

UNIVERSIDAD PRIVADA ANTENOR ORREGO

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS



TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE

INGENIERO DE COMPUTACIÓN Y SISTEMAS

**“SISTEMA INFORMÁTICO WEB PARA LA GESTION DE VENTAS DE LA
BOUTIQUE DETALLITOS E.I.R.L. UTILIZANDO LA METODOLOGÍA AUP Y
FRAMEWORK QCODO DE PHP”**

AUTORES:

Br. Lindley Pedro Julca Diaz

Br. Allinson Frederick Rojas Zarate

ASESOR:

Ing. Freddy Infantes Quiroz

CIP: 139578

TRUJILLO – PERU

2015

“SISTEMA INFORMÁTICO WEB PARA LA GESTION DE VENTAS DE LA BOUTIQUE DETALLITOS E.I.R.L. UTILIZANDO LA METODOLOGÍA AUP Y FRAMEWORK QCODO DE PHP”

PRESENTADO POR:

Br. Lindley Pedro Julca Díaz

Br. Allinson Frederick Rojas Zarate

APROBADO POR:

Ing. Luis Vladimir Urrelo Huimán
PRESIDENTE CIP: 88212

Ing. José Antonio Calderón Sedano
SECRETARIO CIP: 139198

Ing. Henry Antonio Mendoza Puerta
VOCAL CIP: 139568

Ing. Freddy Infantes Quiroz
ASESOR CIP: 139578

DEDICATORIA

A Dios nuestro creador y hacedor de todas las cosas, quién nos da aliento y fuerzas de seguir adelante.

A mis padres quienes nos brindan su apoyo incondicional.

A mis hermanos, quienes siempre me dan aliento para seguir adelante.

Lindley

A Nuestro Padre Celestial por darme las fuerzas para seguir adelante en los momentos difíciles.

A mis padres quienes me formaron, comprendieron mis malos momentos, me tendieron su hombro y son mi ejemplo a seguir.

A mi familia, quienes me inculcaron sus valores, compartieron sus experiencias y me hicieron un mejor persona cada día.

Allinson

AGRADECIMIENTO

La presente Tesis es un esfuerzo conjunto donde participaron varias personas – aconsejando, corrigiendo, brindando su apoyo, etc.- que tuvieron un papel importante en el camino a dar el siguiente paso en nuestra vida profesional: ser Ingenieros.

Agradecemos al Ing. Freddy Infantes Quiroz, por brindarnos su paciencia, experiencia y excelente dirección en el presente trabajo. A nuestros docentes por impartirnos su conocimiento y profesionalismo; y a nuestros compañeros, quienes nos aconsejaron, apoyaron y dieron ánimos en nuestra vida universitaria y profesional; sin los cuales esta experiencia no hubiera sido la misma.

Asimismo, agradecemos a todo el personal de Boutique Detallitos E.I.R.L quienes nos compartieron su conocimiento y siempre se mostraron atentos ante nuestras inquietudes.

A nuestras familias por apoyarnos, por estar siempre presentes y brindarnos sus consejos, experiencia y ganas de superación.

Gracias a todos.

Los Autores.

INDICE GENERAL

DEDICATORIA	3
AGRADECIMIENTO	4
ABSTRACT	11
INTRODUCCIÓN.....	12
CAPITULO I: FUNDAMENTO TEÓRICO	15
1.1. SISTEMAS DE INFORMACIÓN	15
1.2. WEB	16
1.3. SISTEMAS DE INFORMACIÓN WEB	17
1.4. Metodología de Desarrollo AUP	18
1.5. STARUML	21
1.6. PHP	23
1.7. Framework QCode	25
1.8. MySQL.....	26
1.9. APACHE.....	30
1.10. Enterprise Architect	31
CAPÍTULO II: RESULTADOS	32
2. INICIO	33
2.1. Presupuesto	33
2.2. Análisis de requisitos	34
2.3. Modelo de dominio.....	37
2.4. Requerimientos técnicos	38

2.5.	Requerimientos priorizados.....	38
2.6.	Caso de uso de requerimientos	39
2.7.	Diagrama de clases.....	48
3.	ELABORACIÓN	49
3.1.	Modelo de Arquitectura.....	49
3.2.	Prototipo de Interfaz Usuario	49
4.	CONSTRUCCIÓN.....	56
4.1.	Diagrama de colaboración	56
4.2.	Diagrama de secuencia.....	62
4.3.	Modelo de despliegue	65
4.4.	Modelo de componentes	66
4.5.	Modelo Lógico.....	67
4.6.	Modelo físico	68
CAPÍTULO III: DISCUSIÓN.....		80
3.1.	CONTRASTACIÓN DE LA HIPÓTESIS	80
3.1.1.	Identificación de Variables e Indicadores	80
3.2.	ANÁLISIS DE RESULTADOS	88
REFERENCIAS BIBLIOGRAFICAS		91
CONCLUSIONES.....		89
RECOMENDACIONES.....		90

ÍNDICE COMPLEMENTARIOS DE (TABLAS - FIGURAS)

Tabla N° 1: Resumen de Configuración AUP.....	32
Tabla N° 2: Presupuesto	34
Tabla N° 3: Detalle CU “Administrar Usuarios”	41
Tabla N° 4: Detalle CU “Administrar Producto”	42
Tabla N° 5: Detalle CU “Administrar Proveedor.....	43
Tabla N° 6: Detalle CU “Registrar Venta”	44
Tabla N° 7: Detalle CU “Registrar Venta”.....	45
Tabla N° 8: Detalle CU “Registrar Nota de Ingreso”	46
Tabla N° 9: Detalle CU “Registrar Nota de Salida”.....	47
Tabla 10: Variación de datos históricos del indicador Satisfacción del usuario final <i>Fuente:</i> <i>[Elaboración Propia]</i>	81
Tabla 11: Variación de datos históricos del indicador Tiempo <i>Fuente: [Elaboración</i> <i>Propia]</i>	85

Figura N° 1: Modelo general de un sistema (Fernández Alarcón, 2006)	15
Figura N° 2: Ciclo de vida de AUP (AmbySoft, 2006).....	20
Figura N° 3: Diferenciación entre Modelo, Vista y Diagrama.....	22
Figura 4: Interface de StarUML	22
Figura 5: Sintaxis PHP	24
Figura N° 6: Modelo de Dominio.....	37
Figura N° 7: Caso de uso de almacén.....	39
Figura N° 8: Caso de uso de venta por internet.....	40
Figura N° 9: Caso de uso de ventas	40
Figura N° 10: Diagrama de clases	48
Figura N° 11: Autenticar usuario	49
Figura N° 12: Registrar proveedor	50
Figura N° 13: Registrar producto	50
Figura N° 14: Kardex de productos.....	51
Figura N° 15: Nota de ingreso.....	51
Figura N° 16: Nota de salida	52
Figura N° 17: Catálogo de productos	53
Figura N° 18: Carrito de compras	54
Figura N° 19: Diagrama de robustez de gestionar proveedor	56
Figura N° 20: Diagrama de robustez de gestionar producto	57
Figura N° 21: Diagrama de robustez kardex por producto.....	58
Figura N° 22: Diagrama de robustez de nota de salida	59
Figura N° 23: Diagrama de robustez nota de ingreso.....	60
Figura N° 24: Diagrama de robustez carrito de compra.....	61
Figura N° 25: Diagrama de robustez Ventas por internet	61
Figura N° 26: Diagrama de Secuencia Gestionar Cliente	62
Figura N° 27: Diagrama de Secuencia Gestionar Proveedor	63
Figura N° 28: Diagrama de Secuencia Gestionar Producto	63
Figura N° 29: Diagrama de Secuencia Nota de ingreso	64

Figura N° 30: Diagrama de Secuencia Nota de salida	64
Figura N° 31: Modelo de despliegue.....	65
Figura N° 32: Modelo de componentes.....	66
Figura N° 33: Modelo lógico.....	67
Figura 34: Región de aceptación o rechazo Fuente: [Elaboración Propia]	83
Figura N° 36: Región de aceptación o rechazo	87

RESUMEN

“SISTEMA INFORMÁTICO WEB PARA LA GESTION DE VENTAS DE LA BOUTIQUE DETALLITOS E.I.R.L. UTILIZANDO LA METODOLOGÍA AUP Y FRAMEWORK QCODO DE PHP”

PRESENTADO POR:

Bach. Lindley Pedro Julca Diaz

Bach. Allinson Frederick Rojas Zarate

La BOUTIQUE DETALLITOS E.I.R.L. es una empresa dedicada a la gestión comercial de ropa y accesorios para damas enfocada en la dama joven de hoy en día, desde hace dos años ofrece la mejor calidad y variadas marcas tanto nacionales como internacionales Cuenta con un local en Urb. Sol de villa (torre 01) la Libertad - Trujillo el cual es un reducido espacio y debido al incremento de su clientela necesita de mayor espacio y un mejor manejo de sus productos así como de sus gestiones administrativas. Pero se percataron problemas en la boutique, entre los cuales tenemos que mencionar que actualmente no posee un sistema, con el cual se controlen las distintas áreas de la empresa (ventas, compras y almacén), por lo que todos los documentos de información (boletas, facturas, órdenes de compra), son llenados manualmente, llegando a tener pérdida tanto en la información como en el tiempo.

Por lo tanto se vio en la necesidad de proponer como alternativa de solución un Sistema Web que realice la automatización de los procesos de ventas y de Entrada y Salida de productos de boutique detallitos, así mismo controlará el stock, actualizando cada vez que exista una entrada y/o salida de productos.

Para el análisis, diseño, implementación y documentación se usó la metodología AUP, el código fuente y el diseño de los formularios en Framework Qcodo de PHP, para la administración de la base de datos, se usó MySql.

Las principales mejoras y beneficios para la boutique son: Obtener un nuevo canal de ventas por internet para mejorar el servicio del cliente mediante la aceleración del proceso de gestión de ventas, optimizando las operaciones de almacén pues proporciona datos de inventarios precisos y transparentes.

ABSTRACT

“COMPUTER SYSTEM FOR WEB SALES MANAGEMENT OF BOUTIQUE DETALLITOS EIRL AUP USING PHP AND FRAMEWORK OF METHODOLOGY QCODO ”

Bach. Lindley Pedro Julca Diaz

Bach. Allinson Frederick Rojas Zarate

The BOUTIQUE Detallitos E.I.R.L. is a company dedicated to the commercial management of clothing and accessories for women focused on the young lady today, two years offers the best quality and varied national and international has a store in Urb brands. sunshine villa (tower 01) Freedom - Trujillo which is a small space and due to the increase of its customers need more space and better handling of their products and their administrative tasks. But problems noticed in the boutique, among which we must mention that currently does not have a system in which various areas of the company (sales, purchasing and warehouse) are controlled, so that all information documents (ballots , invoices, purchase orders) are completed manually, getting to have both information loss and time.

So he was on the need to propose an alternative solution a Web system to perform automated sales processes and input and output products trifles boutique, also control the stock, updating whenever there is an entry and / or output products.

For the analysis, design, implementation and documentation UPA methodology was used, the source code and designing the forms Qcodo PHP Framework, to manage the database, MySql was used.

The main improvements and benefits to the boutique are: Get a new online sales channel to improve customer service by speeding up the sales management process, optimizing warehouse operations as it provides accurate and transparent data inventories.

INTRODUCCIÓN

En la actualidad, las empresas sea cual sea el giro de su negocio buscan ser cada vez más competitiva, para esto se hace necesario muchas veces modificar la forma de realizar los procesos comunes de la empresa, haciendo uso de las tecnologías de información. En nuestro caso utilizaremos la metodología AUP, la cual abarca con todo el procedimiento de análisis y diseño necesario así como FRAMEWORK QCODO de PHP que nos permitirá cubrir los requerimientos solicitados.

La BOUTIQUE DETALLITOS E.I.R.L. es una empresa dedicada a la gestión comercial de ropa y accesorios para damas enfocada en la dama joven de hoy en día. En esta boutique se encuentran las prendas más selectas del mercado textil; desde hace dos años ofrecen la mejor calidad y variadas marcas tanto nacionales como internacionales.

La Boutique, pensando en la satisfacción de sus clientes así como su comodidad ha pensado en la implementación de un portal web en la que pueda mostrar sus productos al público, con esto pretende atraer nuevos clientes y a la vez informar a sus clientes antiguos de las novedades que llegan a la boutique; así mismo desea que el Portal Web permita a sus clientes la selección y compra de prendas de vestir, esto crearía un nuevo canal de venta a la Boutique, todo esto en conjunto se vería reflejado en un aumento de los ingresos por parte de dicha boutique. El desarrollo de esta aplicación web permitirá a boutique “DETALLITOS” llegar a un público más selectivo en cuanto a ropa, el cual está enfocado en mujeres jóvenes que podrán adquirir una prenda o accesorio desde el sitio donde se encuentren.

La realidad problemática de la gestión comercial de boutique DETALLITOS se caracteriza por lo siguiente:

- Los cambios en los precios son expuestos a confusiones debido que quien lo realiza es cualquier personal de la boutique.
- La empresa no tiene un manejo de stock apropiado tan solo cuenta con un almacén reducido, en la cual el vendedor entra a buscar alguna prenda solicitada.
- La empresa no cuenta con un registro de sus prendas la cual hace tediosa la búsqueda de alguna prenda.

- Al momento en que llegan los lotes de pedidos de los Proveedores a almacén, después de haber sido seleccionados y clasificados manualmente, el gerente debe registrar en un cuaderno los datos de cada lote clasificados por proveedor.
- En lo que respecta a ventas la empresa registra sus ventas en forma manual ante un pedido la vendedora emite un ticket a caja y ahí la cajera realiza manualmente una boleta.
- El gerente no dispone de información inmediata de las ventas, esperándose generalmente el cierre del día para poder realizar y controlar el estado de sus productos.
- Al finalizar el día las vendedoras de cada tienda deben realizar un cuadro de caja manualmente, lo que es muy laborioso y duradero.

Estas circunstancias impulsaron a proponer una solución Web para la Gestión de Ventas de la Boutique Detallitos E.I.R.L, que permitirá mejorar los procesos para así contribuir en la toma de decisiones.

La realidad problemática anterior puede ser directa con la siguiente pregunta de investigación ¿De qué manera se puede mejorar la Gestión de Ventas de la Boutique Detallitos E.I.R.L. utilizando Tecnologías de información?

Formulándose la siguiente hipótesis: “Un Sistema Informático Web utilizando la metodología AUP y tecnología PHP mejorara la Gestión de Ventas de la Boutique Detallitos E.I.R.L”

Siendo el Objetivo General: “Desarrollar un Sistema Informático Web para la Gestión de Ventas de la Boutique Detallitos E.I.R.L. utilizando la metodología AUP y framework QCode de PHP”.

Y los Objetivos Específicos:

- Realizar la Investigación Bibliográfica de la metodología AUP, Framework QCode y Sistemas de información Web.
- Recopilar información propia de la Empresa que permita la posterior identificación de procesos y requerimientos de usuarios.
- Elaborar el análisis y diseño a partir de la información recopilada aplicando las fases de la metodología AUP y la herramienta Enterprise Architect.
- Realizar el diseño y la implementación de la Base de datos usando MySQL.
- Desarrollar el software del Sistema Informático Web utilizando el framework QCode de PHP.

El presente documento consta de 3 capítulos en los cuales se irá viendo cómo se desarrolló la solución y que técnicas fueron usadas terminando con las conclusiones de la solución y recomendaciones, pasaremos a detallar cada capítulo brevemente.

En el Capítulo I: Contiene la descripción de la información de la metodología a usar, en este caso será AUP y de las herramientas tecnológicas que usaremos para la solución, el **Framework QCODO de PHP**, como gestor de base de datos Mysql y para la realización del modelado se usará **Enterprise Architect 7.5**.

En el Capítulo II: Se plantea la solución desarrollada de la metodología AUP y su respectiva documentación, etapa por etapa:

Fase de INICIO: Diagrama de Procesos, Modelo de dominio, Requerimientos técnicos, Requerimientos priorizados, Caso de uso de requerimientos y Diagrama de clases

Fase de ELABORACIÓN: Modelo de Arquitectura y Prototipo de Interfaz Usuario

Fase de CONSTRUCCIÓN: Diagrama de colaboración, Diagrama de secuencia, Modelo de despliegue, Modelo de componentes, Modelo Lógico y Modelo físico.

En el Capítulo III: Comprende la contrastación de la hipótesis.

Finalmente se detallan las principales conclusiones y recomendaciones como resultado del presente trabajo de investigación, así como las referencias bibliográficas y los anexos.

CAPITULO I: FUNDAMENTO TEÓRICO

1.1. SISTEMAS DE INFORMACIÓN

Son Según Vicenç Fernández, Sistemas de Información es un conjunto de componentes interrelacionados que reúne (o recupera), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control de la organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores a analizar problemas, a visualizar asuntos complejos y a crear productos nuevos.

Entre las funciones principales que realiza el sistema de Información son:

- Captura o recolecta datos tanto externos como internos.
- Trata esos datos mediante procesos que operan con ellos.
- Distribuye la información resultante a los usuarios y actividades que la requieran.

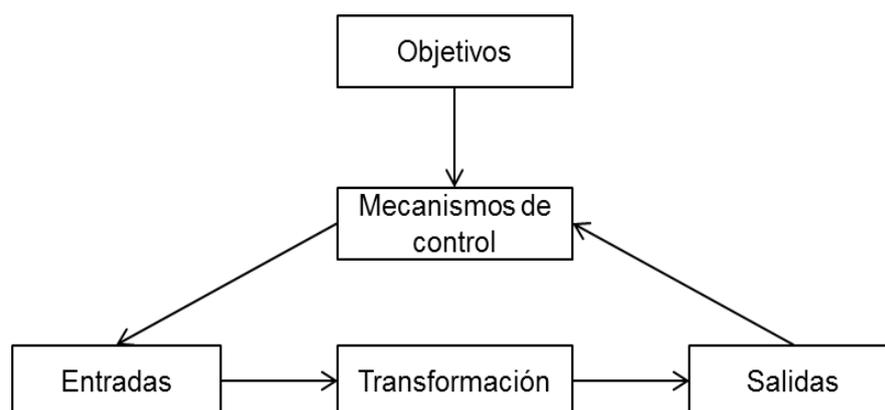


Figura N° 1: Modelo general de un sistema (Fernández Alarcón, 2006)

El modo en que los sistemas de información realizan sus funciones, los recursos con los que cuentan y el tipo de usuarios a los que satisfacen definen el alcance de los mismos. (De Pablos Heredero, 2008)

TIPOS DE SISTEMAS

Laudon y Laudon los clasifica de acuerdo a los diferentes niveles de la organización: Los Sistemas a Nivel Operativo son Sistemas de Información que supervisan las actividades elementales y transacciones de la organización; los Sistemas a Nivel del Conocimiento apoyan los trabajadores del conocimiento y de datos de una organización; los Sistemas a Nivel Administrativo apoyan las actividades de supervisión, control, de tomas de decisiones y administrativas de los gerentes de nivel medio; y por último, los Sistemas a Nivel Estratégico apoyan las actividades de planeación a largo plazo de la dirección general de la empresa. (Laudon & Laudon, 2004)

1.2. WEB

La World Wide Web, más conocida como Web, es una de las aéreas de Internet que se ha desarrollado más rápidamente. Nació en 1989, como parte de un proyecto de CERN de Suiza y con el objetivo de mejorar el intercambio de información dentro de Internet, y vea en lo que se ha convertido actualmente.

