

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

**“DESARROLLO DE UN SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN
DE PRODUCCIÓN DE CALZADOS DE LA EMPRESA JAGUAR S.A.C.
UTILIZANDO LA METODOLOGÍA AUP Y TECNOLOGÍA ASP.NET
FRAMEWORK MVC3”**

AUTORES:

Br. Fernando Edgard Valderrama Guayan

Br. Ricardo Benites Barrientos

ASESOR:

Dr. Walter Aurelio Lazo Aguirre

CIP: 36034

TRUJILLO – PERU

2014

“ DESARROLLO DE UN SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE PRODUCCIÓN DE CALZADOS DE LA EMPRESA JAGUAR S.A.C. UTILIZANDO LA METODOLOGÍA AUP Y TEGNOLOGÍA ASP.NET FRAMEWORK MVC3”

PRESENTADO POR:

Br. Fernando Edgard Valderrama Guayan

Br. Ricardo Benites Barrientos

APROBADO POR:

Ing. José Calderón Sedano
PRESIDENTE CIP: 139198

Ing. Carlos Alberto Jara García
SECRETARIO CIP: 148099

Ing. Freddy Infantes Quiroz
VOCAL CIP: 139578

Dr. Walter Aurelio Lazo Aguirre
ASESOR CIP: 36034

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.

Fernando

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.

Ricardo

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. _____, 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 la EMPRESA JAGUAR S.A.C. 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. PRODUCCIÓN.....	17
1.4. Metodología de Desarrollo AUP	21
1.5. STARUML	24
1.6. UML 20	26
1.6.1. Conceptos básicos sobre UML	26
1.6.2. Estándares que conforman el UML	26
1.6.3. Breve descripción de los diagramas	27
1.7. ASP.NET.....	30
1.8. Microsoft SQL Server 2008	32
1.9. ASP.NET MVC	33
1.10. Enterprise Architect	39
CAPÍTULO II: RESULTADOS	42
2. INICIO	43
2.1. Descripción de la empresa	43
2.2. Presupuesto	45
2.3. Análisis de requisitos	46

2.4.	Modelo de dominio.....	49
2.5.	Requerimientos técnicos	50
2.6.	Caso de uso de requerimientos	50
2.7.	Diagrama de clases.....	58
3.	ELABORACIÓN	59
3.1.	Prototipo de Interfaz Usuario	59
4.	CONSTRUCCIÓN.....	63
4.1.	Diagrama de colaboración	63
4.2.	Diagrama de secuencia.....	68
4.3.	Modelo de despliegue	71
4.4.	Modelo de componentes	72
4.5.	Modelo Lógico.....	73
4.6.	Modelo físico	74
CAPÍTULO III: DISCUSIÓN.....		84
3.1.	CONTRASTACIÓN DE LA HIPÓTESIS	84
3.1.1.	Identificación de Variables e Indicadores	84
3.2.	ANÁLISIS DE RESULTADOS	92
REFERENCIAS BIBLIOGRAFICAS		93
CONCLUSIONES.....		94
RECOMENDACIONES.....		95

ÍNDICE COMPLEMENTARIOS DE TABLAS

Tabla N° 1: Diagrama UML.....	29
Tabla N° 2: Ciclo de Vida de una Página ASP.NET.....	31
Tabla N° 3: Resumen de Configuración AUP.....	42
Tabla N° 4: Presupuesto	46
Tabla N° 5: Detalle CU “Administrar Usuarios”	52
Tabla N° 6: Detalle CU “Administrar Producto”	53
Tabla N° 7: Detalle CU “Administrar Proceso.....	54
Tabla N° 8: Detalle CU “Registrar Orden de Producción”	55
Tabla N° 9: Detalle CU “Registrar requisición de material”	56
Tabla N° 10: Detalle CU “Reporte de producción por operario”	57
Tabla N° 11: Detalle CU “Reporte de hoja de costos”	57
Tabla 12: Variación de datos históricos del indicador Satisfacción del usuario final <i>Fuente: [Elaboración Propia]</i>	85
Tabla 13: Variación de datos históricos del indicador Tiempo <i>Fuente: [Elaboración Propia]</i>	89

ÍNDICE COMPLEMENTARIOS DE FIGURAS

Figura N° 1: Modelo general de un sistema (Alarcón, 2006).....	15
Figura N° 2: Diagrama de la Gestión de Producción	18
Figura N° 3: Diagrama de Flujo del Proceso de Producción.....	20
Figura N° 4: Ciclo de vida de AUP (Ambler, 2005)	23
Figura N° 5: Diferenciación entre Modelo, Vista y Diagrama.....	24
Figura N° 6: Interface de StarUML.....	25
Figura N° 7: Arquitectura Modelo Vista Controlador	34
Figura N° 8: Patrón MVC	35
Figura N° 9: Web y windows forms.....	36
Figura N° 10: Modelos Enterprise Architect.....	41
Figura N° 11: Modelo de Dominio.....	49
Figura N° 12: Planificar Producción	50
Figura N° 13: Caso de uso Controlar la producción.....	51
Figura N° 14: Diagrama de clases	58
Figura N° 15: Autenticar usuario	59
Figura N° 16: Registrar Proceso.....	60
Figura N° 17: Registrar producto	60
Figura N° 18: Orden de producción	61
Figura N° 19: Orden de producción: Asignar Proceso/Operario.....	61
Figura N° 20: Requisición de material	62
Figura N° 21: Requisición de material: Asignar material	62
Figura N° 22: Diagrama de robustez de Administrar Proceso	63
Figura N° 23: Diagrama de robustez de Administrar Producto	64
Figura N° 24: Diagrama de robustez de Administrar Material	65
Figura N° 25: Diagrama de robustez de Registrar Orden de Producción.....	66
Figura N° 26: Diagrama de robustez de Requisición de Material.....	67

Figura N° 27: Diagrama de robustez Hoja de Costos.....	67
Figura N° 28: Diagrama de Secuencia Administrar Proceso	68
Figura N° 29: Diagrama de Secuencia Administrar Producto.....	68
Figura N° 30: Diagrama de Secuencia Administrar Material	69
Figura N° 31: Diagrama de Secuencia Requisición de Material	69
Figura N° 32: Diagrama de Secuencia Hoja de Costos	70
Figura N° 33: Modelo de despliegue	71
Figura N° 34: Modelo de componentes.....	72
Figura N° 35: Modelo lógico.....	73
Figura N° 35: Región de aceptación o rechazo Fuente: [Elaboración Propia].....	87
Figura N° 36: Región de aceptación o rechazo	91

RESUMEN

“ DESARROLLO DE UN SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE PRODUCCIÓN DE CALZADOS DE LA EMPRESA JAGUAR S.A.C. UTILIZANDO LA METODOLOGÍA AUP Y TECNOLOGÍA ASP.NET FRAMEWORK MVC3”

PRESENTADO POR:

Br. Fernando Valderrama Guayan

Br. Ricardo Benites Barrientos

La Empresa Jaguar S.A.C se dedica a la producción y venta en el sector calzado actualmente no posee un control detallado y adecuado de la materia prima utilizada en la producción, asimismo no tiene un conocimiento exacto de los costos de producción, puesto que existe un desequilibrio en la utilización de los materiales e insumos de calzado, causando de esta manera un declive en la producción, por falta de compras oportunas de los materiales e insumos. Se propone desarrollar un Sistema Informático que realice la gestión de producción de calzados de la empresa Jaguar SAC.

Para el análisis, diseño, implementación y documentación del sistema, se usó la metodología AUP, el código fuente y el diseño de los formularios en el framework ASP.NET MVC3, para la administración de la base de datos, se usó el gestor de base de datos SQL Server 2008.

Las principales mejoras y beneficios para la Empresa son: mejor control de los materiales e insumos para la fabricación de calzado, cumplimiento de los pedidos, optimizar las compras oportunas de materiales e insumos de producción, un mejor conocimiento de las utilidades y costos reales del producto terminado.

ABSTRACT

“DEVELOPMENT OF A COMPUTER SYSTEM FOR MANAGING WEB PRODUCTION COMPANY JAGUAR SHOES SAC AUP USING FRAMEWORK METHODOLOGY AND TECHNOLOGY ASP.NET MVC3”

Bach. Fernando Valderrama Guayan

Bach. Ricardo Benites Barrientos

The Company Jaguar SAC is dedicated to the production and sale in the footwear sector currently lacks a detailed and proper control of the raw material used in production, also has an accurate knowledge of production costs, since there is an imbalance in the use of materials and supplies footwear, thus causing a decline in production due to lack of timely procurement of materials and supplies. It is proposed to develop a computer system that make managing footwear production company Jaguar SAC.

For the analysis, design, implementation and documentation of the system, the AUP method was used, the source code and designing the forms in ASP.NET MVC3 framework for the administration of the database, the database manager used SQL Server 2008 data.

Major improvements and benefits for the Company are: better control of materials and supplies for the manufacture of footwear, order fulfillment, optimizing the necessary purchases of materials and production inputs, a better understanding of the real costs and profits of the product completed.

INTRODUCCIÓN

En la actualidad las empresas en el Perú se enfrentan a un mercado global el cual obliga que compitan por calidad y precio; a su vez otro aspecto importante es el manejo oportuno y preciso de la información para llevar a cabo una adecuada toma de decisiones así como reestructurar sus procesos de trabajo utilizando TI, en nuestro caso utilizamos la metodología AUP, la cual abarca con todo el procedimiento de análisis y diseño necesario así como ASP.Net MVC3 que nos permitirá cubrir los requerimientos solicitados.

La empresa en estudio Calzados JAGUAR S.A.C se dedica a la producción y venta en el sector calzado, el proceso de producción actualmente es muy informal, no existe ningún tipo de documentación que respalde la información. Tampoco se lleva un control detallado y adecuado de la materia prima utilizada en la producción. Asimismo, no se tiene un conocimiento exacto de los costos de producción, puesto que existe un desequilibrio en la utilización de los materiales e insumos de calzado; desperdiciándose un 12% de los materiales e insumos adquiridos, causando de esta manera un declive en la producción, por falta de compras oportunas de los materiales e insumos.

La empresa tiene una producción promedio de 180 docenas mensuales, sin embargo esta producción se merma cuando la producción se detiene por falla de maquinarias, falta de personal, falta de fluido eléctrico, falta de control en la producción. Estos inconvenientes, hacen que la producción mensual disminuya en un 5% generando el incumplimiento de los pedidos por los retrasos en la producción y trayendo pérdidas económicas a la Empresa.

La realidad problemática de Calzados JAGUAR S.A.C se caracteriza por lo siguiente:

- Inadecuada utilización de los materiales e insumos para la fabricación de calzado.
- Desconocimiento de las utilidades y costos reales del producto terminado.
- Incumplimiento de los pedidos por retrasos en la producción.
- Falta de control en el proceso de producción
- Falta de materiales e insumos para la producción a causa de compras no oportunas

La realidad problemática anterior puede ser directa con la siguiente pregunta de investigación ¿De qué manera se puede mejorar la Gestión de Producción de calzados de la Empresa de JAGUAR S.A.C. utilizando tecnologías de información?

Formulándose la siguiente hipótesis: “Un Sistema Informático Web utilizando la metodología AUP y tecnología .Net mejorara la Gestión de Producción de calzados de la Empresa de JAGUAR S.A.C”

Siendo el Objetivo General: “Desarrollar un Sistema Informático Web para la Gestión de Producción de calzados de la Empresa de JAGUAR S.A.C. utilizando la metodología AUP y tecnología ASP.Net” y los objetivos específicos:

- Realizar la Investigación Bibliográfica la metodología AUP, tecnología ASP.Net, 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 SQLServer 2008
- Desarrollar el software del Sistema Informático Web utilizando tecnología ASP.Net.

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, ASP.Net como gestor de base de datos SqlServer 2008 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

Según Vicen 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.

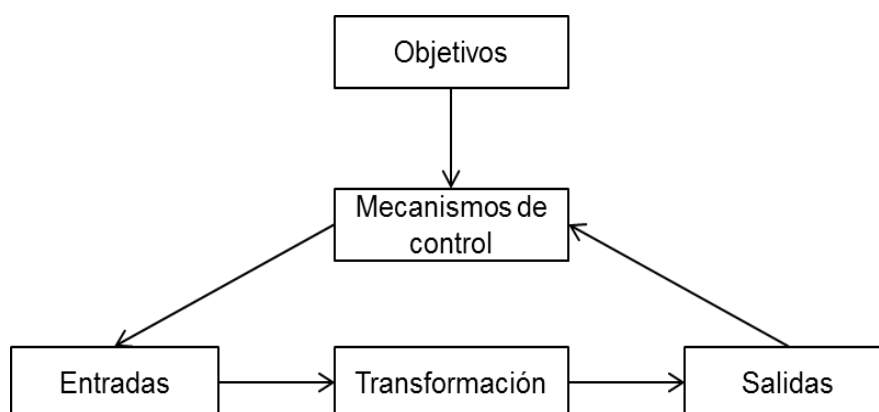


Figura N° 1: Modelo general de un sistema (Alarcón, 2006)

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, 2006)

TIPOS DE SISTEMAS

Se 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.

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. (Hobs, 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.

1.3.PRODUCCIÓN

La Producción es el estudio de las técnicas de gestión empleadas para conseguir la mayor diferencia entre el valor añadido y el coste incorporado consecuencia de la transformación de recursos en productos finales. Así también la producción puede definirse como la técnica de gestión de los sistemas que generan bienes y servicios.

El Sistema de Producción

Se considera a la empresa un sistema de producción. Este sistema puede ser analizado en dos aspectos: su concepción y su administración operativa.

Un sistema de producción empieza a tomar forma desde que se formula un objetivo y se elige el producto que se va a comercializarse.

El producto necesita de un procedimiento específico, el cual debe ser el más económico posible, teniendo en cuenta la capacidad del sistema de producción. Dicha capacidad dependerá de factores tales como los recursos materiales, humanos y financieros de la empresa. Esta capacidad de producción debe permitir el logro del objetivo a un plazo más o menos largo, el cual se fija al inicio de la operación.

La administración operativa engloba funciones esenciales y complementarias que se requieren para asegurar la armonía del sistema de producción. Las funciones esenciales son previsión, planificación y control de la producción; las complementarias son: organización científica del trabajo, administración de la calidad, administración del mantenimiento, seguridad del trabajo e informática. (Vilcarromero Ruiz, 2013)

Gestión de producción en una Empresa

Puede definirse como el proceso de transformación de los factores que ella toma de su entorno, en productos que generan valor agregado. Todo proceso de producción puede subdividirse en tres fases: (González Riesco, 2006)

- **Insumos:** implica la adquisición, recepción y almacenamiento de materias primas. Pueden ser materiales o personas.
- **Procesos:** Conjunto de operaciones a través de las cuales los factores se transforman en productos. Incluye planta, maquinaria y trabajo. Es decir, la tecnología de los activos productivos de materiales indirectos y el conocimiento.
- **Productos:** Bienes físicos y/o servicios entregados del productor al consumidor.

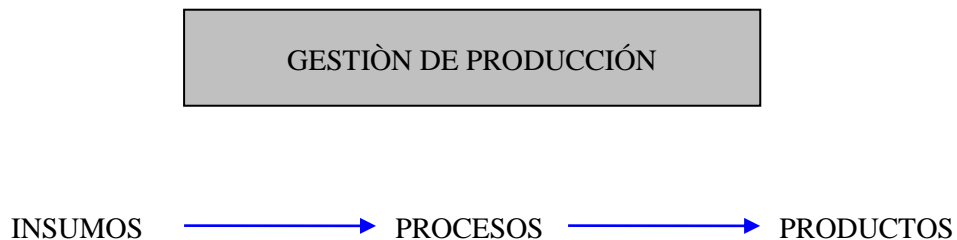


Figura N° 2: Diagrama de la Gestión de Producción

Fuente: Elaboración Propia

Elementos de Producción

Los elementos de producción están conformados por:

- **Capital:** En la gestión de producción el capital es el que designa un conjunto de bienes y una cantidad de dinero de los que se puede obtener, en el futuro, una serie de ingresos.

- **Mano de obra:** En la gestión de producción es necesaria la mano de obra como un elemento fundamental de la producción, igualmente es el esfuerzo humano realizado para asegurar un beneficio económico en la organización.

- **Materiales:** En la gestión de producción utiliza materiales para desarrollar su función esencial, la de transformación de insumos para obtener bienes o servicios.

Proceso de Producción en el Sector Calzado.

El proceso para fabricar calzado no ha variado significativamente a lo largo del tiempo, la elaboración se realiza con máquinas mecánicas y se trata de un proceso artesanal con participación muy reducida de maquinaria ya que la elaboración del producto se realiza básicamente a mano con técnicas rudimentarias. (García Márquez, 2013)

La gestión de producción de calzado es siempre un problema complicado. Por ello se suele abordar en varias etapas. Es frecuente que el número de etapas sea tres:

- Nivel de planificación de la producción.
- Nivel de programación de la producción.
- Nivel de ejecución y control.

