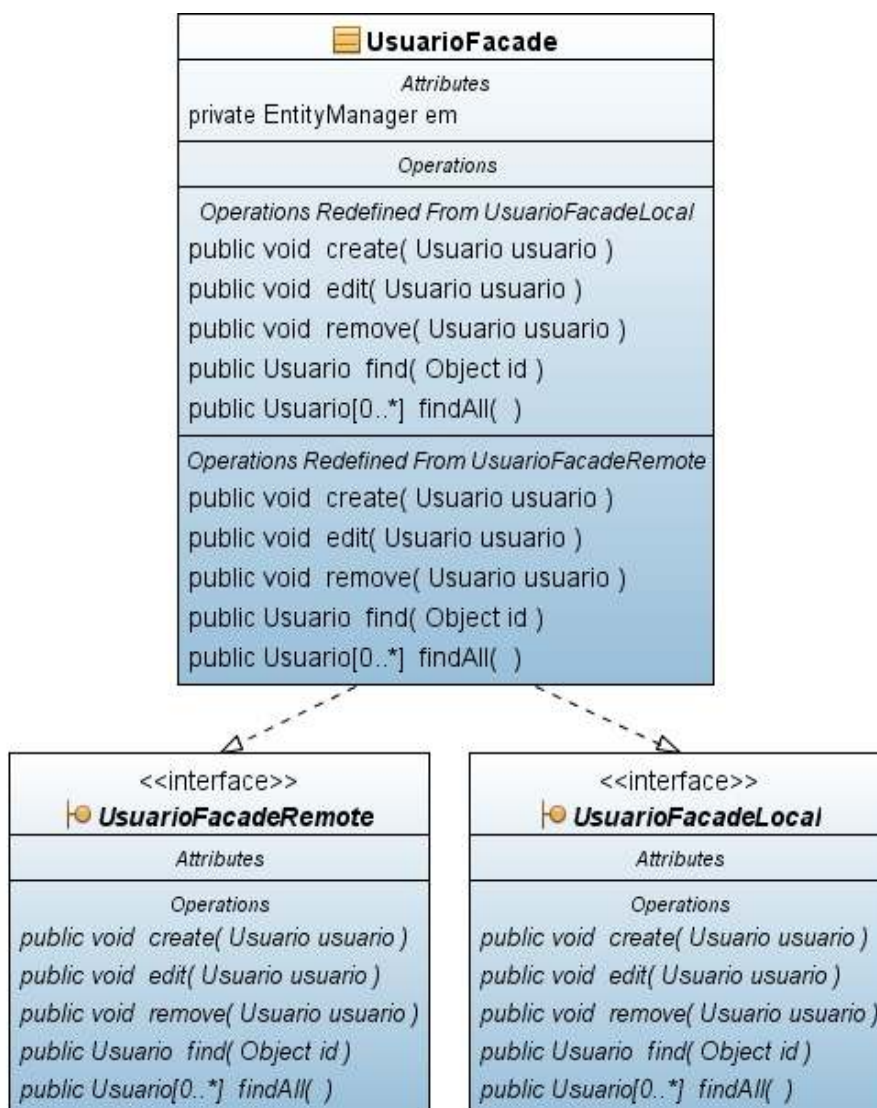
Figura 26: Diagrama de clase Session bean TipoProdFacade



La clase TipoProdFacade (ver figura 26) posee métodos que reciben como parámetro un entity TipoProd para crear, editar o eliminar un registro de la tabla TipoProd de la base de datos del sistema.

La clase TipoProFacade implementa a las interfaces TipoProFacadeRemote y TipoProFacadeLocal.

Figura 27: Diagrama de clase Session bean UsuarioFacade



La clase UsuarioFacade (ver figura 27) posee métodos que reciben como parámetro un entity Usuario para crear, editar o eliminar un registro de la tabla usuario de la base de datos del sistema.

La clase UsuarioFacade implementa a las interfaces UsuarioFacadeRemote y UsuarioFacadeLocal.

- **Modelo relacional**

Presentado en la figura 28 fue diseñado de acuerdo a los requerimientos de los usuarios y normalizado hasta la tercera forma normal,

En dicho diagrama existen relaciones entre cada una de las tablas, por ejemplo un usuario pertenece a una ciudad pero a una misma ciudad pueden pertenecer muchos usuarios.