De ser un lugar en el que se podía encontrar información, ha pasado a ser un gran centro comercial. En muy pocos años, las sencillas paginas estáticas de la Web han evolucionado hasta convertirse en sofisticados sitios (sites) donde se pueden comprar, de forma segura, bienes y otros servicios. Mientras tanto, han nacido nuevas compañías que realizan sus negocios exclusivamente a través de la Web, como por ejemplo la dedicada a la venta de libros www.amazon.com. (Hobbs, 1999)

Los pilares de la web son:

- **HTML** como lenguaje para crear los contenidos de la Web, basado en Estándar Generalized Markup Language (SGML).

- **HTTP** como protocolo de comunicación entre los ordenadores de la Web, encargado de la transferencia de las páginas web y demás recursos.
- **URL** como medio de localización (direccionamiento) de los distintos recursos en Internet. (Luján Mora, 2002)

1.3.SISTEMAS DE INFORMACIÓN WEB

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo del Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet. Los sitios Web, como expresión de sistemas de información, deben poseer los siguientes componentes:

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.

Entonces un sistema de información web es un conjunto de elementos relacionados y ordenados, según ciertas reglas que aporta al sistema objeto- ,es decir, a la organización a la que sirve y que marca sus directrices de funcionamiento- la información necesaria para el cumplimiento de sus fines apoyados de la web para recoger, procesar y almacenar datos, procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. (Rodríguez Perojo & Ronda León, 2012)

1.4. Metodología de Desarrollo AUP

El Proceso Unificado Ágil (AUP por Agile Unified Process en Ingles) es un enfoque de desarrollo de software derivado de RUP que fue desarrollado por Scott Ambler desde 2002 hasta 2006 y combina los principales artefactos de RUP. AUP combina el Modelo de Negocio, Modelo de Casos de Uso, Modelo de Análisis y Diseño de RUP en un solo Modelo. (Stober & Hansmann, 2010)

AUP se caracteriza por ser interactiva y además incremental. Es decir, en el desarrollo de un proyecto importante, éste se divide en pequeños proyectos derivados. Esto sirve para tener control de las pequeñas partes y si surge cualquier problema es posible solucionarlo lo antes posible. Cada pequeña parte de la división del proyecto es una interacción. Esto hace que de solucionarlo lo antes posible.

Cada pequeña parte además, trata de un conjunto de caso de usos, por lo que brinda importancia a la funcionalidad que el sistema debe cumplir para satisfacer los requerimientos del usuario del sistema. Los casos de uso son los que orientan todas las actividades del desarrollo del producto software.

PRINCIPIOS

AUP se basa en:

- **Simplicidad.** Todo se describe concisamente utilizando poca documentación.
- **Agilidad.** El ajuste a los valores y principios de La Alianza Ágil.
- Centrarse en actividades de alto valor: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
- **Herramienta de la independencia.** Usted puede usar cualquier conjunto de herramientas que desea con el AUP. Se sugiere utilizar las herramientas más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.

- **Usted querrá adaptar este producto para satisfacer sus propias necesidades.** La metodología AUP es un producto de fácil uso utilizando cualquier herramienta. No es necesario comprar una herramienta especial, o tomar un curso, para adaptar esta metodología.

Principios de La Alianza Ágil al cual se ajusta AUP:

- La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptar que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregar software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia. (ECURED, 2012)

FASES

AUP se caracteriza por ser “serial in a large”, lo cual se constata con sus cuatro fases:

- **Inicio**, su meta principal es identificar el alcance inicial del proyecto, una arquitectura potencial para el sistema, obtener un presupuesto inicial y la aceptación de los stakeholders.
- **Elaboración**, donde se prueba la arquitectura del sistema.
- **Construcción**, se elaboraran entregables del sistema, priorizando las necesidades de los stakeholders.
- **Transición**, validan y despliegan el sistema en el ambiente de producción.

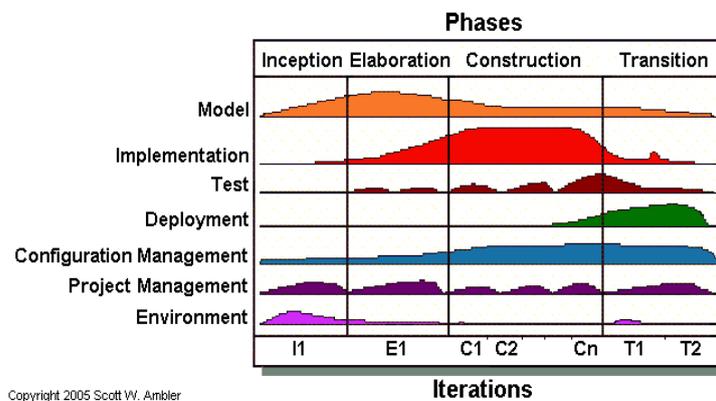


Figura N° 2: Ciclo de vida de AUP (AmbySoft, 2006)

DISCIPLINAS

Son ejecutadas en forma iterativa, definiendo las actividades que el equipo de desarrollo ejecuta para construir, validar y liberar software funcional, el cual cumple con las necesidades del usuario.

ROLES

Los roles no deben ser tomados como puestos, ya que una persona puede realizar varios roles y un rol puede ser asumido por varias personas.

1.5.STARUML

StarUML es un proyecto de código abierto para desarrollo rápido, flexible y extensible, con muchas características y libre disponibilidad UML/ plataforma MDA. Su objetivo es construir una herramienta de modelado de software y plataforma que es un remplazo convincente de las herramientas UML comerciales como Rational Rose, Together, etc. (StarUML, 2005)

Presenta como principales características:

- **UML 2.x:** UML se expande constantemente sus estándares debido a la OMG (Object Management Group).
- **MDA (Model Driven Architecture):** La arquitectura dirigida por modelos fue introducida por la OMG, en la cual StarUML provee tantas variables customizables como perfiles UML.
- **Arquitectura Plug-in:** StarUML provee una simple pero poderosa arquitectura plug-in, en la cual se puede desarrollar módulos plug-in en COM- lenguajes compatibles como C++, Delphi, C# y VB.
- **Usabilidad:** StarUML fue implementado para brindar un fácil uso al usuario como Dialogo Rápido, Manipulación mediante teclado, Resumen de Dialogo, etc. (Jalali, 2008)

StarUML hace una clara distinción conceptual entre modelos, vistas y diagramas.

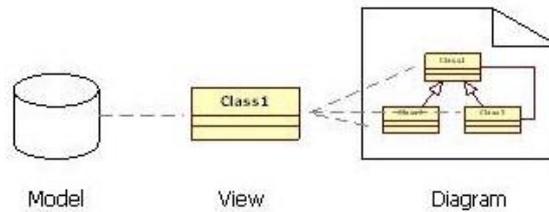


Figura N° 3: Diferenciación entre Modelo, Vista y Diagrama

- **Modelo:** Es un elemento que contiene información para un modelo de software.
- **Vista:** Es una expresión visual de la información contenida en un modelo; y
- **Diagrama:** Es una colección de elementos de la vista que representan pensamientos específicos del usuario de diseño.

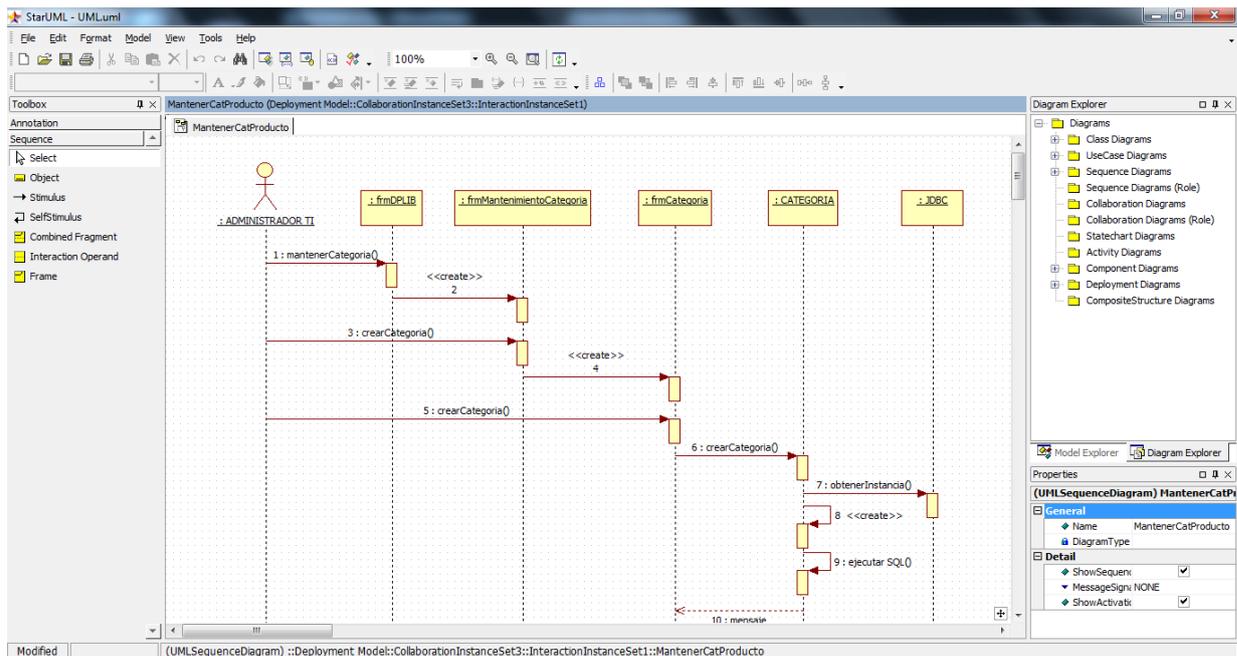


Figura 4: Interface de StarUML

La interfaz del usuario es intuitiva. En el lado derecho superior una ventana permite desplazarse rápidamente entre todos los contenidos de un proyecto, la visualización puede ser como modelo o una vista de diagrama. Permite que varios diagramas puedan estar abiertos al mismo tiempo y las pestañas permiten cambiar rápidamente entre las vistas. La ventana inferior derecha

permite documentar el esquema actual, ya sea con texto o adjuntar un documento de texto externo. Durante la edición del diagrama “wizards” que se encuentran alrededor del objeto le dan los accesos directos a las principales tareas relacionadas con su funcionamiento actual, como por ejemplo: agregar un atributo, crear una clase. (Martinig, 2011)

1.6. PHP

Es un lenguaje de código abierto especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. Una de sus características radica que en lugar de usar muchos comandos para mostrar HTML (como en C o Perl), páginas PHP contienen HTML con código incluido en el mismo que hace "algo".

Lo que distingue a PHP de algo lado-cliente como JavaScript, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. (PHP, 2012)

Siendo un lenguaje multiplataforma; los programas funcionan igual sobre diferentes plataformas, trabajando sobre la mayoría de servidores web y estando preparado para interactuar con más de 20 tipos de base de datos. No obstante, al ser un lenguaje inicialmente concebido para entornos Unix, es sobre este sistema operativo en el que se pueden aprovechar más sus prestaciones.

Entre las tareas para el desarrollo web podemos citar:

- Funciones de correo electrónico que pueden ser utilizadas para programar completos sistemas de correo vía web.
- Funciones de administración y gestión de base de datos específicas para la mayoría de gestores comerciales y funciones para conexiones ODBC con base de datos en sistemas Microsoft.

- Funciones de gestión de directorios y ficheros, incluso para la transferencia mediante FTP.
- Funciones de tratamiento de imágenes y librerías de funciones graficas.
- Funciones de generación y lectura de cookies.
- Funciones para la generación de documentos PDF. (Cobo, et al., 2005)

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

Figura 5: Sintaxis PHP

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario PHP de FastCGI, como lighttpd y nginx. PHP funciona ya sea como un módulo, o como un procesador de CGI.

VENTAJAS

- Es un lenguaje multiplataforma.
- Es completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Por lo que la programación en PHP sea segura y confiable.
- Se puede conectar a la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

- Tiene capacidad de expandir su potencial mediante la utilizando la una gran cantidad de módulos (llamados ext's o extensiones).
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- Tiene manejo de excepciones (desde PHP5).

El programador puede aplicar en su trabajo cualquier metodología de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. (ECURED, 2012).

1.7. Framework QCode

QCode es un framework de desarrollo de código abierto para PHP 5 que construye un mapeo de objetos y una interfaz de ABMS (Alta, baja y modificación) utilizando AJAX desde un modelo existente de una base de datos. Adicionalmente, contiene QForms que son componentes para generar formularios. Es un framework de desarrollo liviano que puede ser utilizado tanto en pequeñas como en grandes aplicaciones.

El framework consiste de dos componentes principales: el generador de código y los Qcomponentes (Qforms). Qcode usa ORM para el acceso a base de datos. El generador de código introspecciona una estructura de base de datos relacional y construye un ORM, así como también varias interfaces remotas (SOAP, AJAX) para el ORM. El ORM en QCode puede ser extendido para proveer funcionalidad mantenida por el usuario (via object subclassing). Qforms es un motor de templates inspirado de .NET en el que cada elemento del formulario es un objeto con su funcionalidad en métodos y atributos. Qforms incluye la habilidad de validar campos, disparar eventos, y asociar llamadas AJAX. Qforms se comunica perfectamente con el ORM, permitiendo a los desarrolladores a rápida e interactivamente cambiar cualquiera de los tres componentes de la arquitectura MVC con poco impacto a los otros componentes.

Qcode es un framework maduro y mantenido por una gran comunidad. Ha sido utilizado en entornos de producción por la NASA y otras organizaciones. El framework fue concebido y desarrollado por Mike Ho y su compañía QuasIdea.

A partir de su versión 0.4, Qcodo incluye novedades orientadas a fomentar una mayor participación de la comunidad en el proyecto, como actualizaciones basadas en paquetes creados por el usuario (QPM) y posibilidad de aportar código vía GitHub.

En noviembre de 2008 algunos desarrolladores de la comunidad de Qcodo lanzaron un fork del proyecto llamado QCubed, basado en la versión 0.3 del framework original.

1.8.MySQL

Es uno de los principales y más populares Motores de Gestión de Base de Datos SQL, la cual es desarrollada y distribuida por Oracle Corporation y actualmente cuenta con más de 100 millones de descargas. (MySQL, 2012)

Sus principales características son:

- Orientado al desarrollo web.
- Escrito en C y C++.
- Permite el desarrollo Aplicado a Objetos.
- Es libre y de código abierto,
- Rápida, confiable, escalable y fácil de usar.
- El código fuente es invisible al navegador y al cliente por ejecutar la lógica. en el servidor y enviar el resultado HTML al navegador.
- Posee gran documentación en la web.

A su vez, trabaja bajo el modelo ACID por:

- **Atomicity (Atomicidad)**, involucra las transacciones InnoDB e incluye las sentencias COMMIT y ROLLBACK.
- **Consistency (Consistencia)**, implica el procesamiento interno InnoDB de protección de datos ante fallas, incluye características cómo: InnoDB doublewrite y InnoDB crash recovery.
- **Isolation (Aislamiento)**, involucra las transacciones InnoDB, en su mayoría del nivel de aislamiento que aplica a cada transacción, incluye

la sentencia SET ISOLATION LEVEL y los detalles de bajo nivel de bloqueo InnoDB.

- **Durability (Durabilidad)**, está referida a las características de MySQL que interactúan con la configuración del hardware, incluye características como: InnoDB doublewrite buffer, prendido o apagado mediante la opción de configuración InnoDB doublewrite y sentencias como InnoDB flush log at trx commit, sync binlog, Innodb file per table, etc. (MySQL, 2012)

VENTAJAS

- **Escalabilidad y flexibilidad.** El servidor de bases de datos MySQL brinda lo último en escalabilidad, pues es capaz de manejar bases de datos empotradas ocupando sólo 1MB, y hacer funcionar data warehouses que contengan terabytes de información. Asimismo, la flexibilidad de plataforma es una característica clásica de MySQL, soportando distintas versiones de Linux, UNIX y Windows Y como es un open source de MySQL permite una personalización completa para aquellos que deseen añadir características al servidor.
- **Alto rendimiento.** Su arquitectura única de motores de bases de datos permite configurar el servidor MySQL para aplicaciones específicas, por lo que MySQL puede cumplir con las expectativas de rendimiento de cualquier sistema, ya sea un sistema de procesamiento transaccional de alta velocidad, o un sitio web de gran volumen sirviendo un billón de consultas diarias MySQL debido a herramientas de carga de alta velocidad, índices full-text y otros mecanismos de mejora del rendimiento
- **Alta disponibilidad.** Solidez y disponibilidad constante son características distintivas de MySQL, con clientes confiando en ellas para garantizar el uptime en todo momento MySQL ofrece una amplia

variedad de soluciones de alta disponibilidad, desde replicación a servidores de cluster especializados, u ofertas de terceros

- **Robusto soporte transaccional.** MySQL ofrece uno de los motores de bases de datos transaccionales más potentes del mercado. Las características incluyen un soporte completo de ACID (atómica, consistente, aislada, duradera), bloqueo a nivel de filas, posibilidad de transacciones distribuidas, y soporte de transacciones con múltiples versiones donde los lectores no bloquean a los escritores y viceversa. También se asegura una integridad completa de los datos mediante integridad referencial, niveles de aislamiento de transacciones especializados, y detección de deadlocks.
- **Fortalezas en Web y Data Warehouses.** MySQL es el estándar de facto para sitios web de gran tráfico por su motor de consultas de alto rendimiento, su posibilidad de insertar datos a gran velocidad, y un buen soporte para funciones web especializadas como las búsquedas fulltext. Estas mismas fortalezas también se aplican a entornos de data warehousing, donde MySQL escala hasta el rango de los terabytes tanto para un solo servidor, como para varios. Otras características como las tablas en memoria, índices B-tree y hash, y tablas comprimidas hasta un 80% hacen de MySQL una buena opción para aplicaciones web y de BI.
- **Fuerte protección de datos.** MySQL ofrece características de seguridad que aseguran una protección absoluta de los datos. En cuanto a autenticación, MySQL ofrece potentes mecanismos para asegurar que sólo los usuarios autorizados tienen acceso al servidor. También se ofrece soporte SSH y SSL para asegurar conexiones seguras. Existe una estructura de privilegios que permite que los usuarios sólo puedan acceder a los datos que se les permite, así como potentes funciones de cifrado y descifrado para asegurarse de que los datos están protegidos. Finalmente, se ofrecen utilidades de backup y recuperación por parte de MySQL y terceros, que permiten copias completas, tanto lógicas como físicas, así como recuperación point-in-time.