Diagrama de Flujo del Proceso de Producción en el Sector Calzado

En el siguiente diagrama de flujo presentaremos el proceso de producción acorde a la Empresa de Calzado “Jaguar”:

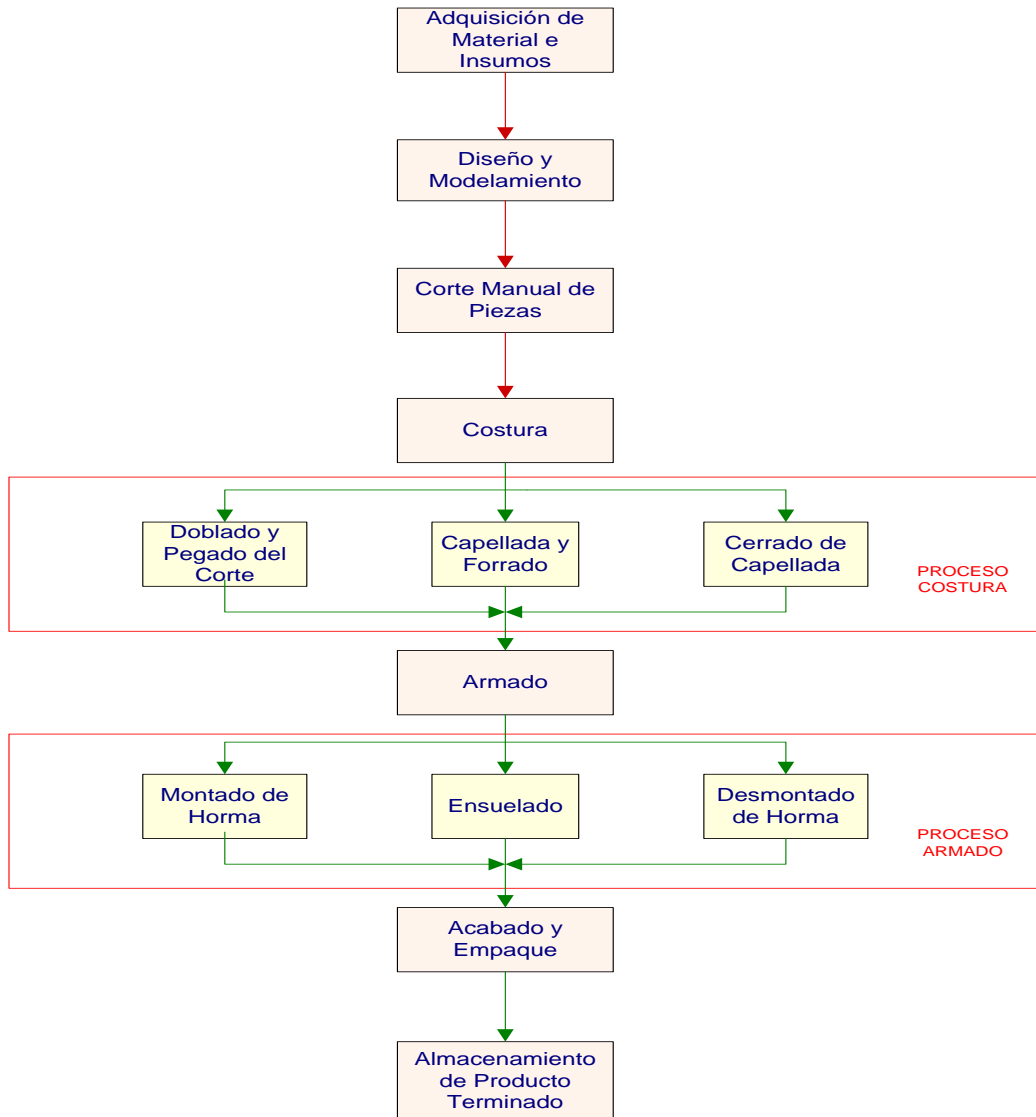


Figura N° 3: Diagrama de Flujo del Proceso de Producción

Fuente: Elaboración Propia

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. (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.

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.

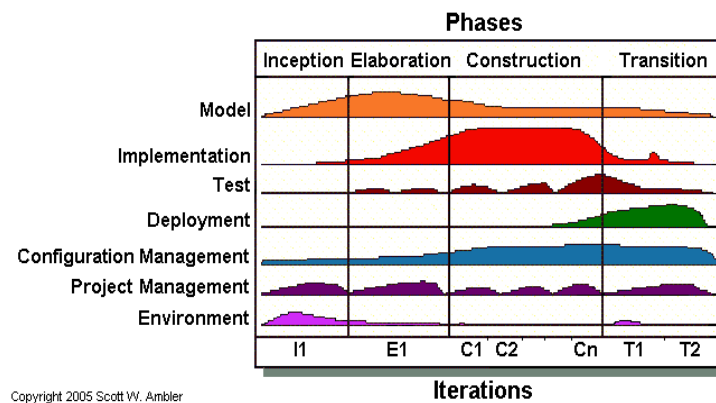


Figura N° 4: Ciclo de vida de AUP (Ambler, 2005)

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.

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

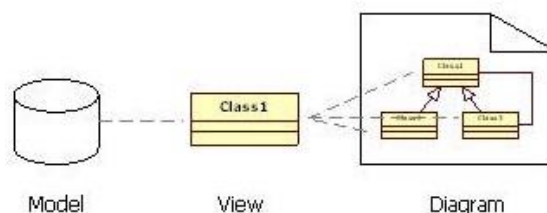


Figura N° 5: Diferenciación entre Modelo, Vista y Diagrama

- **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.

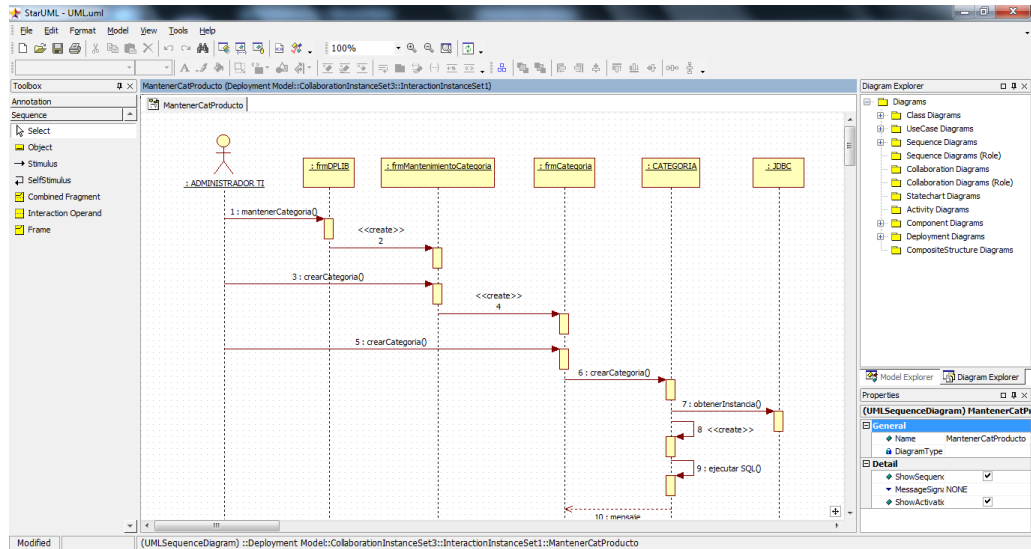


Figura N° 6: Interface de StarUML

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.

1.6. UML 20

Al momento de desarrollar el nuevo estándar 2.0 del UML, la OMG se propuso, entre otros, dos objetivos que podríamos considerar principales debido a la influencia de éstos en la versión final del estándar. Estos objetivos son:

- Hacer el lenguaje de modelado mucho más extensible de lo que era.
- Permitir la validación y ejecución de modelos creados mediante el UML.

UML 2.0 se desarrolla sobre la base de estos dos objetivos, causando un quiebre respecto a versiones anteriores. Para entender la razón del quiebre y el porqué de esta evolución tan marcada, nos profundizaremos un poco en la historia y definición misma del UML. (Laurent Debrauwer, 2010)

1.6.1. Conceptos básicos sobre UML

UML son las siglas para Unified Modeling Language, que en castellano quiere decir: Lenguaje de Modelado Unificado. Para comprender qué es el UML, basta con describir cada una de las palabras que lo componen, por separado:

- **Lenguaje:** el UML es, precisamente, un lenguaje. Lo que implica que éste cuenta con una sintaxis y una semántica. Por lo tanto, al modelar un concepto en UML, existen reglas sobre cómo deben agruparse los elementos del lenguaje y el significado de esta agrupación.
- **Modelado:** el UML es visual. Mediante su sintaxis se modelan distintos aspectos del mundo real, que permiten una mejor interpretación y entendimiento de éste.
- **Unificado:** unifica varias técnicas de modelado en una única.

1.6.2. Estándares que conforman el UML

Superestructura: Es la especificación que usamos todos los días. Aquí se encuentran todos los diagramas que la mayoría de los desarrolladores conocen.

Infraestructura: Conceptos de bajo nivel. **Meta-Modelo** da soporte a la superestructura, entre otras.

OCL: Lenguaje de restricción. De utilidad para especificar conceptos ambiguos sobre los distintos elementos del diagrama.

XMI / Intercambio de diagramas: Permite compartir diagramas entre diferentes herramientas de modelado UML. (Scott, 1999)

1.6.3. Breve descripción de los diagramas

En el siguiente Tabla N° 1 se muestra la importancia que tiene, para un desarrollador, conocer cada una de las nuevas características del UML 2.0:

Diagrama	Descripción	Prioridad
Diagrama de Clases	Muestra una colección de elementos de modelado declarativo (estáticos), tales como clases, tipos y sus contenidos y relaciones.	Alta
Diagrama de Componentes	Representa los componentes que componen una aplicación, sistema o empresa. Los componentes, sus relaciones, interacciones y sus interfaces públicas.	Media
Diagrama de Estructura de Composición	Representa la estructura interna de un clasificador (tal como una clase, un componente o un caso de uso), incluyendo los puntos de interacción de clasificador con otras partes del sistema.	Baja
Diagrama de Despliegue Físico	Un diagrama de despliegue físico muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue.	Media

Diagrama	Descripción	Prioridad
Diagrama de Objetos	Un diagrama que presenta los objetos y sus relaciones en un punto del tiempo. Un diagrama de objetos se puede considerar como un caso especial de un diagrama de clases o un diagrama de comunicaciones.	Baja
Diagrama de Paquetes	Un diagrama que presenta cómo se organizan los elementos de modelado en paquetes y las dependencias entre ellos, incluyendo importaciones y extensiones de paquetes.	Baja
Diagrama de Actividades	Representa los procesos de negocios de alto nivel, incluidos el flujo de datos. También puede utilizarse para modelar lógica compleja y/o paralela dentro de un sistema.	Alta
Diagrama de Colaboraciones	Es un diagrama que enfoca la interacción entre líneas de vida, donde es central la arquitectura de la estructura interna y cómo ella se corresponde con el pasaje de mensajes. La secuencia de los mensajes se da a través de un esquema de numerado de la secuencia.	Baja
Diagrama de Revisión de la Interacción	Los Diagramas de Revisión de la Interacción enfocan la revisión del flujo de control, donde los nodos son Interacciones u Ocurrencias de Interacciones. Las Líneas de Vida los Mensajes no aparecen en este nivel de revisión	Baja
Diagrama de Secuencias	Un diagrama que representa una interacción, poniendo el foco en la secuencia de los mensajes que se intercambian, junto con sus correspondientes ocurrencias de eventos en las	Alta

Diagrama	Descripción	Prioridad
	Líneas de Vida.	
Diagrama de Máquinas de Estado	Un diagrama de Máquina de Estados ilustra cómo un elemento, muchas veces una clase, se puede mover entre estados que clasifican su comportamiento, de acuerdo con disparadores de transiciones, guardias de restricciones y otros aspectos de los diagramas de Máquinas de Estados, que representan y explican el movimiento y el comportamiento.	Media
Diagrama de Tiempos	El propósito primario del diagrama de tiempos es mostrar los cambios en el estado o la condición de una línea de vida (representando una Instancia de un Clasificador o un Rol de un clasificador) a lo largo del tiempo lineal. El uso más común es mostrar el cambio de estado de un objeto a lo largo del tiempo, en respuesta a los eventos o estímulos aceptados. Los eventos que se reciben se anotan, a medida que muestran cuándo se desea mostrar el evento que causa el cambio en la condición o en el estado.	Baja
Diagrama de Casos de Uso	Un diagrama que muestra las relaciones entre los actores y el sujeto (sistema), y los casos de uso.	Media

Tabla N°1: Diagrama UML

Fuente: (Quiron, 2005)

1.7. ASP.NET

ASP.NET es un framework web gratuito que permite a grandes aplicaciones Web, es utilizado por millones de desarrolladores, que se ejecuta algunos de los sitios más grandes del mundo.

Cuando se ejecuta una página ASP.NET, ésta recorre un ciclo de vida en el que realiza una serie de pasos de procesamiento. Entre ellos se incluyen la inicialización, la creación de instancias de controles, la restauración y el mantenimiento del estado, la ejecución del código del controlador de eventos y la representación. Es importante comprender el ciclo de vida de la página para que se pueda escribir código en la fase del ciclo de vida apropiada y conseguir el efecto deseado.

Fase	Descripción
Solicitud de página	La solicitud de página se produce antes de que comience el ciclo de vida de la página. Cuando un usuario solicita la página, ASP.NET determina si ésta se debe analizar y compilar (a fin de que comience el ciclo de vida de la página) o si se puede enviar una versión en caché de la página como respuesta sin ejecutar la página
Inicio	En la fase de inicio, se establecen las propiedades de la página, como Request y Response. En esta fase, la página también determina si la solicitud es una devolución de datos o una nueva solicitud, y establece la propiedad IsPostBack. La página también establece la propiedad UICulture.
Inicialización	Durante la inicialización de la página, los controles incluidos en ella están disponibles y se establece la propiedad UniqueID de cada uno de ellos. También se aplican una página maestra y temas a la página, si

Fase	Descripción
	corresponde. Si la solicitud actual es una devolución de datos, los datos de devolución aún no se han cargado y los valores de las propiedades del control no se han restaurado a los valores del estado de vista.
Carga	Durante la carga, si la solicitud actual es una devolución de datos, las propiedades del control se cargan con información recuperada del estado de vista y del estado del control.
Control de eventos de devolución de datos	Si la solicitud es un postback, se llama a los controladores de eventos de control. Después, se llama al método Validate de todos los controles validadores, que establece la propiedad IsValid de cada uno de los controles validadores y de la página.
Representación	Antes de representar los datos, se guarda el estado de vista de la página y de todos los controles. Durante la fase de presentación, la página llama al método Render para cada control y proporciona un escritor de texto que escribe su resultado en el objeto OutputStream de la propiedad Response de la página.
Descargar	El evento Unload se genera cuando la página se ha representado completamente, se ha enviado al cliente y está lista para ser descartada. Llegado este momento, se descargan las propiedades de la página, como Response y Request, y se llevan a cabo las operaciones de limpieza correspondientes.

Tabla N° 2: Ciclo de Vida de una Página ASP.NET

Fuente: (Desarrollo (ASP.NET), 2013)

1.8. Microsoft SQL Server 2008

Microsoft SQL Server 2008 es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2008 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.

El motor de datos SQL Server 2008 constituye el núcleo de esta solución de administración de datos reales. Asimismo, SQL Server 2008 combina lo mejor en análisis, información, integración y notificación. Esto permite que su negocio cree y despliegue soluciones de BI rentables que ayuden a su equipo a incorporar datos en cada rincón del negocio a través de tableros de comando, escritorios digitales, servicios Web y dispositivos móviles. Con la ampliación de las ventajas ofrece una solución integrada de administración y análisis de datos que ayuda a las organizaciones de cualquier magnitud a realizar lo siguiente: Crear, desplegar y administrar aplicaciones empresariales más seguras, escalables y confiables.

- Maximizar la productividad de IT mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- Compartir datos en múltiples plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.
- Controlar los costes sin sacrificar el rendimiento, la disponibilidad, la escalabilidad o la seguridad.

SQL Server 2008 potencia su infraestructura de datos en tres áreas clave: administración de datos empresariales, productividad del encargado del desarrollo e inteligencia empresarial (BI). También abre nuevos caminos en precios y licencias accesibles, rutas de actualización a SQL Server 2008 y el sistema Microsoft Windows Server. (Misner, 2010)

1.9. ASP.NET MVC

ASP.NET MVC es un framework que combina la efectividad, potencia y claridad de la arquitectura modelo **modelo-vista-controlador** (MVC), las más actualizadas ideas y técnicas del **desarrollo ágil**, y las mejores partes de la **plataforma ASP.NET existente**.

