

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**SISTEMA DE ENTRADAS Y SALIDAS E INVENTARIO  
Caso: BOLITAL S.R.L.**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: CLAUDIA CHIRI HONORIO**  
**TUTOR: LIC. EFRAIN SILVA SANCHEZ**  
**REVISOR: LIC. CARMEN ROSA HUANCA QUISBERT**

LA PAZ - BOLIVIA  
2009

Dedicado:

En especial con mucho cariño a mi madre  
Elena que me apoyo siempre  
incondicionalmente todos los días de mi vida  
dándome su invaluable amor.

A mis hermanos Marcos, Luz, Boris, Lili,  
Gonzalo y Pamela que siempre estuvieron  
ahí para ayudarme

Claudia

## **Agradecimiento**

El Proyecto de Grado que presento, no se hubiera podido realizar sin el apoyo y colaboración de todas aquellas personas que me brindaron su amistad incondicional a lo largo de mis estudios.

Al Lic. Efraín Silva Sánchez, quien como docente tutor, quien me apoyo en la conclusión de este proyecto de grado.

A la Lic. Carmen Rosa Huanca Quisbert, revisora del proyecto de grado, quien me brindo su orientación a través de sus revisiones, impulsando al desarrollo y conclusión de este proyecto.

Al personal de la empresa Bolital S.R.L. quienes me colaboraron y confiaron siempre con el desarrollo de este proyecto, a quien doy mi más sincera gratitud.

A todas las personas que me apoyaron y colaboraron durante la elaboración del presente trabajo.

## RESUMEN

Se ha observado que en la empresa Bolital S.R.L., se tenían problemas en el manejo de la información de todas las actividades que desarrollan debido a que no contaban con un sistema automatizado para todas las actividades ya sean para las entradas o salidas de productos de la empresa.

Se logro un sistema para la empresa el cual para el desarrollo se utilizo la metodología orientada a objetos RUP (Proceso Unificado Racional) y haciendo uso del lenguaje modelo UML, para implementar el sistema se utilizo el lenguaje de programación PHP, la base de datos MySQL y servidor Apache que se adecuan a las necesidades e implementación de la empresa

El presente proyecto de grado presenta una alternativa para contribuir a la empresa Bolital S.R.L. tener un mejor manejo de todos los procesos que desarrolla la empresa como ser entradas y salidas y tener información actualizado de los inventarios de todos los productos y poder realizar reportes detallados de todas las actividades que realiza ya que será de gran utilidad para la empresa.

# ÍNDICE

## CAPÍTULO 1

### PRESENTACIÓN

1.1 Introducción	1
1.2 Antecedentes	2
1.2.1 De la Institución	2
1.3 Del proyecto	3
1.4 Problemática	3
1.4.3 Objetivos	4
1.4.1 Objetivo General	4
1.4.2 Objetivos Específicos	4
1.5 Justificación	5
1.5.1 Justificación Técnica	5
1.5.2 Justificación Económica	5
1.5.3 Justificación Social	6
1.6 Aporte	6
1.6.1 Aporte Teórico	6
1.6.2 Aporte Práctico	6
1.7 Alcance y Límites	6
1.8 Metodología	7
1.9 Herramientas	7

## CAPÍTULO 2

### MARCO TEÓRICO

2.1 Introducción	8
2.2 Modelos de Inventarios	8
2.2.1 Clasificación de Inventarios	9
2.2.2 Modelo Determinista	10
2.2.3 Técnica de valuación para la salida de almacenes	12
2.3 Ingeniería de software	12

2.4 Lenguaje Unificado de Modelado (UML)	14
<b>2.5 Metodología RUP</b>	<b>14</b>
2.5.1 Proceso dirigido por Casos de Uso	15
2.5.2 Proceso centrado en la arquitectura	16
2.5.3 Proceso iterativo e incremental	18
2.5.4 Estructura del proceso	21
2.5.5 Estructura Dinámica del proceso. Fases e iteraciones	22
2.5.5.1 Fase Inicio	24
2.5.5.2 Fase Elaboración	24
2.5.5.3 Fase Construcción	26
2.5.5.4 Fase Transición	27
2.5.6 Estructura Estática del proceso	28
2.5.6.1 Roles	29
2.5.6.2 Actividades	29
2.5.6.3 Artefactos	30
Flujos de trabajo	30
Requisitos	30
Modelo del negocio	33
2.5.6.4.3 Análisis	36
2.5.6.4.4 Diseño	38
Implementación	41
2.5.6.4.6 Pruebas	43
2.6 Herramientas de Desarrollo de Software	44
2.7 Métricas Calidad de Software	47
2.7.1 Métricas Calidad de Software ISO 9126	48

## **CAPÍTULO 3**

### **MARCO APLICATIVO**

3.1 Introducción	53
3.2 Fase Inicio	54
3.2.1 Recopilación de requisitos	54

3.2.2 Casos de uso	57
3.2.3 Análisis	60
3.2.4 Diseño	62
3.3 Fase de elaboración	63
3.3.1 Recopilar requisitos	63
3.3.2 Análisis	65
3.3.3 Diseño	67
3.3.4 Implementación	71
3.3.5 Pruebas	73
3.4. Fase Construcción	73
3.4.1 Recopilar requisitos	74
3.4.2 Análisis	74
3.4.3 Diseño	75
3.4.4 Implementación	76
3.4.5 Pruebas	77
3.5 Fase Transición	80
 <b>CAPÍTULO 4</b>	
<b>CALIDAD DEL SOFTWARE</b>	
4.1 Introducción	81
4.2 Funcionalidad	81
4.3 Confiabilidad	84
4.4 Eficiencia	84
4.5 Portabilidad	85
4.6 Mantenibilidad	85
 <b>CAPÍTULO 5</b>	
<b>CONCLUSIONES Y RECOMENDACIONES</b>	
5.1 Conclusiones	87
5.2 Recomendaciones	88

## **BIBLIOGRAFÍA**

### **ANEXOS**

**ANEXO A:** Aspectos legales de la empresa

**ANEXO B:** Productos Bolital S.R.L.

**ANEXO C:** Proveedores de la empresa

**ANEXO D:** Casos de uso detallados

**ANEXO E:** Análisis de casos de uso

**ANEXO F:** Diseño de casos de uso

**ANEXO G:** Prototipo de interfaz de usuario

**ANEXO H:** Pruebas del sistema





## ÍNDICE DE FIGURAS

<b>Figura. N° 2.1:</b> Modelo estático de un solo artículo	11
<b>Figura N° 2.2:</b> Capas de la ingeniería de software	13
<b>Figura N° 2.3:</b> Los Casos de Uso integran el trabajo	16
<b>Figura N° 2.4:</b> Trazabilidad a partir de los Casos de Uso	16
<b>Figura N° 2.5:</b> Evolución de la arquitectura del sistema	18
<b>Figura N° 2.6:</b> Una iteración RUP	19
<b>Figura N° 2.7:</b> Esfuerzo en actividades según fase del proyecto	20
<b>Figura N° 2.8:</b> Estructura de RUP	21
<b>Figura N° 2.9:</b> Ciclos, releases, baseline	22
<b>Figura N° 2.10:</b> Fases e hitos en RUP	23
<b>Figura N° 2.11:</b> Distribución típicas de esfuerzo y tiempo	24
<b>Figura N° 2.12:</b> Distribución típica de recursos humanos	24
<b>Figura N° 2.13:</b> Relación entre roles, actividades, artefactos	29
<b>Figura N° 2.14:</b> Diagrama de Clase	32
<b>Figura N° 2.15:</b> Diagrama de Casos de Uso	34
<b>Figura N° 2.16:</b> Caso de Uso Detallado	35
<b>Figura N° 2.17:</b> Identificación de paquetes del análisis a partir de los casos de uso	37
<b>Figura N° 2.18:</b> Diagrama de Colaboración del caso de uso: Registrar Clientes	38
<b>Figura N° 2.19:</b> Diagrama de Secuencia	40
<b>Figura N° 2.20:</b> Elemento de diagrama de componente	42
<b>Figura N° 2.21:</b> Elemento de diagrama de despliegue	42
<b>Figura N° 2.22:</b> Cálculo de punto de función para una función	50
<b>Figura N° 3.1:</b> Entrada de Productos	53
<b>Figura N° 3.2:</b> Salida de Productos	54
<b>Figura N° 3.3:</b> Modelado de Dominio	55
<b>Figura N° 3.4:</b> Modelado de caso de uso del negocio general	56
<b>Figura N° 3.5:</b> Diagrama de Casos de Uso – SESI	58

<b>Figura N° 3.6:</b> Paquete del análisis	61
<b>Figura N° 3.7:</b> Diagrama de Colaboración del caso de uso: Registrar Proveedor	62
<b>Figura N° 3.8:</b> Nodos y configuraciones de la red	63
<b>Figura N° 3.9:</b> Dependencias y capas de paquetes del análisis	65
<b>Figura N° 3.10:</b> Subsistemas identificados a partir de paquete del análisis	67
<b>Figura N° 3.11:</b> Capa específica y general de la aplicación	68
<b>Figura N° 3.12:</b> Capa Intermedia y Capa de Software del Sistema	69
<b>Figura N° 3.13:</b> Dependencias y capas de los subsistemas	69
<b>Figura N° 3.14:</b> Diagrama de secuencia: Registro de proveedores	70
<b>Figura N° 3.15:</b> Operaciones de la clase proveedor	70
<b>Figura N° 3.16:</b> Componentes a partir del diseño de la arquitectura	72
<b>Figura N° 3.17:</b> Diagrama de Clases: Sistema de Entrada y Salidas e Inventario	76
<b>Figura N° 3.18:</b> Grafo del proceso de búsqueda en kardex	79

# CAPÍTULO 1

## PRESENTACIÓN

### 1.2 Introducción

Hoy en día las empresas no pueden dejar de lado los sistemas de información, en especial aquellos que manejan gran cantidad de datos porque crece la complejidad al controlar y hacer un seguimiento de toda esa información. Este tipo de información se convierte en un activo muy importante y valioso dentro de todas las instituciones.

Un negocio o una industria por lo común mantienen un inventario razonable de sus productos para asegurar una operación interrumpida.

La naturaleza del problema de inventarios consiste en hacer y recibir pedidos de determinados volúmenes, repetidas veces y a intervalos determinados. Un factor importante en la formulación y la solución de un modelo de inventarios es que la demanda (por tiempo de unidad) de un artículo puede ser determinista (todos los componentes se suponen conocidas y constantes) o probabilísticas (si es que al menos uno de los componentes es aleatorio).

Mantener un sistema de inventarios (existencia de bienes) para su venta o uso futuro es una práctica común en el mundo de los negocios. Las empresas de venta al menudeo los mayoristas, los fabricantes por lo general almacenan bienes o artículos. Con frecuencia utilizan una computadora para mantener un registro de los niveles de inventario y señalar cuando reabastecer, que facilitan la operación a empresas pueden ser estas grandes o pequeñas.

Bolital S.R.L. es una empresa privada, cuenta con una unidad de administración en cuya dependencia se encuentra la unidad de almacén, está a su cargo la entrada y salida de productos terminados que son la Almendra (Castaña) y sus variedades.

Es una necesidad incorporar un sistema de inventarios puesto que en el manejo de información es una ventaja, manejan a niveles óptimos reduciendo los gastos en hacer inspecciones y tiempo para poder tener la información en cualquier momento según lo requiera la administración.

## **1.2 Antecedentes**

### **1.2.1 De la Institución**

Bolital S.R.L. fue creada en Enero de 1997 con recursos propios en la ciudad de Riberalta, provincia Vaca Diez del departamento del Beni en Bolivia; la actividad que realiza la empresa va desde el Acopio<sup>1</sup> de almendra en cáscara hasta el procesamiento de la misma, Almendra (castaña) mayormente conocida como BRAZILIAN NUTS [Anexo B].

Bolital S.R.L. es una empresa dedicada a la comercialización y exportación de la almendra (castaña), producto que es recolectado en un 75% en el Departamento de Pando y el 25% de la provincia Vaca Diez, su producto es 100% de exportación [Anexo C].

La empresa almacena productos de almendra convencional y almendra orgánica la cual cuenta con todos los permisos requeridos tanto para el producto como para la exportación. La unidad de almacenes Bolital S.R.L. se encuentra ubicada en la ciudad del El Alto departamento de La Paz carretera a Oruro.

Actualmente la empresa realiza gran parte de sus actividades de control de forma manual, ocasionando demoras en los registros respectivos que llevan.

---

<sup>1</sup> Acción y efecto de acopiar (Juntar, reunir en cantidad algo, y más comúnmente granos)

### **1.2.2 Del proyecto**

En la Universidad Mayor de San Andrés Facultad de Ciencias Puras y Naturales y en la Carrera de Informática se tiene algunos Proyectos de Grados que están relacionados con el presente, estos son:

Proyecto de Grado “Control de ventas e inventarios para Multicon S.R.L.” su objetivo general es implementar un sistema de control de ventas e inventarios para la empresa. El proyecto abarca el manejo de almacenes y ventas pero se enfoca más en la distribución de productos.

Proyecto de grado “Sistema de Información de Compras e Inventarios SAMA” su objetivo general es desarrollar e implementar un Software que pueda brindar información de las actividades de la fábrica, mediante procesos de recolección de datos brindando información rápida y oportuna para la toma de decisiones de la Gerencia.

“Sistema de información y control de almacenes E.M.M.U.” su propósito fue desarrollar e implementar un sistema computarizado que permita automatizar los procesos de ingreso, salida, despacho y control de bienes de la institución.

“Sistema de información para el control en la gestión y administración de almacenes en INLASA” se desarrollo para el instituto Nacional de Laboratorios de Salud su propósito implementar un sistema de información para la unidades de almacenes, que facilite el registro, actualización y control de movimientos de materiales y reactivos basados en las normas de la institución.