- **Desarrollo de aplicaciones completo.** MySQL es la base de datos open source más popular pues ofrece un soporte completo para cualquier necesidad de desarrollo. En la base de datos se puede encontrar soporte para procedimientos almacenados, triggers, funciones, vistas, cursores, SQL estándar, y mucho más. Existen librerías para dar soporte a MySQL en aplicaciones empotradas. También se ofrecen drivers (ODBC, JDCBC,...) que permiten que distintos tipos de aplicaciones puedan usar MySQL como gestor de bases de datos. No importa si es PHP, Perl, Java, Visual Basic, o .NET, MySQL ofrece todo lo necesario para conseguir el éxito en el desarrollo de sistemas de información basados en bases de datos.
- **Facilidades de gestión.** MySQL ofrece posibilidades de instalación excepcionales, con un tiempo medio desde la descarga hasta completar la instalación de menos de quince minutos. Esto es cierto sin importar que la plataforma sea Windows, Linux, Macintosh, o UNIX Una vez instalado, características de gestión automáticas como expansión automática del espacio, o los cambios dinámicos de configuración descargan parte del trabajo de los atareados administradores. MySQL también ofrece una completa colección de herramientas gráficas de gestión que permiten al DBA gestionar, controlar y resolver problemas en varios servidores desde una misma estación de trabajo. Además, hay multitud de herramientas de terceros que gestionan tareas como el diseño de datos y ETL, administración, gestión de tareas y monitorización.
- **Open Source y soporte 24/7.** Muchas empresas no se atreven a adoptar software open source porque creen que no podrán encontrar el tipo de soporte o servicios profesionales en los que confían con su software propietario actual. Las preguntas sobre indemnizaciones también aparecen. Estas preocupaciones pueden desaparecer con el completo servicio de soporte e indemnización disponibles. MySQL no es un proyecto típico Open Source ya que todo el software es propiedad de MySQL AB, lo que permite un modelo de coste y soporte que ofrece

una combinación única entre la libertad del open source y la confianza de un software con soporte.

- **Coste Total de Propiedad menor.** Los costos son menores al migrar a aplicaciones actuales a MySQL, o usar MySQL. Asimismo, debido a que gracias al servidor MySQL y las arquitecturas scale-out se puede utilizar hardware económico, pueden alcanzar niveles sorprendentes de escalabilidad y rendimiento, y todo a un coste bastante menor que el de los sistemas propietarios. Además, la robustez y facilidad de mantenimiento de MySQL implican que los administradores no pierden el tiempo con problemas de rendimiento o disponibilidad, sino que pueden concentrarse en tareas de mayor impacto en el negocio. (WARP, 2012)

1.9. APACHE

Es un servidor web de código abierto multiplataforma que implementa el protocolo HTTP/1.1 muy popular por su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza. Además, permite la personalización de respuestas ante errores que puedan ocurrir en el servidor, inclusive programarlo para que ejecute scripts cuando esto suceda.

Apache es utilizado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web son diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es altamente configurable, permite bases de datos de autenticación y negociado de contenido.

Apache es usado cuando el contenido necesita ser puesto a disposición en una forma segura y confiable. Ejemplos: Cuando se comparten archivos desde una computadora personal hacia Internet, cuando se comparten documentos importantes dentro de determinadas las áreas de una empresa.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache con el fin de pre-visualizar y probar código mientras éste es desarrollado.

Sus principales ventajas son:

- Modular
- Código Abierto
- Multiplataforma
- Popular (fácil de conseguir ayuda/suporte)

La última versión estable de Apache HTTP Server es la 2.4.1. (APACHE, 2012)

1.10. Enterprise Architect

Enterprise Architect es una herramienta para el lenguaje de modelado UML, que combina el poder de la última especificación UML 2.1 con alto rendimiento e interfaz intuitiva, para concebir un modelado avanzado y completo en el desarrollo e implementación de software (**Sparx System, 2007**)

Características:

- Alta capacidad - cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento.
- Velocidad, estabilidad y buen rendimiento - ayudando a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente.
- Trazabilidad de extremo a extremo - desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue.

Construido sobre las bases de UML 2.1 – Usa Perfiles UML para extender el dominio de modelado, combina Procesos de Negocio, Información y Flujos de trabajo en un modelo. Tiene soporte para los 13 diagramas de UML 2 y más.

CAPÍTULO II: RESULTADOS

Este trabajo de investigación siguió el siguiente método de desarrollo:

	FASES	RESULTADOS
Modelo	Inicio	✓ Diagrama de Flujo de Procesos
		✓ Modelo de Domino General
		✓ Requerimientos Técnicos
		✓ Lista de Requerimientos Priorizados
		✓ Caso de Uso de Requerimientos
		✓ Diagrama de Clases
	Elaboración	✓ Modelo de Arquitectura
		✓ Prototipos de Interfaz de Usuario
	Construcción	✓ Interfaces de Usuario
		✓ Diagrama de Colaboración
		✓ Diagrama de Secuencia
		✓ Modelo de Despliegue
		✓ Diagrama de Clases del diseño
		✓ Modelo Físico de Datos
	✓ Mapa de Navegación	

Tabla N° 1: Resumen de Configuración AUP

2. INICIO

2.1. Presupuesto

RECURSOS HUMANOS		COSTO (S/.)
2 Investigadores		0.00
Asesor		1250.00
BIENES		
Materiales	Útiles de Escritorio	25.00
	Cd's	35.00
Equipos	02 Computadoras Pentium IV de 3 Ghz, 2 Gb RAM, HD 40 Gb	3000.00
	2 memorias USB	160.00
	Herramienta PHP	0.00
Software	Herramienta Mysql	0.00
	Microsoft Windows XP Professional Version 2002 Service Pack 3	800.00
	Microsoft Office 2010	350.00
	Enterprise Architect	200.00
	Biblioteca	00.00
Servicios	Escaneado	50.00
	Espiralado	10.00
	Fotocopiado	20.00
	Grabaciones de cds	40.00
	Impresiones	20.00
	Internet	100.00
	Movilidad	180.00
	Servicio de Luz	50.00
	Telefonía	50.00

PRESUPUESTO TOTAL GENERAL	
CATEGORÍA	COSTO TOTAL (S/.)
Recursos Humanos	1250.00
Materiales	60.00
Equipo	3160.00
Software	1350.00
Servicios	520.00
COSTO TOTAL:	S/. 6340.00

Tabla N° 2: Presupuesto

2.2. Análisis de requisitos

2.2.1. Requerimientos Funcionales

a. Declaración de Requerimientos:

a.1. Registrar Usuario

En este mantenedor se registra el nombre, apellidos, DNI, usuario y clave, correo, fecha de nacimiento y sexo de un nuevo usuario del sistema

a.2. Registrar Cliente

En este mantenedor se registra el nombre, apellido, dirección, DNI, celular, email, fecha de nacimiento y sexo de un nuevo usuario del sistema

a.3. Registrar Proveedor

En este mantenedor se escribe la dirección, el nombre o razón social, RUC, el teléfono fijo, celular y email.

a.4. Registrar Productos

En este mantenedor se ingresa el código interno, la descripción, precio y unidad de medida.

a.5. Registrar Línea de Producto

En este mantenedor se ingresa la descripción y estado.

a.6 Registrar Venta

En este mantenedor se selecciona el tipo de documento (boleta o factura), se la fecha del documento, cliente y se ingresan los productos y su cantidad, automáticamente se calcula el subtotal, el IGV y el total.

a.7 Registrar productos a carrito de compras

Previa consulta del catálogo de productos se procede a registrar los pedidos en el carrito de compras, una vez finalizado se procede a registrar la venta previa validación del software de pasarela de pago de la validez de la transacción.

a.8. Registrar Ingresos

a.8.1. Registrar Nota de Ingreso

En este mantenedor se selecciona el tipo de documento (boleta, factura, guía de remisión), se ingresa el número de documento, la fecha del documento, se selecciona el proveedor, se ingresan los productos y su cantidad, automáticamente se calcula el subtotal, el IGV y el total.

a.9. Registrar Salidas

a.9.1. Registrar Nota de Salida

En este mantenedor se ingresa el tipo documento el número de documento, la fecha del documento, se ingresan los productos y su cantidad, automáticamente se calcula el subtotal, el IGV y el total.

a.10. Consultar Kardex

En este mantenedor se ingresa el código alternativo o una descripción del material, para luego mostrarse una lista del kardex del material.

a.11. Elaborar Reportes

a.11.1. Elaborar Reportes de Notas de Ingresos

En este mantenedor se muestra una lista de ingresos de productos previa selección de rango de fechas a mostrar.

a.11.2. Elaborar Reportes de Notas de Egresos

En este mantenedor se muestra una lista de egresos de productos previa selección de rango de fechas a mostrar.

a.11.3. Elaborar Reportes de Kardex

En este mantenedor se muestra una lista de ingresos y egresos de productos previa selección de material por descripción.

2.2.2. Requerimientos No Funcionales

En la entrevistas sostenida con el administrador de la empresa se pudo obtener una lista de requerimientos que en si no son funcionalidades propias del sistema pero que si es importante considerar en todo en proceso de desarrollo.

- **Transparencia de uso:**

Los colores deben ser agradables a la vista según los estándares establecidos en W3C.

- **Tiempo de respuesta:**

Se debe tener un tiempo de respuesta no mayor de 3 segundos para todas las transacciones de la página.

- **Disponibilidad:**

La página estará activa las 24 horas del día.

- **Seguridad:**

Se realizara backups periódicos.

Se manejan perfiles de acceso a la base de datos.

Se utilizara una clave de acceso para el administrador al sistema.

2.4. Requerimientos técnicos

- Metodología de Desarrollo AUP.
- Herramienta de Modelado: Enterprise Architect.
- FRAMEWORK QCODO DE PHP.
- Base de Datos Mysql 5.0.
- Apache http Server.
- Tiempo de respuesta máximo 4 segundos en cada transacción.

2.5. Requerimientos priorizados

Jefe de Almacén

- Registrar orden de compra
- Consultar orden de compra
- Registrar nota de ingreso
- Registrar categoría
- Registrar producto
- Registrar nota de salida
- Consultar producto
- Reporte de productos por proveedor
- Reporte de productos por categoría
- Reporte de kardex

Cliente

- Consultar categoría
- Registrar venta
- Registrar carrito de compras

Jefe de Ventas

- Reporte de productos más vendidos
- Reporte de ventas por día
- Reporte de productos por stock
- Registrar cliente

2.6. Caso de uso de requerimientos

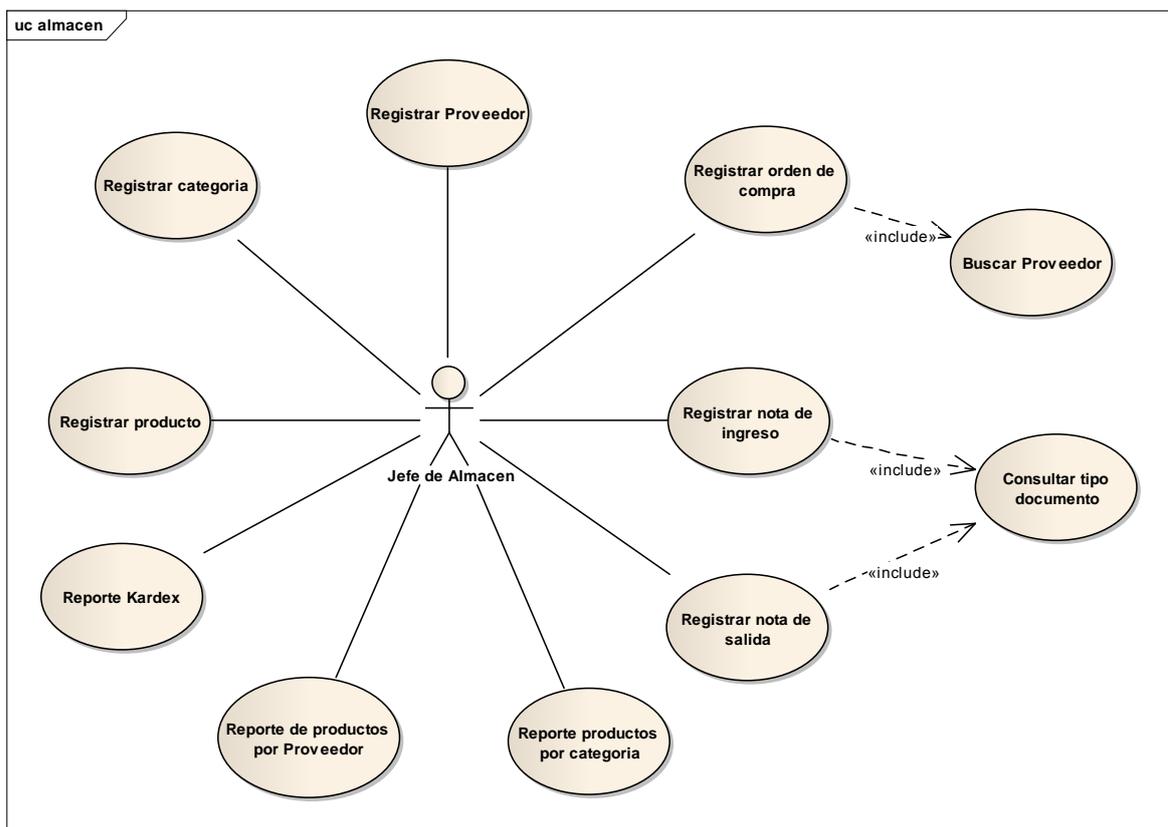


Figura N° 7: Caso de uso de almacén

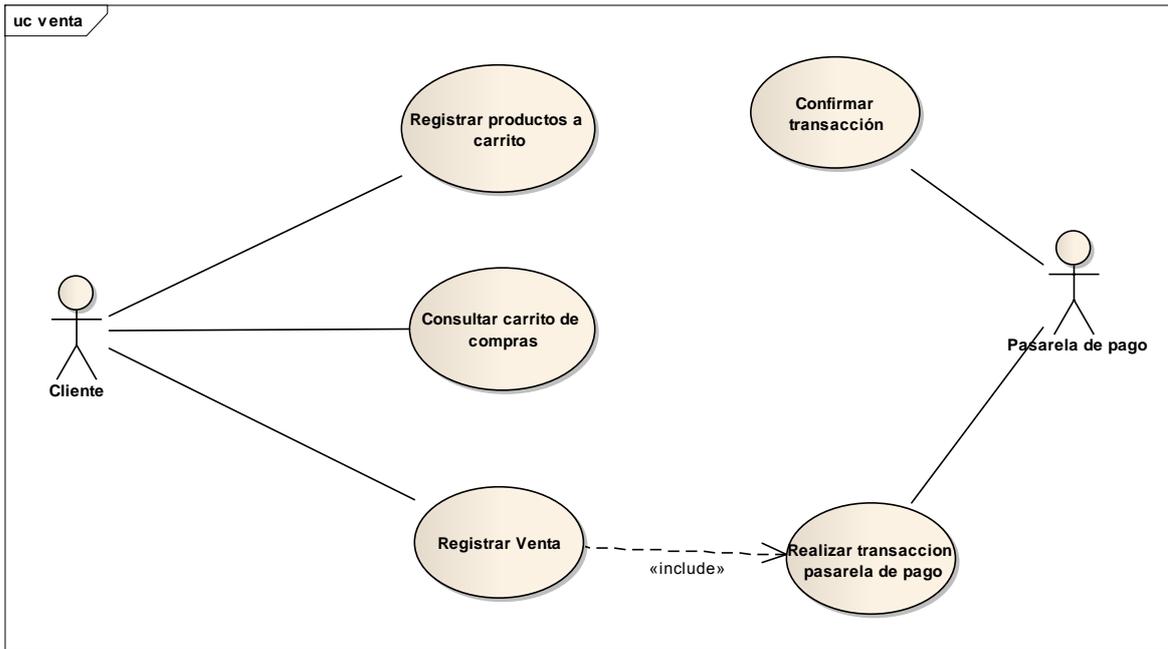


Figura N° 8: Caso de uso de venta por internet

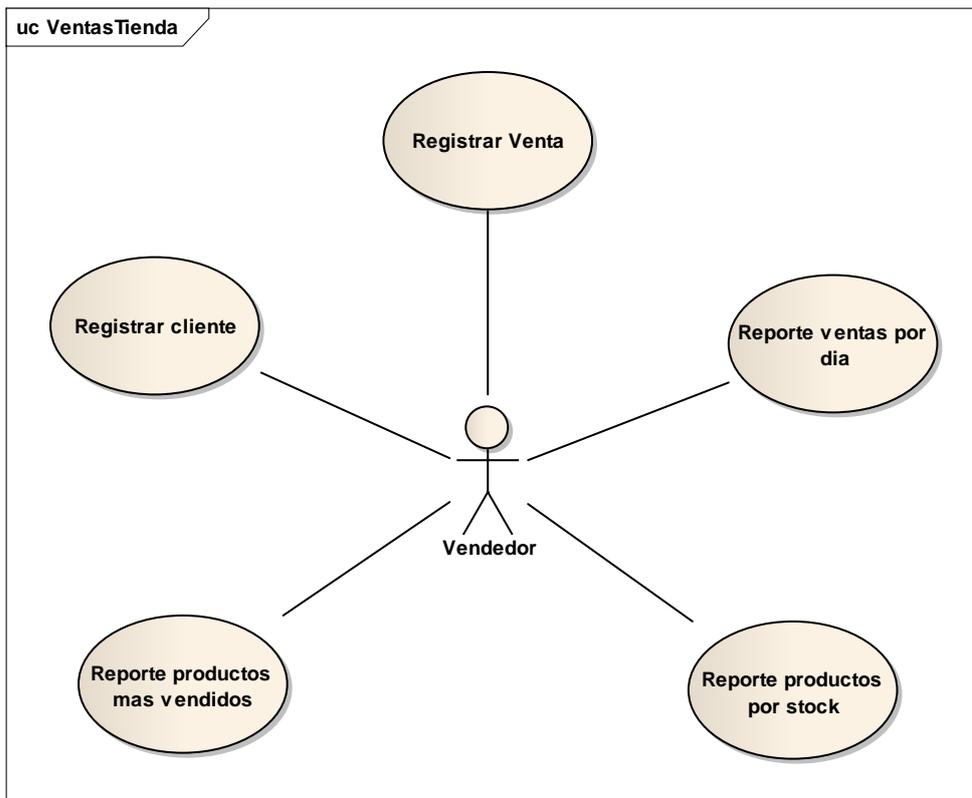


Figura N° 9: Caso de uso de ventas

Nombre	Administrar Usuarios	
Descripción	Permite crear nuevos usuarios al sistema.	
Actor	Jefe de Almacén	
Precondición	El actor debe ser colaborador de la empresa. El actor debe de tener un usuario, contraseña y permisos de acceso.	
Secuencia Normal	Paso	Acción
	1	El actor debe de acceder a la pantalla de logueo.
	2	El actor debe autenticarse con usuario y clave.
	3	El sistema valida credenciales válidas en caso que no sean correcto entonces regresar al paso 2.
	4	Se valida el perfil de usuario que accede al sistema.
	5	Se ingresa a la pantalla respectiva.
	6	El actor selecciona la opción para crear usuarios.
	7	Se despliega la pantalla para crear usuarios.
	8	Se crea un nuevo usuario, ingresando su nombre de usuario, contraseña, email.
	9	Si el usuario desea, puede crear varios usuarios nuevos, volviendo al paso 7.
	10	Una vez que finalice la creación de usuarios, el actor puede salir del sistema o salir de la creación de usuarios, volviendo al paso 5.
	11	El usuario puede buscar usuarios para modificar o eliminar.
	12	Una vez seleccionado el usuario, puede hacer clic en el botón de modificar o eliminar.
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
	1	El sistema valida el usuario y contraseña, y en caso de que no sean correctos se desplegará un mensaje indicando que existió un error.