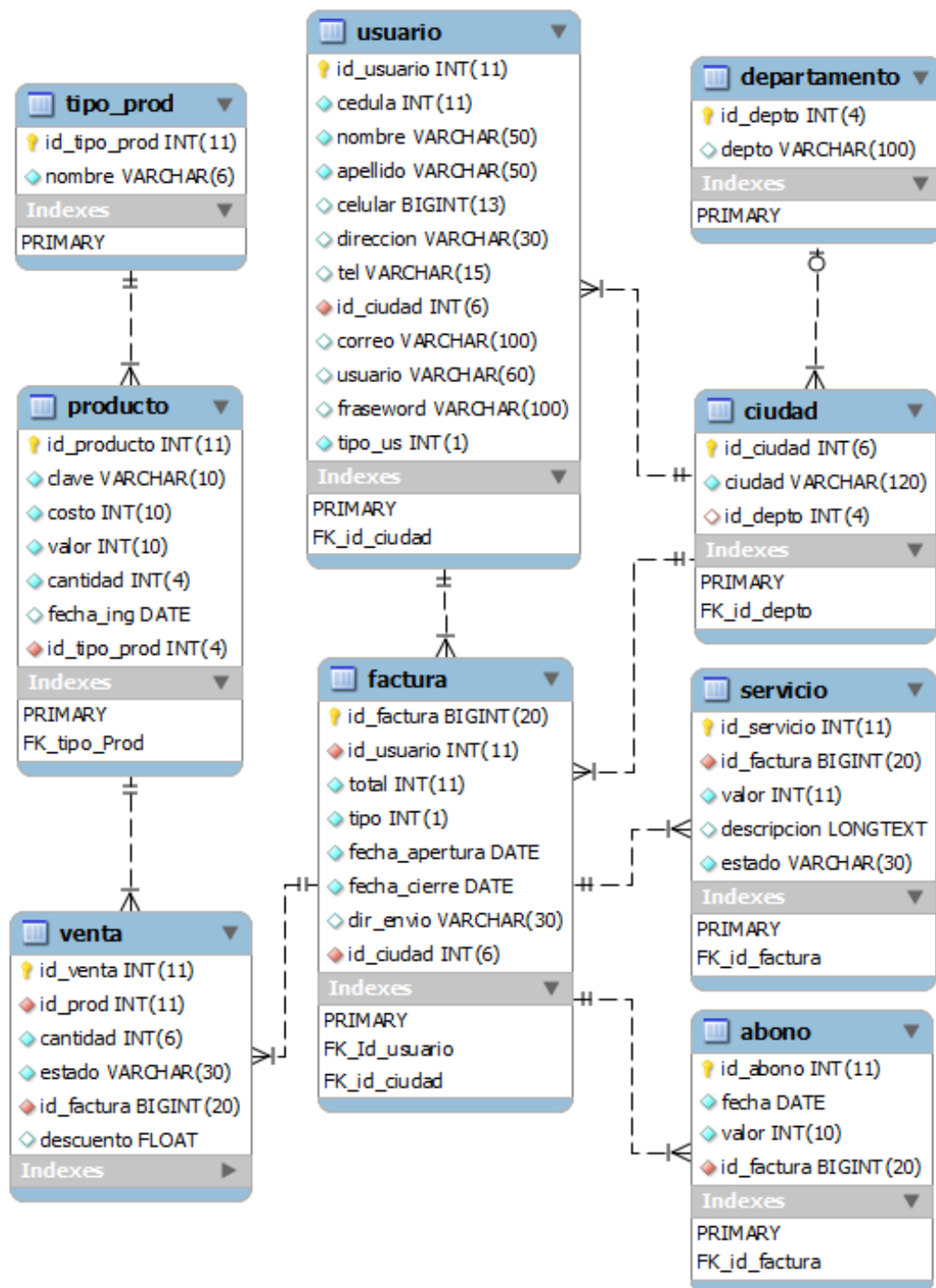
Un usuario por su parte puede estar en muchas facturas pero una factura solo pertenece solo a un usuario.

Un producto solo puede pertenecer a un tipo de producto, pero un tipo de producto puede ser de varios productos.

Por otro lado un departamento posee muchas ciudades y municipios pero una ciudad solo puede pertenecer a un departamento.

Para finalizar una factura puede tener muchos abonos y muchos servicios.

Figura 28: Diagrama entidad relación.



- **Diccionario de datos.** Presentado en las siguientes tablas describe cada una de las tablas de la base de datos para el sistema. Este muestra un pequeño resumen de la tabla, los atributos, el tipo de datos y la longitud de ellos.

Tabla 43: Tabla ciudad

TABLA: CIUDAD				
Descripción: en esta tabla se guarda toda la información de las ciudades.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_ciudad	*	NOT NULL	INT	4
Ciudad		NOT NULL	VARCHAR	40
id_depto		NOT NULL	INT	4

Descripción de los atributos:

id_ciudad: es el identificador único de una ciudad.

ciudad: es el nombre de la ciudad

id_depto: es la llave foránea que hace referencia al campo id_depto de la tabla departamento.

Tabla 44: Tabla departamento

TABLA: DEPARTAMENTO				
Descripción En esta tabla se guarda toda la información relacionada con los departamentos				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_depto	*	NOT NULL	INT	4
Departamento		NOT NULL	VARCHAR	40

Descripción de los atributos:

id_depto: es el identificador único de un departamento.

departamento : es el nombre de la departamento.

Tabla 45: Tabla usuario

TABLA: USUARIO				
Descripción: en esta tabla se guarda toda la información relacionada de los usuarios.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_usuario	*	NOT NULL	INT	11
Cedula		NOT NULL	INT	10
Nombre		NOT NULL	INT	50
Apellido		NOTNULL	INT	50
Cellular		NULL	BIGINT	10
direccion		NOT NULL	DATE	20
Tel		NULL	INT	15
id_ciudad		NOT NULL	INT	6
Correo		NULL	VARCHAR	100
Usuario		NOT NULL	VARCHAR	20
Fraseword		NOT NULL	VARCHAR	30
tipo_us		NOT NULL	INT	1
tipo_identi		NOT NULL	VARCHAR	30

Descripción de los atributos:

id_usuario: es el identificador único de un usuario

cedula: es el documento de identidad de un usuario.

nombre: es el nombre de un usuario.

apellido: es el apellido de un usuario.

celular: es el número del celular de un usuario

direccion: es la dirección que tiene un usuario.

tel: es el teléfono que posee un usuario.

id_ciudad: es la llave foránea que hace referencia al campo id_ciudad de la tabla CIUDAD.

correo: es el correo electrónico que posee un usuario.

usuario: es el login de un usuario.

fraseword: es la contraseña que posee un usuario.

tipo_us: es el tipo de usuario (Administrador, cliente, vendedor).

tipo_identi: es el tipo de documento de identidad de un usuario (cedula, cedula de extranjería, tarjeta de identidad).

Tabla 46: Tabla factura

TABLA: FACTURA				
Descripción: en esta tabla se guarda toda la información de una factura producto de las ventas.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_factura	*	NOT NULL	INT	11
id_usuario		NOT NULL	INT	10
Total		NOTNULL	INT	11
Tipo		NOT NULL	INT	11
fecha_apertura		NOT NULL	DATE	
fecha_cierre		NOT NULL	DATE	
dir_envio		NOT NULL	VARCHAR	30
Id_ciudad		NOT NULL	INT	6

Descripción de los atributos:

id_factura: es el identificador único de una factura.

id_usuario: es la llave foránea que hace referencia al campo id_cliente de la tabla usuario.

tipo: es el concepto por el cual se realiza una factura(Venta, servicio, mixto).

fecha_apertura: es la fecha en que se realiza una factura para adquirir un producto o servicio.

fecha_cierre: es la fecha en que se cancela el monto total de la deuda.

total: es el valor de una factura.

dir_envio: es la dirección donde se enviarán los artículos comprados en la factura.

id_ciudad: es la llave foránea que hace referencia al campo id_ciudad de la tabla ciudad.