Es una completa alternativa al desarrollo tradicional en ASP.NET webforms, otorgando **considerables ventajas** al mismo.

El Patrón Modelo - Vista - Controlador (MVC)

ASP.NET MVC es, básicamente, una implementación del patrón Modelo - Vista - Controlador (MVC) para tecnología

ASP.NET. El patrón MVC no es ni nuevo (data de finales de los años 70) ni está pensado para aplicaciones web, pero en realidad en aplicaciones web encaja perfectamente.

Brevemente podemos decir que el patrón MVC separa la lógica (y acceso a datos) de una aplicación de su presentación, usando 3 componentes:

1. Modelo: Representa las reglas de negocio de la aplicación (y el acceso a datos subyacente).
2. Vistas: Representan la presentación de la aplicación.
3. Controlador: Actúan de intermediario entre el usuario y el Modelo y las Vistas. Recogen las peticiones del usuario, interactúan con el modelo y deciden que vista es la que debe mostrar los datos.

En el contexto de ASP.NET MVC:

- Toda la lógica de negocio y el acceso a datos es el Modelo (en muchos casos el Modelo puede estar en uno o varios assemblies referenciados).
- Las vistas contienen, básicamente, el código que se envía al navegador, es decir el código HTML (y código de servidor asociado, siempre y cuando este código haga cosas de presentación, no de lógica de negocio).

- Los controladores reciben las peticiones del navegador y en base a esas, deciden que vista debe enviarse de vuelta al navegador y con qué datos. (Alvarez, s.f.)

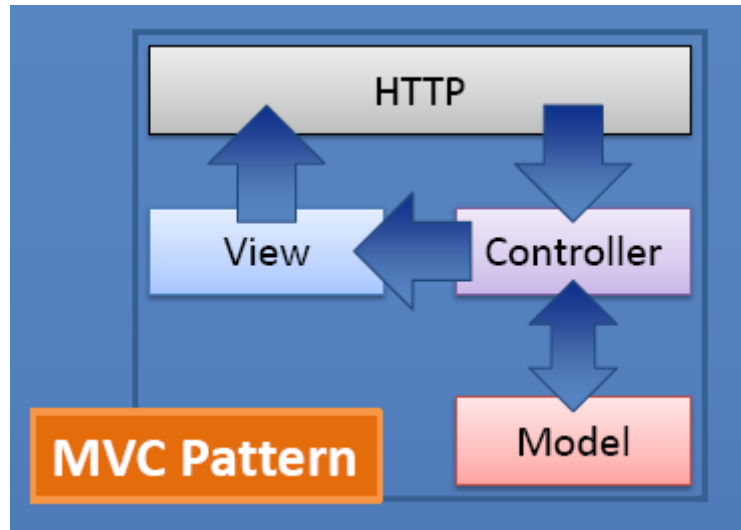


Figura N° 7: Arquitectura Modelo Vista Controlador

MVC viene a ser un patrón de ingeniería de software que distribuye los datos de la aplicación, la interface y la lógica de negocios en tres distintos componentes.

Este patrón arquitectural cuenta con:

Estructura: Pues cuenta con una estructura definida que debe de ser aplicada sin excepciones.

Responsabilidades: Brinda a cada uno de los componentes una limitación en sus acciones, lo que permite mayor orden en el flujo de trabajo.

Relaciones: Integra la funcionalidad de los tres componentes para brindar un solo resultado.

El patrón MVC (III)

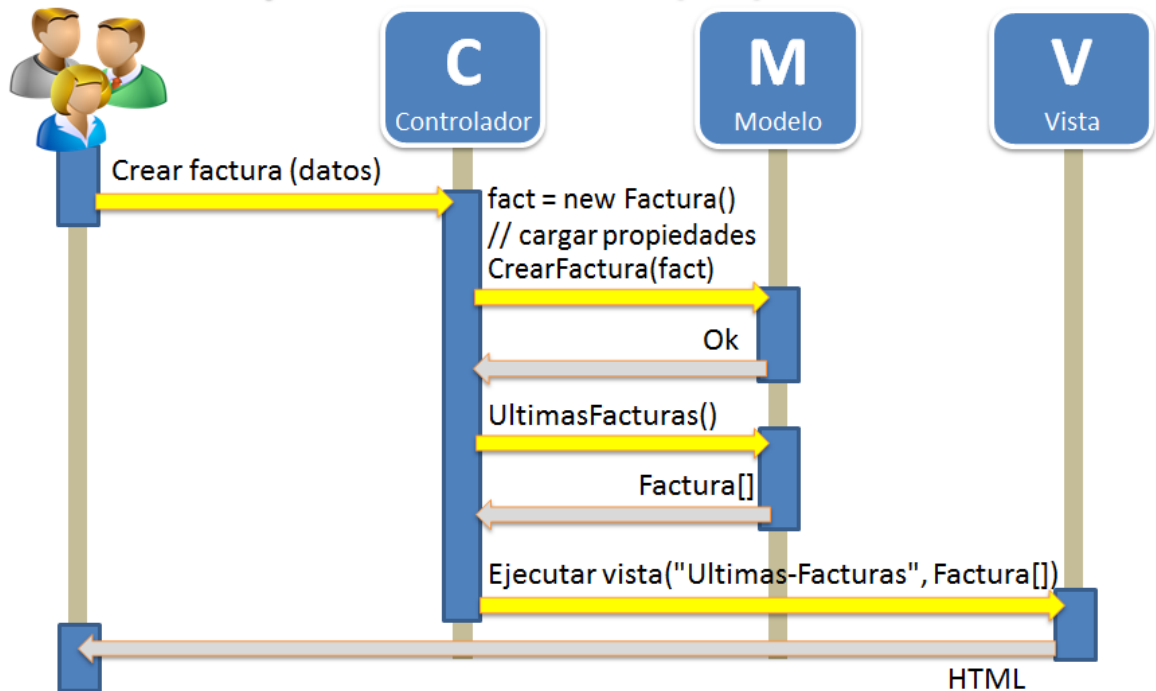


Figura N° 8: Patrón MVC

Como se puede ver en la imagen el proceso de una solicitud se lleva de la siguiente manera:

Usuario solicita la página “Crear Factura”, este es procesado por nuestro servidor y enviado al Controlador correspondiente; quien se encarga de hacer las solicitudes respectivas a el Modelo (Quien se encarga de realizar los procesos respectivos) y es devuelto al Controlador para luego ser enviado a la Vista quien devolverá los resultados en HTML.

ASP.NET Tradicional

ASP.NET tuvo un gran impacto cuando fue lanzado, no solamente por ser una **plataforma multilenguaje**, sino en que llenaba el hueco existente entre las

aplicaciones **Windows Forms, con estado** y orientadas a objetos, y las aplicaciones **web orientadas a HTML, intrínsecamente sin estado**.

Microsoft intentó ocultar tanto el protocolo HTTP (que es intrínsecamente sin estado) como el código HTML generado (que en el momento no era conocido por muchos desarrolladores) modelando una arquitectura de interfaz de usuario que abstraía esos conceptos a un conjunto de controles. Cada control gestionaba su propio estado a través de las diferentes llamadas al servidor, conectaba los diferentes eventos de cliente y servidor, y se encargaba de generar el html correspondiente.

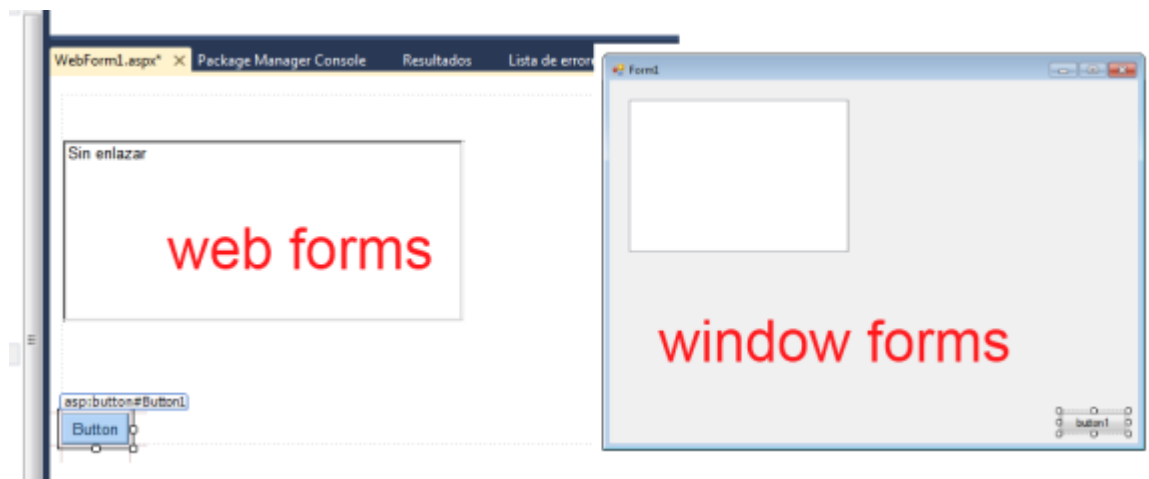


Figura N° 9: Web y windows forms

De esta manera, los desarrolladores no tenían que trabajar con una serie de peticiones HTTP independientes y sus consiguientes respuestas. De esta manera nos olvidábamos de la web, y construíamos nuestro interfaz usando un diseñador de "arrastrar y soltar", y nos imaginábamos que todo sucedía en el servidor.

¿Y qué tiene todo esto de malo?

Aunque la intención fue buena al principio, la realidad resultó poco más complicada. A lo largo de los años ASP.NET webforms ha demostrado tener una serie de debilidades:

- **ViewState:** El mecanismo para mantener estado a través de peticiones web, normalmente resultaba en **bloques gigantes de código** que eran innecesariamente transferidos entre cliente y servidor (muchas veces llegaban a ser cientos de Kb), ralentizando la experiencia del visitante cada vez que clicaban un botón de nuestra aplicación web. ASP.NET Ajax adolece del mismo problema, aún suponiendo que este era el problema que Ajax debía de solucionar.
- **Ciclo de vida de la página:** El mecanismo que conectaba eventos de cliente y servidor, parte del ciclo de vida de la página, llegaba a ser extraordinariamente complicado y delicado, llevando normalmente a errores y problemas de mantenibilidad. Manipular la jerarquía de controles de un webforms comúnmente llevaba a errores de Viewstate o eventos que misteriosamente fallaban al ejecutarse.
- **Limitado control sobre el HTML generado:** Los controles de servidor renderizan HTML, pero nunca el HTML que a ti te gustaría. El código HTML que generan es ineficiente, pesado y no cumple los standards ni hace un correcto uso de CSS. El servidor genera una enfarragosa colección de valores de ID cliente, a los que es muy difícil de acceder via Javascript.
- **No hay separación de responsabilidades:** El modelo de ASP.NET code-behind pretende desacoplar el código HTML del código de servidor, pero en la realidad se acaba mezclando código de presentación (ej: manipulando la jerarquía de controles) con el de lógica de negocio (ej: accediendo a la base de datos) en la misma, enorme, y difícil de mantener clase de code-behind. Sin separación de responsabilidades, el resultado final se frágil e ilegible.

- **Imposible de testear:** Cuando se lanzó ASP.NET, no se anticipó que el desarrollo orientado a pruebas iba a ser una práctica común en el desarrollo de software a nivel mundial. Es por ello que esta arquitectura es **completamente intesteable**.

Beneficios de ASP.NET MVC

Arquitectura MVC

Desarrollada en los años 80, promulga la separación de responsabilidades entre el controlador de peticiones, el modelo o repositorio de datos y reglas de negocio y las vistas. Esta arquitectura facilita la **testeabilidad** .

Extensible

Prácticamente cada pieza que forma el framework ASP.NET MVC puede ser alterado, o modificado por una implementación propia para cubrir nuestras necesidades. Para cada componente del MVC Framework tenemos 3 opciones:

- User la **implementación que trae por defecto** (normalmente suficiente para la mayoría de las aplicaciones)
- Crear una **clase derivada** y modificar el comportamiento deseado.
- Reemplazar el componente deseado por una **nueva y completa implementación** que cubra nuestras necesidades.

Testeable

La arquitectura MVC facilita la creación de pruebas unitarias. Crear aplicaciones usando **metodologías ágiles** o basadas en **TDD** es muy sencillo usando ASP.NET MVC. Ahora nuestras aplicaciones son más robustas.

Control preciso sobre el HTML generado

Usando ASP.NET MVC podemos escribir el código HTML final que deseemos, control que no teníamos con ASP.NET webforms. Eso significa que nuestras

aplicaciones van a generar un código más **limpio**, que siga los estándares, y que se integre fácilmente con nuestro código Javascript.

Desde su versión ASP.NET MVC ha tenido el concepto de motor de vistas (View Engine). A ver, recapitulemos: en ASP.NET MVC las vistas realizan tareas sólo de presentación. No contienen ningún tipo de lógica de negocio y no acceden a datos. Básicamente se limitan a mostrar datos (en el artículo anterior vimos como pasar datos de los controladores a las vistas) y a solicitar datos nuevos al usuario. Si vienes del mundo de webforms, olvídate del concepto de Web Controls: no existen en ASP.NET MVC. No tenemos drag and drop, no configuramos propiedades. Las vistas son básicamente HTML. Y lo que no es HTML son pequeñas porciones de código de servidor destinadas a terminar generando HTML para mostrar información.

Potente sistema de rutado (routing)

ASP.NET webforms ya contaba con un sistema de rutado, pero ha sido con ASP.NET MVC cuando lo hemos exprimido para aprovechar todas sus capacidades. Con ASP.NET MVC, una petición HTTP no tiene que estar necesariamente mapeada a un archivo, sino que podemos crear elegantes y "user-friendly" rutas a nuestro antojo. Ejemplo: Antes con webforms /micoche.aspx?id=3 , ahora con ASP.NET MVC: /Coches/3/Opel-Astra (ASP.NET MVC vs ASP.NET WebForms, 2013)

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 (Enterprise Architect - Herramienta de diseño UML, 2014)

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.

Enterprise Architect permite:

- Modelar los procesos de negocio
- Definir los requisitos de un proyecto.
- Diseñar una solución y construir el esqueleto.
- Codificar utilizando técnicas ágiles y desarrollo guiado por pruebas.

Enterprise Architect nos permite elegir los modelos que queremos utilizar.

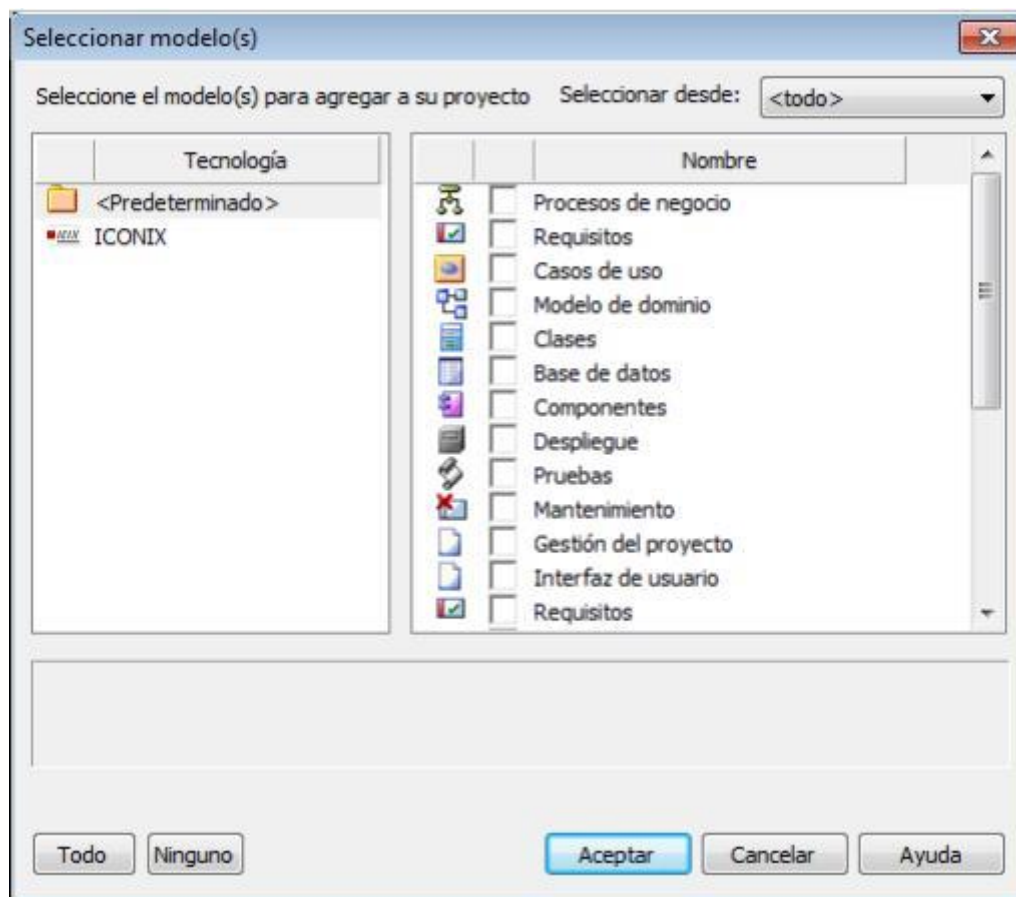


Figura N° 10: Modelos Enterprise Architect

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° 3: Resumen de Configuración AUP

2. INICIO

2.1. Descripción de la empresa

La Empresa JAGUAR S.A.C se dedica a la producción de calzado, desde el año 1990, tiene más de 20 años dentro del rubro del calzado, se encuentra ubicado en la Provincia de Trujillo, Departamento de La Libertad.