Tabla N° 3: Detalle CU “Administrar Usuarios”

Nombre	Administrar Producto	
Descripción	Permite crear nuevos productos al sistema.	
Actor	Jefe de Almacén	
Precondición	El actor debe ser colaborador de la empresa. El actor debe de tener un usuario, contraseña y permisos de acceso.	
Secuencia Normal	Paso	Acción
	1	El actor debe de acceder a la pantalla de logueo.
	2	El actor debe autenticarse con usuario y clave.
	3	El sistema valida credenciales válidas en caso que no sean correcto entonces regresar al paso 2.
	4	Se valida el perfil de usuario que accede al sistema.
	5	Se ingresa a la pantalla respectiva.
	6	El actor selecciona la opción para crear producto.
	7	Se despliega la pantalla para crear producto.
	8	Se crea un nuevo producto, ingresando tipo de producto, descripción, precio y unidad de medida.
	9	Si el usuario desea, puede crear varios productos nuevos, volviendo al paso 7.
	10	Una vez que finalice la creación de productos, el actor puede salir del sistema o salir de la creación de productos, volviendo al paso 5.
	11	El usuario puede buscar productos para modificar o eliminar.
	12	Una vez seleccionado el producto, puede hacer clic en el botón de modificar o eliminar que muestra la pantalla respectiva.
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
	1	El sistema valida el usuario y contraseña, y en caso de que no sean correctos se desplegará un mensaje indicando que existió un error.

Tabla N° 4: Detalle CU “Administrar Producto”

Nombre	Administrar Proveedor	
Descripción	Permite crear nuevos proveedor al sistema.	
Actor	Jefe de Almacén	
Precondición	El actor debe ser colaborador de la empresa. El actor debe de tener un usuario, contraseña y permisos de acceso.	
Secuencia Normal	Paso	Acción
	1	El actor debe de acceder a la pantalla de logueo.
	2	El actor debe autenticarse con usuario y clave.
	3	El sistema valida credenciales válidas en caso que no sean correcto entonces regresar al paso 2.
	4	Se valida el perfil de usuario que accede al sistema.
	5	Se ingresa a la pantalla respectiva.
	6	El actor selecciona la opción para crear proveedor.
	7	Se despliega la pantalla para crear proveedor.
	8	Se crea un nuevo proveedor, ingresando nombre, dirección y RUC.
	9	Si el usuario desea, puede crear varios proveedores nuevos, volviendo al paso 7.
10	Una vez que finalice la creación de proveedores, el actor puede salir del sistema o salir de la creación de proveedores, volviendo al paso 5.	
	11	El usuario puede buscar proveedores para modificar o eliminar.
	12	Una vez seleccionado el proveedor, puede hacer clic en el botón de modificar o eliminar que muestra la pantalla respectiva.
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
	1	El sistema valida el usuario y contraseña, y en caso de que no sean correctos se desplegará un mensaje indicando que existió un error.

Tabla N° 5: Detalle CU “Administrar Proveedor

Nombre	Registrar Venta	
Descripción	Permite al vendedor registrar la venta de ropa y accesorios	
Actor	Vendedor	
Precondición	El actor debe ser colaborador de la empresa. El actor debe de tener un usuario, contraseña y permisos de acceso.	
Secuencia Normal	Paso	Acción
	1	Se muestra una interfaz en la que el vendedor debe seleccionar el tipo de documento Boleta o Factura.
	2	Se procede a buscar al Cliente
	3	Se procede a buscar al producto por descripción.
	4	Una vez seleccionado un producto se puede registrar la cantidad del producto
	5	Si el usuario desea, puede agregar varios productos, volviendo al paso 3.
	6	Al finalizar la venta, el actor puede salir del sistema o regresar a la interfaz inicial.
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
		N/A

Tabla N° 6: Detalle CU “Registrar Venta”

Nombre	Registrar Venta por Internet	
Descripción	Permite al Cliente realizar el pedido de ropa y accesorios y de ser el caso concretar la venta.	
Actor	Cliente	
Precondición		
Secuencia Normal	Paso	Acción
	1	Se consulta el catálogo de los productos (ropa y/o accesorios).
	2	Se selecciona un producto y se añade al carrito de pedido.
	3	Una vez seleccionado el producto en el carrito de pedido se puede registrar la cantidad a pedir.
	4	Si el usuario desea, puede agregar varios productos, volviendo al paso 2.
	5	Al finalizar el pedido, el cliente hace clic en el botón comprar.
	6	Haciendo clic en comprar el software de pasarela de pago se encarga de la validación de las credenciales en la entidad financiera y retorna un parámetro de conformidad o inconformidad.
7	Si el parámetro que retorna es de conformidad, se registra la venta en la boutique y se programa la entrega.	
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
		N/A

Tabla N° 7: Detalle CU “Registrar Venta”

Nombre	Registrar Nota de Ingreso	
Descripción	Permite al jefe de almacén registrar nota de ingreso de ropa y accesorios	
Actor	Jefe de Almacén	
Precondición	El actor debe ser colaborador de la empresa. El actor debe de tener un usuario, contraseña y permisos de acceso.	
Secuencia Normal	Paso	Acción
	1	Se muestra una interfaz en la que el vendedor debe seleccionar el tipo de movimiento (Ingreso por Compra), fecha movimiento, numero de documento de referencia.
	2	Se procede a seleccionar el proveedor
	3	Se procede a hacer clic en el botón adicionar producto, el cual muestra la pantalla para buscar producto por descripción, una vez seleccionado se adiciona el producto.
	4	Si el usuario desea, puede agregar varios productos, volviendo al paso 3.
	5	Al finalizar el registro de la nota de ingreso, el actor puede salir del sistema o regresar a la interfaz inicial.
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
		N/A

Tabla N° 8: Detalle CU “Registrar Nota de Ingreso”

Nombre	Registrar Nota de Salida	
Descripción	Permite al jefe de almacén registrar ropa y accesorios.	
Actor	Jefe de Almacén	
Precondición	El actor debe ser colaborador de la empresa. El actor debe de tener un usuario, contraseña y permisos de acceso.	
Secuencia Normal	Paso	Acción
	1	Se muestra una interfaz en la que el vendedor debe seleccionar el tipo de movimiento, fecha movimiento, numero de documento de referencia.
	2	Se procede a hacer clic en el botón adicionar producto, el cual muestra la pantalla para buscar producto por descripción, una vez seleccionado se adiciona el producto.
	3	Si el usuario desea, puede agregar varios productos, volviendo al paso 3.
	4	Al finalizar el registro de la nota de salida, el actor puede salir del sistema o regresar a la interfaz inicial.
Post-condición	La nueva información ha sido almacenada en la Base de Datos.	
Excepciones	Paso	Acción
		N/A

Tabla N° 9: Detalle CU “Registrar Nota de Salida”

2.7. Diagrama de clases

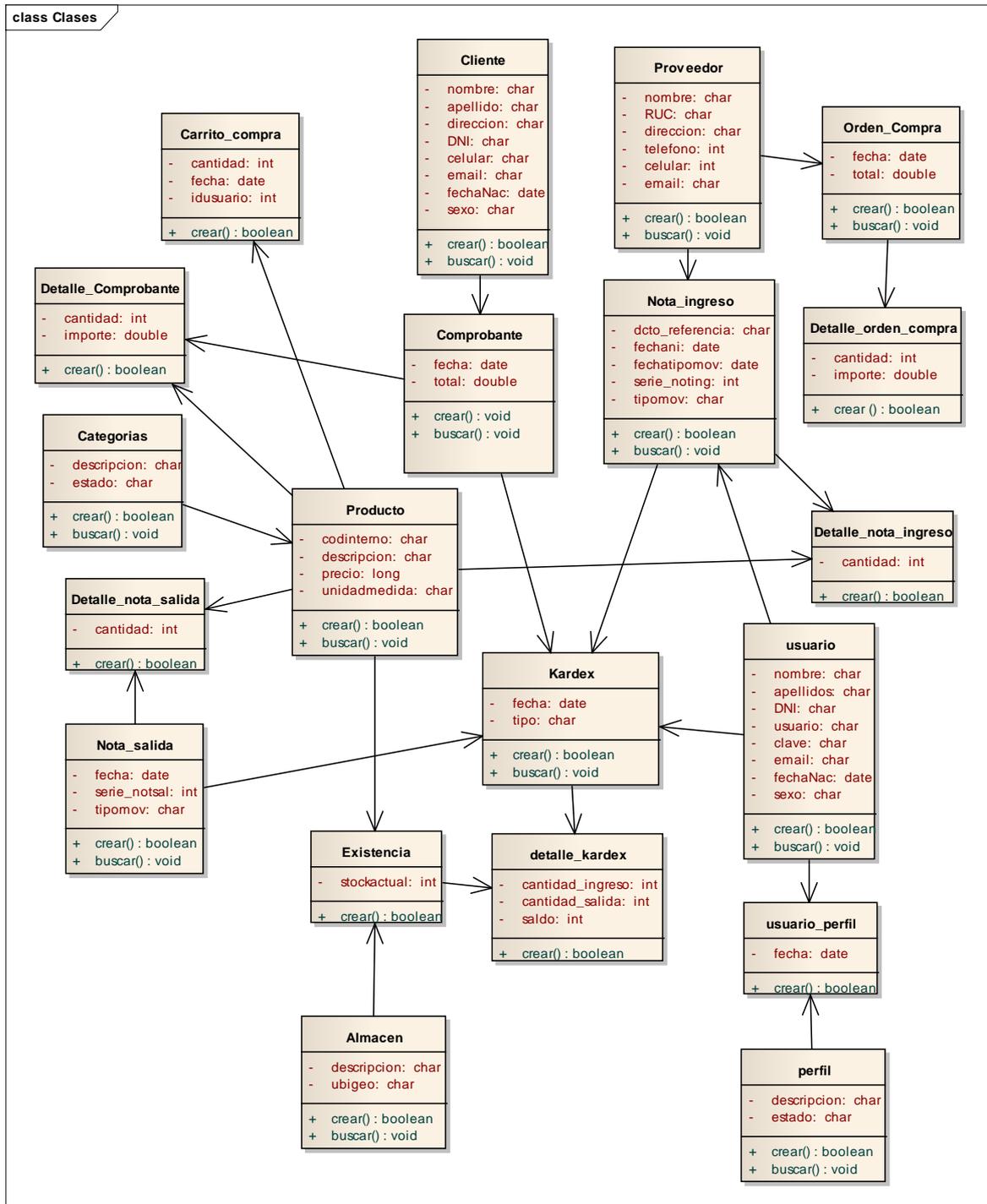


Figura N° 10: Diagrama de clases

3. ELABORACIÓN

3.1. Prototipo de Interfaz Usuario

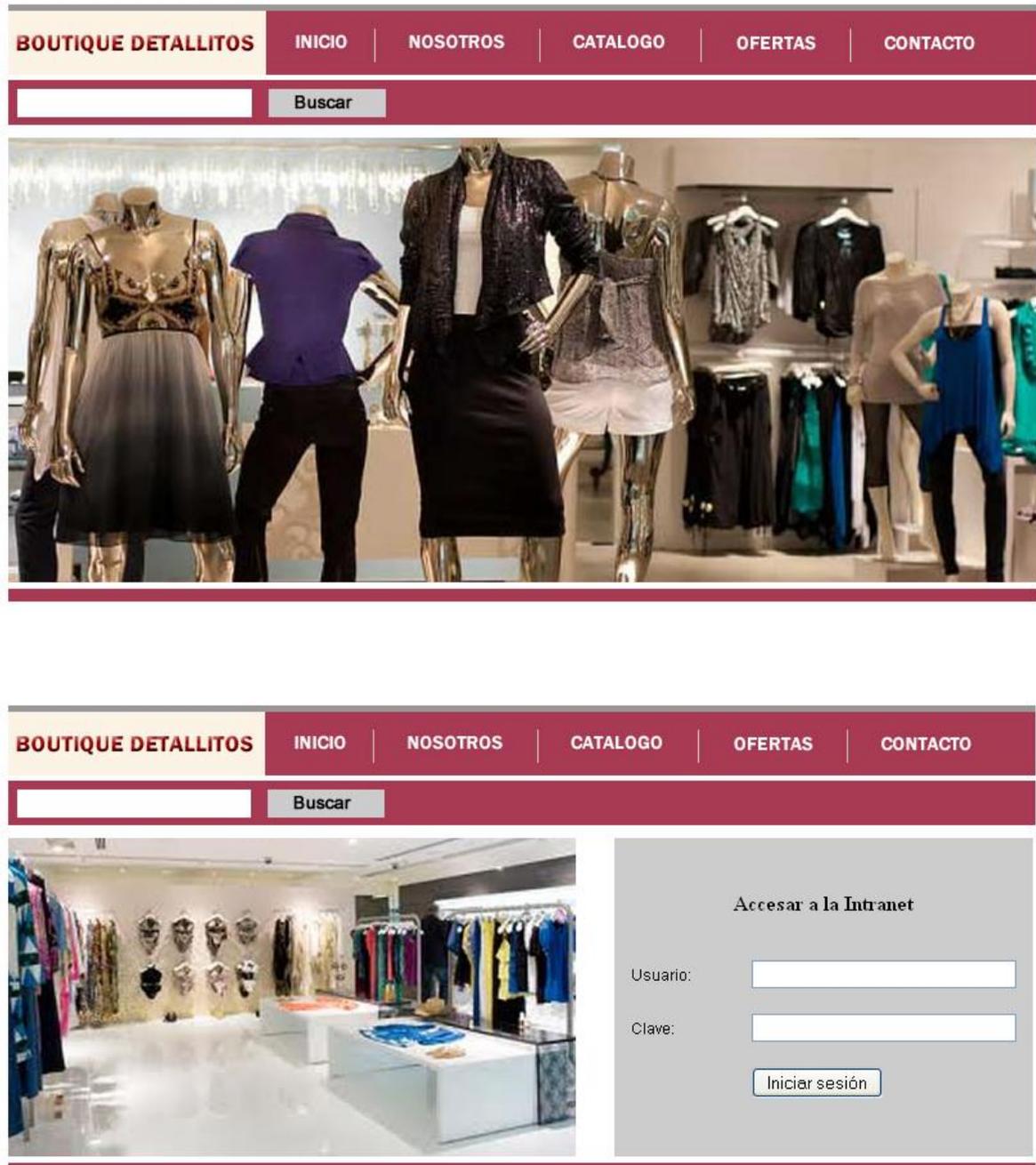


Figura N° 11: Autenticar usuario

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Buscar

GESTIONAR PROVEEDOR

PROVEEDOR: **Buscar**  **Agregar**

Código	nombre	dirección	RUC	Acciones
				 

Figura N° 12: Registrar proveedor

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Buscar

GESTIONAR PRODUCTO

PRODUCTOS: **Buscar**  **Agregar**

Código inter.	descripción	unidad de medida	precio	tipo producto	Acciones
					 

Figura N° 13: Registrar producto

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

KARDEX DE PRODUCTOS

Producto:

Codigo	Descripción	Fecha inicial:
		01/06/2014
		Fecha final: 18/06/2014

Fecha Movimient.	Tipo Movimiento	Documento	Cantidad ingreso	Cantidad salida	Saldo

Movimientos: Existencias:

Figura N° 14: Kardex de productos

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Nota de ingreso

Numero: Movimiento: Nrodoc.ref:
 Fecha: Fechamov:

Proveedor:

Codigo	Descripción	Cantidad
120	Pantalones Jeans Lee	6

Figura N° 15: Nota de ingreso

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Nota de salida

Numero: Movimiento: Nro doc.ref.
 Fecha: Fechamov.

Codigo	Descripción	Cantidad
0000056	Polos	6

Figura N° 16: Nota de salida

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Buscar

ROPA

- Blusas
- Polos
- Pantalones
- Shorts
- Ropa de baño
- Casacas

Damas

Aquí encontrará todas las colecciones de mujer de moda.
Esta categoría incluye todos los elementos básicos de su guardarropa y mucho más: zapatos, pantalones vaqueros, accesorios, camisetas impresas, vestidos femeninos, las mujeres!

NUEVO



Camisetas manga corta
Manga corta con escote alto. Material suave y elástico para un ajuste cómodo. Accesorios con un sombrero de paja y ya está listo para el verano!