### **1.7 Problemática**

En la unidad de almacenes de la empresa Bolital S.R.L., en todas las actividades que lleva acabo la empresa, se viene utilizando un sistema manual de registro, en distintos libros, ya sea para la recepción de productos terminados o de salida de los mismos.

La rutina en la recepción de productos, deben controlar y registrar su origen (proveedores), procedencia, mas el conjunto de papeles de constancia como el permiso de la superintendencia forestal, certificado del SENASAG, etc., el manejo del sistema manual provoca demora en la elaboración y entrega de la información actualizada que sea capaz de indicar la cantidad que se tiene en almacenes [Anexo A].

- No se dispone de información oportuna acerca del destino de los productos
- No se dispone de información oportuna referente al ingreso de productos.
- No cuentan con un sistema de inventarios para que les proporcione información actualizada.

Existe un gran volumen de información referente a entradas y salidas e inventario de productos, que ocasiona demoras en los procesos de búsqueda de información y retrasos en la omisión de informes debido a que no se cuenta con registros adecuados, ni actualizaciones para el control de movimiento de productos en las unidades de almacenes de Bolital S.R.L.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar e implementar un sistema de entradas y salidas e inventario para la empresa Bolital S.R.L. de tal forma que se mejoren los procesos operativos y administrativos, brindando a la empresa información actualizada precisa sobre los productos.

### **1.4.2 Objetivos Específicos**

- Diseñar y desarrollar un subsistema de información para el control de las entradas y salidas de los productos.
- Aplicar el modelo de inventario determinístico para obtener un pronóstico acerca de los productos que se tiene en almacenes.



- Diseñar módulos que permita realizar consultas sobre todo la información respecto a los productos que tiene en la empresa (stock).
- Diseñar y desarrollar un sistema de información para generar reportes e informes de los productos.

## **1.5 Justificación**

### **1.5.1 Justificación Técnica**

La unidad de administración que controla la unidad de almacenes como de inventarios, dispone de recursos para el desarrollo del sistema que son equipos computacionales que tiene las siguientes características: Microprocesador Pentium 4, capacidad de memoria 512 MB y utilizan sistema operativo Windows XP, de la que no están siendo utilizados adecuadamente debido a que los registros de ingresos y salidas se realiza de forma manual o semiautomatizada utilizando la hoja de Excel.

La empresa Bolital S.R.L. Se actualizara con los avances tecnológicos en el campo de la informática, tomando en cuenta de los equipos que disponen.

### **1.5.2 Justificación Económica**

La empresa Bolital S.R.L. será beneficiada con el desarrollo e implementación del nuevo sistema de inventarios para mejorar los procedimientos que se realizan de forma semiautomática y manual. Se busca incrementar el grado de eficiencia para las tareas específicas, por otro lado se busca reducir horas de trabajo del personal en el registro de todas las actividades que tiene la empresa en el control de inventarios, ayudara a agilizar los procesos manuales en la unidad de almacenes, tener un mejor control y seguimiento de la mercadería tanto de entradas como de salidas de productos lo que permitirá obtener información rápida, oportuna, confiable y fácil de comprender beneficiando a la unidad de administración y al personal de la empresa.

La implementación del software se realizara con software libre, lo que reducirá el costo de desarrollo.

### **1.5.3 Justificación Social**

Con el desarrollo e implementación del nuevo sistema ayuda a la institución en tener un mayor prestigio en la sociedad y tener mayor competitividad en el mercado.

En cuando al personal de la empresa tendrán mayor noción en el manejo del sistema puesto que es una herramienta lo que le permitirá tener mayor conocimiento en los avances tecnológicos.

## **1.6 Aporte**

### **1.6.1 Aporte Teórico**

La aplicación de modelo de inventario denominado Modelo Deterministico, permitirá poder tener un pronóstico acerca de la cantidad óptima de pedido y la frecuencia de tiempo.

### **1.6.2 Aporte Práctico**

Se implementara un modelo de inventario para el manejo de los productos, apoyando en un soporte grafico el cual contribuirá a la administración para la toma de decisiones, porque el uso de estos componentes visuales permite una percepción más clara sobre el estado de los almacenes para que así se pueda tener una idea clara sobre qué productos son los más solicitados, y de esta manera poder reabastecer a los almacenes de la empresa.

## **1.7 Alcance y Límites**

El presente proyecto de grado “Sistema de Entradas y Salidas e Inventario” Caso: Bolital S.R.L. se elaborará solo para la gestión de los productos que son: las almendras y sus diferentes tipos de variedades.

- Un módulo de control de entradas de producto.
- Un módulo de control de salida de producto.



- Un módulo de control de inventarios de almacenes.

## **1.8 Metodología**

Se utilizara como metodología de desarrollo de Software RUP, que es uno de los procesos de desarrollo de software que transforma los requisitos de un usuario en un sistema de software [JBR-00].

Para el Diseño y modelado de Casos de uso, tanto del sistema como del negocio, análisis, implementación, y otros diagramas se hará uso de una herramienta case y como lenguaje de modelado UML.

## **1.9. Herramientas**

Se tiene planificado implementar la Base de Datos en el Gestor de BD MySQL, para el desarrollo de la aplicación se utilizará PHP (lenguaje de programación) y como servidor Apache.

# CAPÍTULO 2

## MARCO TEÓRICO

### 2.1 Introducción

En este capítulo se definirá conceptos, definiciones para el desarrollo del proyecto de grado para un mejor entendimiento.

Como el proyecto trata de un Sistema de entradas y salidas e inventario empezaremos con las definiciones de lo referente al tema.

Posteriormente se definirá conceptos y definiciones más importantes acerca de la metodología RUP (*Proceso Unificado Racional*) y UML como lenguaje de Modelado para el desarrollo del proyecto y herramientas que se utilizarán tales como php, mysql, servidor apache y herramientas case para el diseño y modelado.

### 2.2 Modelos de Inventarios

Un negocio o una industria por lo común mantienen un inventario razonable de sus productos, para asegurar una operación continua.

Un factor importante en la formulación y la solución de un modelo de inventario es que la demanda (por tiempo de unidad) de un artículo puede ser determinista (conocida con cierto grado de incertidumbre) o probabilística (descrita mediante una distribución de probabilidades) [TAHA96].

Para cumplir a tiempo con la demanda, las empresas mantienen con frecuencia existencias a la espera de su venta. El objetivo de la teoría de inventarios es determinar reglas que se pueda aplicar la gerencia para reducir al mínimo los costos relacionados con el mantenimiento de existencias y cumplir con la demanda del

consumidor. Los modelos de inventarios responden las siguientes preguntas: ¿Cuándo se debe pedir un producto?, ¿Cuánto se debe pedir del producto? [WLW92].

### 2.2.1 Clasificación de Inventarios

De acuerdo a las características de la empresa encontramos tipos de inventarios.

- ✓ *Inventario de Mercancías*, lo constituyen todos aquellos bienes que le pertenecen a la empresa bien sea comercial o mercantil, los cuales los compran para luego venderlos sin ser modificados. En esta cuenta se mostrarán todas las mercancías disponibles para la venta. Las que tengan otras características y estén sujetas a condiciones particulares se deben mostrar en cuentas separadas, tales como las mercancías en camino (las que han sido compradas y no recibidas aún), las mercancías dadas en consignación o las mercancías pignoradas (aquellas que son propiedad de la empresa pero que han sido dadas a terceros en garantía de valor que ya ha sido recibido en efectivo u otros bienes)
- ✓ *Inventario de productos terminados*, son todos aquellos bienes adquiridos por las empresas manufactureras o industriales, los cuales son transformados para ser vendidos como productos elaborados.
- ✓ *Inventario de productos en proceso de fabricación*, lo integran todos aquellos bienes adquiridos por las empresas manufactureras o industriales, los cuales se encuentran en proceso de manufactura. Su cuantificación se hace por la cantidad de materiales, mano de obra y gastos de fabricación, aplicables a la fecha de cierre.
- ✓ *Inventario de materias primas*, lo conforman todos los materiales con los que se elaboran los productos, pero que todavía no han recibido procesamiento.
- ✓ *Inventario de suministros de fábrica*, son los materiales con los que se elaboran los productos, pero que no pueden ser cuantificados de una manera exacta (Pintura, lija, clavos, lubricantes, etc.).

### 2.2.2 Modelo Determinista

La gestión de inventario preocupa a la mayoría de las empresas cualquiera sea el sector de su actividad y dimensión, por tres factores imperativos.

- No hacer esperar al cliente.
- Realizar la producción a un ritmo regular, aun cuando fluctúe la demanda.
- Comprar los insumos a precios más bajos.

Una buena gestión de los inventarios es definir perfectamente:

- Mercadería a pedir.
- Fechas de pedido.
- Lugar de almacenamiento.
- La manera de evaluar el nivel de stock.
- Modo de reaprovisionamiento.

Sistema de reaprovisionamiento

Reglas de Gestión Cuándo y Cómo Pedir.

1. Cuando es necesario el reaprovisionamiento del inventario; a fecha fija o fecha variable, según el nivel de stock.
2. Cuando es necesario pedir por cantidades fijas o variables según el nivel de stock.

Modelos determinista, es difícil idear un modelo general de inventarios que tome en cuenta todas las variaciones de los sistemas reales, incluso, aun si puede ser formulado un modelo lo suficientemente general tal vez no sea posible su resolución analítica, por consiguiente, estos modelos tratan de ser ilustrativos de algunos sistemas de inventarios.

## Modelos estáticos de un solo artículo

El modelo de inventarios que implica un índice de la demanda constante con un reabastecimiento instantáneo de pedidos y sin faltante. Digamos que:

$y$  = Cantidad de pedido (número de unidades)

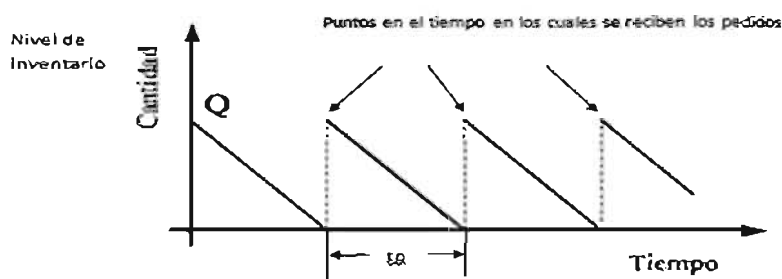
$D$  = Índice de demanda (unidades por tiempo de unidad)

$t_o$  = Duración del ciclo de pedidos (unidades de tiempo)

Utilizando estas definiciones, el nivel de inventario sigue el patrón representado en la Figura 2.2. Se hace un pedido de un volumen de  $Y$  unidades al instante cuando el nivel del inventario es cero. De esta manera, las existencias se agotan de manera uniforme según el índice de la demanda constante  $D$ .

El ciclo de pedidos para este patrón es:

$$t_o = \frac{Y}{D} \text{ unidades de tiempo}$$



**Figura. N° 2.1:** Modelo estático de un solo artículo

**Fuente:** [TAHA96]

El nivel resultante del inventario promedio se da como:

$$\text{Nivel del inventario promedio} = \frac{Y}{2} \text{ unidades}$$

### **2.2.3 Técnica de valuación para la salida de almacenes**

#### **El Método Primero en Entrar, Primero en Salir (PEPS)**

Este método identificado también como "PEPS", se basa en el supuesto de que los primeros artículos y/o materias primas en entrar al almacén o a la producción son los primeros en salir de él.

Se ha considerado conveniente este método porque da lugar a una evaluación del inventario concordante con la tendencia de los precios; puesto que se presume que el inventario está integrado por las compras más recientes y esta valorizado a los costos también más recientes, la valorización sigue entonces la tendencia del mercado.

Método PEPS, tipo de inventario perpetuo que detalla por medio de la Tarjeta de Control de inventario, las salidas y entradas de las mercancías. Establece que la primera mercancía que se compra es la primera en venderse o salir.

En tales condiciones, el costo está absorbiendo materiales a precio más reciente, que son los más bajos. El objetivo final de esta técnica es que las utilidades sean más conservadoras.

#### **Primero en Entrar, Primero en Salir (PEPS)**

En cuanto se agota el producto más antiguo, con su correspondiente costo de adquisición. En inventario tiende a quedar valorado al costo de adquisición mas reciente. Considera que las primera unidad adquirida, son las primeras surtidas al ser vendidas. La existencia en el inventario corresponde a las compras más reciente.

### **2.3 Ingeniería de software**

La ingeniería de software es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre maquinas reales.



El fundamento de la ingeniería de software es la capa de proceso. El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería de software. El proceso define un marco de trabajo para un conjunto de Áreas clave de procesos (ACPs) [PAU93] que se deben establecer para la entrega efectiva de la tecnología de la ingeniería de software. Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos del trabajo(modelos, documentos, datos , informes, formularios etc.), se establecen hitos, se asegura la calidad y el cambio se gestionan adecuadamente.



**Figura N° 2.2:** Capas de la ingeniería de software

**Fuente:** [PRES02]

Las herramientas de la Ingeniería de Software proporcionan un enfoque automático o semiautomático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamada ingeniería del Software asistida por computadora (CASE).

Los métodos de la ingeniería de software indican “como” construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería del software dependen de un conjunto de principios básicos

que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

En el proceso de software, se establece un marco común del proceso definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos del software, con independencia de su tamaño o complejidad [PRES02].

## **2.4 Lenguaje Unificado de Modelado (UML)**

El lenguaje unificado de maquetación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (OMG) y es el estándar de descripción de esquemas de software.

UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema.

## **2.5 Metodología RUP**

El proceso de desarrollo RUP (Rational Unified Process) aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y organizaciones. Provee a cada miembro del equipo, un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo de software. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje modelado común: UML.