Tabla 47: Tabla servicio

TABLA: SERVICIO				
Descripción: en esta tabla se guarda la información relacionada con un servicio prestado a un cliente.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_servicio	*	NOT NULL	INT	6
id_factura		NOT NULL	DATE	
Valor		NOT NULL	INT	8
descripcion		NULL	INT	11
estado		NOT NULL	VARCHAR	40

Descripción de los atributos:

id_servicio: es el identificador único de un servicio.

id_factura: es la llave foránea que hace referencia al campo id_factura de la tabla factura.

valor: es el valor de un servicio.

descripcion: es una definición del servicio que se está prestando.

estado: es el estado en que esta el servicio (arreglado, entregado, sin arreglado, en proceso, y devuelto, en garantía).

Tabla 48: Tabla abono

TABLA: ABONO				
Descripción: en esta tabla se guarda toda la información relacionada con el abono a una deuda de un separado o una fianza.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_abono	*	NOT NULL	INT	6
Fecha		NOT NULL	DATE	
Valor		NOT NULL	INT	8
id_factura		NULL	INT	11

Descripción de los atributos:

id_abono: es el identificador único de un abono.

fecha: es la fecha en que se realiza el abono

valor: es el valor que se abona.

id_factura: es la llave foránea que hace referencia al campo id_factura de la tabla factura.

Tabla 49: Tabla Tipo_prod

TABLA: TIPO_PROD				
Descripción: en esta tabla se guarda toda la información relacionada con los tipos de productos.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_tipo_prod	*	NOT NULL	INT	11
Nombre		NOT NULL	VARCHAR	20

Descripción de los atributos:

id_tipo_prod: es el identificador único de un tipo de artículo.

nombre: es el nombre del tipo de artículo.

Tabla 50: Tabla producto

TABLA: Producto				
Descripción: en esta tabla se guarda toda la información relacionada con los productos.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_producto	*	NOT NULL	INT	11
Clave		NOT NULL	VARCHAR	10
Costo		NOT NULL	INT	10
Valor		NOTNULL	INT	10
Cantidad		NOT NULL	INT	4
fecha_ing		NOT NULL	DATE	
Id_tipo_prod		NOT NULL	INT	4

Descripción de los atributos

id_producto: es el identificador único de un artículo.

clave: es el precio de costo de un artículo

valor: es el valor unitario de un artículo.

costo: es el valor de compra de un artículo.

cantidad: es la cantidad de productos almacenados

fecha_ing: es la fecha de ingreso de un producto a stock.

id_tipo_prod: es la llave foránea que hace referencia al campo id_tipo_prod de la tabla tipo_prod.

Tabla 51: Tabla ventas

TABLA: VENTAS				
Descripción: en esta tabla se guarda una venta.				
<i>Atributos</i>	<i>Llave Primaria</i>	<i>Opcional</i>	<i>Tipo de Dato</i>	<i>Longitud</i>
id_venta	*	NOT NULL	BIGINT	20
id_factura		NOT NULL	INT	10
Estado		NOT NULL	VARCHAR	40
Valor		NOT NULL	INT	10
Descuento		NOT NULL	INT	10
id_producto		NOT NULL	INT	10
Cantidad		NOT NULL	INT	10

id_venta: es el identificador único de una venta.

valor: es el valor de la venta.

estado: define el estado en el que se encuentra un artículo en una venta (Cambiado, separado, entregado).

id_factura: es la llave foránea que hace referencia al campo id_factura de la tabla factura.

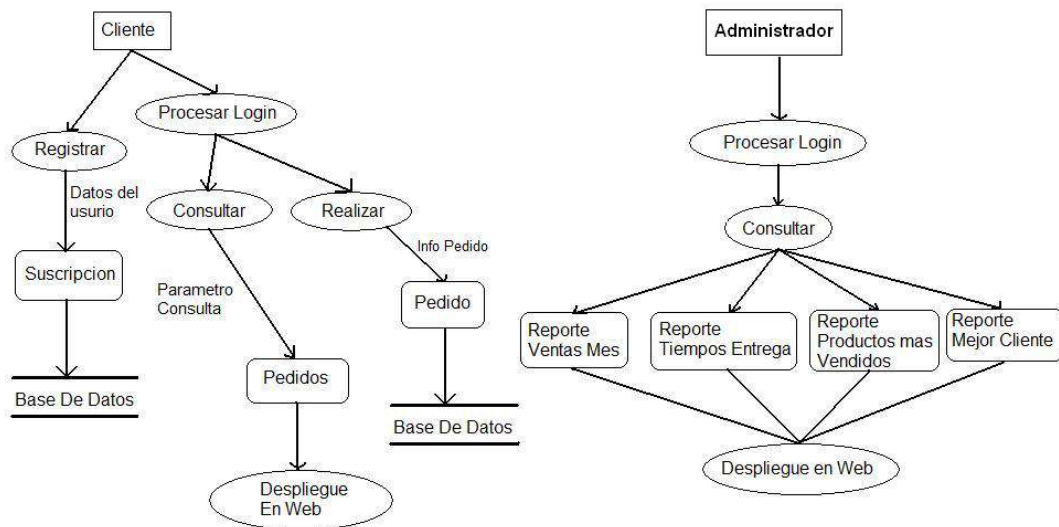
descuento: descuento que se hace sobre un artículo específico

id_producto: es la llave foránea que hace referencia al campo id_producto de la tabla producto.

cantidad: es la cantidad de productos que se van a comprar.

- **Modelos de Flujo de Información.** A través de estos diagramas se identifica el flujo de los datos en cada uno de los procesos principales de la aplicación.

Figura 29: Modelos de flujo de información.