S/. 19

[Añadir al carrito](#)

[Más](#)

NUEVO



Blusa
Blusa de manga corta con manga femenina

S/. 32

[Añadir al carrito](#)

[Más](#)

Figura N° 17: Catálogo de productos

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Buscar

ROPA

- Blusas
- Polos
- Pantalones
- Shorts
- Ropa de baño
- Casacas

Damas

Aquí encontrará todas las colecciones de mujer de moda.
Esta categoría incluye todos los elementos básicos de su guardarropa y mucho más: zapatos, pantalones vaqueros, accesorios, camisetas impresas, vestidos femeninos, las mujeres!

PRODUCTOS EN SU CARRITO Su carrito contiene 2 productos

01. Resumen | 02. Iniciar sesión | 03. Envío | 04. Pago

producto	Descripción	Disp.	Precio unitario	Cant.	Total	
	Vestido	In stock	S/. 36	1	S/. 36	
	Polo	In stock	S/. 19	1	S/. 19	
					S/. 55	
TOTAL					S/. 55	

< Continuar la compra Ir a la caja >

Figura N° 18: Carrito de compras

BOUTIQUE DETALLITOS | INICIO | NOSOTROS | CATALOGO | OFERTAS | CONTACTO

Buscar

ROPA

- Blusas
- Polos
- Pantalones
- Shorts
- Ropa de baño
- Casacas

Damas

Aquí encontrará todas las colecciones de mujer de moda. Esta categoría incluye todos los elementos básicos de su guardarropa y mucho más: zapatos, pantalones vaqueros accesorios, camisetas impresas, vestidos formales, las mujeres!



PRODUCTOS EN SU CARRITO Se están mostrando 2 productos

01. Resumen | **02. Iniciar sesión** | 03. Envío | 04. Pago

Inicia sesión

Ingresa tus datos para iniciar el proceso de compra

usuario

clave

[¿Olvidaste tu clave?](#)