La Empresa se ha dedicado a brindar los factores que el cliente busca:

- Exclusividad en los diseños.
- Calidad.
- Precio.

Cuenta con la dirección de la Gerencia que tiene a su cargo diferentes áreas como son: Contabilidad, Ventas, Almacén y Producción.

Área de Contabilidad: Departamento donde se lleva el control de las cuentas de la empresa, asimismo detallada cada operación; siendo necesario planear el sistema contable que se utilizará.

Área de Ventas: Departamento donde se reciben los pedidos.

El Área de ventas sirve para medir la actuación de la fuerza de ventas en relación con las ventas pronosticadas, de tal forma que sea posible detectar las variaciones significativas, y adoptar las medidas correctivas adecuadas, tales como mejorar el servicio al cliente en cuanto a rapidez de entrega del producto, mejorar las facilidades de crédito, aumentar el número de visitas, reforzar al vendedor a través de campañas publicitarias, aumentar sus áreas de acción y ampliar su perspectiva.

Área de Almacén: Departamento donde se llevan los productos terminados y los insumos que se emplearán en la producción, para su posterior distribución a la ciudad de Lima y provincias. Almacén lleva un control sobre las entradas y salidas de productos terminados y materiales. El Área de Almacén brinda en constante información al gerente, sobre las existencias reales de la materia prima.

Área de Producción: Departamento donde se desarrollan los métodos mas adecuados para la elaboración del calzado.

Se cuenta con 21 Trabajadores:

2 diseñadores (modelistas)

4 cortadores de cuero

5 aparadores

5 armadores

5 acabadores

El diseñador (modelista) es el encargado de los nuevos diseños, el gerente es la persona quien da la aprobación para la fabricación, en muchas ocasiones se producen y se llevan a Lima para ofrecerlos.

El cortador de cuero es el encargado de cortar el cuero de acuerdo al modelo seguido por el diseñador.

El aparador (perfilador) es el encargado de realizar las costuras de los recortes de cuero.

El armador es el encargado de colocar la costura dentro de la horma para realizar el armado del calzado.

El acabador es el encargado de hacer los acabados del calzado para la venta y distribución del mismo.

Maquinaria empleada:

La maquinaria que se emplea para la fabricación de calzado son las siguientes:

- 2 maquina rematadora de suela
- 2 maquina devastadora de cuero
- 2 maquina devastadora de badana
- 2 maquina devastadora de suela
- 5 máquinas de coser
- 5 módulos para armar
- 2 remachadora
- 3 selladoras de cuero
- 3 selladoras de suela
- 3 cortadoras de suela

2.2. Presupuesto

RECURSOS HUMANOS		COSTO (S/.)
2 Investigadores		0.00
Asesor		1250.00
BIENES		COSTO MENSUAL(S/.)
Materiales	Útiles de Escritorio	25.00
	Cd's	35.00
Equipos	02 Computadoras Pentium IV de 3 Ghz, 512 Mb RAM, HD 40 Gb	3000.00
	2 memorias USB	160.00
Software	Microsoft Visual Studio.Net 2010	300.00
	Microsoft Sql Server 2008	200.00
	Microsoft Windows XP Professional Version 2002 Service Pack 3	800.00
	Microsoft Office 2010	350.00

Servicios	Biblioteca	00.00
	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	1650.00
Servicios	520.00
COSTO TOTAL:	S/. 6640.00

Tabla N° 4: Presupuesto

2.3. Análisis de requisitos

2.3.1. Requerimientos Funcionales

a. Declaración de Requerimientos:

a.1. Registrar Usuario

En este mantenedor se registra el nombre, apellido paterno, apellido materno, DNI, usuario y clave de un nuevo usuario del sistema

a.2. Registrar Operario

En este mantenedor se registra el nombre, apellido paterno, apellido materno, DNI, fecha de nacimiento y tipo de operario.

a.3. Registrar Tipo de operario

En este mantenedor se registra nombre, descripción.

a.4. Registrar Proceso

En este mantenedor se registra nombre, descripción y tiempo de proceso.

a.5. Registrar Producto

En este mantenedor se registra serie, talla, color, precio, descripción, unidad de medida y stock.

a.6 Consultar Pedido

En este requerimiento se consulta el pedido para realizar la orden de producción.

a.7 Registrar Orden de Producción

En este mantenedor se basa en la búsqueda del pedido y se procede a registrar fecha, producto y cantidad, así como el proceso y operario.

a.8. Registrar Requisición de Material

En este mantenedor se registrar la fecha del documento y la cantidad del material.

a.9. Registrar Hoja de Costos

En este mantenedor se registrar la fecha, cantidad, subtotal y costo de los materiales.

a.10. Elaborar Reportes

a.10.1. Elaborar Reportes de producción por operario

En este mantenedor se muestra una lista de la producción de los operarios previa selección de rango de fechas a mostrar.

a.10.2. Elaborar Reportes de Hoja de Costos

En este reporte se muestra de hoja de costos por orden de producción.

a.10.3. Elaborar Reportes de Materiales por orden de producción

En este reporte se muestra el reporte de materiales por orden de producción.

2.3.2. 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.

- **Hosting**

Contar con un hosting que de soporte al entorno ASP.Net

- **Manual de Usuario**

Debe existir un manual de usuario relacionado con el área de ventas.

2.4. Modelo de dominio

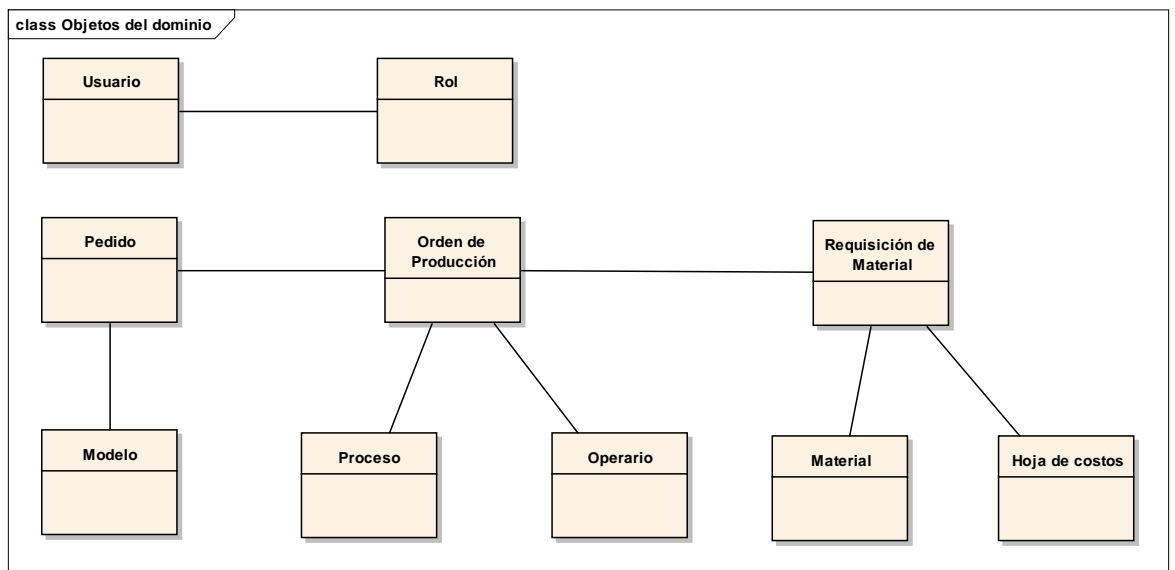


Figura N° 11: Modelo de Dominio

2.5. Requerimientos técnicos

- Metodología de Desarrollo AUP.
- Herramienta de Modelado: Enterprise Architect.
- ASP.Net
- Base de Datos SQLServer 2008
- Tiempo de respuesta máximo 4 segundos en cada transacción.

2.6. Caso de uso de requerimientos

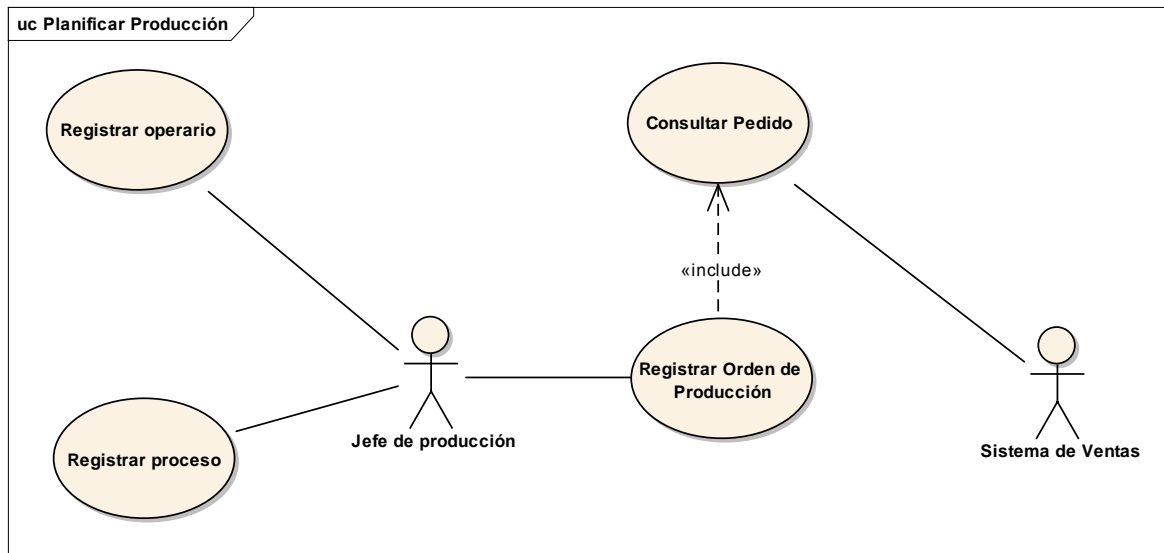


Figura N° 12: Planificar Producción

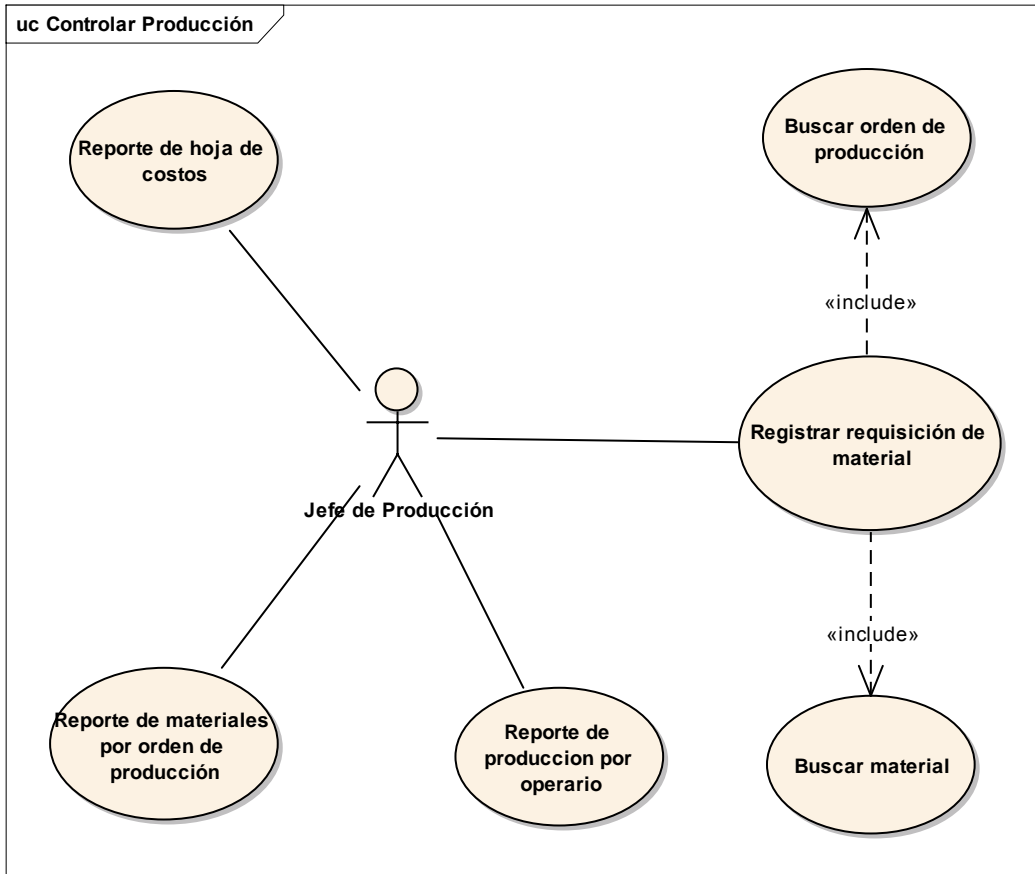


Figura N° 13: Caso de uso Controlar la producción

Nombre	Administrar Usuarios	
Descripción	Permite crear nuevos usuarios al sistema.	
Actor	Jefe de Producción	
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° 5: Detalle CU “Administrar Usuarios”

Nombre	Administrar Producto	
Descripción	Permite crear nuevos productos al sistema.	
Actor	Jefe de Producción	
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 serie, talla, color, precio, descripción, unidad de medida, stock.
	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° 6: Detalle CU “Administrar Producto”

Nombre	Administrar Proceso	
Descripción	Permite crear nuevo proceso al sistema.	
Actor	Jefe de Producción	
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 proceso.
	7	Se despliega la pantalla para crear proceso.
	8	Se crea un nuevo proceso, ingresando nombre, descripción y tiempo de proceso.
	9	Si el usuario desea, puede crear varios procesos nuevos, volviendo al paso 7.
	10	Una vez que finalice la creación de procesos, el actor puede salir del sistema o salir de la creación de procesos, volviendo al paso 5.
	11	El usuario puede buscar procesos para modificar o eliminar.
	12	Una vez seleccionado el proceso, 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° 7: Detalle CU “Administrar Proceso

Nombre	Registrar Orden de Producción	
Descripción	Permite crear la orden de producción	
Actor	Jefe de Producción	
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 jefe de producción buscar el pedido de producción por el número.
	2	Se muestra el detalle del pedido en el detalle de la orden de producción.
	3	Se procede a cada línea de detalle de la orden de producción buscar proceso y asignar.
	4	Se procede a cada línea de detalle de la orden de producción buscar operario y asignar.
	5	Al finalizar la orden de producción, 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.	
Includes	Buscar Pedido	
Excepciones	Paso	Acción
		N/A

Tabla N° 8: Detalle CU “Registrar Orden de Producción”

Nombre	Registrar requisición de material	
Descripción	Permite crear la requisición de material	
Actor	Jefe de Producción	
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 jefe de producción buscar la orden de producción por el número.
	2	Se muestra el detalle de la orden de producción en el detalle de la orden de requisición de material.
	3	Se procede a cada línea de detalle de la orden de requisición asignar materiales.
	4	Al finalizar la orden de requisición de material, 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.	
Includes	Buscar orden de producción Buscar material	
Excepciones	Paso	Acción
		N/A

Tabla N° 9: Detalle CU “Registrar requisición de material”

Nombre	Reporte de producción por operario	
Descripción	Permite al jefe de producción realizar reporte de producción por operario.	
Actor	Jefe de producción	
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 se muestra la producción por operario.
	2	Buscar producción por operario y mostrar.
Post-condición		
Excepciones	Paso	Acción
		N/A

Tabla N° 10: Detalle CU “Reporte de producción por operario”

Nombre	Reporte de hoja de costos	
Descripción	Permite al jefe de producción realizar reporte de hoja de costos	
Actor	Jefe de producción	
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 se muestra la hoja de costos por orden de producción.
	2	Buscar Hojas de costo y mostrar.
Post-condición		
Excepciones	Paso	Acción
		N/A