El RUP es un proceso de desarrollo de software:

Provee un enfoque estructurado para realizar tareas y responsabilidades en una organización de desarrollo. Su principal objetivo es asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales, que sea realizado en las fechas acordadas y con el presupuesto disponible.

El RUP es un producto:

IBM comercializa un producto que permite instanciar al RUP según las características del proyecto, siendo una referencia en la metodología que sirve como repositorio único de información.

El RUP incorpora muchas de las conocidas como “buenas prácticas” en el desarrollo de software moderno, las cuáles se deben tener presentes en el desarrollo de aplicaciones empresariales para garantizar el éxito del proyecto, tales como:

Desarrollo iterativo, Gestión de Requerimientos, Arquitectura basada en componentes, Modelo Visual, Verificación de la calidad en forma continua y control de cambios.

El RUP presenta 3 características que constituyen la esencia de todo el proceso de desarrollo:

### **2.5.1 Proceso dirigido por Casos de Uso**

Según [KRU00], los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en la Figura 2.3.

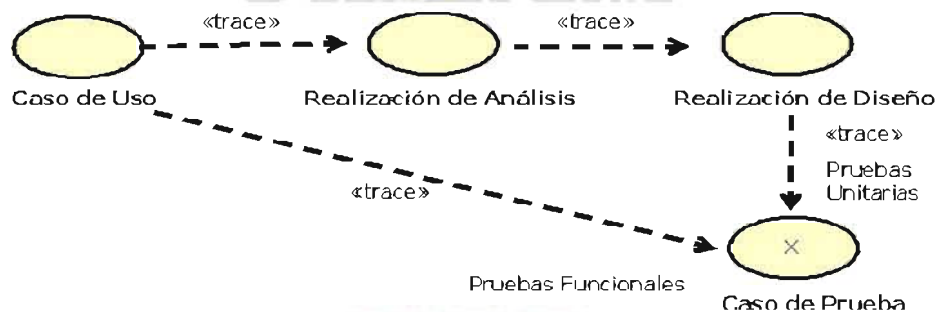


**Figura N° 2.3:** Los Casos de Uso integran el trabajo

**Fuente:** [KRU00]

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Como se muestra en la Figura 2.3, basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso.



**Figura N° 2.4:** Trazabilidad a partir de los Casos de Uso

**Fuente:** [KRU00]

## 2.5.2 Proceso centrado en la arquitectura

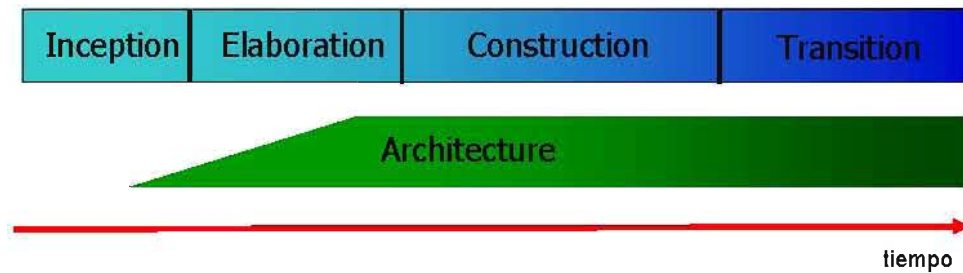
La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [KRU00].

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

En la Figura 2.5 se ilustra la evolución de la arquitectura durante las fases de RUP. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de *baselines* y se va modificando dependiendo de las necesidades del proyecto.



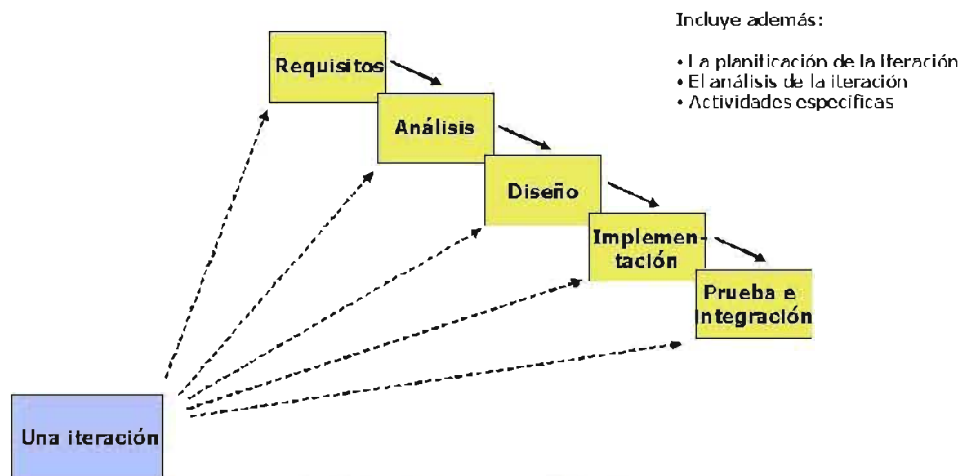
**Figura N° 2.5:** Evolución de la arquitectura del sistema

**Fuente:** [KRU00]

### 2.5.3 Proceso iterativo e incremental

Según [JBR00] el equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en la Figura 2.6. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.



**Figura N° 2.6:** Una iteración RUP

**Fuente:** [KRU00]

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 2.7 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



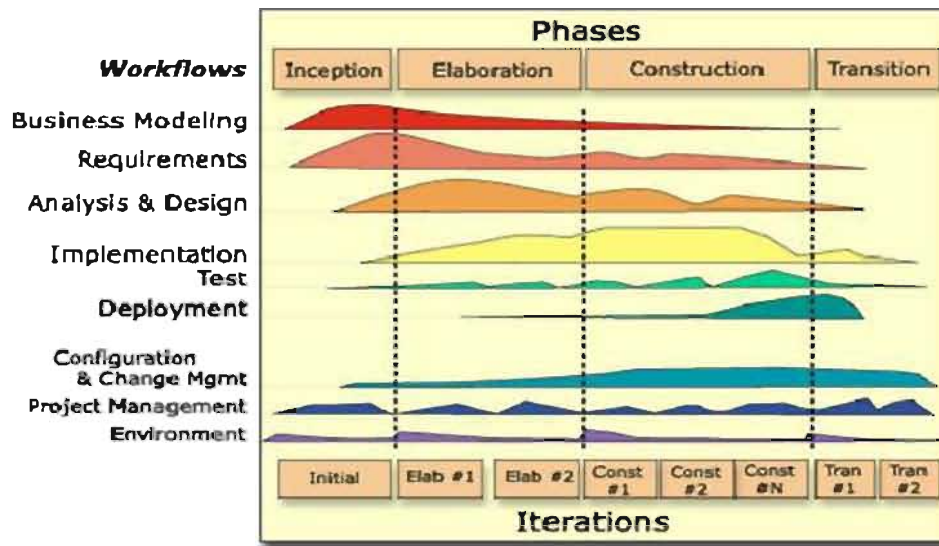


Figura N° 2.7: Esfuerzo en actividades según fase del proyecto

Fuente: [KRU00]

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una *baseline* de la arquitectura.

Durante la fase de inicio las iteraciones ponen mayor énfasis en actividades como el modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la *baseline* de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la *baseline* de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones. Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

## 2.5.4 Estructura del proceso

El proceso puede ser descrito en dos dimensiones o ejes [RSC98]:

**Eje horizontal:** Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la Figura 2.8 que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.

**Eje vertical:** Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

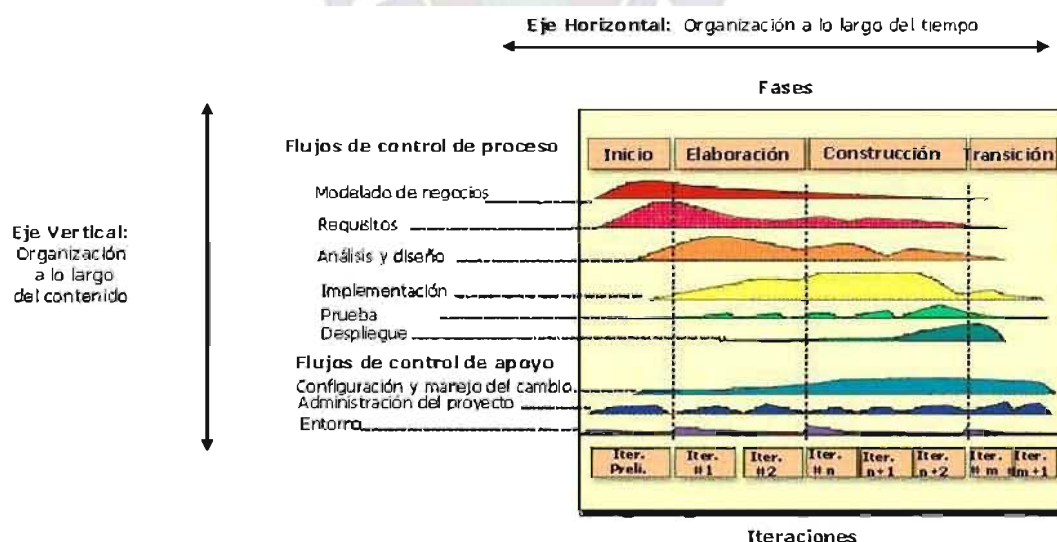
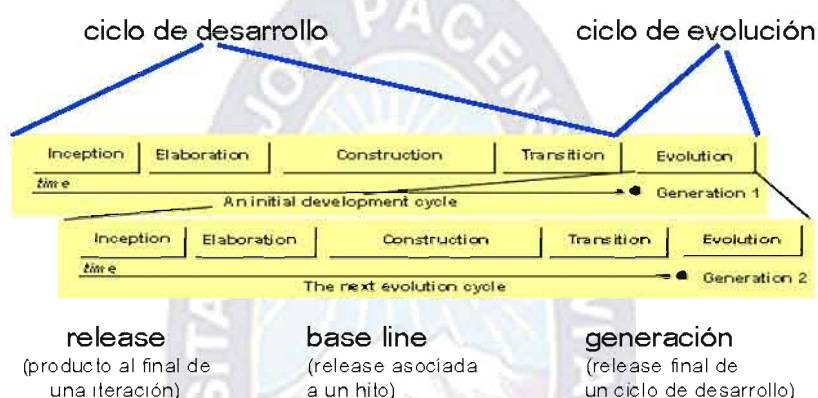


Figura N° 2.8: Estructura de RUP

Fuente: [KRU00]

### 2.5.5 Estructura Dinámica del proceso. Fases e iteraciones

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

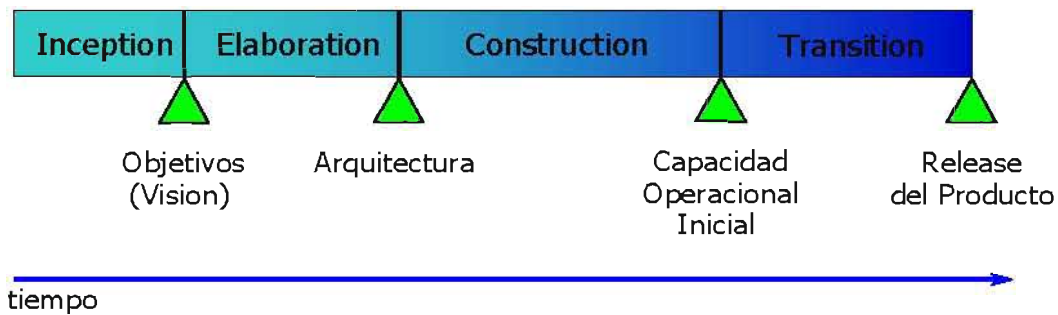


**Figura N° 2.9: Ciclos, releases, baseline**

**Fuente:** [KRU00]

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son: Inicio - *Lifecycle Objectives*, Elaboración - *Lifecycle Architecture*, Construcción - *Initial Operational Capability*, Transición - *Product Release*. Las fases y sus respectivos hitos se ilustran en la Figura 2.10





**Figura N° 2.10: Fases e hitos en RUP**

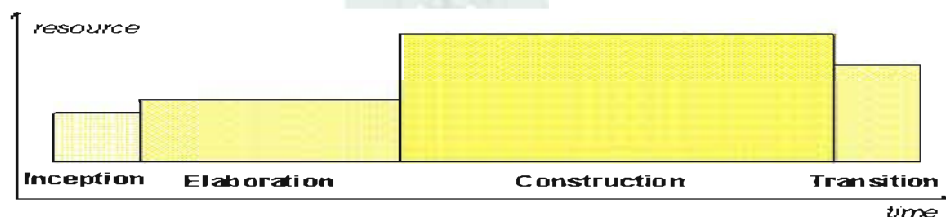
**Fuente:** [RSC98]

La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto. Sin embargo, la Figura 2.11 ilustra porcentajes frecuentes al respecto. Consecuente con el esfuerzo señalado, la Figura 2.12 ilustra una distribución típica de recursos humanos necesarios a lo largo del proyecto.

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10%
Tiempo Dedicado	10 %	30 %	50 %	10%

**Figura N° 2.11: Distribución típicas de esfuerzo y tiempo**

**Fuente:** [RSC98]



**Figura. N° 2.12: Distribución típica de recursos humanos**

**Fuente:** [KRU00]

### **2.5.5.1 Fase Inicio**

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar qué recursos deben ser asignados al proyecto.

Los objetivos de esta fase son [KRU00]:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.

Los resultados de la fase de inicio deben ser [RSC98]:

- Un documento de visión: Una visión general de los requerimientos del proyecto, características clave y restricciones principales.
- Modelo inicial de Casos de Uso (10-20% completado).
- El caso de negocio.
- Plan del proyecto, mostrando fases e iteraciones.
- Modelo de negocio, si es necesario
- Prototipos exploratorios para probar conceptos o la arquitectura candidata.