Ingresar

Inicio de sesión

4. CONSTRUCCIÓN

4.1. Diagrama de colaboración

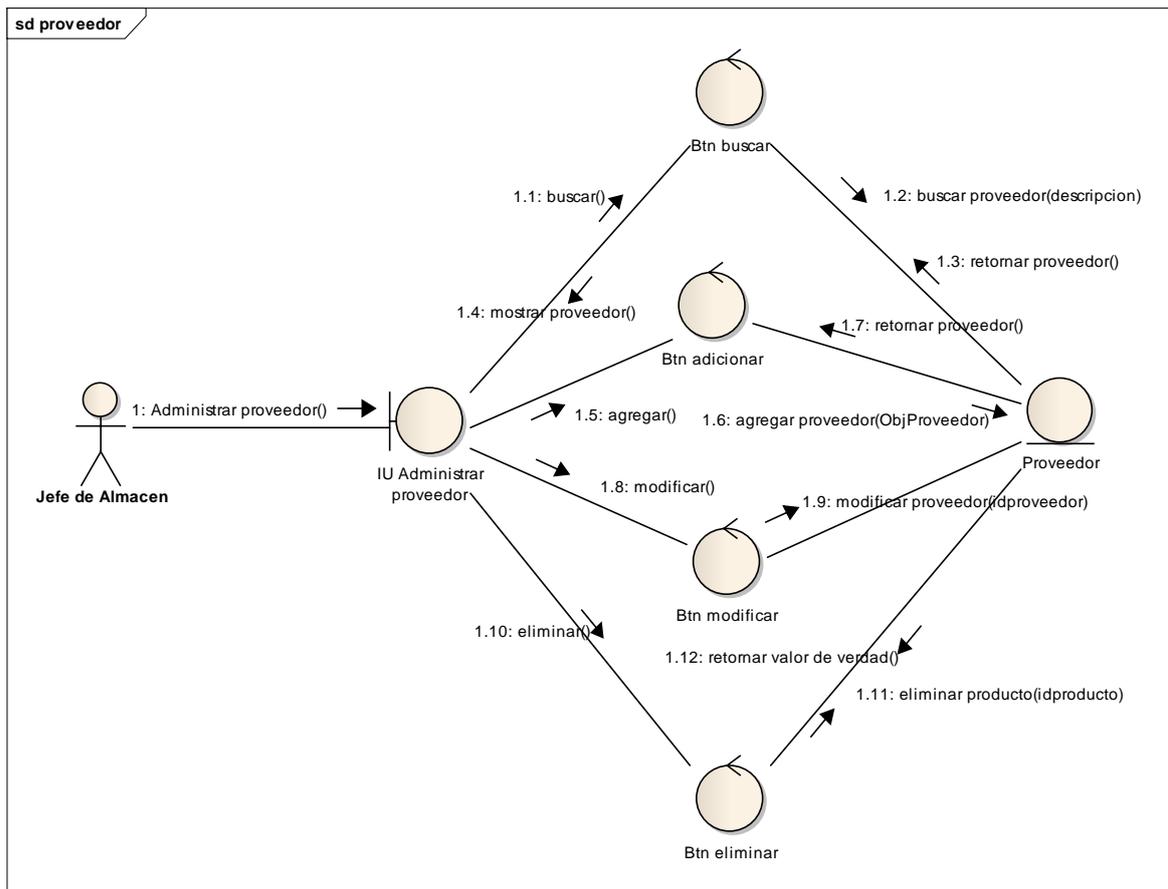


Figura N° 19: Diagrama de robustez de gestionar proveedor

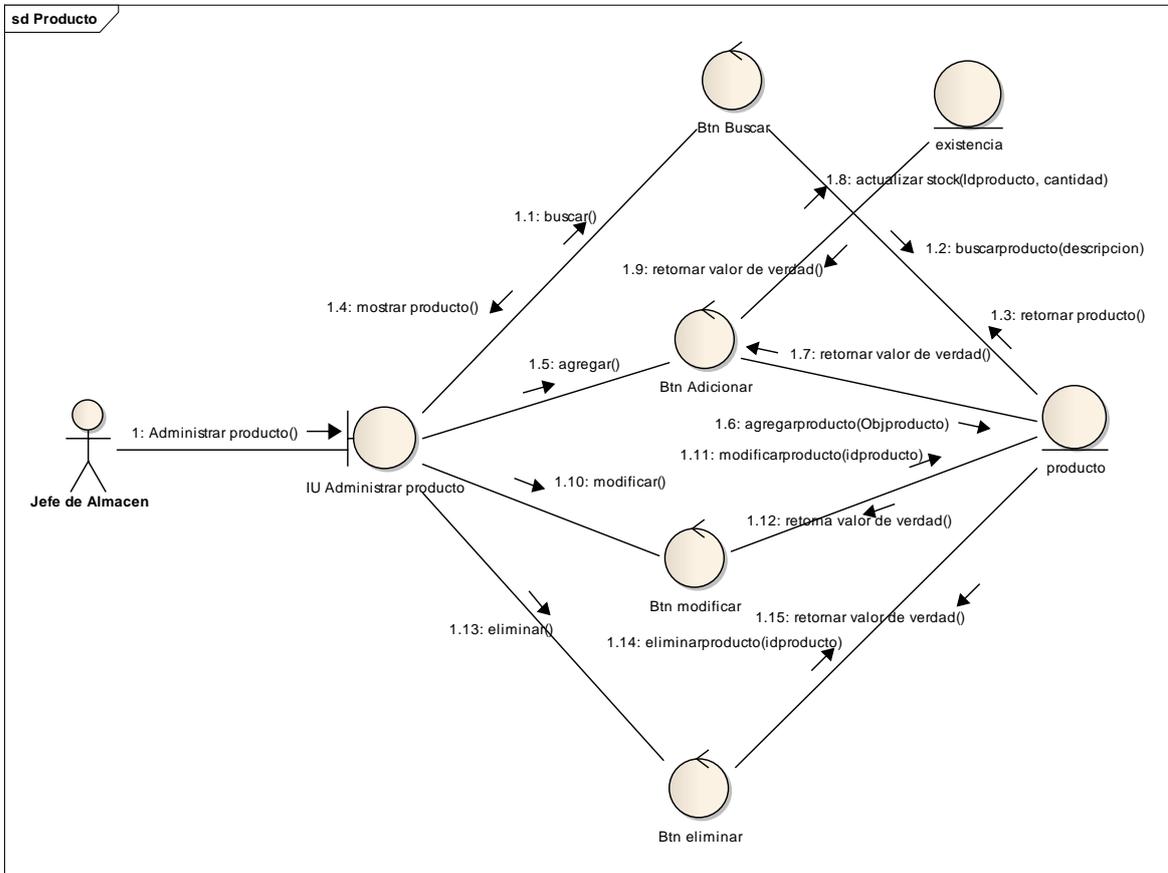


Figura N° 20: Diagrama de robustez de gestionar producto

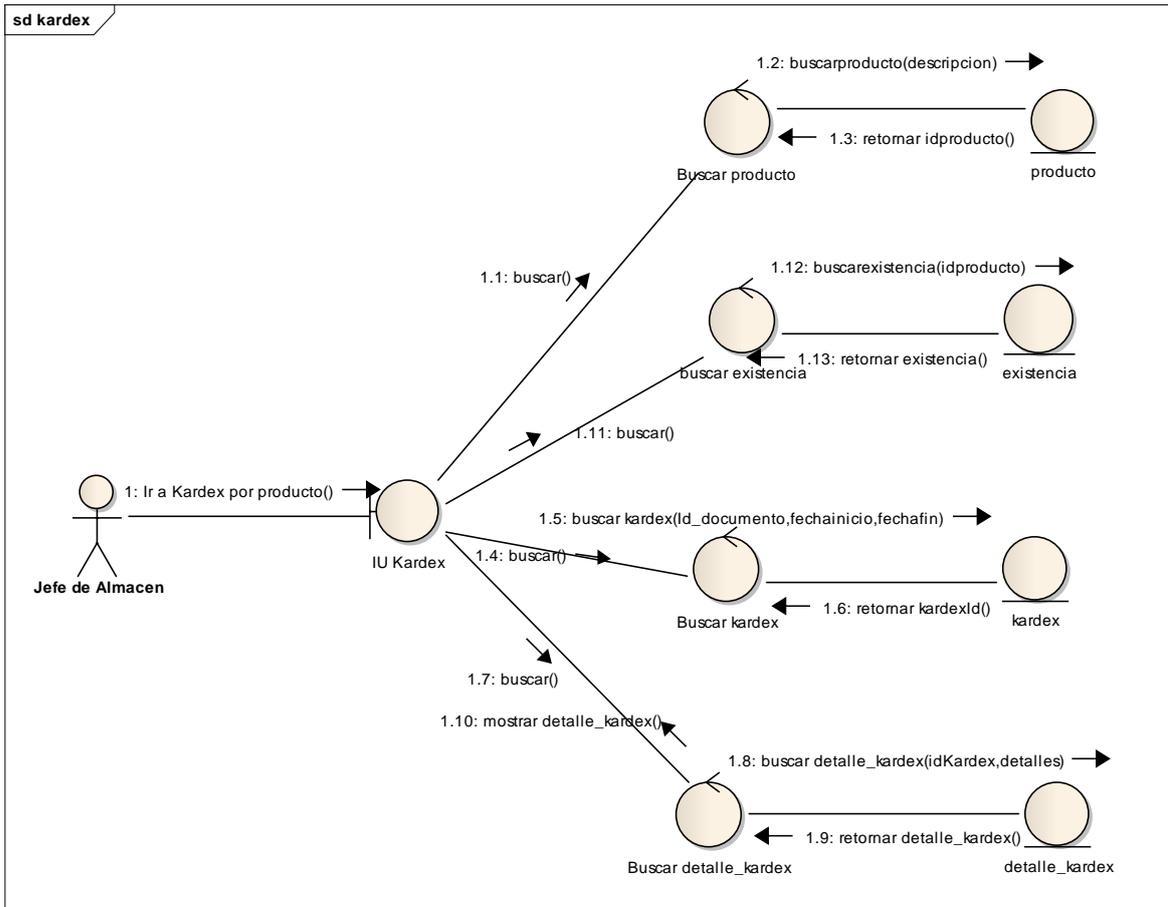


Figura N° 21: Diagrama de robustez kardex por producto

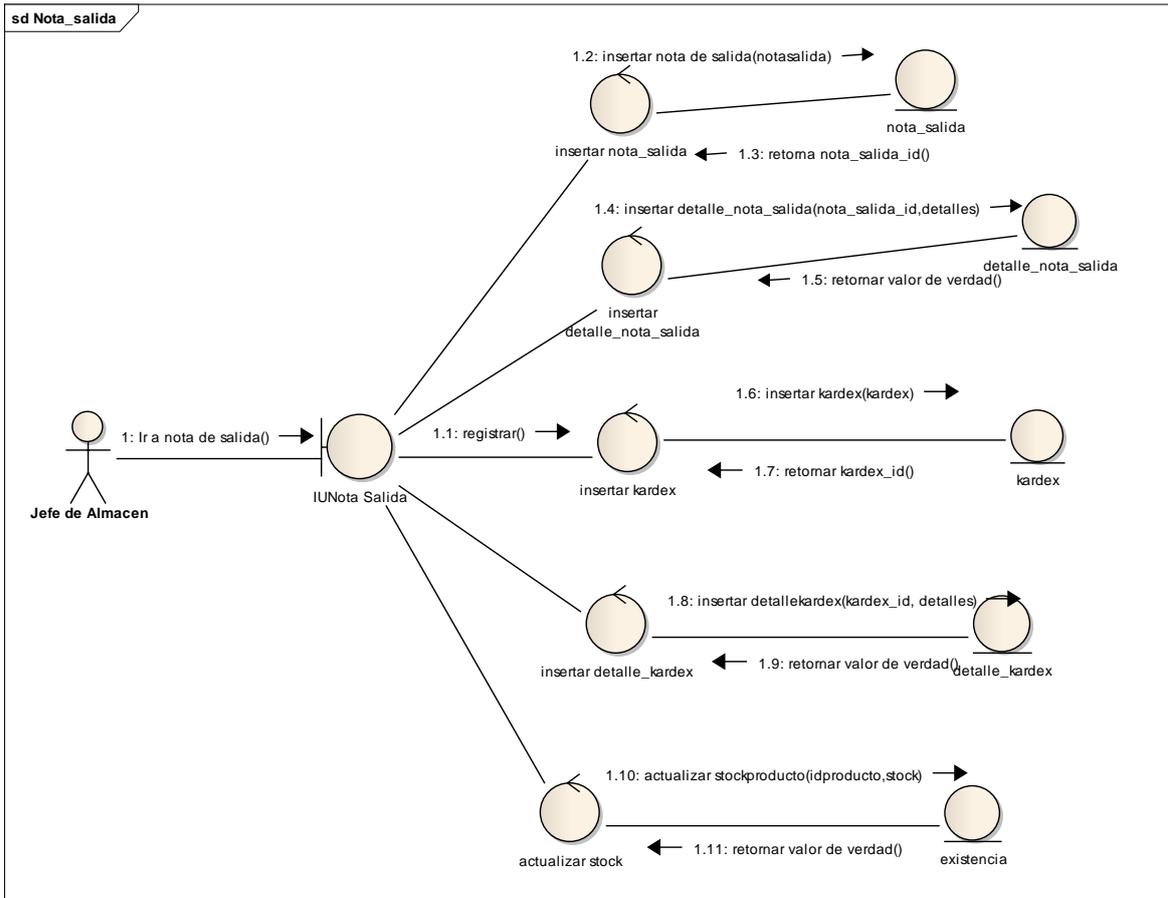


Figura N° 22: Diagrama de robustez de nota de salida

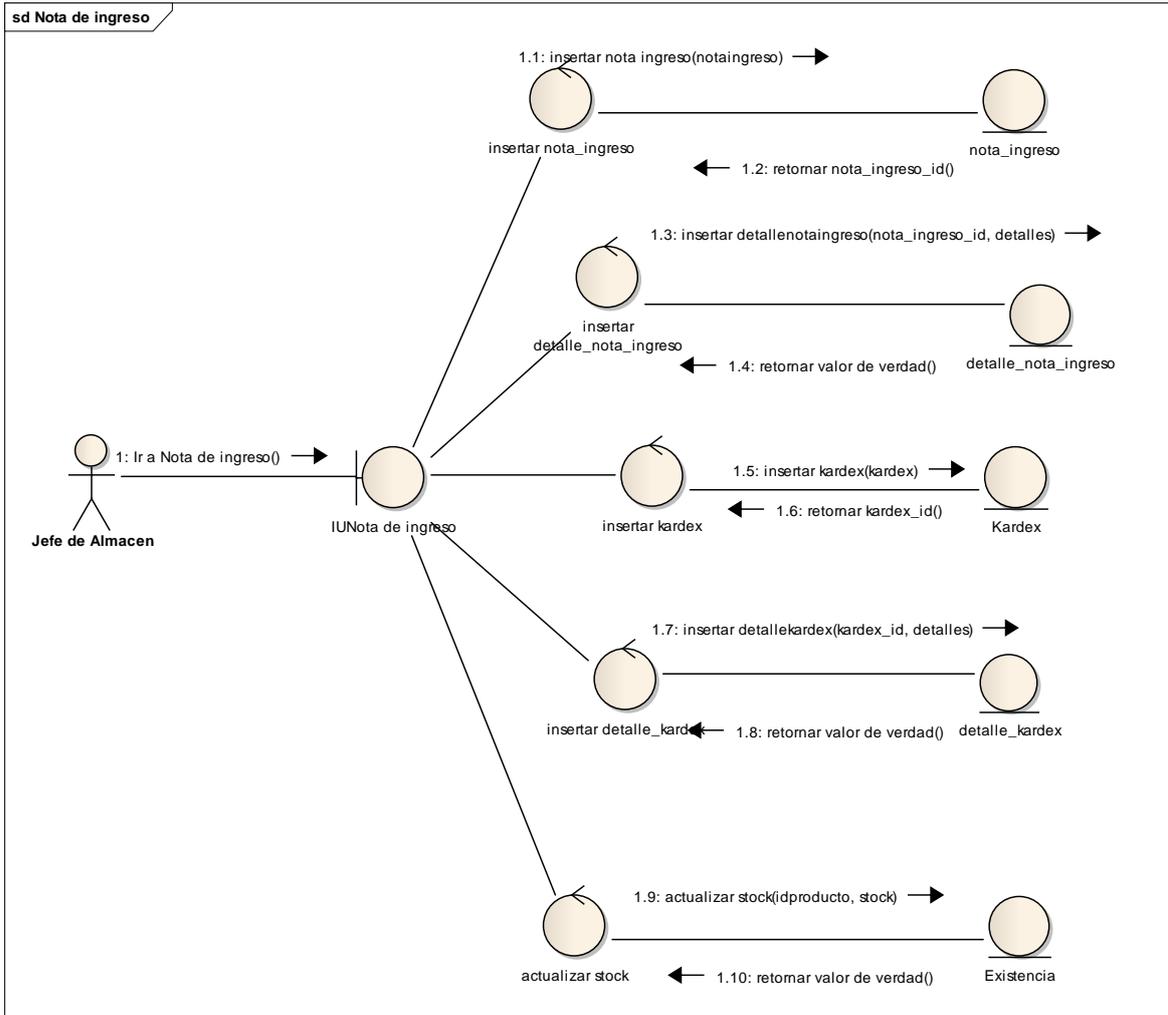


Figura N° 23: Diagrama de robustez nota de ingreso

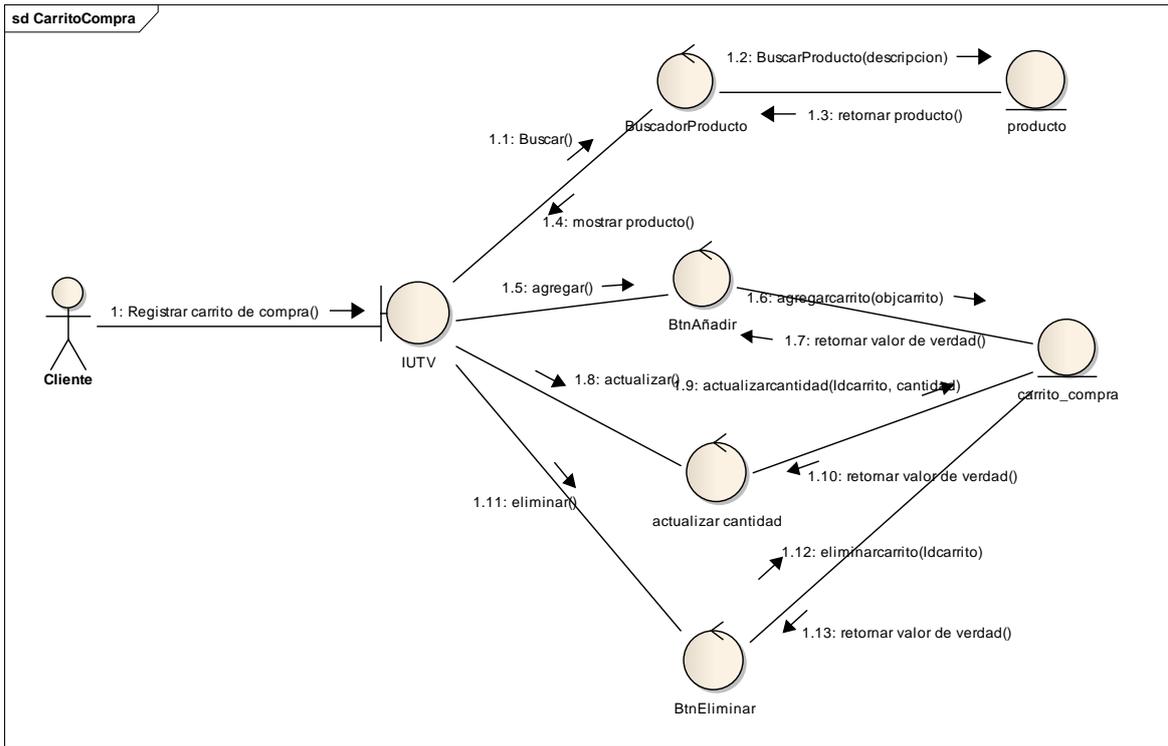


Figura N° 24: Diagrama de robustez carrito de compra

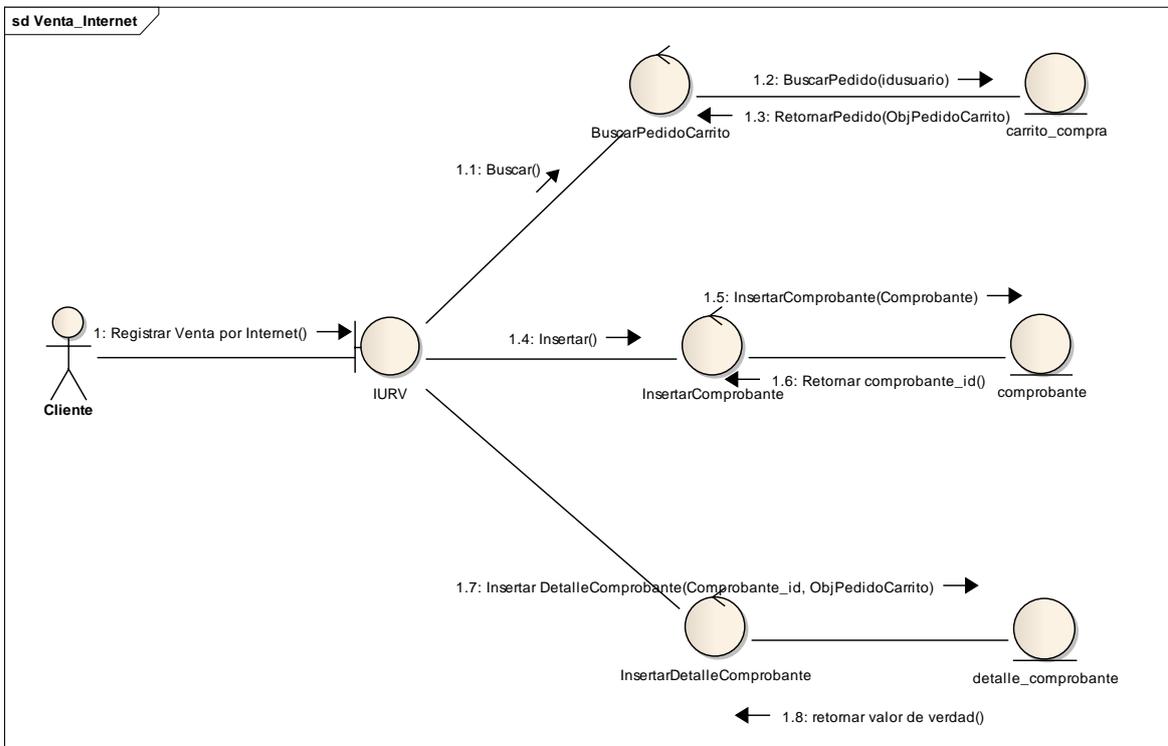


Figura N° 25: Diagrama de robustez Ventas por internet

4.2. Diagrama de secuencia

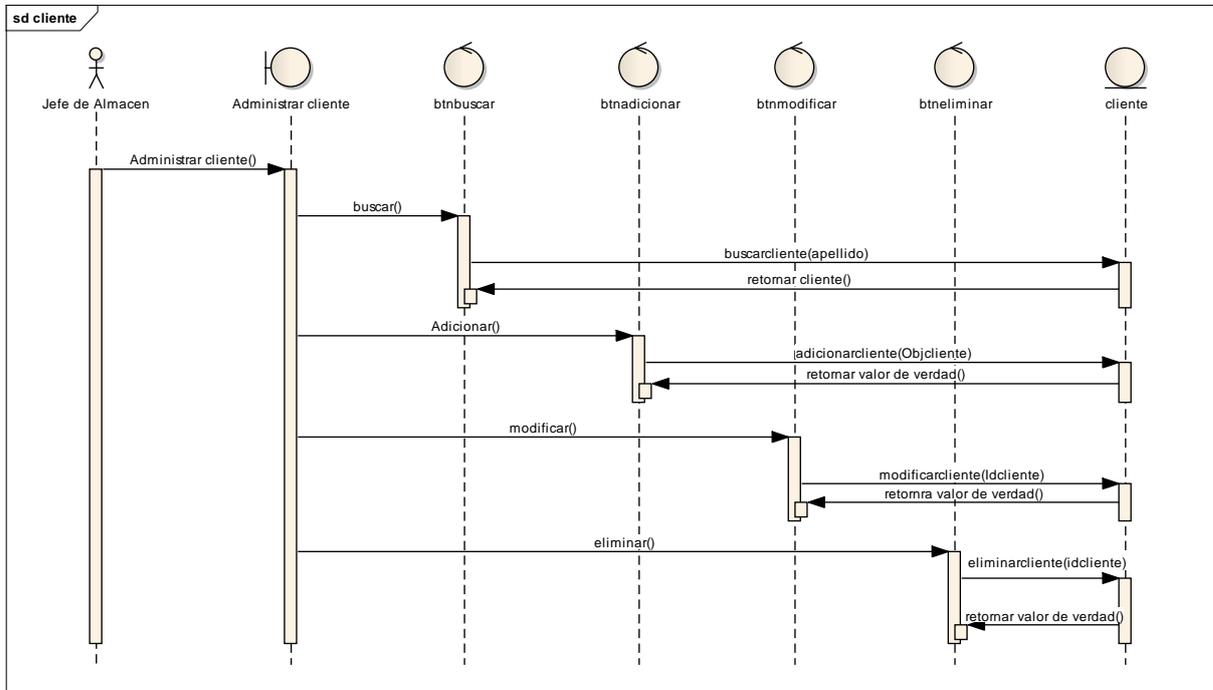


Figura N° 26: Diagrama de Secuencia Gestionar Cliente

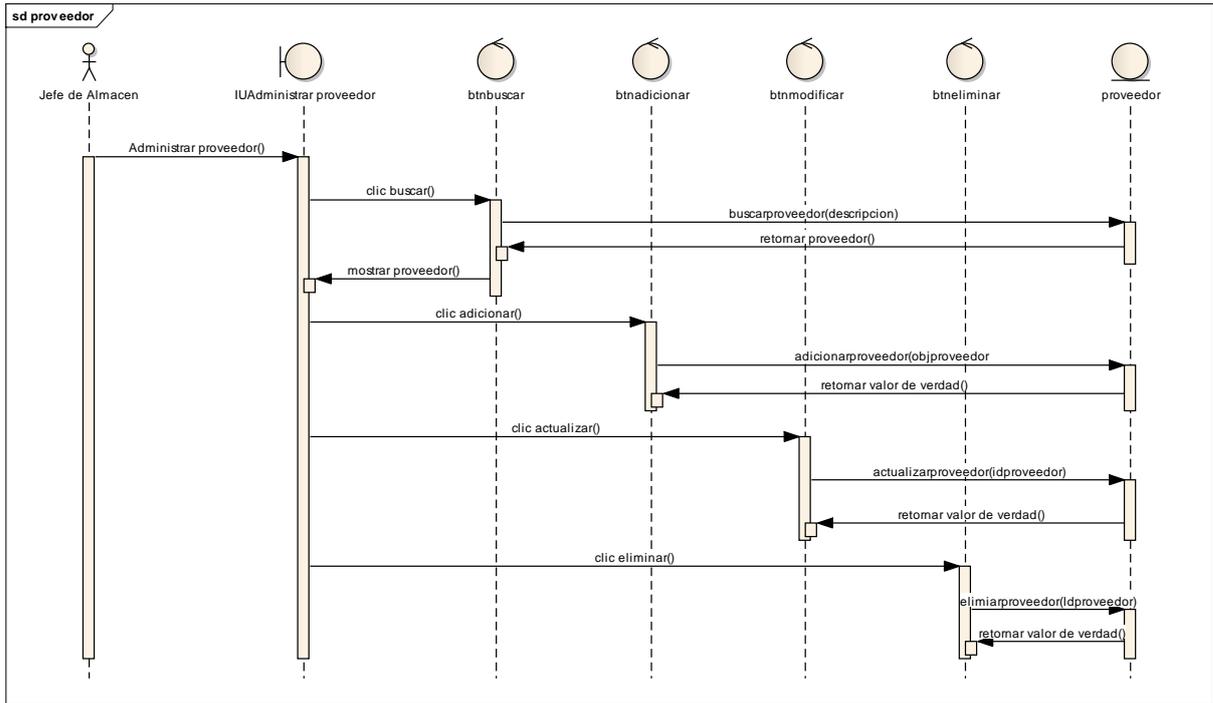


Figura N° 27: Diagrama de Secuencia Gestionar Proveedor

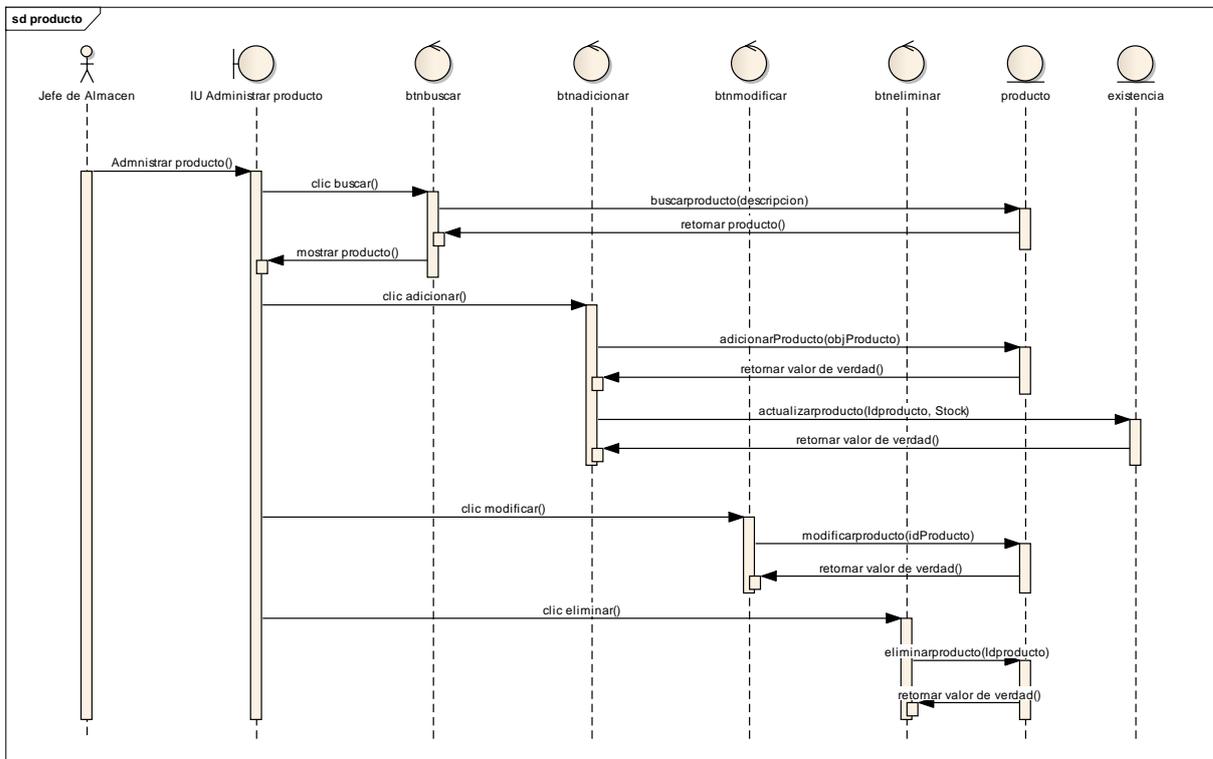


Figura N° 28: Diagrama de Secuencia Gestionar Producto

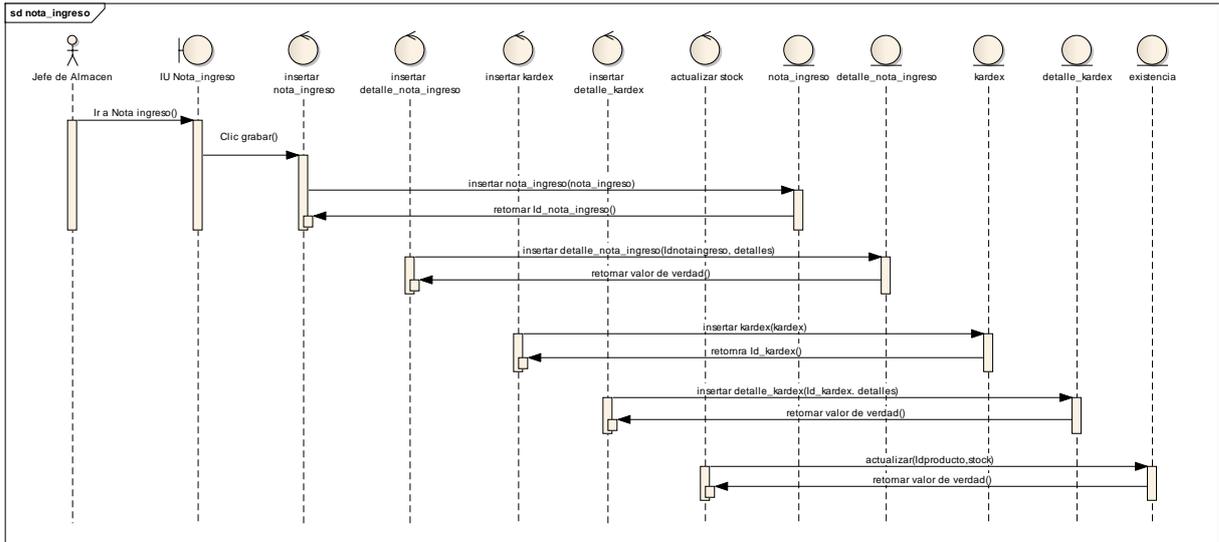


Figura N° 29: Diagrama de Secuencia Nota de ingreso

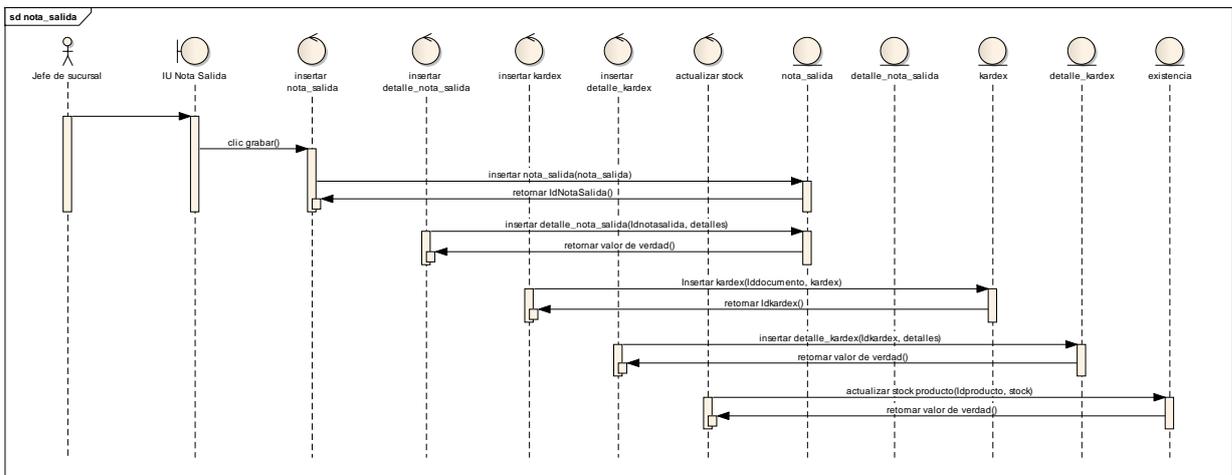


Figura N° 30: Diagrama de Secuencia Nota de salida

4.3. Modelo de despliegue

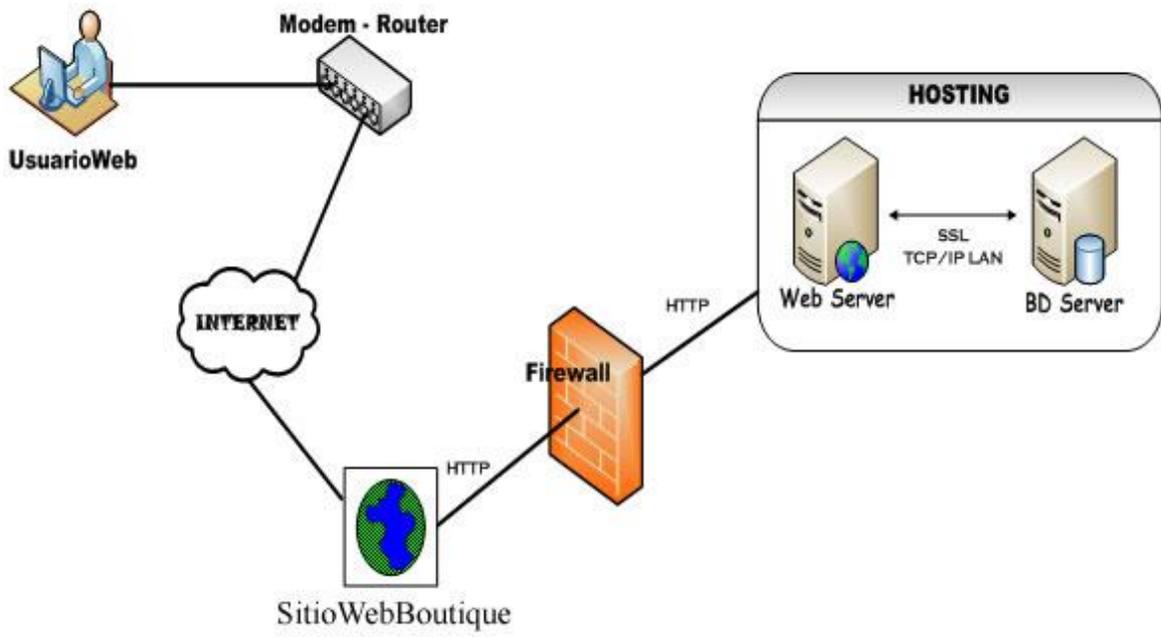


Figura N° 31: Modelo de despliegue

4.4. Modelo de componentes

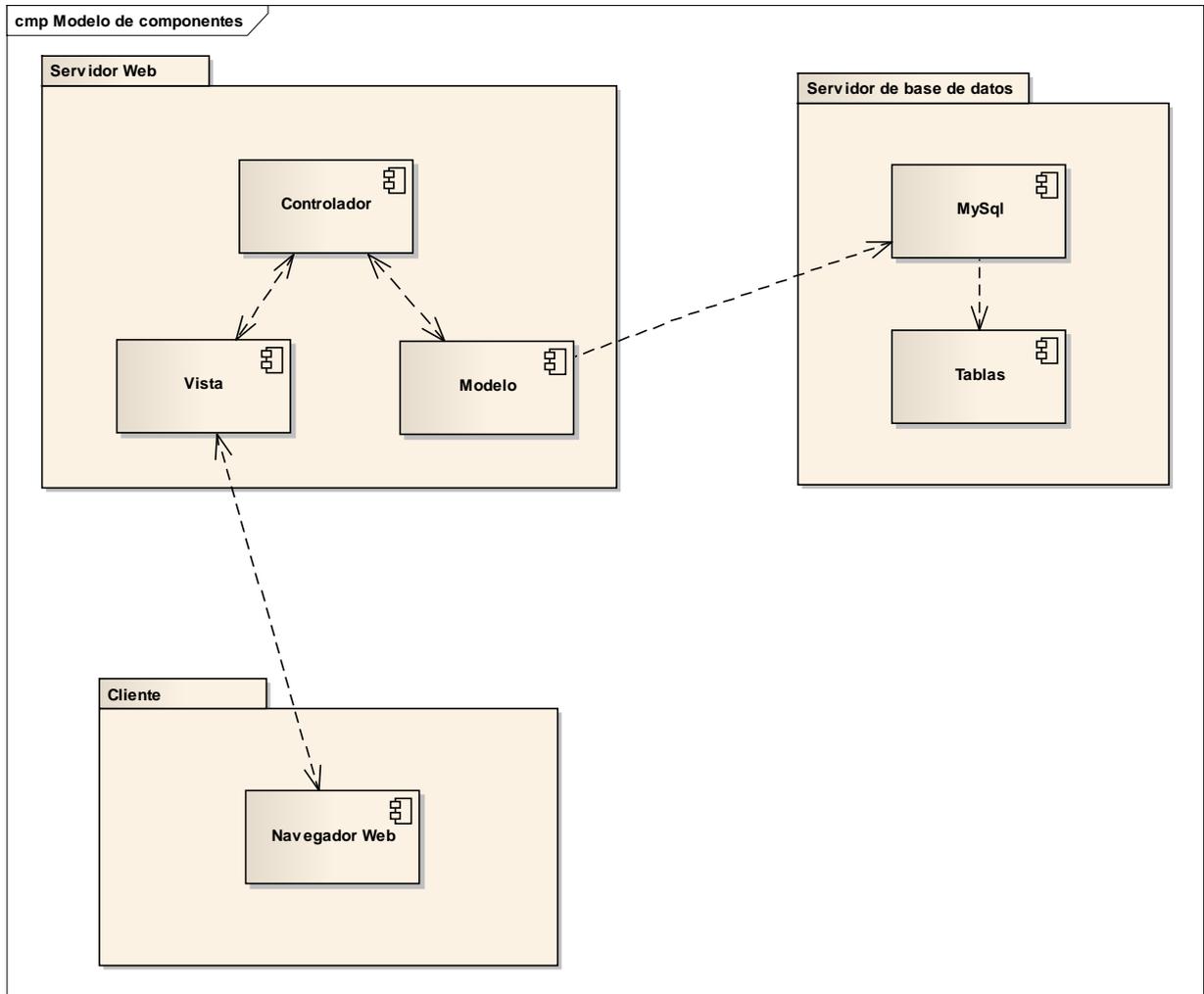


Figura N° 32: Modelo de componentes

4.6. Modelo físico

```
CREATE TABLE Almacen (  
    descripcion char(10),  
    ubigeo char(10),  
    almacenID int NOT NULL  
);  
  
CREATE TABLE Carrito_compra (  
    cantidad int,  
    fecha date,  
    carrito_compraID int NOT NULL,  
    productoID int  
);  
  
CREATE TABLE Categorías (  
    descripcion char(10),  
    estado char(10),  
    categoríasID int NOT NULL  
);  
  
CREATE TABLE Cliente (  
    nombre char(10),  
    apellido char(10),  
    dirección char(10),  
    DNI char(10),
```

```
        celular char(10),  
        email char(10),  
        fechaNac date,  
        sexo char(10),  
        clienteID int NOT NULL
```

```
);
```

```
CREATE TABLE Comprobante (
```

```
        fecha date,  
        total real,  
        comprobanteID int NOT NULL,  
        clienteID Integer
```

```
);
```

```
CREATE TABLE Detalle_Comprobante (
```

```
        cantidad int,  
        importe real,  
        detalle_ComprobanteID int NOT NULL,  
        comprobanteID Integer,  
        productoID Integer
```

```
);
```

```
CREATE TABLE detalle_kardex (
```

```
        cantidad_ingreso int,  
        cantidad_salida int,
```

```

        saldo int,

        detalle_kardexID int NOT NULL,

        kardexID Integer,

        existenciaID Integer

    );

CREATE TABLE Detalle_nota_ingreso (

    cantidad int,

    detalle_nota_ingresoID int NOT NULL,

    nota_ingresoID Integer,

    productoID Integer

);

CREATE TABLE Detalle_nota_salida (

    cantidad int,

    detalle_nota_salidaID int NOT NULL,

    nota_salidaID Integer,

    productoID Integer

);

CREATE TABLE Detalle_orden_compra (

    cantidad int,

    importe real,

    detalle_orden_compraID int NOT NULL,

    orden_CompraID int

```

);

```
CREATE TABLE Documento (  
    fecha date,  
    tipodcto char(10),  
    documentoID int NOT NULL,  
    comprobanteID Integer,  
    nota_ingresoID Integer,  
    nota_salidaID Integer
```

);

```
CREATE TABLE Existencia (  
    stockactual int,  
    existenciaID int NOT NULL,  
    almacenID Integer,  
    productoID Integer
```

);

```
CREATE TABLE Interfaz1 (  
    interfaz1ID int NOT NULL
```

);

```
CREATE TABLE Kardex (  
    fecha date,  
    tipo char(10),  
    kardexID int NOT NULL,
```

```

        documentoID Integer
    );

CREATE TABLE Nota_ingreso (
    dcto_referencia char(10),
    fechani date,
    fechatipomov date,
    serie_noting int,
    tipomov char(10),
    nota_ingresoID int NOT NULL,
    proveedorID Integer
);

CREATE TABLE Nota_salida (
    fecha date,
    serie_notsal int,
    tipomov char(10),
    nota_salidaID int NOT NULL
);

CREATE TABLE Orden_Compra (
    fecha date,
    total real,
    orden_CompraID int NOT NULL,
    proveedorID int

```

);

CREATE TABLE perfil (

descripcion char(10),

estado char(10),

perfilID int NOT NULL

);

CREATE TABLE Producto (

codinterno char(10),

descripcion char(10),

precio long,

unidadmedida char(10),

productoID int NOT NULL,

categoriasID int,

tipo_ProductoID Integer

);

CREATE TABLE Proveedor (

nombre char(10),

RUC char(10),

direccion char(10),

telefono int,

celular int,

email char(10),

```
        proveedorID int NOT NULL
    );

CREATE TABLE usuario (
    nombre char(10),
    apellidos char(10),
    DNI char(10),
    usuario char(10),
    clave char(10),
    email char(10),
    fechaNac date,
    sexo char(10),
    usuarioID int NOT NULL
);

CREATE TABLE usuario_perfil (
    fecha date,
    usuario_perfilID int NOT NULL,
    formulariosID int,
    usuarioID int,
    perfilID int
);
```

ALTER TABLE Almacen ADD CONSTRAINT PK_Almacen

PRIMARY KEY CLUSTERED (almacenID);

ALTER TABLE Carrito_compra ADD CONSTRAINT PK_Carrito_compra

PRIMARY KEY CLUSTERED (carrito_compraID);

ALTER TABLE Categorias ADD CONSTRAINT PK_Categorias