Tabla N° 11: Detalle CU “Reporte de hoja de costos”

2.7. Diagrama de clases

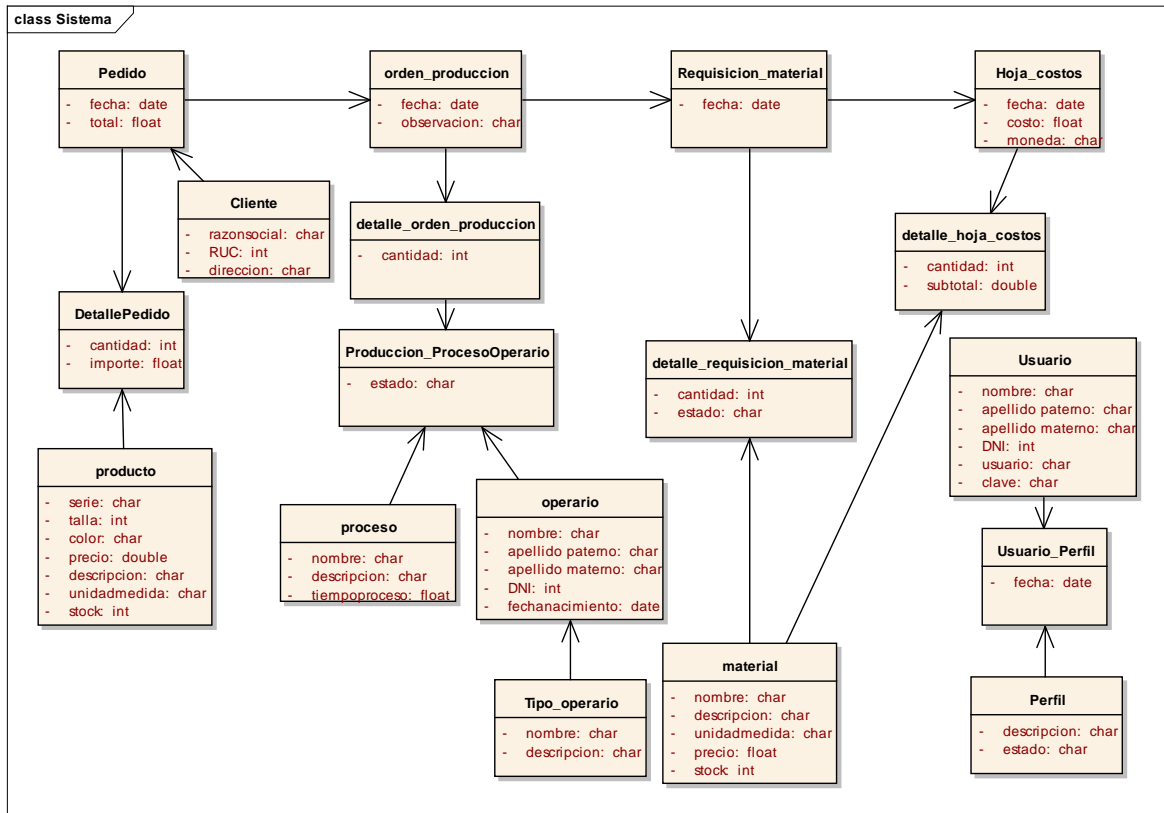


Figura N° 14: Diagrama de clases

3. ELABORACIÓN

3.1. Prototipo de Interfaz Usuario



The image shows a web interface for 'Calzados Jaguar SAC'. The header features the company name and 'Venta de calzado' with a search bar and 'BUSCAR' button. A navigation menu includes 'Inicio', 'Planificación', 'Control', and 'Ayuda'. A breadcrumb trail reads 'Planificación > Orden de producción'. On the left is a photograph of a jaguar on a rock. On the right is a grey box titled 'Accesar a la Intranet' containing fields for 'Usuario:' and 'Clave:', and an 'Iniciar sesión' button.

Figura N° 15: Autenticar usuario

Calzados Jaguar SAC **BUSCAR**

Venta de calzado

Inicio Planificación Control Ayuda

Planificación > Registrar Proceso

PROCESO:



Nombre	descripcion	pagoxdocena	tiempoxdocena	Acciones
				 

Figura N° 16: Registrar Proceso

Calzados Jaguar SAC **BUSCAR**

Venta de calzado

Inicio Planificación Control Ayuda

Planificación > Registrar Producto

PRODUCTO:

serie	talla	color	precio	descripcion	unidad medida	stock	Acciones
							 

Figura N° 17: Registrar producto

Calzados Jaguar SAC **BUSCAR**

Venta de calzado

Inicio Planificación Control Ayuda

Planificación > Orden de producción

Numero de orden: Numero de pedido: **Buscar**

Fecha:

Detalle de producción: **Asignar Proceso/Operario**

Código	Descripción	Cantidad
00010	Zandalia, 35, Blanco	1(doc)
00011	Zandalia, 37, Blanco	1(doc)

Figura N° 18: Orden de producción

Calzados Jaguar SAC **BUSCAR**

Venta de calzado

Inicio Planificación Control Ayuda

Planificación > Orden de producción

Numero de orden: Numero de pedido: **Buscar**

Fecha:

Detalle de producción: **Asignar Proceso/Operario**

Código:	<input type="text" value="00010"/>	Proceso:	<input type="text" value="Cortado"/>
Descripción:	<input type="text" value="Zandalia, 35, Blanco"/>	Operario:	<input type="text" value="Juan García"/>
Cantidad:	<input type="text" value="6"/>	<input type="button" value="Asignar"/>	

Código	Proceso	Operario	Estado	Cantidad	Fecha	Acción
00010	Cortado	Juan García Perez	Pendiente	1(doc)		<input type="checkbox"/>
00010	Aparado	Luis Flores Clavijo	Pendiente	1(doc)		<input type="checkbox"/>
00010	Armado	Ernesto Ulco Garcia	Pendiente	1(doc)		<input type="checkbox"/>
00010	Acabado	Antonio Mendoza Perez	Pendiente	1(doc)		<input type="checkbox"/>

Figura N° 19: Orden de producción: Asignar Proceso/Operario

Calzados Jaguar SAC
Venta de calzado

BUSCAR

Inicio Planificación Control Ayuda

Planificación > Requisición de material

Numero requisición: Numero de orden: **Buscar**

Fecha:

Detalle requisición **Asignar material**

Código	Descripción	Cantidad
00010	Zandalia, 35, Blanco	1(doc)
00011	Zandalia, 37, Blanco	1(doc)

Grabar **Salir**

Figura N° 20: Requisición de material

Calzados Jaguar SAC
Venta de calzado

BUSCAR

Inicio Planificación Control Ayuda

Planificación > Requisición de material

Numero requisición: Numero de orden: **Buscar**

Fecha:

Detalle requisición **Asignar material**

<p>Código: <input type="text" value="00010"/></p> <p>Descripción: <input type="text" value="Zandalia, 35, Blanco"/></p> <p>Cantidad: <input type="text" value="1(doc)"/></p>	<p>Material: <input type="text" value="Bencina"/> Buscar</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Nombre</th> <th>Unidad</th> </tr> </thead> <tbody> <tr> <td>Bencina</td> <td>1 Litro</td> </tr> <tr> <td>Bencina</td> <td>1/2 litro</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th>Nombre</th> <th>Unidad</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr> <td>Bencina</td> <td>1 litro</td> <td><input type="text" value="2"/></td> </tr> </tbody> </table> <p style="text-align: center;">Asignar</p>	Nombre	Unidad	Bencina	1 Litro	Bencina	1/2 litro	Nombre	Unidad	Cantidad	Bencina	1 litro	<input type="text" value="2"/>
Nombre	Unidad												
Bencina	1 Litro												
Bencina	1/2 litro												
Nombre	Unidad	Cantidad											
Bencina	1 litro	<input type="text" value="2"/>											

Código	Nombre	Cantidad	Fecha	Acción
00010	Bencina	2		<input type="button" value="X"/>