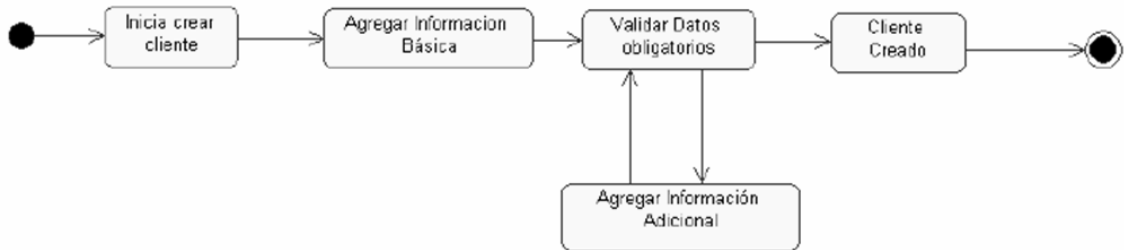


La figura 29 muestra los flujos de datos para dos actores del sistema. En el caso de ser un cliente este se debe registrar en la base de datos y para consultar o realizar un pedido desplegado en web debe pasar por el proceso de login.

En el caso del administrador, este debe validarse ante el sistema para acceder al módulo de consultas.

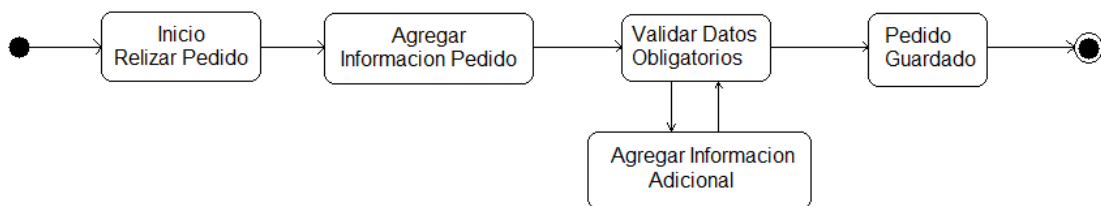
- **Modelos de Ciclo de Vida de Información.** A través de estos diagramas se analizó la vida (donde nacen, si son borrados o almacenados) de los datos a través de los procesos de la aplicación.

Figura 30: Modelo de ciclo de vida de información, Registro cliente



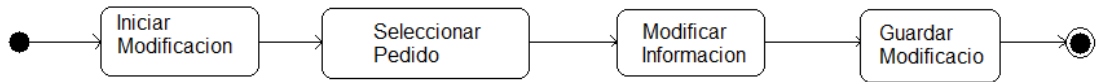
Ciclo de vida de los datos del cliente. Los datos del cliente tienen dos flujos de información básicos: creación y modificación. La creación implica agregar los datos requeridos para crear al cliente con la información básica, validar estos datos, revisar si es necesario ingresar más información y crear el cliente. La modificación implica seleccionar el cliente que va a ser modificado, modificar los datos requeridos y actualizar la nueva información.

Figura 31: Modelo de ciclo de vida de información, Realizar Pedido



Ciclo de vida de los datos del pedido. Los datos del pedido tienen dos flujos de información básicos: creación y modificación. La creación implica seleccionar el cliente quien realiza el pedido, registrar los datos básicos del pedido (producto, fecha, usuario), y luego finalizar su registro. La modificación implica seleccionar el pedido que va a ser modificado, modificar los datos requeridos y actualizar la nueva información.