### **2.5.5.2 Fase Elaboración**

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe

contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

Los objetivos de esta fase son [KRU00]:

- Definir, validar y cimentar la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.
- Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.

Al terminar deben obtenerse los siguientes resultados [RSC98]:

- Un modelo de Casos de Uso completa al menos hasta el 80%: todos los casos y actores identificados, la mayoría de los casos desarrollados.
- Requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.
- Descripción de la arquitectura software.
- Un prototipo ejecutable de la arquitectura.
- Lista de riesgos y caso de negocio revisados.
- Plan de desarrollo para el proyecto.
- Un caso de desarrollo actualizado que especifica el proceso a seguir.
- Un manual de usuario preliminar (opcional).

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- La visión del producto es estable.

- La arquitectura es estable.
- Se ha demostrado mediante la ejecución del prototipo que los principales elementos de riesgo han sido abordados y resueltos.
- El plan para la fase de construcción es detallado y preciso. Las estimaciones son creíbles.
- Todos los interesados coinciden en que la visión actual será alcanzada si se siguen los planes actuales en el contexto de la arquitectura actual.
- Los gastos hasta ahora son aceptables, comparados con los previstos.
- Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

### **2.5.5.3 Fase Construcción**

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Los objetivos concretos según [KRU00] incluyen:

- Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.

Los resultados de la fase de construcción deben ser [RSC98]:

- Modelos Completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)
- Arquitectura íntegra (mantenida y mínimamente actualizada)
- Riesgos Presentados Mitigados
- Plan del Proyecto para la fase de Transición.

- Manual Inicial de Usuario (con suficiente detalle)
- Prototipo Operacional – beta
- Caso del Negocio Actualizado

Los criterios de evaluación de esta fase son los siguientes:

- El producto es estable y maduro como para ser entregado a la comunidad de usuario para ser probado.
- Todos los usuarios expertos están listos para la transición en la comunidad de usuarios.
- Son aceptables los gastos actuales versus los gastos planeados.

#### **2.5.5.4 Fase Transición**

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

En [KRU00] se citan algunas de las cosas que puede incluir esta fase:

- Prueba de la versión Beta para validar el nuevo sistema frente a las expectativas de los usuarios
- Funcionamiento paralelo con los sistemas legados que están siendo sustituidos por nuestro proyecto.
- Conversión de las bases de datos operacionales.
- Entrenamiento de los usuarios y técnicos de mantenimiento.
- Traspaso del producto a los equipos de marketing, distribución y venta.

Los principales objetivos de esta fase son:

- Conseguir que el usuario se valga por sí mismo.

- Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.

Los resultados de la fase de transición son [RSC98]:

- Prototipo Operacional
- Documentos Legales
- Caso del Negocio Completo
- Línea de Base del Producto completa y corregida que incluye todos los modelos del sistema
- Descripción de la Arquitectura completa y corregida
- Las iteraciones de esta fase irán dirigidas normalmente a conseguir una nueva versión.

Los criterios de evaluación de esta fase son los siguientes:

- El usuario se encuentra satisfecho.
- Son aceptables los gastos actuales versus los gastos planificados.

### **2.5.6 Estructura Estática del proceso**

Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? Figura 2.13 [RSC98].





**Figura N° 2.13:** Relación entre roles, actividades, artefactos

**Fuente:** [RSC98]

**2.5.6.1 Roles:** Un rol define el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas.

Las responsabilidades de un rol son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos [RSC98].

### 2.5.6.2 Actividades

Una actividad en concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.

### 2.5.6.3 Artefactos

Un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados

tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final [MMA].

- Un artefacto puede ser cualquiera de los siguientes [RSC02]:
- Un documento, como el documento de la arquitectura del software.
- Un modelo, como el modelo de Casos de Uso o el modelo de diseño.
- Un elemento del modelo, un elemento que pertenece a un modelo como una clase, un Caso de Uso o un subsistema.

#### **2.5.6.4 Flujos de trabajo**

Con la enumeración de roles, actividades y artefactos no se define un proceso, necesitamos contar con una secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables. A continuación se dará una explicación de cada flujo de trabajo.

##### **2.5.6.4.1 Requisitos**

El propósito fundamental de flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre que debe y que no debe hacer el sistema. [JACOBSON00]

##### **Enumerar los requisitos candidatos**

Se mantiene una lista de ideas, que se considera requisitos candidatos que se puede decir implementar en una versión futura del sistema. Esta lista de características crece a medida que se añaden nuevos elementos que se convierten en requisitos y se transforman en otros artefactos como casos de uso. [JACOBSON00]



## Comprender el contexto del sistema

Para capturar los requisitos correctos y construir el sistema correcto, los desarrolladores requieren un firme conocimiento del contexto en el que se emplaza el sistema. Para expresar el contexto de un sistema en una forma utilizable hay por lo menos dos aproximaciones: modelo del dominio y modelo del negocio.

### ✓ Modelo de Dominio

Un modelo de dominio describe los conceptos importantes del contexto como objetos del dominio, y enlaza estos objetos unos con otros. El modelo de dominio se describe mediante diagramas de UML, mediante diagrama de clases.

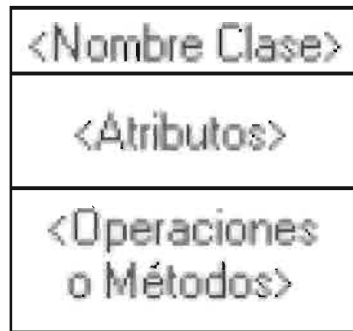
## Diagramas de Clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

## Elementos

*Clase*: Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).

En UML, una clase es representada por un rectángulo que posee tres divisiones:



**Figura N° 2.14:** Diagrama de Clase

**Fuente:** [SCHMU]

*Nombre de clase:* Contiene el nombre de la Clase

*Atributos:* Representa alguna propiedad de la clase, que se encuentra en todas las instancias de la clase.

Definen la estructura de una clase y de sus correspondientes objetos. Los atributos corresponden a sustantivos y sus valores pueden ser sustantivos o adjetivos. Dentro de una clase, los nombres de los atributos deben ser únicos (aunque puede aparecer el mismo nombre de atributo en diferentes clases).

*Operaciones o Métodos:* Las operaciones son funciones o transformaciones que se aplican a todos los objetos de una clase particular. La operación puede ser una acción ejecutada por el objeto o sobre el objeto.

### ***Relaciones entre clases***

Una asociación es una conexión estructural simple entre clases. La instancia de las clases implicadas en una asociación, estarán probablemente comunicándose en el momento de ejecución. Los enlaces entre dos objetos pueden representarse entre las respectivas clases

Forma de relación entre clases: Asociación (conexión entre clases), dependencia (relación de uso) y generalización/especialización (relaciones de herencia)

#### 2.5.6.4.2 Modelo del negocio

El objetivo del modelo del negocio es describir los procesos existentes u observados con el objetivo de comprenderlos. El modelo de negocio especifica que procesos de negocio soportara el sistema. Aparte de identificar los objetos del dominio o del negocio implicados en el negocio, este modelo también establece las competencias requeridas encada proceso: sus trabajadores, sus responsabilidades, y en las operaciones que llevan a cabo. [JACOBSON00]

Para realizar este modelo de debe realizar un modelo de casos de uso del negocio que identifique los actores del negocio y los casos de uso del negocio que utilicen los actores [JACOBSON00], se utilizara diagramas de UML (diagrama de casos de uso).

##### **Diagrama de Casos de Uso**

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Un diagrama de caso de uso explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre estos y los casos de uso. Estos últimos se muestran en óvalos y os actores son Figuras estilizadas. Hay líneas de comunicaciones entres los casos y los actores; las flechas indican el flujo de la información o el estímulo.

El diagrama tiene por objetivo ofrecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan.

**Elementos:** Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: Actores, casos de uso y relaciones entre casos de uso.

**Actores:** Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una Figura humana dibujada con

palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).

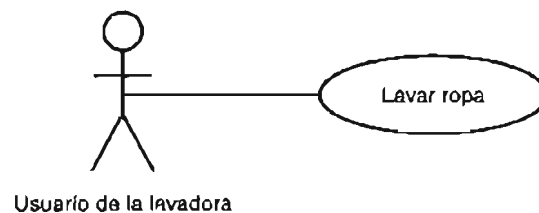
**Casos de Uso:** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

### Relaciones entre Casos de Uso

Entre dos casos de uso puede haber las siguientes relaciones:

- **Extiende:** Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- **Usa:** Cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la Figura 2.15 se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.



**Figura N° 2.15:** Diagrama de Casos de Uso

**Fuente:** [SCHMU]

### Ejemplo TPV: Especificación del caso de uso "compra de productos en efectivo"

**Caso de uso:** Compra de productos en efectivo.

**Actores:** Cliente (Iniciador), Cajero.

**Propósito:** Capturar una venta y su pago en efectivo.

**Resumen:** Un cliente llega a la caja con productos para comprar

El cajero registra los productos y gestiona el pago en efectivo

Al acabar, el cliente se va con los productos

**Tipo:** Primario y esencial

**Referencias cruzadas:** R1.1, R1.2, R1.3, R1.7, R2.1

**Curso típico de eventos**

Acciones de los actores	Respuesta del sistema
1. El caso de uso comienza cuando un Cliente llega a la caja con los productos para comprar.	
2. El Cajero indica que comienza una nueva venta	3. Registra el inicio de una nueva venta.
4. El Cajero registra el identificador de cada producto. Si hay más de una unidad del producto el Cajero puede introducir la cantidad.	5. Determina el precio del producto y añade su información a la cuenta.
6. Al acabar la entrada de productos el Cajero lo indica.	7. Calcula y muestra el total de la cuenta.

**Figura N° 2.16:** Caso de Uso Detallado

**Fuente:** [TCR03]

### Capturar requisitos funcionales

La técnica inmediata para identificar los requisitos del sistema se basa en los casos de uso. Estos casos de uso capturan tanto los requisitos funcionales como los no funcionales que son específicos de cada caso de uso. [JACOBSON00]

### Capturar requisitos no funcionales

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento y dependencias de la plataforma, etc. La mayoría de los requisitos de rendimiento afectan solo a ciertos casos de uso [JACOBSON00].

## **Encontrar actores y casos de uso**

Esta actividad se identifica los actores y los casos de uso para delimitar el sistema de su entorno; esbozar que (actores) interactúan con el sistema y que funcionalidad (casos de uso) se espera del sistema.

La identificación de actores y casos de uso es la actividad más decisiva para obtener adecuadamente los requisitos [JACOBSON00].

Encontrar actores y casos de uso: en este paso se identifican a los actores y casos de uso y se realiza una pequeña descripción de lo que hacen.

Describir el modelo de casos de uso completo: después de haber identificado a actores y casos de uso, se realiza el modelo de casos de uso del sistema realizando una descripción general del modelo de caso de uso.

### **2.5.6.4.3 Análisis**

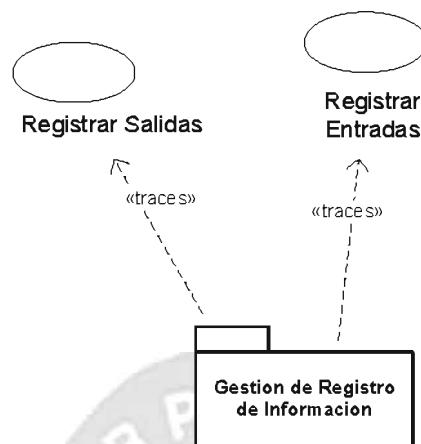
El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe como implementar el sistema.

#### **Análisis de la arquitectura**

El propósito del análisis de la arquitectura es esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes de análisis, clases del análisis evidente. [JACOBSON00]

Los paquetes de análisis se los identifican a partir de los casos de uso, agrupando a aquellos casos de uso que tengan alguna característica en común. Los paquetes de análisis se representan utilizando notación UML.





**Figura N° 2.17:** Identificación de paquetes del análisis a partir de los casos de uso

**Fuente:** [JACOBSON00]

Una vez analizado los paquetes de análisis después se debe definir las dependencias entre estos, para ver si sus contenidos están relacionados. La identificación de clases del análisis evidente se la puede realizar basándose en las clases del dominio o las entidades del negocio que se identifican durante la captura de requisitos aunque la mayoría de las clases de entidad serán identificadas en la actividad Análisis de casos de uso.

**Analizar un caso de uso** Esta actividad se la realiza para identificar las clases del análisis cuyos objetivos son necesarios para llevar a cabo el flujo de sucesos del caso de uso [JACOBSON00].

Identificar clases de entidad mediante el estudio en detalle de la descripción del caso de uso, y después considerar que información debe utilizarse y manipularse en la realización del caso de uso.

Identificar una clase de interfaz central para cada actor, y dejar que esta clase represente la venta principal de interfaz de usuario con el que interactúa el actor.

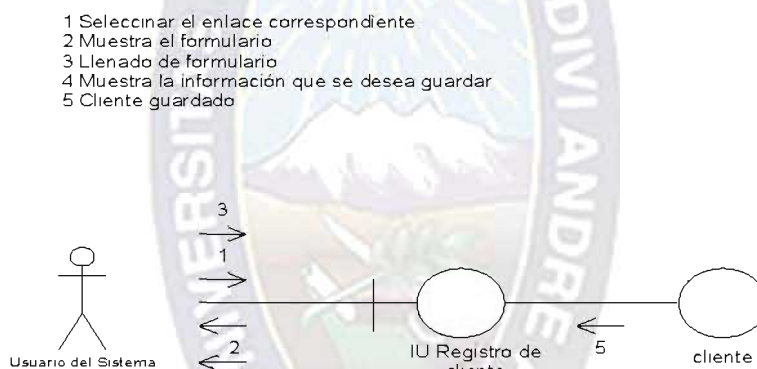
Identificar una clase de interfaz primitiva para cada clase de entidad que hayamos encontrado anteriormente.