Grabar **Salir**

Figura N° 21: Requisición de material: Asignar material

4. CONSTRUCCIÓN

4.1. Diagrama de colaboración

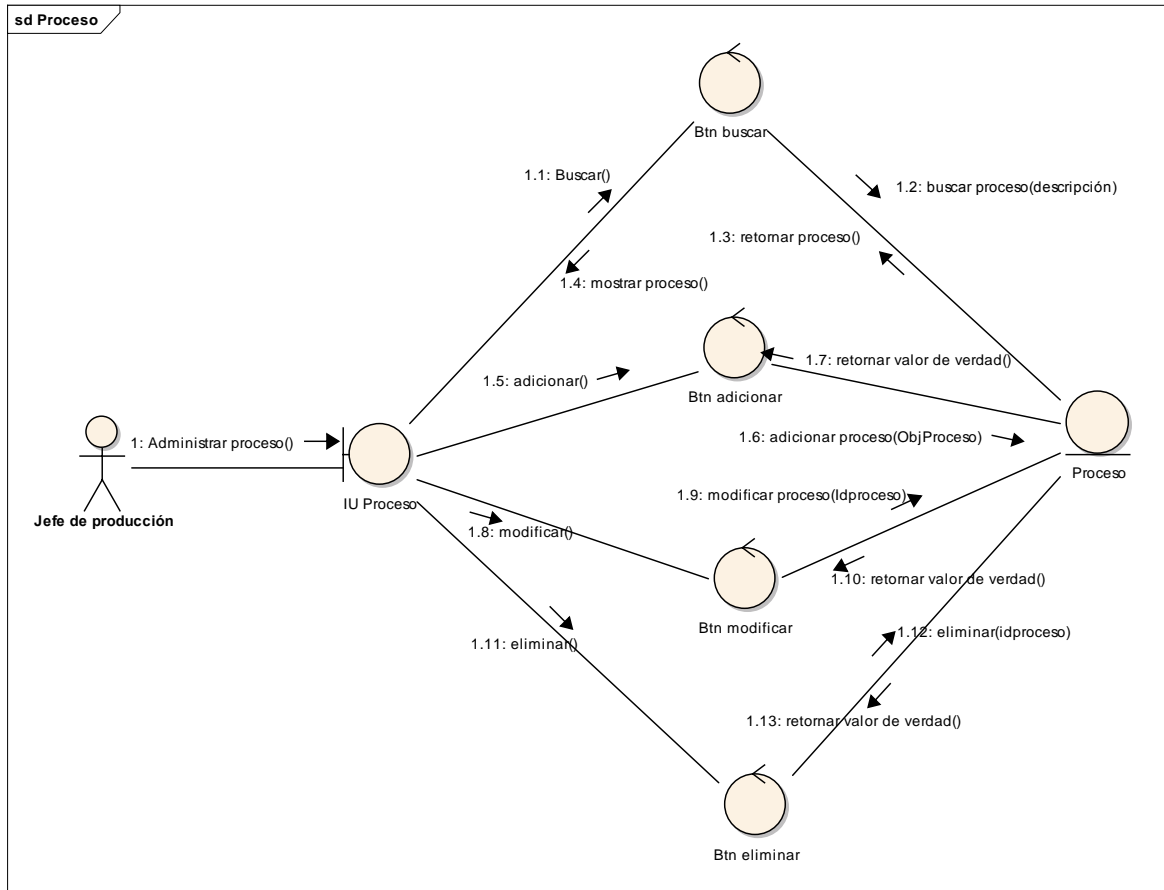


Figura N° 22: Diagrama de robustez de Administrar Proceso

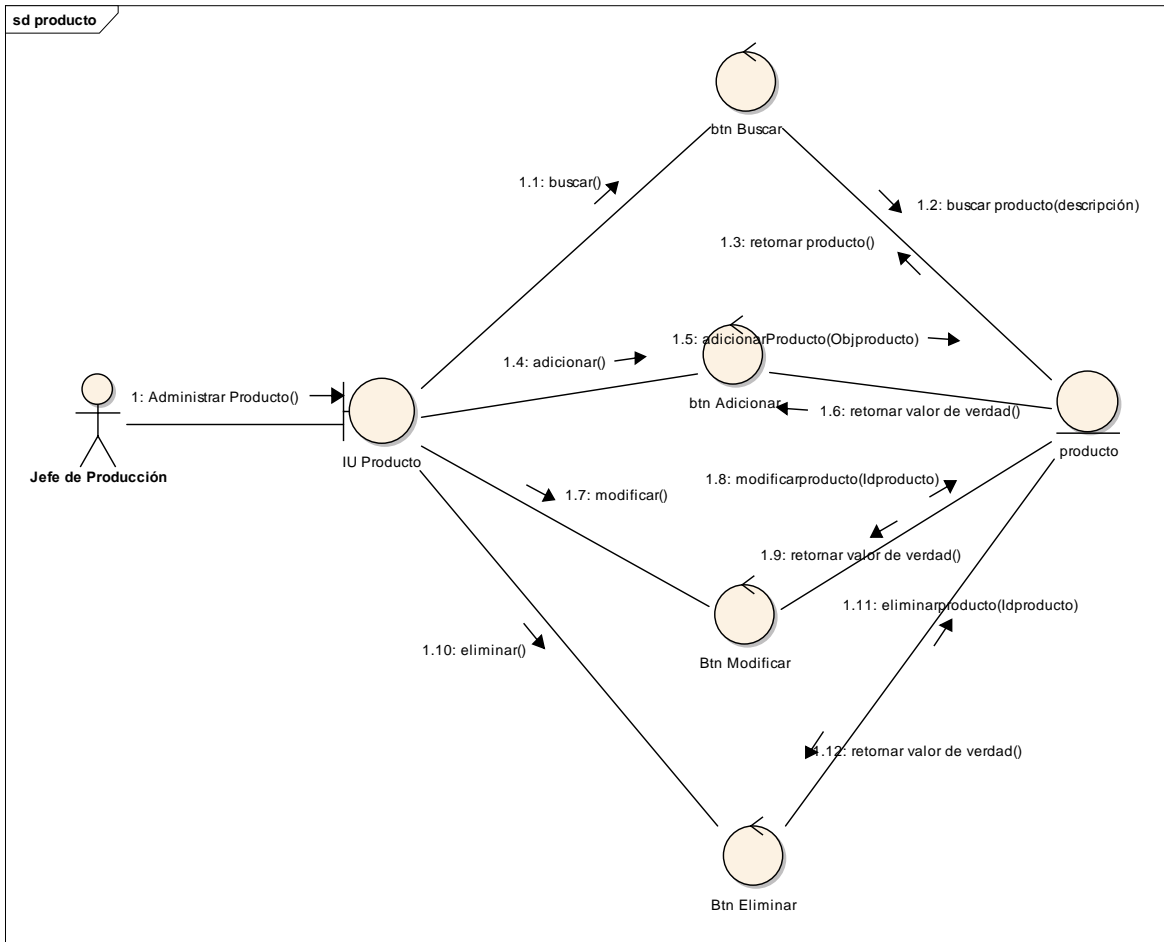


Figura N° 23: Diagrama de robustez de Administrar Producto

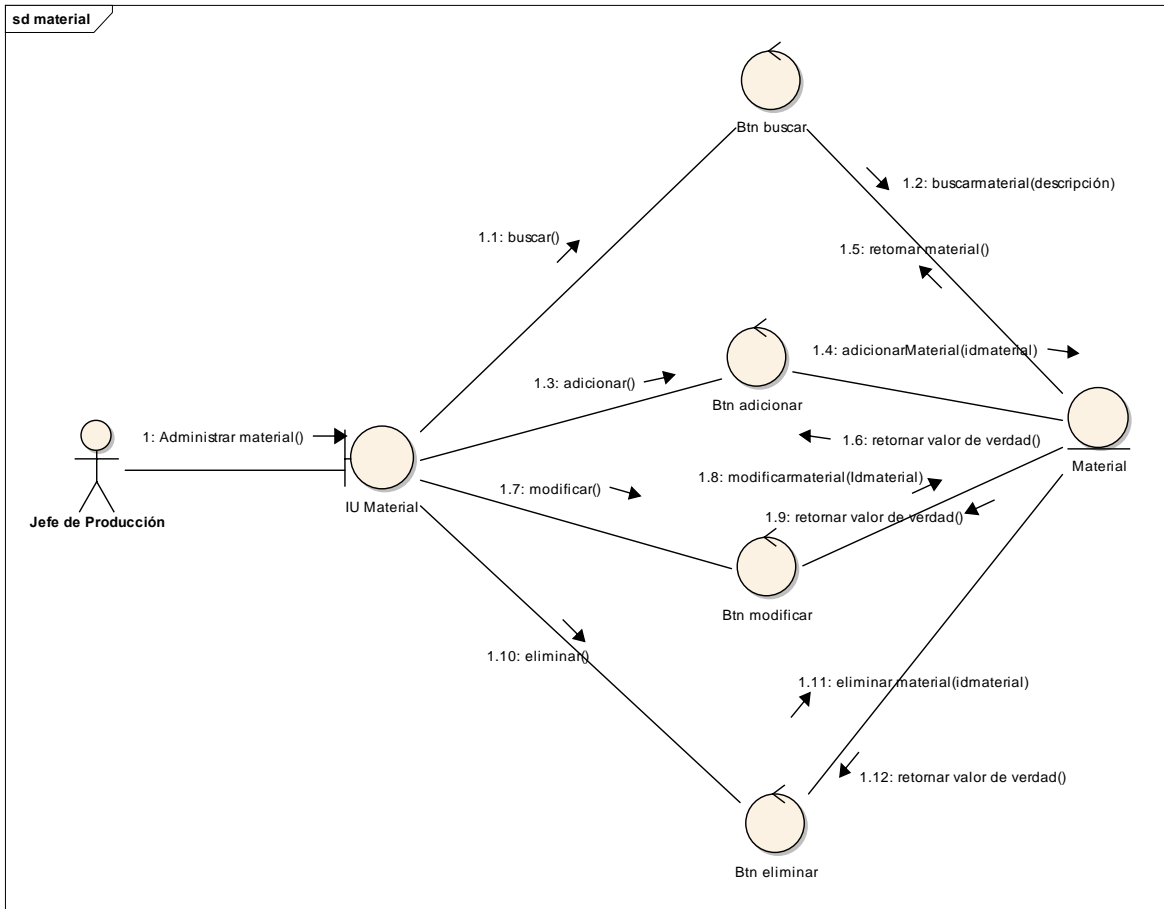


Figura N° 24: Diagrama de robustez de Administrar Material



Figura N° 25: Diagrama de robustez de Registrar Orden de Producción

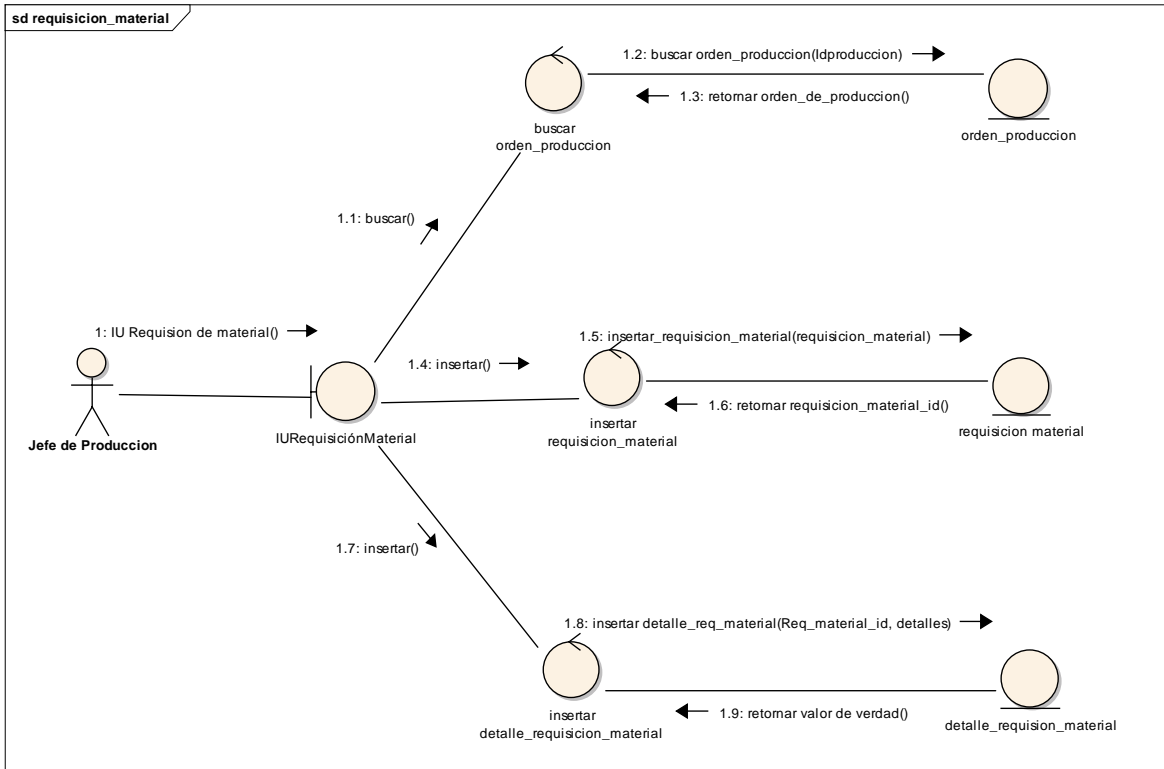


Figura N° 26: Diagrama de robustez de Requisición de Material

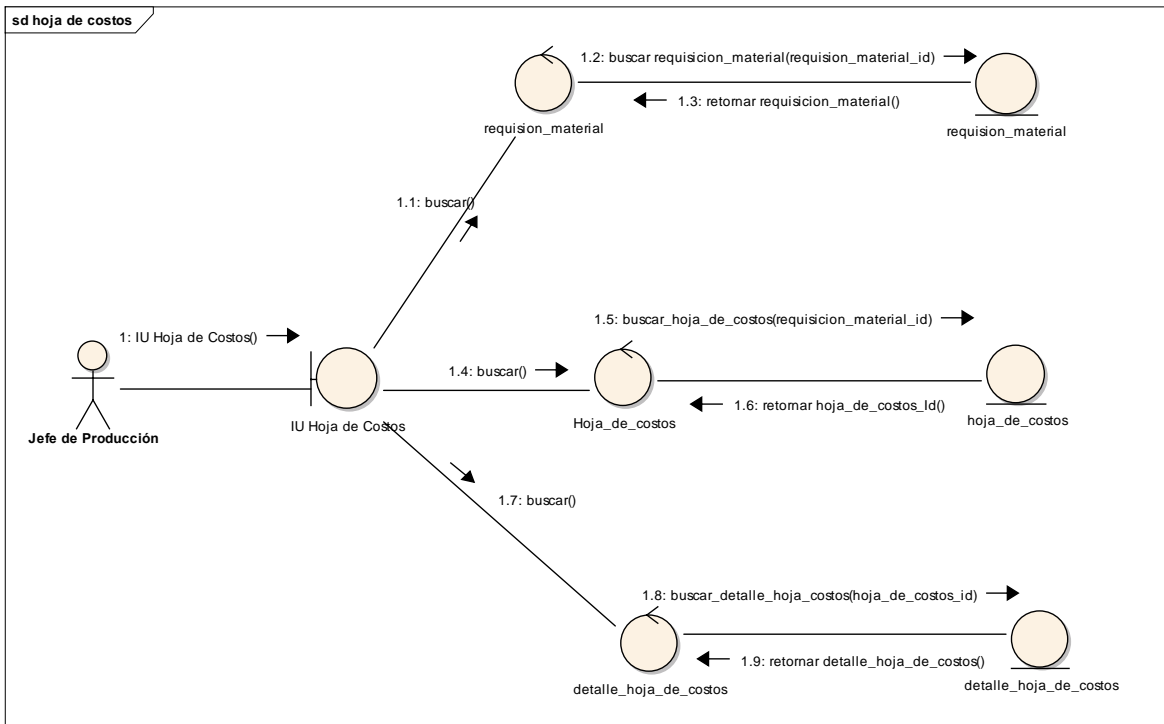


Figura N° 27: Diagrama de robustez Hoja de Costos

4.2. Diagrama de secuencia

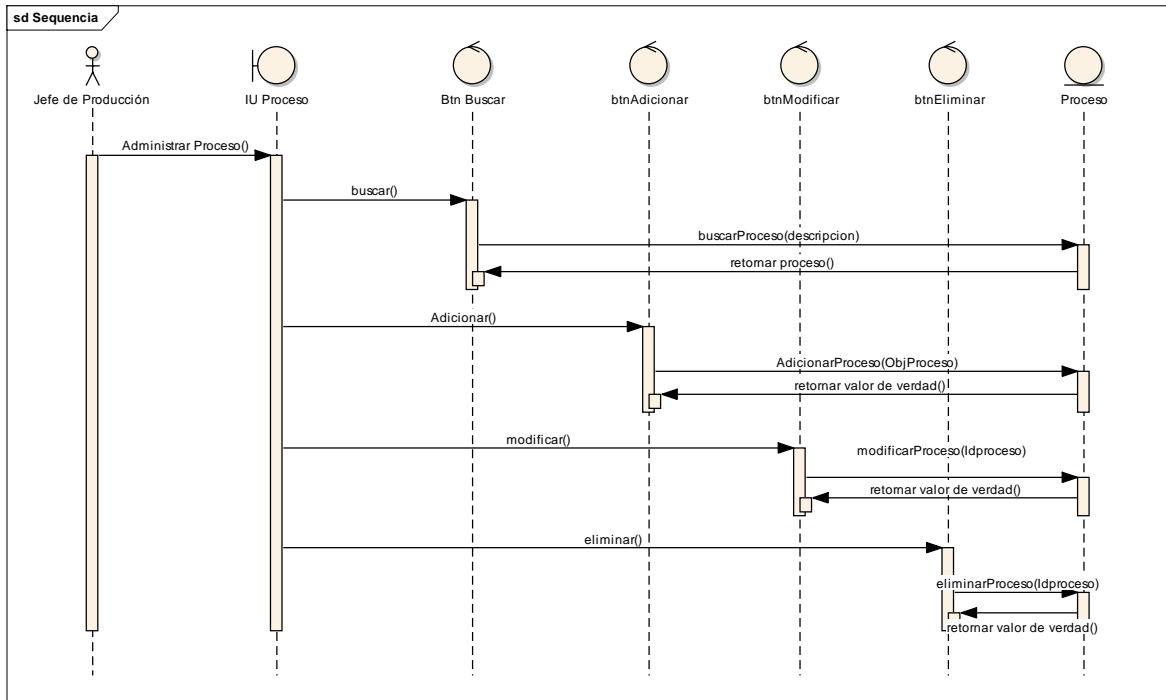


Figura N° 28: Diagrama de Secuencia Administrar Proceso

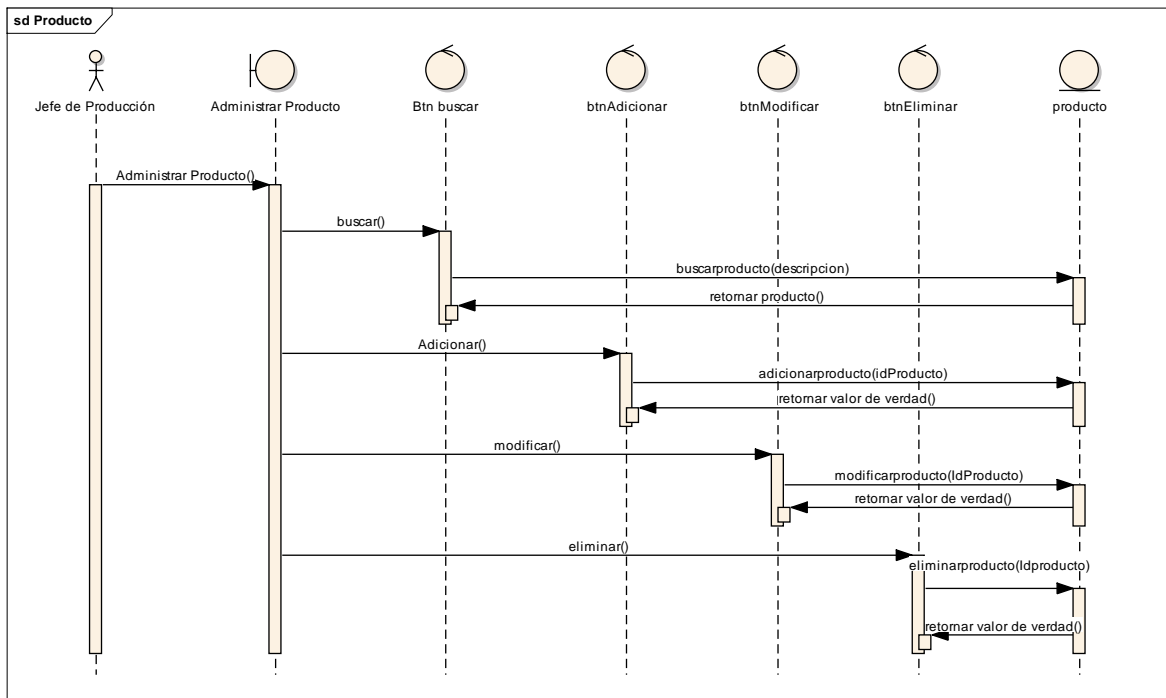


Figura N° 29: Diagrama de Secuencia Administrar Producto

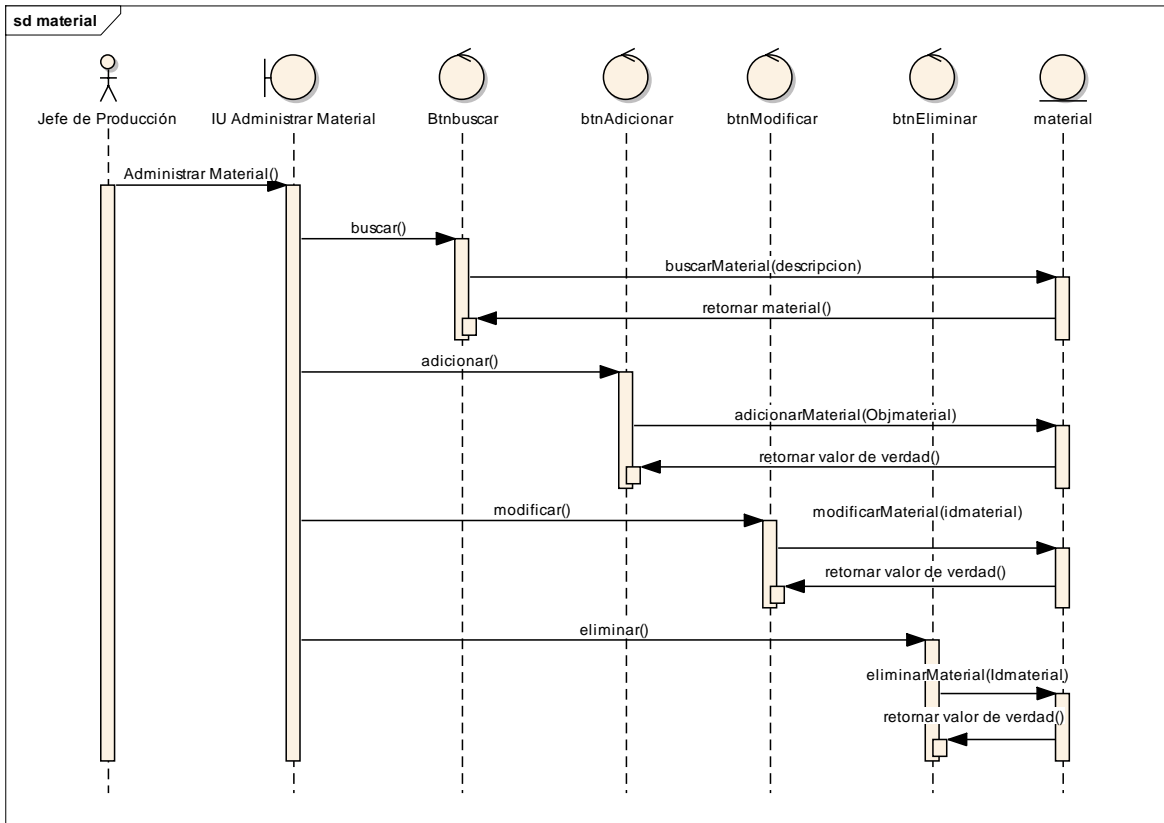


Figura N° 30: Diagrama de Secuencia Administrar Material

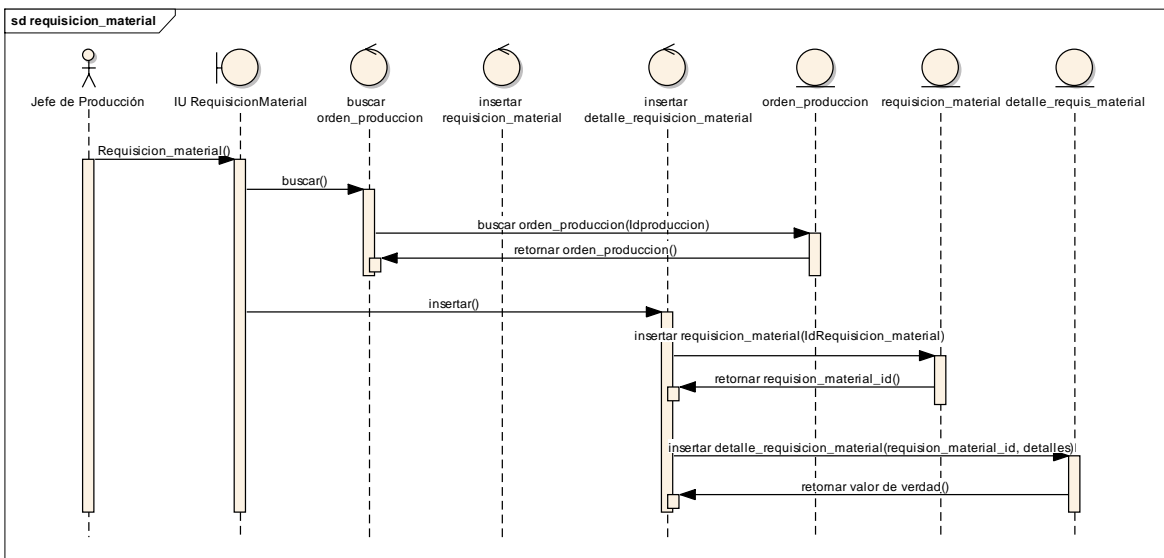


Figura N° 31: Diagrama de Secuencia Requisición de Material

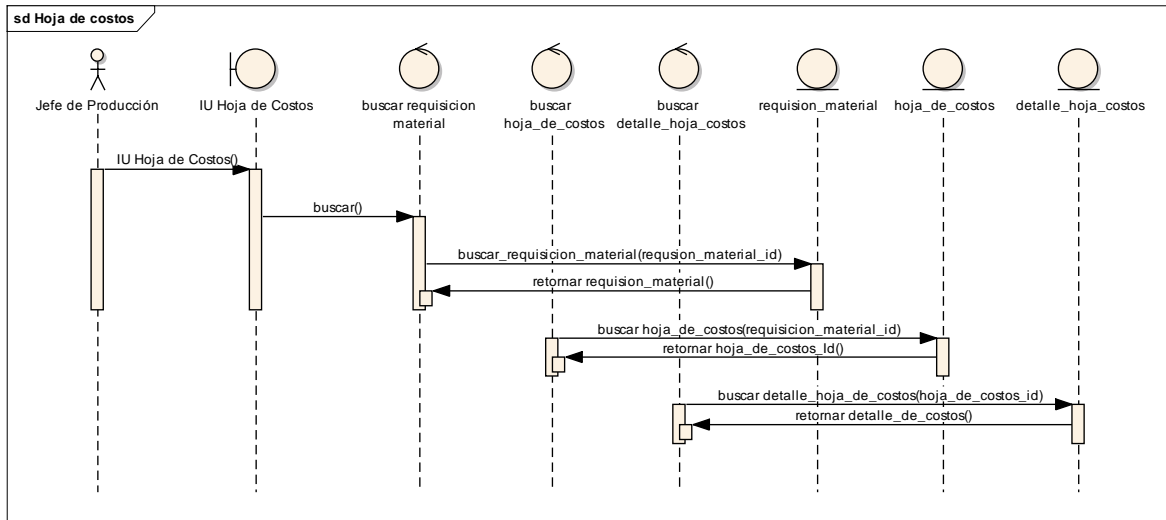


Figura N° 32: Diagrama de Secuencia Hoja de Costos

4.3. Modelo de despliegue

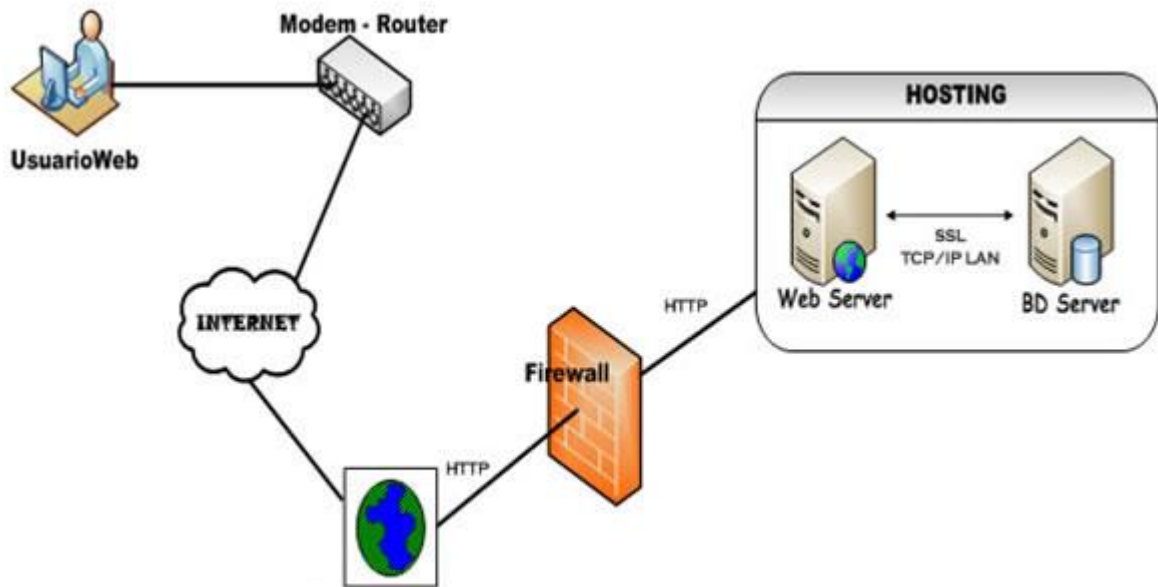


Figura N° 33: Modelo de despliegue

4.4. Modelo de componentes

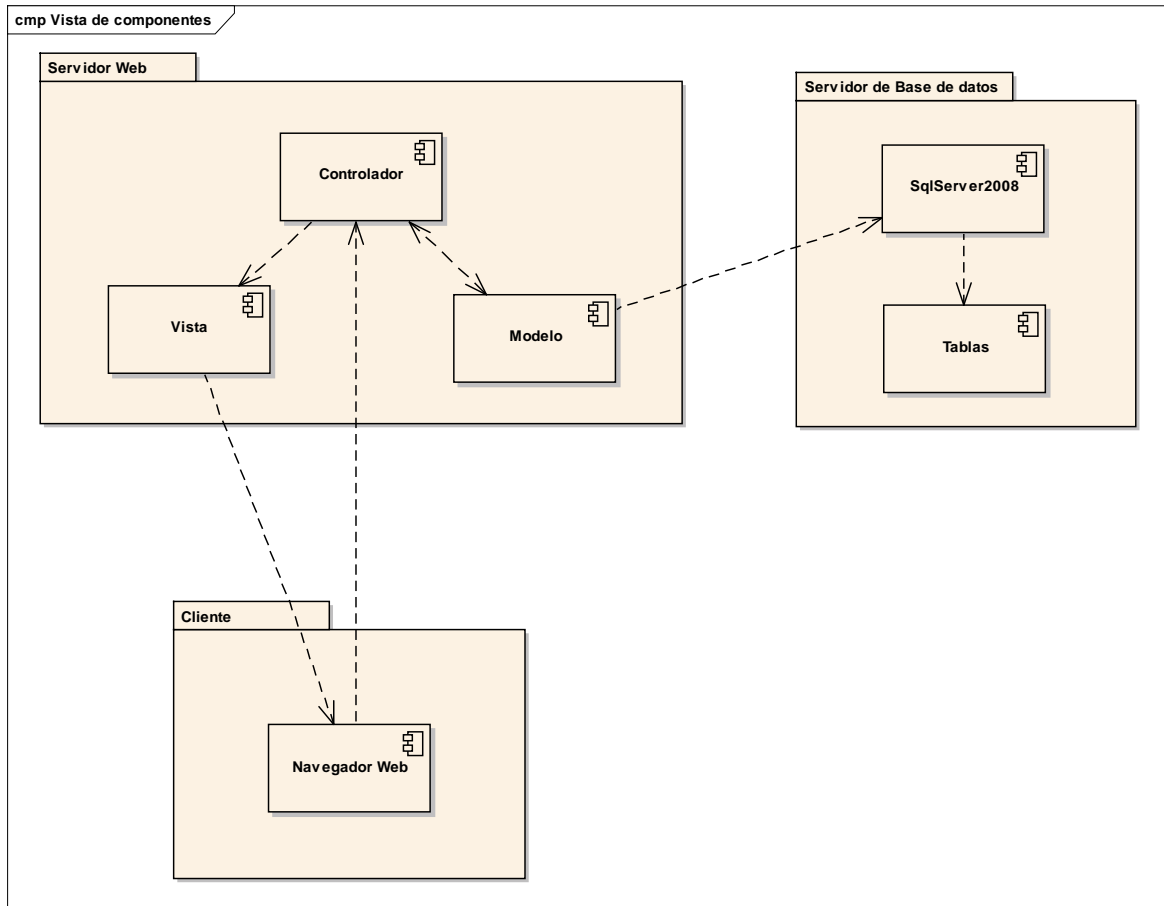


Figura N° 34: Modelo de componentes

4.5. Modelo Lógico

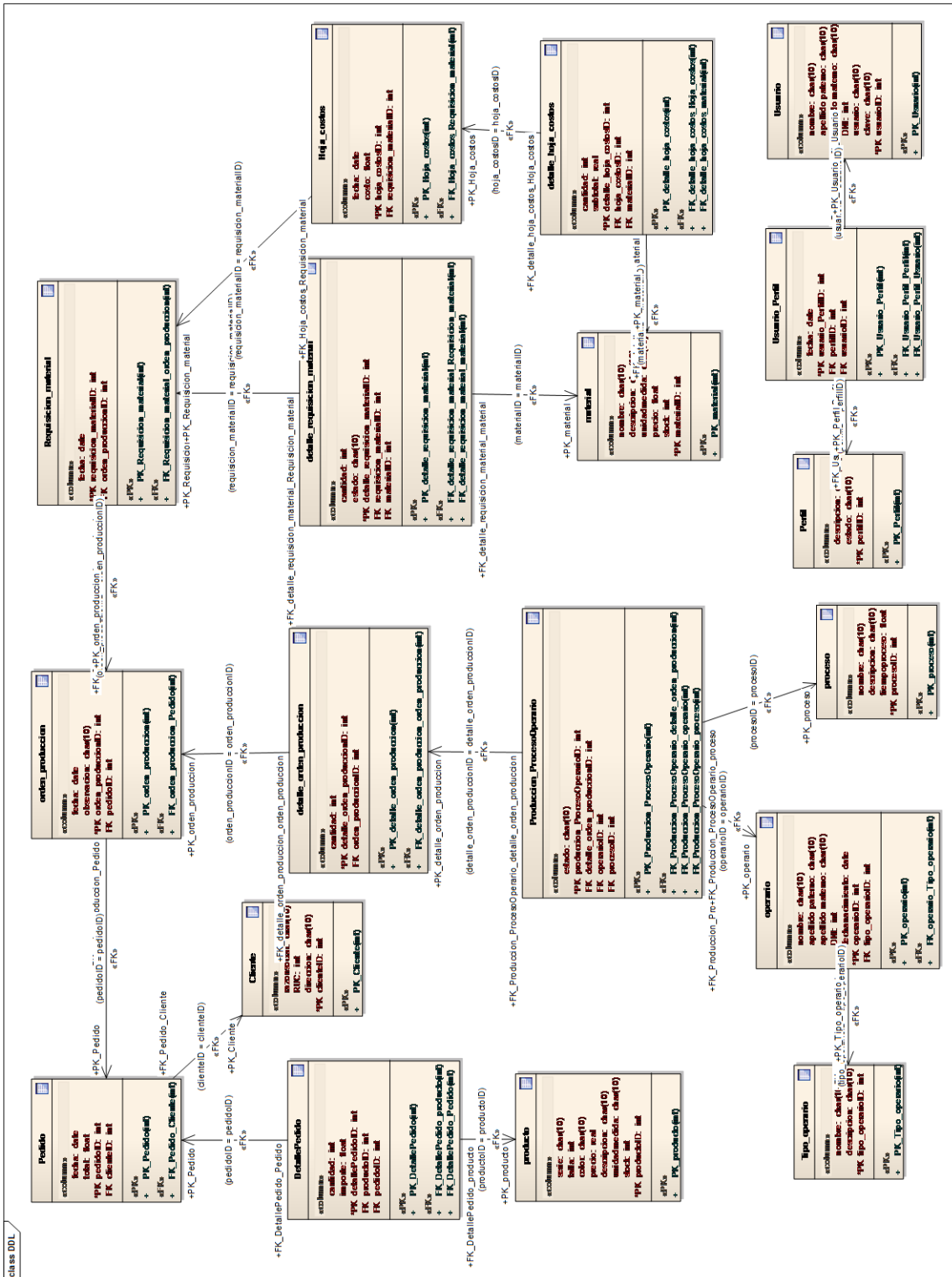


Figura N° 35: Modelo lógico

4.6.Modelo físico

```
CREATE TABLE Cliente (  
    razonsocial char(10),  
    RUC int,  
    direccion char(10),  
    clienteID int NOT NULL  
);
```

```
CREATE TABLE detalle_hoja_costos (  
    cantidad int,  
    subtotal real,  
    detalle_hoja_costosID int NOT NULL,  
    hoja_costosID int,  
    materialID int  
);
```

```
CREATE TABLE detalle_orden_produccion (  
    cantidad int,  
    detalle_orden_produccionID int NOT NULL,  
    orden_produccionID int  
);
```

```
CREATE TABLE detalle_requisicion_material (  
    cantidad int,  
    estado char(10),  
    detalle_requisicion_materialID int NOT NULL,  
    requisicion_materialID int,
```

```
materialID int
);

CREATE TABLE DetallePedido (
    cantidad int,
    importe float,
    detallePedidoID int NOT NULL,
    productoID int,
    pedidoID int
);
```

```
CREATE TABLE Hoja_costos (
    fecha date,
    costo float,
    moneda float,
    hoja_costosID int NOT NULL,
    requisicion_materialID int
);
```

```
CREATE TABLE material (
    nombre char(10),
    descripcion char(10),
    unidadmedida char(10),
    precio float,
```

```
        stock int,  
        materialID int NOT NULL  
);
```

```
CREATE TABLE operario (  
        nombre char(10),  
        apellido paterno char(10),  
        apellido materno char(10),  
        DNI int,  
        fechanacimiento date,  
        operarioID int NOT NULL,  
        tipo_operarioID int  
);
```

```
CREATE TABLE orden_produccion (  
        fecha date,  
        observacion char(10),  
        orden_produccionID int NOT NULL,  
        pedidoID int  
);
```

```
CREATE TABLE Pedido (  
        fecha date,  
        total float,  
        pedidoID int NOT NULL,  
        clienteID int  
);
```

```
CREATE TABLE Perfil (  
    descripcion char(10),  
    estado char(10),  
    perfilID int NOT NULL  
);
```

```
CREATE TABLE proceso (  
    nombre char(10),  
    descripcion char(10),  
    tiempo proceso float,  
    procesoID int NOT NULL  
);
```

```
CREATE TABLE Produccion_ProcesoOperario (  
    estado char(10),  
    produccion_ProcesoOperarioID int NOT NULL,  
    detalle_orden_produccionID int,  
    operarioID int,  