Figura 32: Modelo de ciclo de vida de información, modificar Pedido

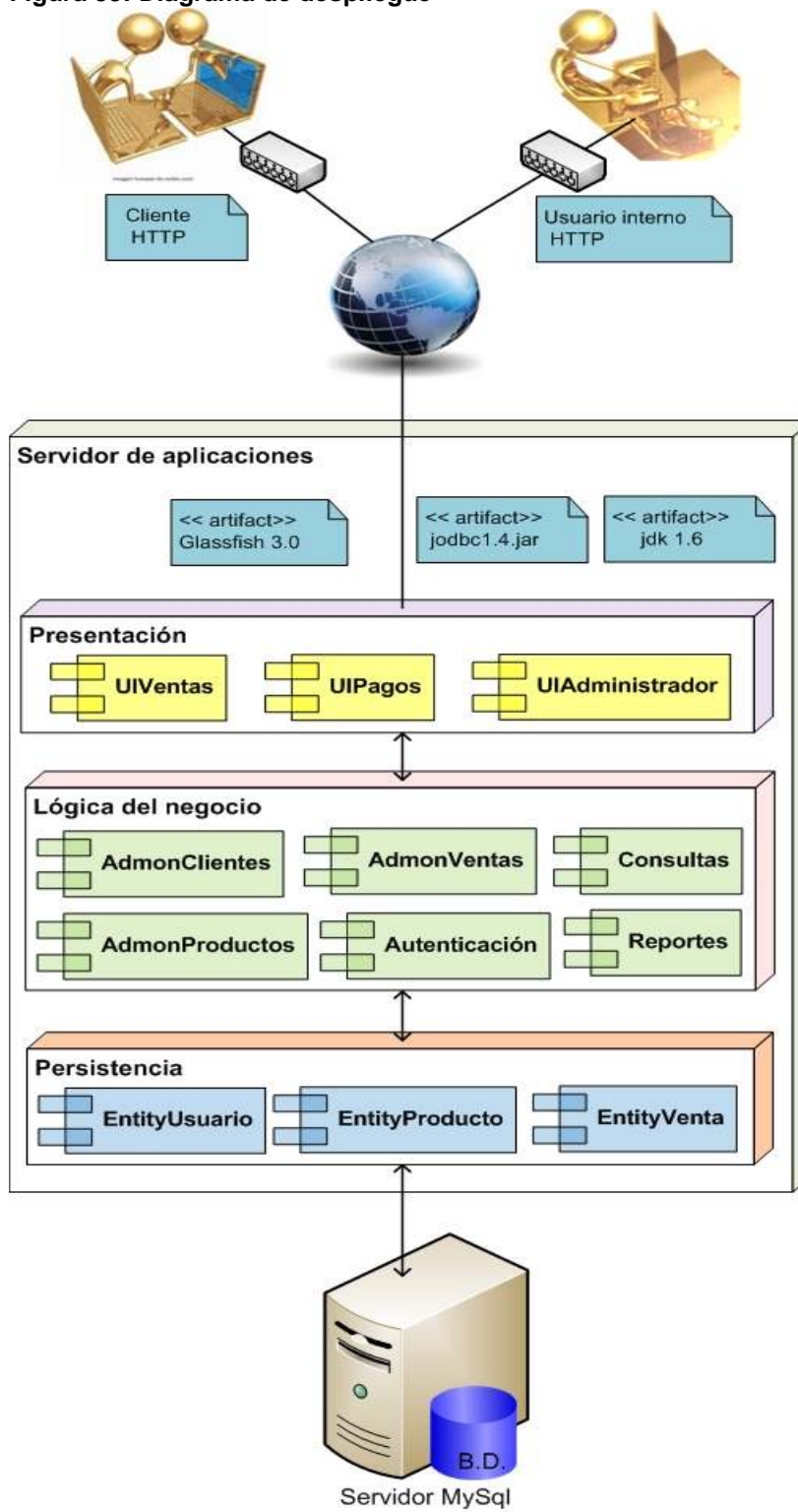


5.3.3 Punto de Vista de Despliegue. Describe el ambiente en el cual el sistema será desplegado incluyendo dependencias del sistema en su ambiente de ejecución, teniendo en cuenta los modelos de plataforma de ejecución, red y dependencias tecnológicas, en el que se define un modelo completo de despliegue que incluye infraestructura, distribución de componentes y requisitos tecnológicos.

- **El Diagrama de despliegue.** Presentado en la figura 33, fue diseñado con los requerimientos y los atributos de calidad del sistema e incluye las especificaciones de las capas de la arquitectura implementada y la distribución de los componentes.

La capa de presentación despliega la información a los usuarios aceptando a su vez peticiones y respuestas de los mismos a través de Interfaces de Usuario o JSP (*Java Server Pages*). La capa de Lógica del negocio gestiona todas las actividades del sistema. Mientras que la capa de Persistencia manipula la información del negocio para su consulta y almacenamiento, en esta capa se encuentran los componentes *Entity* creados a partir de cada tabla de la Base de datos.

Figura 33: Diagrama de despliegue



- **Relaciones entre los Puntos de Vista**

Esta sección describe las relaciones generales entre los puntos de vista escogidos para representar la arquitectura. Adicionalmente en esta sección, se discute la consistencia entre dichas vistas.

En las secciones 5.2.5, 5.3.1, 5.3.2 y 5.3.3 de este documento se desarrollaron varios puntos de vista para dar a conocer la arquitectura candidata planteada. La interrelación entre las vistas es fundamental para poder describir un sistema complejo ilustrando sus características funcionales y propiedades de calidad. Cada una de estas, orientada a un *stakeholder* o grupo de *stakeholders* con *concerns*.

A lo largo de este documento se han desarrollado los puntos de vista de despliegue, funcional y de información.

En el desarrollo del punto de vista de despliegue se incorporaron un diagrama que cumple las características fundamentales del punto de vista. Si se observa, en el diagrama (ver figura 4) se hizo una alta integración de los componentes funcionales de la arquitectura. Cada componente funcional en este diagrama hace relación a los componentes mostrados en el punto de vista funcional de la arquitectura.

Por otro lado en el punto de vista funcional es importante resaltar que los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema y el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema (características funcionales). En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema.

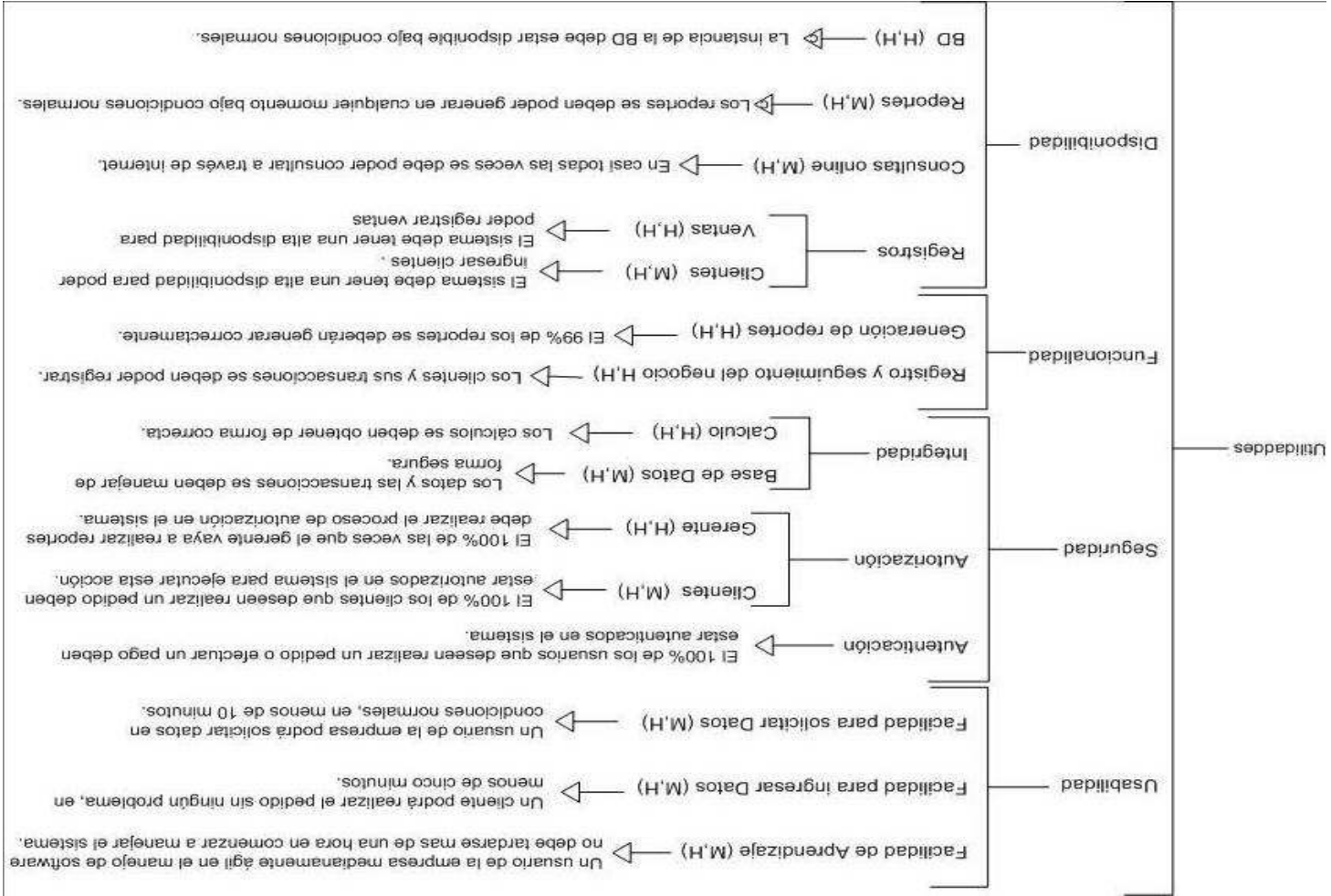
Pero aún más importante es que un caso de uso describe (desde el punto de vista del actor) un grupo de actividades de un sistema que produce un resultado concreto y tangible. Es por ello que se puede observar que los diagramas de casos de uso y los de actividades tienen una estrecha relación (por ejemplo el caso de uso de crear un cliente y su respectivo diagrama de actividades de crear un cliente).