Identificar una clase central para cada actor que sea un sistema externo, y dejar que esta clase represente la interfaz de comunicación.

Identificar una clase de control responsable del tratamiento del control y de la coordinación de la realización del caso de uso [JACOBSON00]. Utilizaremos la notación UML (Diagramas de colaboración).

### Diagrama de colaboración

Los diagramas de colaboración, muestra las interacciones entre objetos creando enlaces entre ellos añadiendo mensajes a esos enlaces. El nombre de un mensaje debería denotar el propósito del objeto invocante en la interacción con el objeto invocado.



**Figura N° 2.18:** Diagrama de Colaboración del caso de uso: Registrar Clientes

**Fuente:** [JACOBSON00].

#### 2.5.6.4.4 Diseño

En este flujo de trabajo se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos. Una entrada esencial en el diseño es el resultado del análisis, esto es, el modelo de análisis. El modelo de análisis proporciona una comprensión detallada de los requisitos, y lo más importante, impone una estructura del sistema.

En concreto, los propósitos del diseño son: Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos, crear una entrada apropiada y un

punto de partida para actividades de implementación; ser capaces de descomponer los trabajos de implementación en partes más manejables. [JACOBSON00]

## **Diseño de la arquitectura**

El objetivo del diseño de la arquitectura es esbozar los modelos de diseño y despliegue y su arquitectura mediante la identificación de los siguientes elementos:

- Nodos y sus configuraciones de red.
- Subsistema y sus interfaces.
- Clase del diseño significativo para la arquitectura, como las clases activas [JACOBSON00].

## **Diseño de un caso de uso**

Los objetivos del diseño de un caso de uso son: Identificar las clases del diseño y los subsistemas cuyas instancias son necesarias para llevar a cabo el flujo de sucesos del caso de uso; distribuir el comportamiento del caso de uso entre los objetos del diseño que interactúan entre los subsistemas participantes; definir los requisitos sobre las operaciones de las clases del diseño y sobre los subsistemas y sus interfaces, capturar los requisitos de implementación del caso de uso [JACOBSON00].

Para realizar esta actividad se debe identificar clases de diseño que participan en la realización del caso de uso, después de haber realizado esto se debe describir las interacciones entre los objetos del diseño, esta descripción debe hacerse utilizando diagramas de secuencia utilizando notación UML.

**Diagrama de Secuencia**, Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

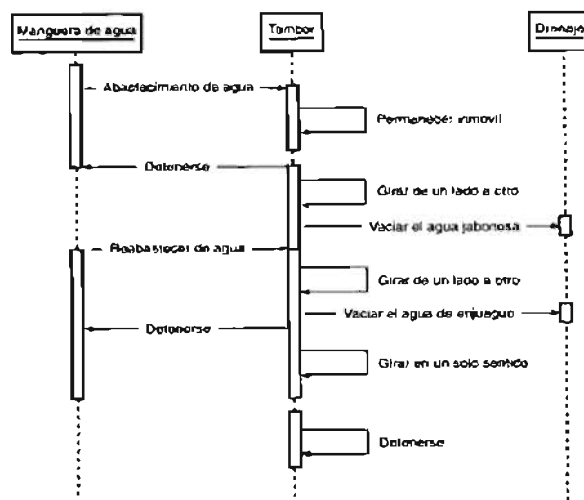


Figura N° 2.19: Diagrama de Secuencia

Fuente: [SCHMU]

## Diseño de clases

Esta actividad tiene como propósito crear una clase de diseño que cumpla su papel en las realizaciones de los casos de uso y los requisitos no funcionales que se aplican a estos. Las tareas que realizan esta actividad son: Identificar operaciones, atributos, asociaciones, agregaciones, métodos y estados [JACOBSON00].

## Diseño de un subsistema

Los objetivos del diseño de un subsistema son: Garantizar que el subsistema es tan independiente como sea posible de otros subsistemas, garantizar que el subsistema proporciona las interfaces correctas, garantizar que el subsistema cumple su

propósito de ofrecer una realización correcta de las operaciones tal y como se definen en las interfaces que proporciona [JACOBSON00].

#### **2.5.6.4.5 Implementación**

En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, el propósito principal de la implementación es desarrollar la arquitectura del sistema como un todo. [JACOBSON00]

##### **Implementación de la arquitectura**

El propósito de la implementación de la arquitectura es esbozar el modelo de implementación y su arquitectura mediante la identificación de componentes significativos arquitectónicamente y la asignación de componentes a los nodos en las configuraciones de redes relevantes. [JACOBSON00]

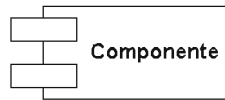
Durante el diseño de la arquitectura se esbozan los subsistemas de diseño, sus contenidos y sus interfaces. Durante la implementación utilizamos subsistemas de implementación que siguen la traza uno a uno a estos subsistemas de diseño y proporcionan las mismas interfaces.

En esta actividad se mantiene, refinada y actualizada la descripción de la arquitectura y las vistas de la arquitectura de los modelos de implementación y despliegue. [JACOBSON00].

Para representar el modelo de implementación se utilizarán diagramas de componentes y para el modelo de despliegue se utilizarán diagramas de despliegue.

##### **Diagrama de componentes**

Muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones



**Figura N° 2.20:** Elemento de diagrama de componente

**Fuente:** [JACOBSON00]

## Diagrama de despliegue

Representan la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Muestran la vista de despliegue estática de una arquitectura y se relacionan con los componentes ya que, por lo común, los nodos contienen uno o más componentes.



**Figura N° 2.21:** Elemento de diagrama de despliegue

**Fuente:** [JACOBSON00]

## Integrar el sistema

Los objetivos de la integración del sistema son: crear un plan de integración de construcción que describa las construcciones necesarias en una iteración y los requisitos de cada construcción, integrar cada construcción antes de que sea sometida a pruebas de integración.

## Implementar un subsistema

El propósito de implementar un subsistema es el de asegurar que un subsistema cumple su papel en cada construcción, tal y como se especifica en el plan de integración de la construcción.



## **Implementar una clase**

El propósito de la implementación de una clase es implementar una clase de diseño en un componente fichero. Esto incluye lo siguiente: Esbozo de un componente fichero que contendrá el código fuente, generación de código fuente a partir de la clase de diseño y de las relaciones en que participa, implementación de las operaciones de clase y comprobación de que el componente proporciona las mismas interfaces que la clase de diseño [JACOBSON00].

### **2.5.6.4.6 Pruebas**

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

En este flujo de trabajo de la prueba verificamos el resultado de la implementación probando cada construcción, incluyendo tantas construcciones internas, así como las versiones finales del sistema.

#### **Planificar prueba**

El propósito de la planificación de la prueba es planificar los esfuerzos de prueba en una iteración llevando a cabo las siguientes tareas: describiendo una estrategia de prueba, estimando los requisitos para el esfuerzo de la prueba y planificando el esfuerzo de la prueba.

#### **Diseñar prueba**

Los objetivos de esta actividad son: Identificar y describir los casos de prueba, identificar y estructurar los procedimientos de prueba especificando como realizar los casos de prueba.

## **Realizar prueba del sistema**

El propósito de esta actividad es el de realizar las pruebas de sistema necesarias en cada iteración y el de recopilar los resultados de las pruebas. La prueba de sistema puede empezar cuando las pruebas de integración indican que el sistema satisface los objetivos de calidad de integración fijados en el plan de prueba de la iteración actual.

### **Prueba de software**

Las pruebas de software son un elemento crítico para la garantía del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

La creciente percepción del software como un elemento del sistema y la importancia de los <<costes>> asociados a un fallo del propio sistema están motivando la creación de pruebas minuciosas y bien planificadas. [PRES02]

### **Prueba de la caja blanca**

La prueba de la caja negra, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que se usa la estructura de control de diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero de software puede obtener casos de prueba que (1) garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo; (2) ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; (3) ejecuten todos los bucles en sus límites y con sus límites operacionales; y (4) ejerciten las estructuras internas de datos para asegurar su validez. [PRES02]

## **2.6 Herramientas de Desarrollo de Software**

Herramientas para la elaboración de software que es de principal importancia.

## ➤ MySQL

MySQL es un sistema de gestión de bases de datos relacional, el software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multihilo, multiusuario y robusto. El servidor está proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo masivo de software.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's (Interfaz de Programación de Aplicaciones) en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

## ➤ HTML(*Lenguaje de Marcas de Hipertexto*),

HTML, es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

## ➤ PHP (Hypertext Preprocessor)

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Sus principales características son:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- (*Lenguaje de Marcas de Hipertexto*), es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

#### ➤ Apache

Apache servidor web, (Acrónimo de "a patchy server") distribución libre y de código abierto, siendo el más popular del mundo desde abril de 1996, con una penetración actual del 50% del total de servidores web del mundo.

La principal competencia de Apache es el IIS (Microsoft Internet Information Services) de Microsoft.

Apache fue la primera alternativa viable para el servidor web de Netscape Communications, actualmente conocido como Sun Java System Web Server.

Apache es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la Apache Software Foundation.

La aplicación permite ejecutarse en múltiples sistemas operativos como Windows, Novell NetWare, Mac OS X y los sistemas basados en Unix.

Sus principales características son:

- Soporte para los lenguajes perl, python, tcl y PHP.
- Módulos de autenticación: mod\_access, mod\_auth y mod\_digest.
- Soporte para SSL y TLS.
- Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- Permite autenticación de base de datos basada en SGBD.

#### Uso de Apache

Apache es principalmente usado para servir páginas web estáticas y dinámicas en la WWW. Apache es el servidor web del popular sistema XAMP, junto con MySQL y los lenguajes de programación PHP/Perl/Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP, etc.

## 2.7 Métricas Calidad de Software

La calidad de software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e

integridad. La calidad de software es medible y varia de un sistema a otro o de un programa a otro.

Las actividades de control de calidad pueden ser manuales, completamente automáticas o una combinación de herramientas automáticas e interacción humana. Un concepto clave del control de calidad es que hayan definido todos los productos y las especificaciones mensurables en las que se puedan comparar los resultados de cada proceso. El bucle de realimentación es esencial para reducir los defectos producidos

La calidad de software es una compleja mezcla de factores que variaran a través de diferentes aplicaciones y según los clientes que las pidan [PRES02].

### **2.7.1 Métricas Calidad de Software ISO 9126**

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos clave de la calidad para, el software. El estándar identifica seis atributos clave de calidad:

- *Funcionalidad.* El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.
- *Confiabilidad.* Cantidad de tiempo que el software está disponible para su uso. Esta referido por los siguientes subatributos: madurez tolerancia fallos y facilidad de recuperación.
- *Usabilidad.* Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: Facilidad de comprensión, facilidad de aprendizaje y operatividad.
- *Eficiencia.* Grado en que el software hace optimo el uso los recursos del sistema. Esta indica por los siguientes subatributos: tiempo de uso y recursos utilizados.



- *Facilidad de mantenimiento.* La facilidad con que una modificación puede ser realizada. Esta indica por los siguientes subatributos: Facilidad de análisis, facilidad de cambio, estabilidad y facilidad de pruebas.
- *Portabilidad.* La facilidad con que el software puede ser llevado de un entorno a otro. Esta referido por los siguientes subatributos: Facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.[PRES02]

## Funcionalidad

La métrica de punto de función (PF) se puede usar como medio, para predecir el tamaño de un sistema que se va a obtener de un modelo de análisis.

Para ilustrar el empleo de la métrica de PF en este contexto, consideramos una sencilla representación del modelo de análisis mostrado en la Figura. En la Figura se representa un diagrama de flujo de datos de una función del sistema Hogar Seguro. La función gestiona la interacción con el usuario, aceptando una contraseña de usuario para activar/desactivar el sistema y permitiendo consultas sobre el estado de las zonas de seguridad y varios sensores de seguridad. La función muestra una serie de mensajes de petición y envía señales apropiadas de control a varios componentes del sistema de seguridad.

El diagrama de flujo de datos se evalúa para determinar las medidas clave necesarias para el cálculo de la métrica de punto de función:

- número de entradas del usuario
- número de salidas del usuario
- número de consultas de usuario
- número de archivos
- número de interfaces externas

La cuenta total debe ajustarse utilizando la siguiente ecuación:

$$PF = \text{cuenta-total} * (0,65 + 0,01 * \sum Fi)$$

Donde cuenta-total es la suma de todas las entradas PF obtenidas de la Figura y

Fi (i=1 a 14) son los “valores de ajuste de complejidad”.

Parámetro de medición	Cuenta	Factor de ponderación				
		Simple	Media	Compleja		
Número de entradas del usuario	<input type="text"/>	X	3	4	6	= <input type="text"/>
Número de salidas del usuario	<input type="text"/>	X	4	5	7	= <input type="text"/>
Número de consultas del usuario	<input type="text"/>	X	3	4	6	= <input type="text"/>
Número de archivos	<input type="text"/>	X	7	10	15	= <input type="text"/>
Número de interfaces externas	<input type="text"/>	X	5	7	10	= <input type="text"/>
Cuenta total						→ <input type="text"/>

**Figura N° 2.22:** Cálculo de punto de función para una función

**Fuente:** [PRES02].

Valores de ajuste de la complejidad según respuestas a las siguientes preguntas:

- 1.- ¿Requiere el sistema copias de seguridad y de recuperación fiables?
- 2.- ¿Se requiere comunicación de datos?
- 3.- ¿Existen funciones de procesamiento distribuido?
- 4.- ¿Es crítico el rendimiento?
- 5.- ¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?
- 6.- ¿Requiere el sistema entrada de datos interactiva?
- 7.- ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
- 8.- ¿Se actualizan los archivos maestros de forma interactiva?
- 9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
- 10.- ¿Es complejo el procesamiento interno?
- 11.- ¿Se ha diseñado el código para ser reutilizable?
- 12.- ¿Están incluidas en el diseño la conversión y la instalación?
- 13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?