    procesoID int  
);
```

```
CREATE TABLE producto (  
    serie char(10),  
    talla int,  
    color char(10),  
    precio real,
```

```
descripcion char(10),
unidadmedida char(10),
stock int,
productoID int NOT NULL
);
```

```
CREATE TABLE Requisicion_material (
    fecha date,
    requisicion_materialID int NOT NULL,
    orden_produccionID int
);
```

```
CREATE TABLE Tipo_operario (
    nombre char(10),
    descripcion char(10),
    tipo_operarioID int NOT NULL
);
```

```
CREATE TABLE Usuario (
    nombre char(10),
    apellido_paterno char(10),
    apellido_materno char(10),
    DNI int,
    usuario char(10),
    clave char(10),
    usuarioID int NOT NULL
);
```

```
CREATE TABLE Usuario_Perfil (  
    fecha date,  
    usuario_PerfilID int NOT NULL,  
    perfilID int,  
    usuarioID int  
);
```

```
ALTER TABLE Cliente ADD CONSTRAINT PK_Cliente  
    PRIMARY KEY CLUSTERED (clienteID);
```

```
ALTER TABLE detalle_hoja_costos ADD CONSTRAINT PK_detalle_hoja_costos  
    PRIMARY KEY CLUSTERED (detalle_hoja_costosID);
```

```
ALTER TABLE detalle_orden_produccion ADD CONSTRAINT PK_detalle_orden_produccion  
    PRIMARY KEY CLUSTERED (detalle_orden_produccionID);
```

```
ALTER TABLE detalle_requisicion_material ADD CONSTRAINT  
PK_detalle_requisicion_material  
    PRIMARY KEY CLUSTERED (detalle_requisicion_materialID);
```

```
ALTER TABLE DetallePedido ADD CONSTRAINT PK_DetallePedido  
    PRIMARY KEY CLUSTERED (detallePedidoID);
```

```
ALTER TABLE Hoja_costos ADD CONSTRAINT PK_Hoja_costos  
    PRIMARY KEY CLUSTERED (hoja_costosID);
```

```
ALTER TABLE material ADD CONSTRAINT PK_material  
    PRIMARY KEY CLUSTERED (materialID);
```

```
ALTER TABLE operario ADD CONSTRAINT PK_operario  
PRIMARY KEY CLUSTERED (operarioID);
```

```
ALTER TABLE orden_produccion ADD CONSTRAINT PK_orden_produccion  
PRIMARY KEY CLUSTERED (orden_produccionID);
```

```
ALTER TABLE Pedido ADD CONSTRAINT PK_Pedido  
PRIMARY KEY CLUSTERED (pedidoID);
```

```
ALTER TABLE Perfil ADD CONSTRAINT PK_Perfil  
PRIMARY KEY CLUSTERED (perfilID);
```

```
ALTER TABLE proceso ADD CONSTRAINT PK_proceso  
PRIMARY KEY CLUSTERED (procesoID);
```

```
ALTER TABLE Produccion_ProcesoOperario ADD CONSTRAINT  
PK_Produccion_ProcesoOperario  
PRIMARY KEY CLUSTERED (produccion_ProcesoOperarioID);
```

```
ALTER TABLE producto ADD CONSTRAINT PK_producto  
PRIMARY KEY CLUSTERED (productoID);
```

```
ALTER TABLE Requisicion_material ADD CONSTRAINT PK_Requisicion_material  
PRIMARY KEY CLUSTERED (requisicion_materialID);
```

```
ALTER TABLE Tipo_operario ADD CONSTRAINT PK_Tipo_operario  
PRIMARY KEY CLUSTERED (tipo_operarioID);
```



```
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 detalle_hoja_costos ADD CONSTRAINT FK_detalle_hoja_costos_Hoja_costos  
FOREIGN KEY (hoja_costosID) REFERENCES Hoja_costos (hoja_costosID);
```

```
ALTER TABLE detalle_hoja_costos ADD CONSTRAINT FK_detalle_hoja_costos_material  
FOREIGN KEY (materialID) REFERENCES material (materialID);
```

```
ALTER TABLE detalle_orden_produccion ADD CONSTRAINT  
FK_detalle_orden_produccion_orden_produccion  
FOREIGN KEY (orden_produccionID) REFERENCES orden_produccion  
(orden_produccionID);
```

```
ALTER TABLE detalle_requisicion_material ADD CONSTRAINT  
FK_detalle_requisicion_material_Requisicion_material  
FOREIGN KEY (requisicion_materialID) REFERENCES Requisicion_material  
(requisicion_materialID);
```

```
ALTER TABLE detalle_requisicion_material ADD CONSTRAINT  
FK_detalle_requisicion_material_material  
FOREIGN KEY (materialID) REFERENCES material (materialID);
```

```
ALTER TABLE DetallePedido ADD CONSTRAINT FK_DetallePedido_producto  
FOREIGN KEY (productoID) REFERENCES producto (productoID);
```