PRIMARY KEY CLUSTERED (categoriasID);

ALTER TABLE Cliente ADD CONSTRAINT PK_Cliente

PRIMARY KEY CLUSTERED (clienteID);

ALTER TABLE Comprobante ADD CONSTRAINT PK_Comprobante

PRIMARY KEY CLUSTERED (comprobanteID);

ALTER TABLE Detalle_Comprobante ADD CONSTRAINT PK_Detalle_Comprobante

PRIMARY KEY CLUSTERED (detalle_ComprobanteID);

ALTER TABLE detalle_kardex ADD CONSTRAINT PK_detalle_kardex

PRIMARY KEY CLUSTERED (detalle_kardexID);

ALTER TABLE Detalle_nota_ingreso ADD CONSTRAINT PK_Detalle_nota_ingreso

PRIMARY KEY CLUSTERED (detalle_nota_ingresoID);

ALTER TABLE Detalle_nota_salida ADD CONSTRAINT PK_Detalle_nota_salida

PRIMARY KEY CLUSTERED (detalle_nota_salidaID);

ALTER TABLE Detalle_orden_compra ADD CONSTRAINT PK_Detalle_orden_compra

PRIMARY KEY CLUSTERED (detalle_orden_compraID);

ALTER TABLE Documento ADD CONSTRAINT PK_Documento

PRIMARY KEY CLUSTERED (documentoID);

```
ALTER TABLE Existencia ADD CONSTRAINT PK_Existencia
    PRIMARY KEY CLUSTERED (existenciaID);

ALTER TABLE Kardex ADD CONSTRAINT PK_Kardex
    PRIMARY KEY CLUSTERED (kardexID);

ALTER TABLE Nota_ingreso ADD CONSTRAINT PK_Nota_ingreso
    PRIMARY KEY CLUSTERED (nota_ingresoID);

ALTER TABLE Nota_salida ADD CONSTRAINT PK_Nota_salida
    PRIMARY KEY CLUSTERED (nota_salidaID);

ALTER TABLE Orden_Compra ADD CONSTRAINT PK_Orden_Compra
    PRIMARY KEY CLUSTERED (orden_CompraID);

ALTER TABLE perfil ADD CONSTRAINT PK_perfil
    PRIMARY KEY CLUSTERED (perfilID);

ALTER TABLE Producto ADD CONSTRAINT PK_Producto
    PRIMARY KEY CLUSTERED (productoID);

ALTER TABLE Proveedor ADD CONSTRAINT PK_Proveedor
    PRIMARY KEY CLUSTERED (proveedorID);

ALTER TABLE usuario ADD CONSTRAINT PK_usuario
    PRIMARY KEY CLUSTERED (usuarioID);

ALTER TABLE usuario_perfil ADD CONSTRAINT PK_usuario_perfil
    PRIMARY KEY CLUSTERED (usuario_perfilID);
```

```
ALTER TABLE Carrito_compra ADD CONSTRAINT FK_Carrito_compra_Producto
```

```
FOREIGN KEY (productoID) REFERENCES Producto (productoID);
```

```
ALTER TABLE Comprobante ADD CONSTRAINT FK_Comprobante_Cliente
```

```
FOREIGN KEY (clienteID) REFERENCES Cliente (clienteID);
```

```
ALTER TABLE Detalle_Comprobante ADD CONSTRAINT
```

```
FK_Detalle_Comprobante_Comprobante
```

```
FOREIGN KEY (comprobanteID) REFERENCES Comprobante (comprobanteID);
```

```
ALTER TABLE Detalle_Comprobante ADD CONSTRAINT
```

```
FK_Detalle_Comprobante_Producto
```

```
FOREIGN KEY (productoID) REFERENCES Producto (productoID);
```

```
ALTER TABLE detalle_kardex ADD CONSTRAINT FK_detalle_kardex_Kardex
```

```
FOREIGN KEY (kardexID) REFERENCES Kardex (kardexID);
```

```
ALTER TABLE detalle_kardex ADD CONSTRAINT FK_detalle_kardex_Existencia
```

```
FOREIGN KEY (existenciaID) REFERENCES Existencia (existenciaID);
```

```
ALTER TABLE Detalle_nota_ingreso ADD CONSTRAINT
```

```
FK_Detalle_nota_ingreso_Nota_ingreso
```

```
FOREIGN KEY (nota_ingresoID) REFERENCES Nota_ingreso (nota_ingresoID);
```

```
ALTER TABLE Detalle_nota_ingreso ADD CONSTRAINT
```

```
FK_Detalle_nota_ingreso_Producto
```

```
FOREIGN KEY (productoID) REFERENCES Producto (productoID);
```

```
ALTER TABLE Detalle_nota_salida ADD CONSTRAINT
```

```
FK_Detalle_nota_salida_Nota_salida
```

```
FOREIGN KEY (nota_salidaID) REFERENCES Nota_salida (nota_salidaID);
```

ALTER TABLE Detalle_nota_salida ADD CONSTRAINT
FK_Detalle_nota_salida_Producto

FOREIGN KEY (productoID) REFERENCES Producto (productoID);

ALTER TABLE Detalle_orden_compra ADD CONSTRAINT
FK_Detalle_orden_compra_Orden_Compra

FOREIGN KEY (orden_CompraID) REFERENCES Orden_Compra
(orden_CompraID);

ALTER TABLE Documento ADD CONSTRAINT FK_Documento_Comprobante

FOREIGN KEY (comprobanteID) REFERENCES Comprobante (comprobanteID);

ALTER TABLE Documento ADD CONSTRAINT FK_Documento_Nota_ingreso

FOREIGN KEY (nota_ingresoID) REFERENCES Nota_ingreso (nota_ingresoID);

ALTER TABLE Documento ADD CONSTRAINT FK_Documento_Nota_salida

FOREIGN KEY (nota_salidaID) REFERENCES Nota_salida (nota_salidaID);

ALTER TABLE Existencia ADD CONSTRAINT FK_Existencia_Almacen

FOREIGN KEY (almacenID) REFERENCES Almacen (almacenID);

ALTER TABLE Existencia ADD CONSTRAINT FK_Existencia_Producto

FOREIGN KEY (productoID) REFERENCES Producto (productoID);

ALTER TABLE Kardex ADD CONSTRAINT FK_Kardex_Documento

FOREIGN KEY (documentoID) REFERENCES Documento (documentoID);

ALTER TABLE Nota_ingreso ADD CONSTRAINT FK_Nota_ingreso_Proveedor

FOREIGN KEY (proveedorID) REFERENCES Proveedor (proveedorID);

ALTER TABLE Orden_Compra ADD CONSTRAINT FK_Orden_Compra_Proveedor

FOREIGN KEY (proveedorID) REFERENCES Proveedor (proveedorID);

```
ALTER TABLE Producto ADD CONSTRAINT FK_Producto_Categorias
    FOREIGN KEY (categoriasID) REFERENCES Categorias (categoriasID);

ALTER TABLE usuario_perfil ADD CONSTRAINT FK_usuario_perfil_perfil
    FOREIGN KEY (perfilID) REFERENCES perfil (perfilID);

ALTER TABLE usuario_perfil ADD CONSTRAINT FK_usuario_perfil_usuario
    FOREIGN KEY (usuarioID) REFERENCES usuario (usuarioID);
```

CAPÍTULO III: DISCUSIÓN

3.1. CONTRASTACIÓN DE LA HIPÓTESIS

3.1.1. Identificación de Variables e Indicadores

La contrastación de la hipótesis sólo se podría realizar una vez que el sistema sea puesto en línea. Lo que ahora se presenta es una simulación para poder establecer cómo podrá la hipótesis responder a la situación problema.