5.3.4 Evaluación de la Arquitectura. Para realizar la evaluación de la arquitectura se comenzó por identificar diferentes escenarios que fueron propuestos en la sección 5.2.6 (atributos de calidad), en estos se especifica el atributo de calidad implicado, el punto de sensibilidad y los riesgos que pueden presentar, para luego llevarlos a un árbol de utilidad (ver figura 34), desde el cual se procederá a realizar a hacer una evaluación de la arquitectura con la evaluación ATAM descrita en la sección 5.3.5 .

- **Árbol de Utilidad**

A través del análisis de este árbol se visualiza la jerarquía de los atributos de calidad con la identificación de los escenarios de evaluación, luego de realizar este análisis se pueden determinar los puntos más sensibles de la arquitectura propuesta.

Figura 34: Árbol de utilidades



5.3.5 Evaluación ATAM (*Architecture Tradeoff Analysis Method*) Este método de evaluación permite hacer un análisis de resultados previo de la arquitectura para visualizar qué tanto una arquitectura satisface los atributos de calidad requeridos para el sistema; también provee ideas de cómo esas metas de calidad interactúan entre ellas y cómo realizar intercambios de prioridad mutuos (*tradeoff*) entre ellas.

Este método de evaluación de la arquitectura comienza identificando los diferentes *stakeholders* (Actores) y sus respectivos requerimientos. Se identifican los atributos de calidad y se elabora un árbol de utilidades en donde se predicen y ubican los diferentes escenarios de calidad que facilitan la identificación de los puntos más sensibles de la Arquitectura (o *tradeoff*) del sistema.

En esta sección se analizarán dos escenarios importantes para identificar los dos puntos más sensibles de la arquitectura.

Tabla 52: Evaluación ATAM del atributo sobre disponibilidad

Escenario	Ingreso de una ventas al sistema por parte del administrador.
Objetivo del Negocio	Permitir el ingreso de las ventas realizadas a la base de datos.
Atributo de Calidad	Disponibilidad
Aproximación Arquitectural	El sistema deberá permitir ingresar las ventas sin ninguna anomalía.
Riesgos	El 99 % de las veces las ventas deben ser ingresadas correctamente por parte del administrador, algún error en el ingreso podrá afectar otras operaciones.
Tradeoffs	Asegurar una alta disponibilidad en el sistema para este caso, afectará la seguridad del sistema ya que este tratará de resistir usos no autorizados mientras se sigue proveyendo servicios a usuarios legítimos.

La tabla 53, muestra que para realizar un reporte el administrador debe estar autorizado de allí se describen los riesgos y cual atributo será afectado si se desea tener alta seguridad de la información.

Tabla 53: Evaluación ATAM del atributo sobre seguridad

Resumen Escenario	Todas las veces que el administrador vaya a realizar un reporte debe estar autorizado.
Objetivo del Negocio	Permitir el ingreso del administrador al sistema para realizar reportes.
Atributo de Calidad	Seguridad
Aproximación Arquitectural	El sistema no deberá permitir ingresar a un usuario sin validarlo primero.
Riesgos	Inseguridad a la hora de chequear claves o validar contraseñas. Que el sistema permita generar reportes a algún usuario no autorizado.
Tradeoffs	Tener una alta seguridad en la generación de reportes afectará la usabilidad del sistema debido a las reglas de seguridad implementadas para la autenticación de los usuarios.

5.4 FASE DE IMPLEMENTACIÓN

En esta fase se prepara el entorno para garantizar el buen desarrollo del proyecto y la disponibilidad de todos los medios requeridos; entre ellos: el servidor de aplicaciones, el gestor de bases de datos y las herramientas de generación de código.

Para este proyecto se utilizó el servidor web *Glassfish* versión 3 y la herramienta *Netbeans* 6.9 como ambiente de desarrollo integrado para la creación del sistema y las páginas web, por ser un IDE de distribución gratuita, código abierto y con características para el programación como por ejemplo soporte para compilación de código fuente y corrección de errores.

Como servidor de aplicaciones se decidió por *Glassfish* V3 por ser un servidor de aplicaciones de código abierto, licencia gratis, soporte de funciones avanzadas. Además es compatible con Java EE, proporcionando una pequeña base con todas las funciones para la implementación de Java EE 6.