ALTER TABLE DetallePedido ADD CONSTRAINT FK_DetallePedido_Pedido

FOREIGN KEY (pedidoID) REFERENCES Pedido (pedidoID);

ALTER TABLE Hoja_costos ADD CONSTRAINT FK_Hoja_costos_Requisicion_material

FOREIGN KEY (requisicion_materialID) REFERENCES Requisicion_material
(requisicion_materialID);

ALTER TABLE operario ADD CONSTRAINT FK_operario_Tipo_operario

FOREIGN KEY (tipo_operarioID) REFERENCES Tipo_operario (tipo_operarioID);

ALTER TABLE orden_produccion ADD CONSTRAINT FK_orden_produccion_Pedido

FOREIGN KEY (pedidoID) REFERENCES Pedido (pedidoID);

ALTER TABLE Pedido ADD CONSTRAINT FK_Pedido_Cliente

FOREIGN KEY (clienteID) REFERENCES Cliente (clienteID);

ALTER TABLE Produccion_ProcesoOperario ADD CONSTRAINT
FK_Produccion_ProcesoOperario_detalle_orden_produccion

FOREIGN KEY (detalle_orden_produccionID) REFERENCES detalle_orden_produccion
(detalle_orden_produccionID);

ALTER TABLE Produccion_ProcesoOperario ADD CONSTRAINT
FK_Produccion_ProcesoOperario_operario

FOREIGN KEY (operarioID) REFERENCES operario (operarioID);

ALTER TABLE Produccion_ProcesoOperario ADD CONSTRAINT
FK_Produccion_ProcesoOperario_proceso

FOREIGN KEY (procesoID) REFERENCES proceso (procesoID);

```
ALTER TABLE Requisicion_material ADD CONSTRAINT  
FK_Requisicion_material_orden_produccion
```

```
    FOREIGN KEY (orden_produccionID) REFERENCES orden_produccion  
(orden_produccionID);
```

```
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 producción.

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:

S₁ _____ O _____ S₂

S1: Satisfacción del usuario final con el proceso actual.

O: (Aplicación de la variable independiente) Desarrollo de un Sistema Informático Web para la Gestión de Producción de Calzados de La

Empresa Jaguar S.A.C. utilizando la metodología AUP y tecnología ASP.Net Framework MVC3.

S2: Satisfacción del usuario final con el Desarrollo de un Sistema Informático Web para la Gestión de Producción de Calzados de La Empresa Jaguar S.A.C. utilizando la metodología AUP y tecnología ASP.Net Framework MVC3.

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 sistema le ayuda a la oportuna toma de decisiones en la producción.	27	64	37	1369
Lleva un efectivo control de la producción.	36	82	46	2116
Tiene conocimiento de la cantidad de materiales a utilizar en la producción.	45	73	28	784
Cumple con la producción de los pedidos requeridos por sus clientes.	18	82	64	4096
Tiene conocimiento de los costos que demanda la producción.	9	91	82	6724
	Totales		257	15089

Tabla 12: Variación de datos históricos del indicador Satisfacción del usuario final
Fuente: [Elaboración Propia]

$$\sum d_i = 257$$

$$\sum d_i^2 = 15089$$

$$\bar{d}_i = \frac{\sum d_i}{n} = \frac{257}{5} = 51.4$$

Encontrando la desviación estándar:

$$s_d = \sqrt{\frac{n \sum d_i^2 - (\sum d_i)^2}{n(n-1)}} = \sqrt{\frac{5(15089) - (257)^2}{5(5-1)}} = 21.6$$

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:

$$t_o = \frac{\bar{d}_i}{s_d} \sqrt{n}$$

Valor Crítico de “t” de student:

$$t(1-\alpha)(n-1)$$
$$t(1-0.05)(5-1)$$
$$t(0.95)(4) = 3.8$$
$$t_1 = 3.8$$

$$t_0 = \frac{51.4}{21.6} \sqrt{5} = 5.3$$

Región de aceptación o rechazo:

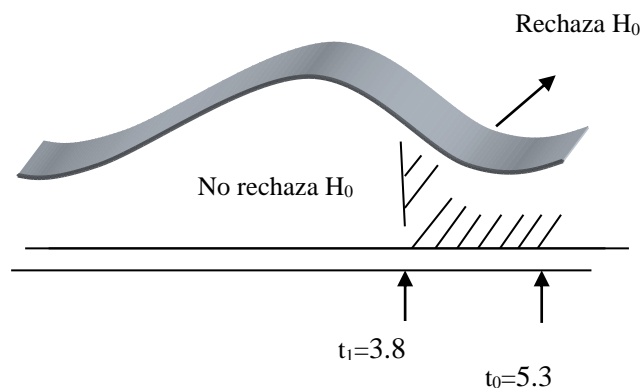


Figura N° 36: Región de aceptación o rechazo
Fuente: [Elaboración Propia]

Se concluye $t_0 = 5.3$ es mayor que $t_1 = 3.8$

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 producción de calzados ha 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 Producción.

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 producción de calzados de la Empresa Jaguar S.A.C. utilizando la metodología AUP y tecnología ASP.Net framework MVC3.

T2: Tiempo en hh:mm:ss (horas:minutos:segundos) que toma realizar los requerimientos funcionales con el Sistema Informático Web para la Gestión de producción de calzados de la Empresa Jaguar S.A.C. utilizando la metodología AUP y tecnología ASP.Net framework MVC3. (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 Usuario	00:01:50	00:00:50	60	3600
Registrar Operario	00:01:40	00:00:55	45	2025
Registrar Proceso	00:01:50	00:00:55	55	3025
Registrar Producto	00:01:55	00:00:45	70	4900
Registrar Orden de Producción	00:02:10	00:00:50	80	6400
Registrar Requisición de Material	00:01:55	00:00:50	65	4225
Elaborar Reportes de Hoja de Costos	00:01:50	00:00:55	55	3025
		Totales	430	27200

Tabla 13: Variación de datos históricos del indicador Tiempo
Fuente: [Elaboración Propia]

$$\sum d_i = 430$$

$$\sum d_i^2 = 27200$$

$$\bar{d}_i = \frac{\sum d_i}{n} = \frac{430}{7} = 61.4$$

Encontrando la desviación estándar:

$$s_d = \sqrt{\frac{n \sum di^2 - (\sum di)^2}{n(n-1)}} = \sqrt{\frac{7(27200) - (430)^2}{7(7-1)}} = 11.4$$

Hipótesis estadística:

$H_0: O_1 - O_2 = 0$ Hipótesis nula

No hay diferencia alguna entre dos muestras.

$H_1: O_2 - O_1 > 0$ Hipótesis alternativa

Si hay diferencia

Nivel de significancia $\alpha = 0.05$

Función de prueba:

$$t_o = \frac{\bar{d}_i}{s_d} \sqrt{n}$$

Valor Crítico de “t” de student:

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

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

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

$$t_1 = 5.7$$

$$t_o = \frac{61.4}{11.4} \sqrt{7} = 14.2$$

Región de aceptación o rechazo:

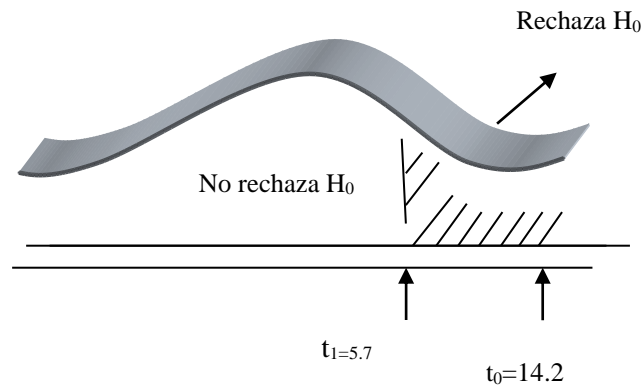


Figura N° 37: Región de aceptación o rechazo

Fuente: [Elaboración Propia]

Se concluye $t_0 = 14.2$ 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 para la Gestión de producción de calzados de la Empresa Jaguar S.A.C. utilizando la metodología AUP y tecnología ASP.Net framework MVC3 *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 GESTIÓN DE PRODUCCIÓN DE CALZADOS DE LA EMPRESA JAGUAR S.A.C. UTILIZANDO LA METODOLOGÍA AUP Y TECNOLOGÍA ASP.NET FRAMEWORK MVC3”

Es ACEPTADA.

REFERENCIAS BIBLIOGRAFICAS

Alarcón, V. F. (2006). *Desarrollo de Sistemas de Información*. Catalunya: UPC Editorial.

Alvarez, M. A. (s.f.). *Manual del framework ASP.NET MVC*. Obtenido de <http://www.desarrolloweb.com>

Ambler, S. (2005). www.ambyssoft.com/unifiedprocess/.

De Pablos Heredero, C. (2006). *Dirección y Gestión de los Sistemas de Información en la Empresa*. Madrid: ESIC Editorial.

Desarrollo (ASP.NET). (2013). Obtenido de [http://msdn.microsoft.com/es-es/library/vstudio/ms178093\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/vstudio/ms178093(v=vs.100).aspx)

Enterprise Architect - Herramienta de diseño UML. (2014). Obtenido de <http://www.sparxsystems.com.ar/products/ea.html>

García Márquez, F. (2013). *Dirección y Gestión de la Producción*. Barcelona.

González Riesco, M. (2006). *Gestión de la Producción*. Ideaspropias.

Hansmann, T. a. (2010). *Agile Software Development*. Berlin.

Hobs, L. (1999). *Designing Internet Home Pages Made Simple*. Butterworth Heinemann.

Laurent Debrauwer, E. (2010). *UML 2: Practique la modelación*.

Misner, R. M. (2010). *Introducing Microsoft SQLServer 2008*. Online Training Solutions, Inc.

Pérez, M. Á. (2006). *GESTIÓN DE LA PRODUCCIÓN*.

Quiron. (2005). *Introducción a UML 2.0*. Obtenido de http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15

Scott, M. F. (1999). *UML Gota a Gota*. Addison Wesley Longman.

StarUML. (2005). Obtenido de <http://staruml.sourceforge.net/v1/>

Vilcarromero Ruiz, R. (2013). *Administración de la Producción*. Fundación Universitaria Andaluza Inca Garcilazo.

(2013). Obtenido de ASP.NET MVC vs ASP.NET WebForms: <http://www.vitaminasdev.com/Recursos/8/asp-net-mvc-vs-asp-net-webforms>

CONCLUSIONES

1. Se realizó la Investigación Bibliográfica sobre la Metodología AUP, tecnología ASP.Net y Sistemas de información Web.
2. Se determinó e identificó con el desarrollo de la metodología AUP, QUINCE (15) 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 DIECIOCHO (18) clases.
 - DIECISEIS (10) Casos de Uso y OCHO (8) Diagramas de Robustez principales.
4. Utilizando SQLServer2008 como gestor de base de datos se obtuvieron DIECIOCHO (18) tablas.
5. Con la Tecnología.Net se construyó las Vistas del software.

RECOMENDACIONES

- Se recomienda usar AUP como metodología para estos tipos de desarrollo de sistemas informáticos web de gestión de producción; por ser el más adaptable y apropiado para proporcionar los entregables con el menor tiempo y esfuerzo posible en comparación a otras metodologías clásicas; además de facilitar respuestas rápidas a cambios de requisitos a lo largo del desarrollo del sistema.
- Realizar un plan de reuniones con el encargado del área para la evaluación de procesos que se realizan en la gestión para así obtener un mejor análisis de procesos y requerimientos de usuario.
- Se recomienda crear índices y realizar periódicamente limpieza de tablas históricas a fin de mejorar el rendimiento de SQLServer2008.

ANEXO

ANEXO 01 - ANÁLISIS DE RESULTADOS DE ENCUESTA

Tabulación por pregunta y opción:

Análisis en el Planteamiento Actual

Pregunta 1

¿El sistema le ayuda a la oportuna toma de decisiones en la producción?

	# Personal Empresa	%
SI	3	27
NO	8	73
Total	11	100.00

Pregunta 2

¿Lleva un efectivo control de la producción?

	# Personal Empresa	%
SI	4	36
NO	7	64
Total	11	100.00

Pregunta 3

¿Tiene conocimiento de la cantidad de materiales a utilizar en la producción?

	# Personal Empresa	%
SI	5	45
NO	6	55
Total	11	100.00

Pregunta 4

¿Cumple con la producción de los pedidos requeridos por sus clientes?

	# Personal Empresa	%
SI	2	18
NO	9	82
Total	11	100.00

Pregunta 5

¿Tiene conocimiento de los costos que demanda la producción?

	# Personal Empresa	%
SI	1	9
NO	10	91
Total	11	100.00

Tabulación por pregunta y opción:

Análisis en Planteamiento Propuesto

Pregunta 1

¿El sistema le ayuda a la oportuna toma de decisiones en la producción?

	# Personal Empresa	%
SI	7	64
NO	4	36
Total	11	100.00

Pregunta 2

¿Lleva un efectivo control de la producción?

	# Personal Empresa	%
SI	9	82
NO	2	18
Total	11	100.00

Pregunta 3

¿Tiene conocimiento de la cantidad de materiales a utilizar en la producción?

	# Personal Empresa	%
SI	8	73
NO	3	27
Total	11	100.00

Pregunta 4

¿Cumple con la producción de los pedidos requeridos por sus clientes?

	# Personal Empresa	%
SI	9	82
NO	2	18
Total	11	100.00

Pregunta 5

¿Tiene conocimiento de los costos que demanda la producción?

	# Personal Empresa	%
SI	10	91
NO	1	9
Total	11	100.00