14. -¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Cada una de las preguntas anteriores es respondida usando una escala con rangos desde 0 (no importante aplicable) hasta 5 (absolutamente esencial). Los valores constantes de la ecuación del punto de función y los factores de peso que se aplican a las cuentas de los dominios de información se determinan empíricamente. Una vez que se han calculado los puntos de función, se utilizan de forma análoga a las LDC como forma normalizar las medidas de productividad, calidad y otros atributos del software. [PRES02].

### **Mantenibilidad**

Todas las métricas del software pueden usarse para el desarrollo de nuevo software y para el mantenimiento del existente. Sin embargo, se han propuesto métricas diseñadas explícitamente para actividades de mantenimiento. [PRES02].

El estándar IEEE 982.1-1988 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software (basada en los cambios que ocurren con cada versión del producto).

Se determina la siguiente información:

$M_t$  = número de módulos en la versión actual

$F_c$  = número de módulos en la versión actual que se han cambiado

$F_a$  = número de módulos en la versión actual que se han añadido

$F_d$  = número de módulos de la versión anterior que se han borrado en la versión actual

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = [Mt - (Fc + Fa + Fd)] / Mt$$

A medida que el IMS se aproxima a 1,0 el producto se empieza a estabilizar. El IMS puede emplearse también como métrica para la planificación de las actividades de mantenimiento del software. [PRES02].

## Confiabilidad

La confiabilidad del software se refiere a la cantidad de tiempo que el software está disponible para su uso. Entre sus atributos están: madurez, tolerancia a fallos y facilidad de recuperación. Se dice que la confiabilidad del software es la probabilidad de que un sistema opere sin fallas bajo determinadas condiciones para un intervalo de tiempo dado.

Supongamos que el sistema comienza a trabajar en el instante  $t=0$  y observamos hasta que falla. Supongamos que la falla ocurre en el tiempo  $t$ . Se asignará a la variable  $T$  como "la duración del tiempo de trabajo sin falla". Entonces la probabilidad de falla del sistema en el tiempo  $t$  es:

$$P [T \sim 1] = F (t)$$

Donde  $F (1)$  es la función de distribución de la variable aleatoria  $T$ . Por lo tanto, la probabilidad de que no haya falla en el sistema en el tiempo  $t$  es

$$P [T \leq t] = 1 - F (t)$$

Para el cálculo de la confiabilidad del sistema se utilizará la siguiente función exponencial, donde  $t$  será el periodo de tiempo en el que se prueba el sistema.

$$F (t) = \begin{cases} PF \times e^{-t/10} & , x > 0 \\ 0 & , e.o.c \end{cases}$$

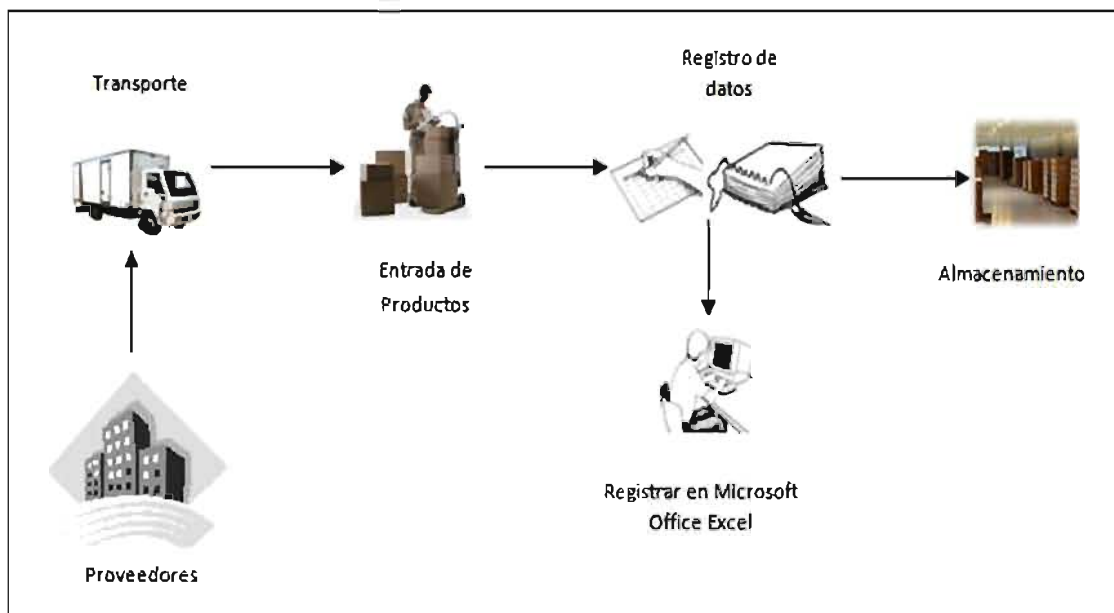
# CAPÍTULO 3

## MARCO APLICATIVO

### 3.1 Introducción

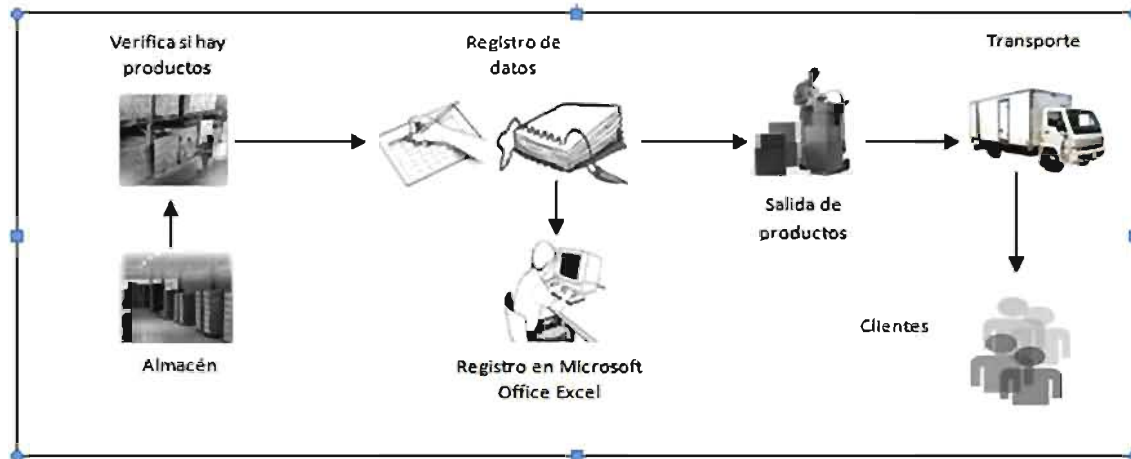
Para llegar a desarrollar el sistema, es fundamental la comprensión, de los requerimientos funcionales, sus alcances y restricciones que se requiere para el desarrollo del sistema. Este capítulo describe el marco de trabajo y la construcción de los diferentes artefactos generados en las fases e iteraciones de la Metodología RUP.

Es por esta razón que este capítulo es uno de los más importantes dentro del trabajo para el desarrollo del software.



**Figura N° 3.1:** Entrada de Productos

**Fuente:** [Elaboración Propia]



**Figura N° 3.2:** Salida de Productos

**Fuente:** [Elaboración Propia]

### 3.2 Fase Inicio

Se define el ámbito del sistema y se realiza el esbozo de la arquitectura que tendrá el sistema, a continuación se desarrolla los flujos de trabajo, llevando a cabo cada una de sus actividades.

#### 3.2.1 Recopilación de requisitos

##### ***Requisitos Candidatos:***

Los requisitos candidatos para el Sistema de Entrada y Salida e Inventario son los siguientes:

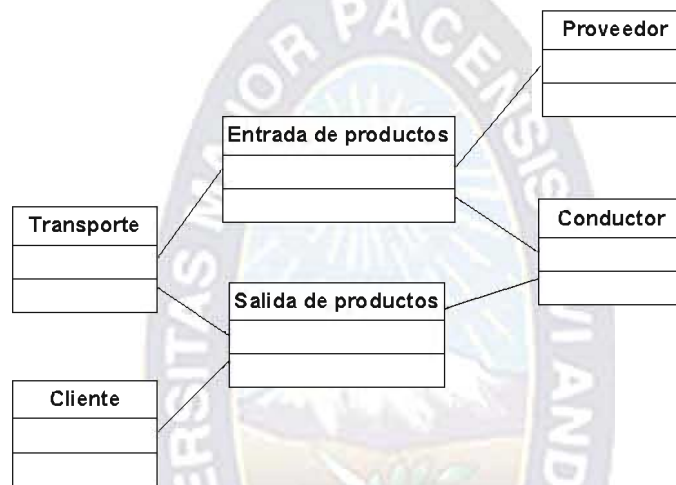
- Tener el registro ordenados de los productos de salida.
- Tener el registro ordenados de los productos de entrada.
- Tener registro e información del inventario.
- Genera consultas y reportes sobre la existencia de producto que se tiene en inventario (Stock).
- Registrar a los proveedores.
- Registrar los productos.
- Registrar los clientes.



- Registrar medio de transporte y registrar conductor.
- Tener control sobre altas y bajas de los productos.

### **Contexto del sistema:**

El contexto del Sistema de Entradas y salidas e Inventario representado en el Modelo del Dominio, está representado con diagramas de clase se muestra en la Figura 3.3.



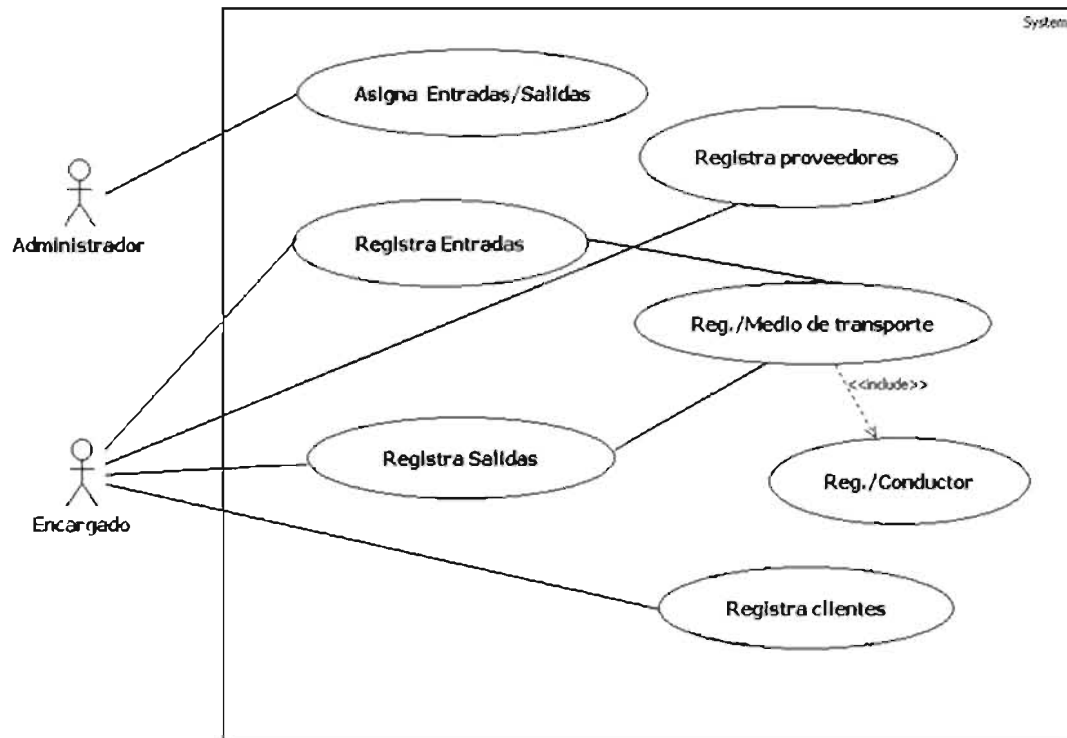
**Figura N° 3.3:** Modelado de Dominio

**Fuente:** [Elaboración Propia]

### **Modelado del Negocio**

El modelado del negocio es una técnica que nos ayuda identificar los procesos que observamos en la unidad a ser automatizada y entender la actividad de trabajo de la empresa.

Se definen también los actores y los casos de uso de la empresa, en la Figura.N°3.4 podemos observar a los principales actores que interactúan con el sistema.



**Figura N° 3.4:** Modelado de caso de uso del negocio general

**Fuente:** [Elaboración Propia]

### Requisitos funcionales

Para esta actividad se ha seleccionado de los requisitos candidatos a aquellos requisitos que serán representados como casos de uso los cuales son los siguientes:

- Registrar las entradas de productos
- Registrar las salidas de productos
- Actualización del inventario de productos
- Registrar clientes
- Registrar proveedores
- Registrar los medios de transporte
- Registrar conductor
- Generar Reportes

## Requisitos no funcionales

Las características que tendrá el sistema de entradas y salida e inventario son los siguientes:

- El sistema debe ser controlado por el administrador
- Debe ser portable
- El sistema debe tener documentación para la instalación del sistema

*Requisitos en hardware*, como mínimo una Pentium IV.

*Requisitos de software*, sistema operativo de cliente como de servidor Windows XP.

### 3.2.2 Casos de uso

#### Descripción de los actores

Vamos a definir a los actores que intervienen en los procesos identificados:

**Administrador:** Persona que toma las decisiones y da la autorización de nuevas entradas o salidas de productos, dando a conocer esta información al Encargado.

**Encargado:** El encargado es una persona que realiza las tareas de registrar las entradas y salidas de productos que estas van acompañadas de los registro de clientes, proveedores y medio de transporte (conductor).