Para acceder a los módulos web del sistema se utilizó *Internet Explorer* 8.0, el navegador predeterminado de *Windows*, y *Mozilla* el cual es un navegador de código abierto con versiones para múltiples plataformas.

Las interfaces (figura 35) se hicieron con *JSP* utilizando la tecnología *JSF* definida anteriormente.

Figura 35: Interfaz gráfica



The screenshot shows a web browser window with the URL `http://localhost:8080/Sijec_V11/faces/Pages/producto/List.jsp`. The browser tabs include 'Catalogo de Relojes', 'Página principal - W...', 'Editing TipoProd', 'New TipoProd', and 'Error - The class 'En...'. The website header features the logo 'Plateria Joyeria Alvarez' and a navigation menu with links: 'Inicio', 'Quienes Somos', 'Registro de Clientes', 'Catálogo de Productos', 'Pagos', and 'Administración'. The main content area is titled 'Lista de Productos' and includes a small image of two figures with a laptop. Below the title, it indicates 'Producto 1..10 de 12 Remaining 2'. The table below lists the following products:

Nombre	Costo	Valor	Cantidad	Tipo	Ver	Editar
Anillo Diamantado	100	500	20	1	Ver	Editar
Anillo Rubis	200000	400000	3	1	Ver	Editar
Anillo Circones	10000	20000	3	1	Ver	Editar
Anillo Solitario de Diamantes	10000	20000	3	1	Ver	Editar
Casio metal plata	100000	150000	1	7	Ver	Editar
Cadena mediana plata	25000	50000	12	4	Ver	Editar
Cadena gruesa	30000	60000	10	4	Ver	Editar

Para el manejo de la persistencia se utilizó MySQL como motor de bases de datos.

En esta fase también, se realizaron pruebas unitarias para asegurar la funcionalidad de cada módulo de código y pruebas de integración con el propósito de garantizar que todo el sistema cumple con los requerimientos.

5.5 FASE DE TRANSICIÓN

En esta fase se realizó la configuración del sistema de información web y se socializó el manual de usuario (ver anexo B).

6 ANÁLISIS Y RESULTADOS.

6.1 ANÁLISIS DE RESULTADOS PREVIOS A LA IMPLEMENTACIÓN

Los análisis de resultados para este proyecto se iniciaron de forma temprana dentro de la fase de implementación permitiendo realizar un estudio abstracto para encontrar posibles errores y fallas antes de implementar el sistema, pues hacer una evaluación de la arquitectura es la manera más económica de evitar desastres ya que estas evaluaciones no producen resultados cuantitativos.

Este análisis se presentó entre las secciones 5.2.4 a 5.2.9 durante el proceso que finaliza con la evaluación ATAM.

6.2 ANÁLISIS DE RESULTADOS DEL PRODUCTO FINAL

El uso del *framework* JSF permitió desarrollar un sistema Web que cumple con los requerimientos funcionales y no funcionales (seguridad y desempeño) con características de alta calidad en el software. Además permite la aplicación de conceptos relacionados con arquitecturas y patrones de software.

La base de datos siguió el modelo relacional normalizado hasta la tercera forma normal y se implementó con el motor de bases de datos de MySQL. Este Motor de bases de datos da un alto desempeño al sistema pues no exige mayores recursos de máquina.

En la implementación de las páginas (JSP) del sistema se tuvo inconvenientes al inicio con el manejo de las bibliotecas de JSF pues este *framework* presenta conflictos si no se tiene un riguroso cuidado con las diferentes versiones que existen. También se debe tener cuidado al tratar de mezclar los componentes ofrecidos por JSF con los ofrecidos por Icefaces componente que ofrece bibliotecas para el diseño de interfaces web y otros de su estilo, pues se mezclan componentes similares dentro de sus bibliotecas que generan errores al momento de desplegarlos en el servidor.

Como experimento se realizó la implementación del proyecto en los sistemas operativos Windows 7 y Ubuntu 10.04 obteniendo diferencia en el desempeño de la aplicación, siendo Ubuntu un sistema que logra desplegar más rápidamente las páginas aunque a la hora de instalarlo hay que entrar a manipular las carpetas de instalación por consola para asignarle permisos de súper usuario para escritura de archivos.

Como resultado de aplicar un estilo arquitectural por capas, se obtiene un bajo acoplamiento y alta cohesión entre los componentes del sistema.

El sistema de información permite registrar, consultar y modificar información en la base de datos. Según el rol de un usuario, se permiten diferentes acciones sobre el sistema. Por ejemplo si es un administrador el sistema le permite registrar, modificar y consultar la información de productos, usuarios, tipos de productos, ciudades y departamentos y generar reportes. Si es un cliente le permite registrarse, consultar y modificar su información, consultar y comprar productos. Sin embargo para que estas funcionalidades se lleven a cabo todos deben autenticarse ante el sistema.

7 CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

La arquitectura y la tecnología implementadas aumentan la reusabilidad y mantenibilidad lo que permite futuras actualizaciones y la integración de nuevas tecnologías, por ejemplo aplicaciones móviles.

La adopción de este diseño de aplicaciones empresariales, en la que no se manipulan los datos en la capa de presentación aumenta la seguridad e integridad de los mismos.

El uso de J2EE en MIPYMES incrementa las posibilidades de expansión de mercados y reduce las limitaciones de recursos humanos, físicos y económicos.

El aporte del proyecto se fundamenta en la entrega de un documento que incluye de manera detallada los diferentes conceptos de arquitectura de software entregados en cada fase de desarrollo.

Se logró implementar el carrito de compras utilizando interfaces de usuario desarrolladas con J2EE a través de las diferentes capas del sistema.

7.2 RECOMENDACIONES

Las arquitecturas basadas en modelos de componentes y/o modelos orientados a objetos como la arquitectura *N-Tier* implementada en este proyecto permiten futuras actualizaciones y acoplamientos a nuevas tecnologías.

En el desarrollo de proyectos que implementen JSF se recomienda un manejo riguroso de las bibliotecas pertenecientes a la versión utilizada.