La prueba de hipótesis del proyecto será empleando el Método de Pre-Test y Post-Test.

- **Indicador: *Satisfacción del usuario final***

Se mide el nivel de satisfacción del usuario final con el proceso actual y con el sistema informático web propuesto. Esta medición se realiza mediante la aplicación de encuesta a los principales involucrados en la gestión de ventas.

Pre-Test (S1): Medición previa de la variable dependiente a ser utilizada.

Post-Test (S2): Corresponde a la nueva medición de la variable dependiente a ser utilizada.

Donde:

$$S_1 \text{ _____ } O \text{ _____ } S_2$$

S1: Satisfacción del usuario final con el proceso actual.

O: (Aplicación de la variable independiente) Sistema Informático Web para la gestión de ventas de la boutique detallitos EIRL utilizando la Metodología AUP y Framework Qcodo de PHP.

S2: Satisfacción del usuario final con Sistema Informático Web para la gestión de ventas de la boutique detallitos EIRL utilizando la Metodología AUP y Framework Qcodo de PHP.

A continuación, se muestran las mediciones hechas a la variable correspondiente (Ver detalle en Anexos):

Donde:

d_i : diferencia medida en porcentaje de la satisfacción del usuario final.

Descripción	Con el proceso actual (%)	Con el sistema propuesto (%)	d_i (%)	d_i^2 (%)
El Cliente se mantiene informado y realiza sus compras de manera rápida y eficiente.	7.14	92.86	85.72	7347.92
Existen actividades en la gestión de ventas que le ocasionan demoras o fallas en el momento de su desarrollo.	78.57	14.29	64.28	4131.92
El Sistema Web para la gestión de ventas le ayuda en el desarrollo de sus actividades de forma eficiente.	28.57	85.71	57.14	3264.98
Encuentra problemas en la gestión de ventas que se lleva en la actualidad en la boutique detallitos.	50.00	14.29	35.71	1275.20
	Totales		242.85	16020.02

Tabla 10: Variación de datos históricos del indicador Satisfacción del usuario final
Fuente: [Elaboración Propia]

$$\sum d_i = 242.85$$

$$\sum d_i^2 = 16020.02$$

$$\bar{di} = \frac{\sum di}{n} = \frac{242.85}{4} = 60.71$$

Encontrando la desviación estándar:

$$s_d = \sqrt{\frac{n \sum di^2 - (\sum di)^2}{n(n-1)}} = \sqrt{\frac{4(16020.02) - (242.85)^2}{4(4-1)}} = 20.62$$

Hipótesis estadística:

$H_0: O_1 - O_2 = 0$ Hipótesis nula

No hay diferencia alguna entre dos muestras.

$H_i: O_2 - O_1 > 0$ Hipótesis alternativa

Si hay diferencia

Nivel de significancia $\alpha = 0.05$

Función de prueba:

$$t_o = \frac{\bar{di}}{s_d} \sqrt{n}$$

Valor Crítico de “t” de student:

$$t(1-\alpha)(n-1)$$
$$t(1-0.05)(4-1)$$
$$t(0.95)(3) = 2.85$$
$$t_1 = 2.85$$

$$t_0 = \frac{60.71}{20.62} \sqrt{4} = 5.89$$

Región de aceptación o rechazo:

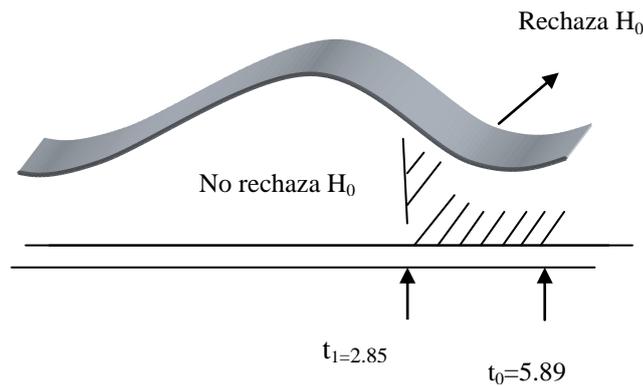


Figura 34: Región de aceptación o rechazo
Fuente: [Elaboración Propia]

Se concluye $t_0 = 5.89$ es mayor que $t_1 = 2.85$

Se rechaza H_0 y se acepta H_i : $O_2 - O_1 > 0$

Es decir, se concluye que los usuarios finales están satisfechos con el sistema propuesto y opinan que con el uso de éste software en la gestión de ventas han solucionado los principales problemas que ocurrían, además de hacerles más eficiente el desarrollo de sus actividades.

- **Indicador: *Tiempo***

Se mide el tiempo que toma realizar los requerimientos funcionales en la Gestión de ventas.

Pre-Test (T1): Medición previa de la variable dependiente a ser utilizada.

Post-Test (T2): Corresponde a la nueva medición de la variable dependiente a ser utilizada.

Donde:

T1: Tiempo en hh:mm:ss (horas:minutos:segundos) que toma realizar los requerimientos funcionales en el sistema de trabajo actual (toma de tiempo realizado con cronómetro).

O: (Aplicación de la variable independiente) Sistema Informático Web para la gestión de ventas de la boutique detallitos EIRL utilizando la Metodología AUP y Framework Qcodo de PHP.

T2: Tiempo en hh:mm:ss (horas:minutos:segundos) que toma realizar los requerimientos funcionales con el Sistema Informático para la gestión de ventas de la boutique detallitos EIRL utilizando la Metodología AUP y Framework Qcodo de PHP (toma de tiempo realizado con cronómetro).

T1_____O_____T2

A continuación, se muestran las mediciones hechas a la variable correspondiente:

Donde:

d_i : diferencia medida en segundos del tiempo que toma realizar los requerimientos funcionales. *(hh:mm:ss): Horas:Minutos:Segundos*

Requerimiento Funcional	Tiempo actual (hh:mm:ss)	Tiempo propuesto (hh:mm:ss)	d_i (Segundos)	d_i^2 (Segundos)
Registrar producto	00:01:30	00:00:55	35	1225
Registrar cliente	00:01:40	00:00:55	45	2025
Registrar Venta	00:02:00	00:00:55	65	4225
Registrar Orden de compra	00:01:45	00:00:45	60	3600
Registrar Nota de ingreso	00:01:55	00:00:50	65	4225
Registrar Nota de salida	00:01:55	00:00:50	65	4225
Registrar Proveedor	00:01:30	00:00:55	35	1225
		Totales	370	20750

Tabla 11: Variación de datos históricos del indicador Tiempo
Fuente: [Elaboración Propia]

$$\sum d_i = 370$$

$$\sum d_i^2 = 20750$$

$$\bar{d_i} = \frac{\sum d_i}{n} = \frac{370}{7} = 52.85$$

Encontrando la desviación estándar:

$$s_d = \sqrt{\frac{n \sum di^2 - (\sum di)^2}{n(n-1)}} = \sqrt{\frac{7(20750) - (370)^2}{7(7-1)}} = 14.09$$

Hipótesis estadística:

$H_0: O_1 - O_2 = 0$ Hipótesis nula

No hay diferencia alguna entre dos muestras.

$H_i: O_2 - O_1 > 0$ Hipótesis alternativa

Si hay diferencia

Nivel de significancia $\alpha = 0.05$

Función de prueba:

$$t_o = \frac{\bar{di}}{s_d} \sqrt{n}$$

Valor Crítico de “t” de student:

$$t(1-\alpha)(n-1)$$

$$t(1-0.05)(7-1)$$

$$t(0.95)(6) = 5.7$$

$$t_1 = 5.7$$

$$t_o = \frac{52.8}{14.09} \sqrt{7} = 9.91$$

Región de aceptación o rechazo:

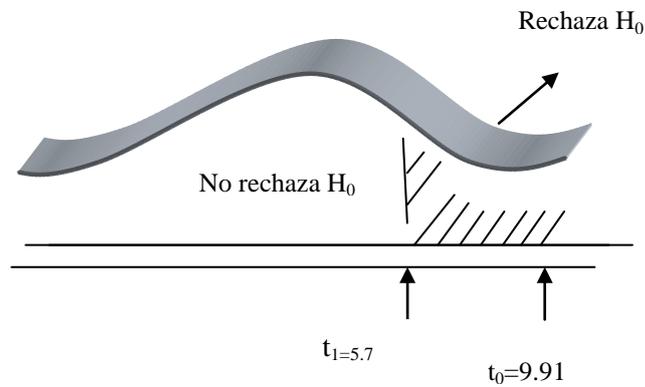


Figura N° 35: Región de aceptación o rechazo

Fuente: [Elaboración Propia]

Se concluye $t_0 = 9.91$ es mayor que $t_1 = 5.7$

Se rechaza H_0 y se acepta H_i : $O_2 - O_1 > 0$

Es decir, el tiempo que toma realizar los requerimientos funcionales con el sistema informático web desarrollado usando framework Qcodo de PHP bajo la metodología AUP es menor al tiempo que toma realizar los requerimientos funcionales en el esquema actual de trabajo.

3.2. ANÁLISIS DE RESULTADOS

Los resultados obtenidos en las mediciones hechas a los indicadores son los siguientes:

Indicadores:

- Satisfacción del usuario final:

Se concluye $S1 < S2$, por tanto SE ACEPTA $H_i: O_2 - O_1 > 0$

- Tiempo:

Se concluye $T1 > T2$, por tanto SE ACEPTA $H_i: O_2 - O_1 > 0$

Por tanto, de acuerdo a estos resultados se cumple la regla de inferencia establecida, ya que de los indicadores *son aceptados* y se concluye que la HIPÓTESIS:

“SISTEMA INFORMÁTICO WEB PARA LA GESTION DE VENTAS DE LA BOUTIQUE DETALLITOS E.I.R.L. UTILIZANDO LA METODOLOGÍA AUP Y FRAMEWORK QCODO DE PHP”

Es ACEPTADA.

CONCLUSIONES

1. Se realizó la Investigación Bibliográfica sobre la Metodología AUP, Framework CODO y Sistemas de información Web.
2. Se determinó e identificó con el desarrollo de la metodología AUP, DIECISIETE (17) requerimientos funcionales y SEIS (06) no funcionales.
3. Utilizando el ENTERPRISE ARCHITECT 7.5 en el desarrollo de la metodología AUP se obtuvieron los siguientes resultados:
 - El diagrama de clases se compuso de VEINTE (20) clases.
 - DIECISEIS (16) Casos de Uso y OCHO (8) Diagramas de Robustez principales.
4. Utilizando MySQL como gestor de base de datos se obtuvieron VEINTE (20) tablas.
5. Con el Framework QCODO de PHP se construyó las Interfaces de Usuarios. OCHO (8) Interfaces de Usuario principales.

RECOMENDACIONES

- Se recomienda realizar el análisis y diseño utilizando metodologías ágiles como AUP y XP dado que la documentación que presentan es flexible y de alta calidad.
- Elaborar un cronograma de reuniones del equipo de trabajo y el cliente, para tener una constante retroalimentación del avance del software.
- Se recomienda crear índices y realizar periódicamente limpieza de tablas históricas a fin de mejorar el rendimiento de MySQL.

REFERENCIAS BIBLIOGRAFICAS

- AmblySoft. (2006). *The Agile Unified Process v1.1*. Recuperado el 29 de 03 de 2012, de Amblysoft: <http://www.amblysoft.com/unifiedprocess/agileUP.html>
- APACHE. (2012). *APACHE*. Recuperado el 06 de 04 de 2012, de <http://www.apache.org/dist/httpd/Announcement2.4.txt>
- Cobo, Á., Gómez, P. P., & Rocha, R. (2005). *PHP y MySQL* (1era ed.). Madrid: Ediciones Días de Santos.
- De Pablos Heredero, C. (2008). *Dirección y gestión de los sistemas de información en la empresa* (2da (Reimpresión) ed.). Madrid: ESIC.
- ECURED. (2012). *Enciclopedia Cubana*. Recuperado el 30 de 05 de 2012, de http://www.ecured.cu/index.php/Agile_Unified_Process#Caracter.C3.ADstica_de_AUP
- Fernández Alarcón, V. (2006). *Desarrollo de sistemas de Información: Una metodología basada en el modelado* (1ra ed.). Barcelona: UPC.
- Hobbs, L. (1999). *Designing Internet Home Pages Made Simple* (2da ed.). Barcelona: MARCOMBO S.A.
- Jalali, A. H. (2008). *A Methodology for Risk Assessment of Product Line Architectures* (1era ed.). Morgantown: West Virginia University.
- Laudon, K. C., & Laudon, J. P. (2004). *Sistemas de información gerencial: Administración de la empresa digital* (8va ed.). México: PEARSON EDUCACIÓN.
- Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web* (1era ed.). Alicante: Editorial Club Universitario.
- Martinig, F. (2011). *StarUML - Open Source UML Tool*. Recuperado el 30 de 05 de 2012, de <http://www.methodsandtools.com/tools/staruml.php>
- MySQL. (2012). *MySQL and the ACID Model*. Recuperado el 30 de 04 de 2012, de <http://dev.mysql.com/doc/refman/5.6/en/mysql-acid.html>
- MySQL. (2012). *What is MySQL?* Recuperado el 01 de 04 de 2012, de MySQL: <http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>
- PHP. (2012). *PHP*. Recuperado el 31 de 03 de 2012, de PHP: <http://www.php.net/manual/es/intro-what-is.php>
- Rodríguez Perojo, K., & Ronda León, R. (2012). *El web como sistema de información*. Recuperado el 01 de 04 de 2012, de http://bvs.sld.cu/revistas/aci/vol14_1_06/aci08106.htm

StarUML. (2005). *The Open Source UML/MDA Platform*. Recuperado el 31 de 03 de 2012, de StarUML: <http://staruml.sourceforge.net/en/about.php>

Stober, T., & Hansmann, U. (2010). *Agile Software Development: Best Practices for Large Software Development* (1era ed.). New York: Springer.

WARP. (2012). *10 razones para usar MySQL*. Recuperado el 12 de 06 de 2012, de <http://warp.es/mysql/productos/razones/>

ANEXO

ANEXO 01 - ENCUESTA

Sistema Informático Web para la gestión de ventas de la boutique detallitos EIRL
utilizando la Metodología AUP y Framework Qcodo de PHP

1. ¿Encuentra problemas en la gestión de ventas que se lleva en la actualidad en la boutique detallitos?

SI ()

NO ()

2. ¿Existen actividades en la gestión de ventas que le ocasionan demoras o fallas en el momento de su desarrollo?

SI ()

NO ()

3. ¿El Sistema Web para la gestión de ventas le ayuda en el desarrollo de sus actividades de forma eficiente?

SI ()

NO ()

4. ¿El Cliente se mantiene informado y realiza sus compras de manera rápida y eficiente?

SI ()

NO ()

ANEXO 02 - ANÁLISIS DE RESULTADOS DE ENCUESTA

Tabulación por pregunta y opción:

Análisis en Planteamiento Actual

Pregunta #1

¿Encuentra problemas en la gestión de ventas que se lleva en la actualidad en la boutique detallitos?

	# Clientes	# Personal Empresa	# Total	%
SI	5	2	7	50.00
NO	7	0	7	50.00
Total	12	2	14	100.00

Pregunta #2

¿Existen actividades en la gestión de ventas que le ocasionan demoras o fallas en el momento de su desarrollo?

	# Clientes	# Personal Empresa	# Total	%
SI	9	2	11	78.57
NO	3	0	3	21.43
Total	12	2	14	100.00

Pregunta #3

¿El Sistema Web para la gestión de ventas le ayuda en el desarrollo de sus actividades de forma eficiente?

	# Clientes	# Personal Empresa	# Total	%
SI	4	0	4	28.57
NO	8	2	10	71.43
Total	12	2	14	100.00

Pregunta #4

¿El Cliente se mantiene informado y realiza sus compras de manera rápida y eficiente?

	# Clientes	# Personal Empresa	# Total	%
SI	1	0	1	7.14
NO	11	2	13	92.86
Total	12	2	14	100.00

Tabulación por pregunta y opción:

Análisis en Planteamiento Propuesto

Pregunta #1

¿Encuentra problemas en la gestión de ventas que se lleva en la actualidad en la boutique detallitos?

	# Clientes	# Personal Empresa	# Total	%
SI	2	0	2	14.29
NO	10	2	12	85.71
Total	12	2	14	100.00

Pregunta #2

¿Existen actividades en la gestión de ventas que le ocasionan demoras o fallas en el momento de su desarrollo?

	# Clientes	# Personal Empresa	# Total	%
SI	2	0	2	14.29
NO	10	2	12	85.71
Total	12	2	14	100.00

Pregunta #3

¿El Sistema Web para la gestión de ventas le ayuda en el desarrollo de sus actividades de forma eficiente?

	# Clientes	# Personal Empresa	# Total	%
SI	10	2	12	85.71
NO	2	0	2	14.29
Total	12	2	14	100.00

Pregunta #4

¿El Cliente se mantiene informado y realiza sus compras de manera rápida y eficiente?

	# Clientes	# Personal Empresa	# Total	%
SI	11	2	13	92.86
NO	1	0	1	7.14
Total	12	2	14	100.00

CALCULO DEL VAN, R B/C Y TIR CON UNA TASA DE DESCUENTO DEL 12%

Año de operación	Costos totales (S/.)	Beneficios totales (S/.)	Factor de actualización 12.0%	Costos actualizados (S/.)	Beneficios actualizados (S/.)	Flujo neto de efectivo act. (S/.)
0	6,340	0	1.000	6,340.00	0.00	-6,340.00
1	10,000	13,000	0.893	8,928.57	11,607.14	2,678.57
2	10,000	13,000	0.797	7,971.94	10,363.52	2,391.58
3	10,000	13,000	0.712	7,117.80	9,253.14	2,135.34
4	10,000	13,000	0.636	6,355.18	8,261.74	1,906.55
5	10,000	13,000	0.567	5,674.27	7,376.55	1,702.28
Total	46,340	65,000		42,387.76	46,862.09	4,474.33

Los indicadores financieros que arroja el proyecto son:

VAN=	4,474.33	Se acepta
TIR =	37.79%	Se acepta
B/C =	1.11	Se acepta