#### Descripción de los casos de uso.

**Asignación de entradas/salidas:** En este caso se hace la asignación de entradas y salidas de productos.

**Registro de entradas:** Registra las entradas de productos que lleva acabo la empresa que son autorizados.

**Registro de proveedores:** Registra a todos los proveedores que tiene la empresa.

**Registro de salidas:** Registra todas las salidas de productos que son autorizados por el administrador.

**Registro de clientes:** Registra a todos los clientes que tiene la empresa.

**Registro de conductor:** Registra a todos los conductores que realizan las salidas o entradas de productos.

**Registro de medio de transporte:** Registra los medios de transporte que realizan el traslado de los productos para la entrada y salida de producto con un respectivo conductor.

### Modelado de Casos de Uso

A continuación se muestra el Diagrama de Casos de Uso del Sistema de entradas y salidas e inventario.



Figura N° 3.5: Diagrama de Casos de Uso - SESI

Fuente: [Elaboración Propia]

## Detallar un caso de uso

Como en esta fase solo se detalla el diez por ciento de los casos de uso, entonces se realizara esta actividad en un solo caso de uso. Para ver todos los casos de uso detallados ver [Anexo D].

Caso de Uso:	Registro de Proveedores
Actores:	Encargado
Propósito	Capturar datos de los Proveedores
Resumen	Actor registra proveedores que tiene la empresa.
Tipo	Primario y Esencial
Referencias	R1, R6,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1. Comienza cuando el actor requiere transcribir proveedores de la empresa.	
2. Este caso de uso comienza un actor se Identifica	
3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este	
4. Realiza petición de acuerdo a la necesidad que tenga:	
• Registrar Proveedor	
5. Muestra:	• Listado de Proveedores

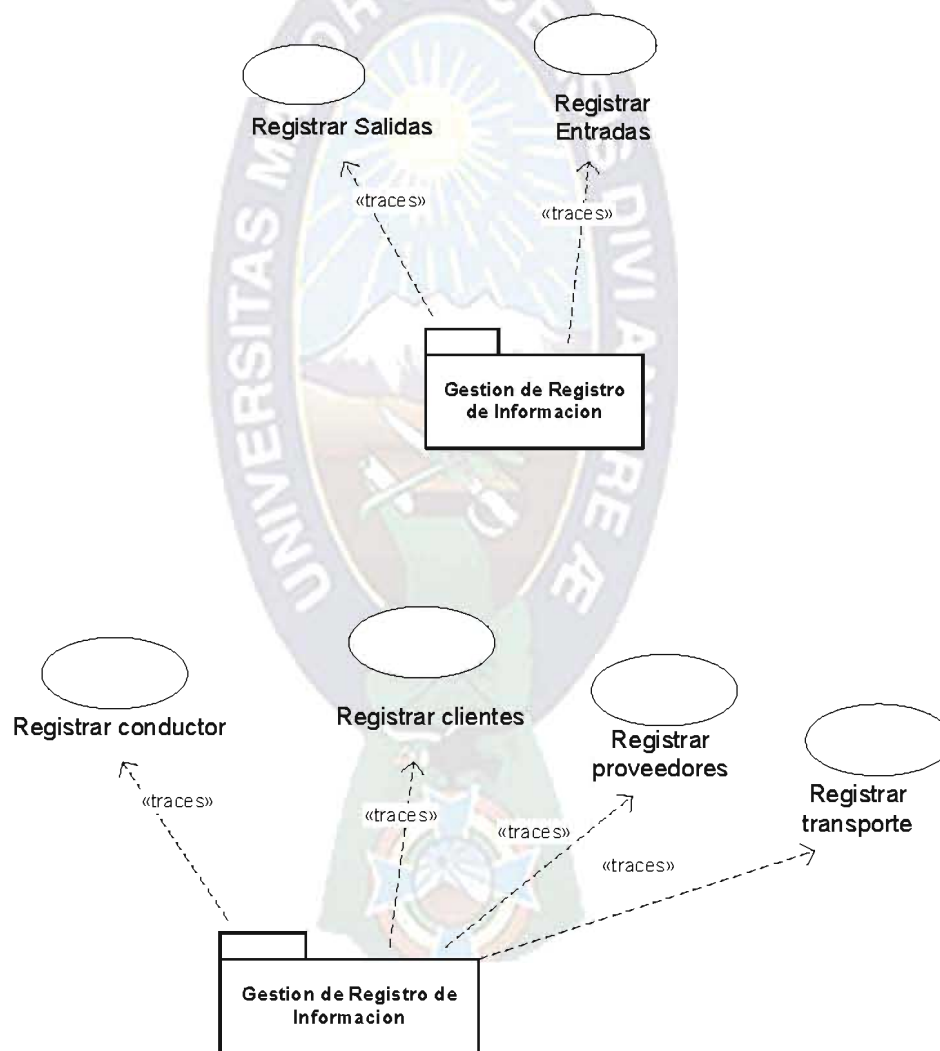
**Tabla N° 3.1:** Caso de uso expandido: Registro de Proveedores

**Fuente:** [Elaboración Propia]

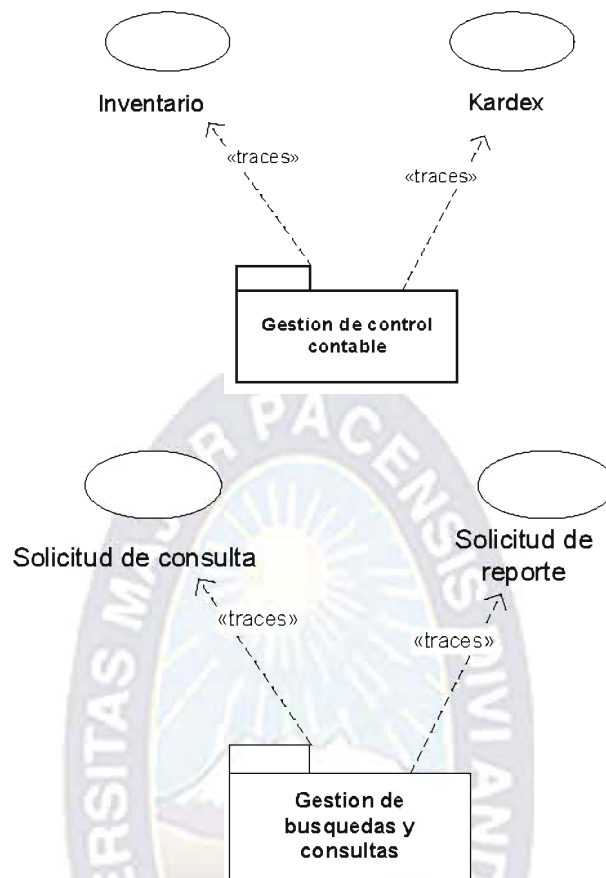
### 3.2.3 Análisis

#### *Análisis de la Arquitectura:*

A partir de los casos de uso hallados en el anterior flujo, se procede a la clasificación de estos, para esto, se agrupa en paquetes de análisis. Los paquetes del análisis identificados para el Sistema de entradas y salidas e inventario se muestran en la figura N°3.6.







**Figura N° 3.6:** Paquete del análisis

**Fuente:** [Elaboración Propia]

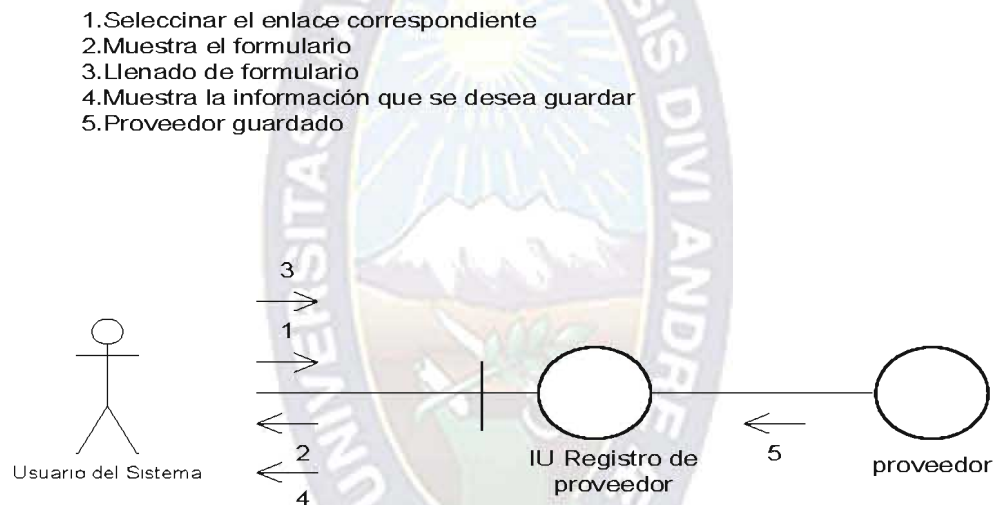
### Analizar un caso de uso

Al igual que la actividad Detallar un caso de uso, en un diez por ciento de todos los casos de uso se irá detallando más adelante en las siguientes fases. Se detallara el caso de uso Registrar proveedor. Para ver todos los casos de uso analizados ir al [Anexo E].

La figura N° 3.7 muestra como estas clases del análisis interactúan entre sí y con el usuario del sistema.

### Caso de uso: Registrar Proveedor

- Clase de entidad: Proveedor
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de Proveedor.



**Figura N° 3.7:** Diagrama de Colaboración del caso de uso:  
Registrar Proveedor

**Fuente:** [Elaboración Propia]

### 3.2.4 Diseño

#### Diseño de arquitectura

Para esta fase, solamente se identificarán los nodos y configuraciones de la red.

## Identificación de nodos y configuración de red

El Sistema de entradas y salidas e inventario será implementado en un solo servidor, el cual el usuario tendrá el acceso, este servidor estará conectado a una impresora y se lo utilizara para la impresión de reportes



**Figura N° 3.8:** Nodos y configuraciones de la red

**Fuente:** [Elaboración Propia]

### 3.3 Fase de elaboración

El propósito de la fase es analizar el dominio del problema, establecer una arquitectura estable que se utilizara para guiar el sistema a lo largo de su vida

#### 3.3.1 Recopilar requisitos

##### Actores y Casos de uso

Para esta fase de desarrollo de software, se continuara con la identificación de actores y casos de uso adicionales a las establecidas en la fase de inicio.

**Actores**, no se han identificado a otros actores de las que ya se ha definido en la fase de inicio.

**Casos de uso**, se mantendrá con los casos de uso identificado en la fase anterior, debido a que no se han identificado otros casos de uso.

Al no haber encontrado nuevos actores y casos de uso, el modelo de casos de uso no sufre ningún cambio.

### Detallar un caso de uso

Para esta fase ya se detallara un ochenta por ciento de los casos de uso para una mejor comprensión de todos los casos de uso ver [Anexo D].

Caso de Uso:	Registro de Clientes
Actores:	Encargado
Propósito	Capturar datos de los Clientes que tiene la empresa
Resumen	Actor registra clientes que tiene la empresa.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R7,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1 Comienza cuando el actor requiere transcribir al cliente	3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este
2. Este caso de uso comienza un actor se Identifica	
4. Realiza petición de acuerdo a la necesidad que tenga:	

<ul style="list-style-type: none"> <li>• Registrar Cliente</li> </ul>	5. Muestra: <ul style="list-style-type: none"> <li>• Listado de Cliente</li> <li>• Formulario de Registro de Clientes</li> </ul>
6. Registra datos necesarios.	7. Valida y registra datos ingresados 8. Almacena registro en la Base de Datos.

**Tabla N° 3.2:** Caso de uso expandido: Registro de Clientes

**Fuente:** [Elaboración Propia]

### 3.3.2 Análisis

#### Análisis de la arquitectura

Como no se ha encontrado nuevos casos de uso, los paquetes del análisis no sufren ningún cambio, ahora se procede a definir las dependencias entre los paquetes del análisis.

Podemos observar en la Figura 3.9, dependencias y capas de paquetes del análisis del sistema de entradas y salidas e inventario.



**Figura N° 3.9:** Dependencias y capas de paquetes del análisis

**Fuente:** [Elaboración Propia]

## **Analizar un caso de uso**

Para esta fase de desarrollo del software ya se tiene analizado un ochenta por ciento del total de casos de uso. Para ver todos los casos de uso analizados ver [Anexo E].

## **Analizar una clase**

En esta actividad, se realizara el análisis de cada clase de análisis identificados en la actividad Analizar un caso de uso, se identifican sus responsabilidades, sus atributos y relaciones con otras clases del análisis

*Clase: Proveedor*

*Rol:* Esta clase se crea en el caso de uso Registrar Proveedor. Debe tener un encargado el cual está encargado de registrar proveedor.

*Responsabilidades:* Del usuario del sistema

Registrar: Proveedor

*Atributos:* Los atributos de la clase de análisis proveedor son:

Código de proveedor, nombre del proveedor, dirección del proveedor, teléfono del proveedor, ciudad y país del proveedor.

## **Analizar un paquete**

Para esta fase se realizara el análisis de paquete: Gestión de registro de información.

Este paquete contiene a las clases de entidad. Proveedor, cliente, conductor, transporte, entrada de productos, salida de productos .y para poder registrar la información, es necesario ingresas al sistema, introduciendo el código y contraseña del usuario del sistema. Los paquetes: Gestión de Registro de información, Gestión de control contable y gestión de búsquedas y consultas, ya que para que se puedan



ejecutar los paquetes mencionados anteriormente, es necesario que la información este almacenada en el sistema.