Para el manejo de la persistencia se recomienda analizar las cantidades de datos requeridos para cada sistema, para no implementar motores de datos muy robustos sin

que estos sean necesarios pues requieren más recursos de máquina y disminuyen el desempeño del sistema.

8 BIBLIOGRAFÍA

CLEMENTS, Paul. Documenting Software Architectures: Views and BeyondsII, Addison-Wesley, 2002.

CONALLEN, J. Modeling Web Application Architectures with UML. Communications of the ACM, 42(10), October, 1999, pp. 63-70.

CONECTIVA, e-Commerce con Linux Guía práctica de comercio electrónico, Prentice Hall, 2001, ISBN: 958-699-039-7.

DEITEL, Harvey M. y DEITEL Paul J. Cómo programar en java. Quinta Edición, Prentice Hall, 2004.

GEARY, David y HORSTMANN, Cay. Core Java Server™ Faces, Second Edition. Prentice Hall. 2007

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Documentación. Presentación de tesis, trabajos de grado y otros trabajos de investigación. Sexta actualización. Bogotá: ICONTEC, 2008 41 p.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Referencias bibliográficas. Contenido forma y escritura. NTC 5613. Bogotá 2008, 38p.

PRESSMAN S, Roger. Ingeniería de Software, McGraw Hill, Sexta Edición en Español, 2008. ISBN: 84-481-3214-9.

ROZANSKI, Nick y WOODS, Eoin. Software Systems Architectures – Working with *Stakeholders* Using Viewpoints and Perspectives. Addison-Wesley. 2005.

JACOBSON, Ivar ; BOOCH, Grady y RUMBAUGH, James. El Proceso unificado de desarrollo de software, Addison-Wesley , 2000.

CIBERGRAFÍA

OMG Unified Modeling Language TM (OMG UML), Infraestructure, Version 2.2 [Libro en línea] disponible desde internet en: <<http://www.omg.org/docs/formal/09-02-04.pdf>> [con acceso el 19 marzo de 2009].

Plan Nacional de Tecnologías de la Información y las Comunicaciones [documento en línea] disponible desde internet en: <<http://www.colombiaplantic.org/docs/080409-Plan%20Nacional%20de%20TIC.pdf>> [con acceso el 3 de mayo de 2009]

Tutoriales Cupi 2 – Motivación [documento en línea] disponible desde internet en: <http://cupi2.uniandes.edu.co/tutoriales/publicados/JSF_ColegioWeb/Motivacion.htm> [con acceso el 10 de mayo de 2009]

Struts y JavaServer Faces, cara a cara [documento en línea] disponible desde internet en: < <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/109.pdf> > [con acceso el 11 de mayo de 2009]

The data URL scheme, [documento en línea] disponible desde internet en: <https://developer.mozilla.org/en/The_data_URL_scheme> [con acceso el 1 de mayo de 2009].

Arquitectura de Software [documento en línea] disponible desde internet en: <<http://materias.fi.uba.ar/7572/#contenidos>> [con acceso el 30 de noviembre de 2010]

GLOSARIO

ATRIBUTO DE CALIDAD: característica de un componente o sistema a la cual se le puede asignar una métrica, dichas características pueden ser: mantenibilidad, fiabilidad, usabilidad, seguridad, eficiencia, funcionabilidad entre otros.

BROWSER: programa utilizado para visualizar las páginas web.

CASO DE USO: en ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema.

COMPONENTE: los componente de Software son todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otro/s, un entorno funcional requerido por cualquier proceso predefinido. Son independientes entre ellos, y tienen su propia estructura e implementación.

CONCERNS: éste término se utiliza en los análisis arquitecturales de software para identificar intereses, necesidades y requerimientos de un sistema.

DESEMPEÑO: habilidad del sistema de funcionar dentro de sus parámetros aceptables de funcionamiento.

E-BUSINESS: engloba toda la actividad ejecutada a través de Internet, y también algo que debe entenderse no solamente como venta y compra, sino que permite que la empresa estreche sus relaciones con sus clientes y proveedores, comparta información, y facilite, por ejemplo, la toma de decisiones. Uno de los sectores.

FRAMEWORK: estructura software compuesta de componentes *personalizables* e intercambiables que permiten desarrollar una aplicación.

HTML: lenguaje de marcas de hipertexto.

HTTP:(Protocolo de transferencia de hipertexto) protocolo usado para acceder a la Web. Se encarga de procesar y dar respuestas a las peticiones para visualizar una página web.

JAVASERVER PAGES: tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

JAVASERVER FACES: es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

PROTOCOLO: es un conjunto de normas y/o procedimientos para la transmisión de datos que ha de ser observado por los dos extremos de un proceso comunicacional (emisor y receptor)

SERVIDOR WEB: es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

ARQUITECTURA DE SOFTWARE: son la estructura o estructuras del sistema que incluyen, las propiedades visibles de estos elementos y la relación entre ellos.

SERVLET: es una clase java que se ejecuta desde un servidor J2EE como tomcat o glassfish ofreciendo contenido dinámico desde el servidor web, generalmente HTML.

STAKEHOLDER: son actores o entidades que pueden afectar o ser afectados por las actividades de un sistema o empresa.

URL: (Localizador universal de recursos) método de identificación de documentos o lugares en Internet, es una cadena de caracteres que identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.

VISTA: es la representación de uno o más aspectos de la estructura de una arquitectura con el propósito de abarcar uno o más requerimientos para los interesados de estos requerimientos (stakeholder).

PUNTO DE VISTA: es una colección de patrones, plantillas, y convenciones para construir un tipo de vista. En él se definen las partes interesadas cuyas preocupaciones se reflejan en el punto de vista y las directrices, principios, modelos y plantillas para la construcción de sus vistas.

XML: extensible Markup Language es un lenguaje más flexible y adaptable que el HTML. XML es un lenguaje de lenguajes (un metalenguaje), porque es utilizado para describir otros lenguajes que permiten construir documentos electrónicos.