### 3.3.3 Diseño

#### Diseño de arquitectura

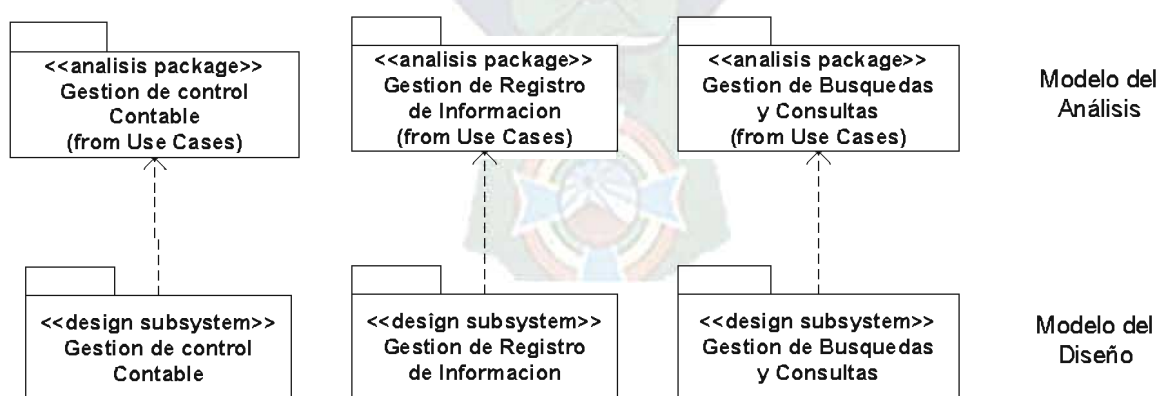
Para esta fase, se continuara con el diseño de la arquitectura, en la fase anterior se identifico los nodos y configuración de red y en esta fase se identificara los subsistemas e interfaces.

#### Identificación de subsistemas e interfaces

En esta actividad se trabaja con los paquetes del análisis identificados en el flujo de trabajo anterior, para la realización de esta actividad se lleva a cabo la siguiente tarea:

#### Identificación de subsistema de aplicación:

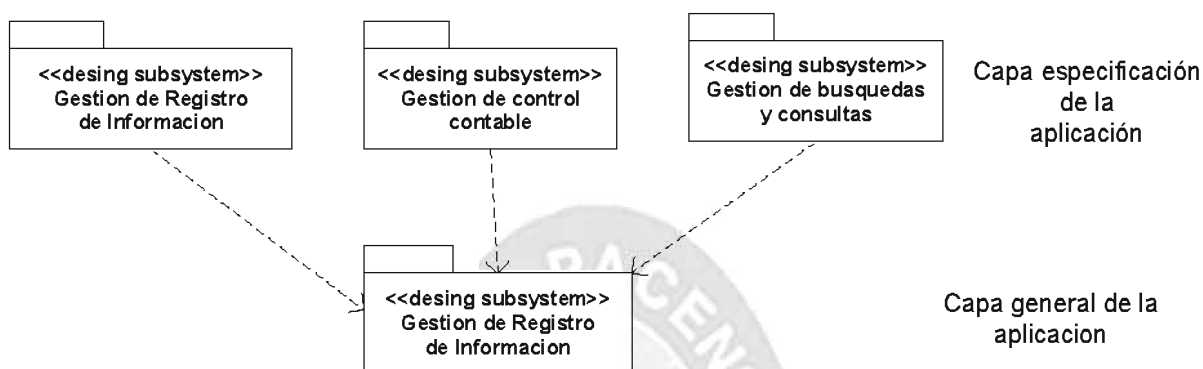
A partir de los paquetes de análisis identificados se hallan los subsistemas de la capa específica de la aplicación y general de la aplicación, como se muestra en la figura 3.10.



**Figura N° 3.10:** Subsistemas identificados a partir de paquete del análisis

**Fuente:** [Elaboración Propia]

Una vez identificados los subsistemas, se determinan las capas específicas y general de la aplicación, como lo muestra la figura 3.11.



**Figura N° 3.11:** Capa específica y general de la aplicación

**Fuente:** [Elaboración Propia]

### Identificación de subsistemas intermedios y de software del sistema

La Figura 3.12 muestra los subsistemas de la capa intermedia y de software del sistema identificados para el Sistema Entrada y Salida e Inventario

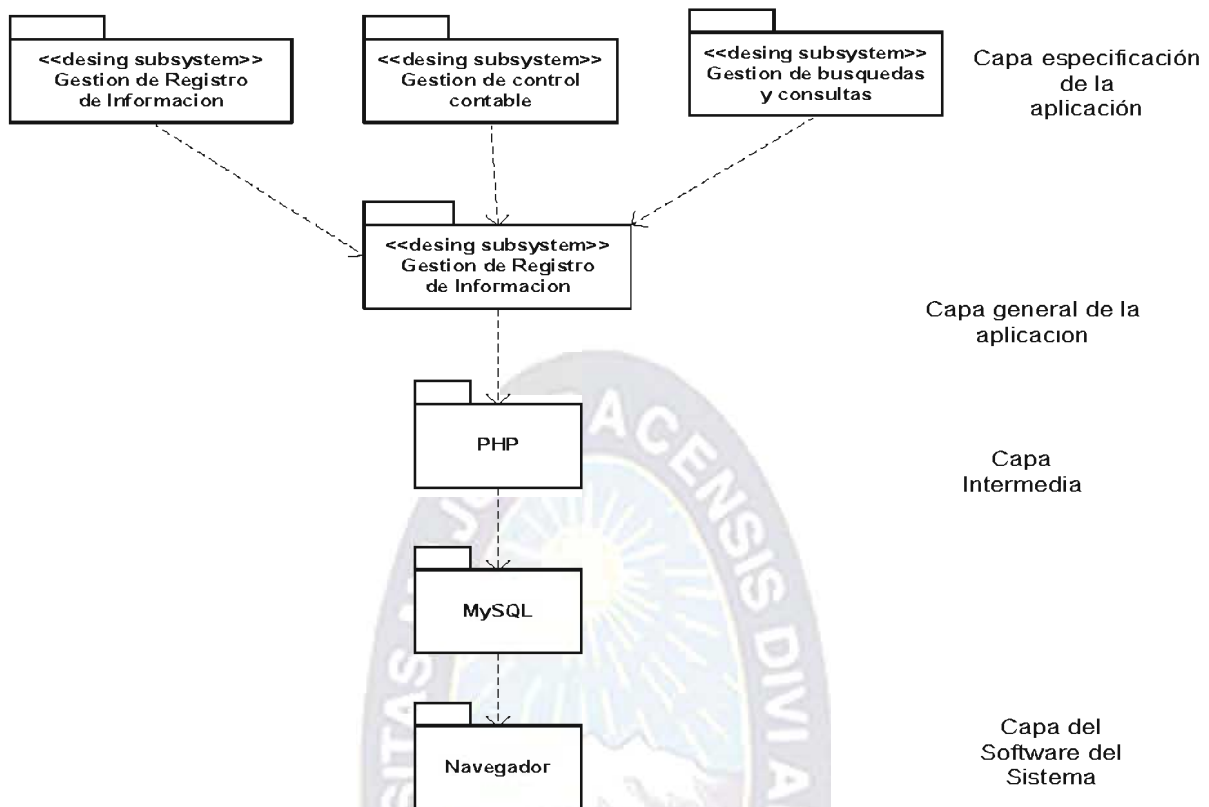


**Figura N° 3.12:** Capa Intermedia y Capa de Software del Sistema

**Fuente:** [Elaboración Propia]

### Definición de dependencias entre subsistemas

Luego de haber identificado las capas, ahora se definirán las dependencias entre los subsistemas, tomando en cuenta que sus contenidos deben tener relación de unos con otros.



**Figura N° 3.13:** Dependencias y capas de los subsistemas

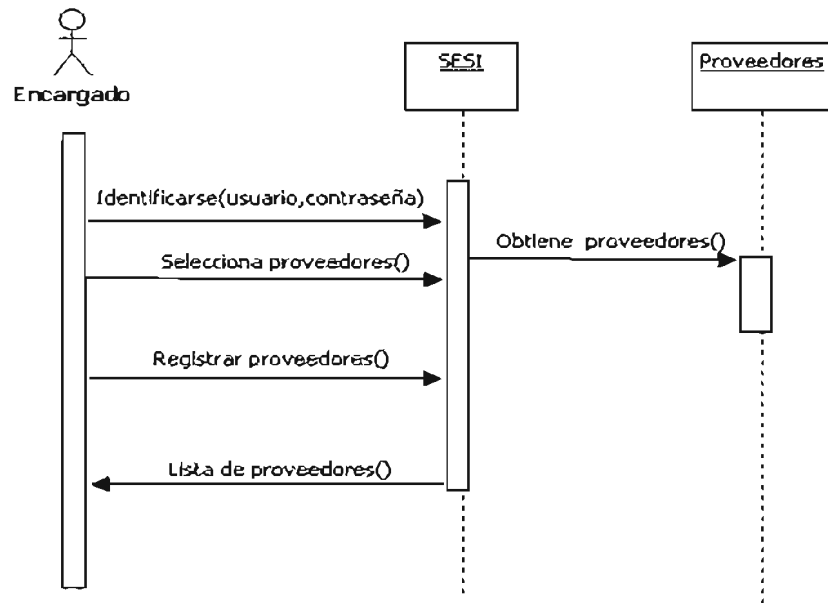
**Fuente:** [Elaboración Propia]

### Diseño de un caso de uso

Para la realización de esta actividad primero se debe identificar a las clases de diseño que participan en la realización del caso de uso, para luego hacer la descripción de interacciones entre objetos del diseño, utilizando diagramas de secuencia.

*Caso de uso:* Registro de proveedor



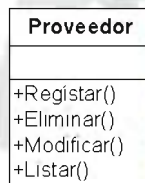


**Figura N° 3.14:** Diagrama de secuencia: Registro de proveedores

**Fuente:** [Elaboración Propia]

### Diseño de una clase

En esta fase se diseña la clase de entidad Proveedor. Para realizar el diseño de esta clase primero se identifica las operaciones que esta clase de diseño va a necesitar. La figura muestra las operaciones de la clase de diseño Proveedor.



**Figura N° 3.15:** Operaciones de la clase proveedor

**Fuente:** [Elaboración Propia]

Después de haber identificado las operaciones de la clase de diseño Proveedor, se identifica los atributos de dicha clase. Los atributos para la clase de diseño

Proveedor son: Código proveedor, nombre proveedor, dirección proveedor, ciudad proveedor.

## **Diseño de un subsistema**

En esta fase, se va a diseñar el subsistema Gestión de Registro de información, para ello se debe realizar las siguientes tareas:

### *Mantenimiento de las dependencias entre subsistemas:*

Durante el diseño de la arquitectura se han identificado las dependencias que existen entre los subsistemas; para el caso del subsistema Gestión de Registro de Información estas dependencias se mantienen.

### *Mantenimiento de interfaces proporcionadas por el subsistema:*

En la fase de construcción, se realiza la actividad Desarrollar Prototipo de Interfaz de usuario ver [Anexo G], en esta actividad también se identifican las interfaces que existen para cada subsistema. Para el subsistema Gestión de Registro de Información, estas interfaces no sufrirán ningún cambio, por lo tanto, se mantendrán.

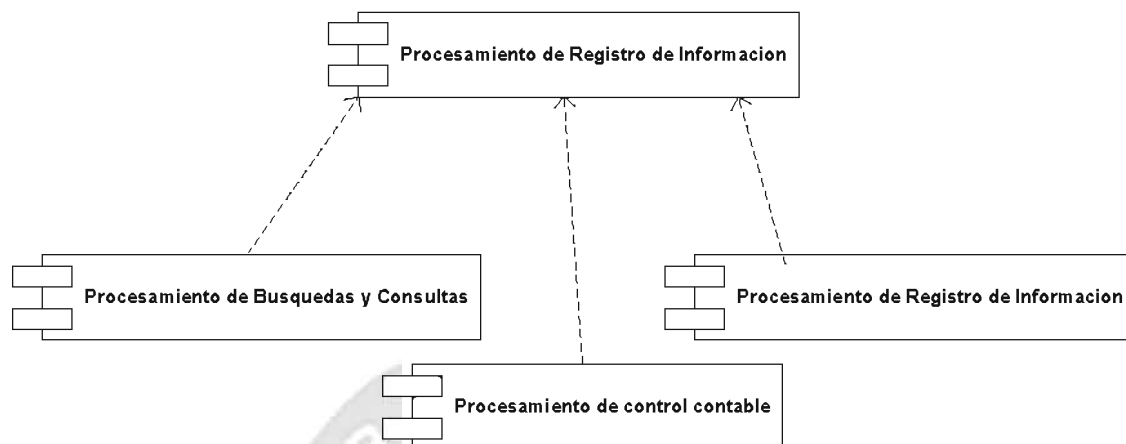
### *Mantenimiento de los contenidos de los subsistemas:*

El contenido del subsistema Gestión de Registro de Información, no sufre ningún cambio, ya que cumple con el objetivo de que ofrece una realización correcta de las operaciones.

## **3.3.4 Implementación**

### **Implementación de la arquitectura**

Los componentes identificados en la figura 3.16 serán asignados al nodo del servidor del usuario del sistema.



**Figura N° 3.16:** Componentes a partir del diseño de la arquitectura

**Fuente:** [Elaboración Propia]

#### **Integrar el sistema:**

- Construir el subsistema Gestión de registro de Información, para llevar a cabo esta construcción se debe implementar los casos de uso, en el orden siguiente: Registrar proveedor, registrar conductor, registrar transporte, registrar entradas de productos, registrar cliente registrar salidas de productos.

-Posteriormente se construirá el subsistema Gestión de control Contable, realizando los casos de uso Inventario.

-Luego se construirá el subsistema de Búsquedas y consultas, llevando a cabo el caso de uso Generar reporte.

En la siguiente fase se llevara a cabo este plan de construcción, el cual al integrar cada construcción, permitirá la construcción del Sistema de Entradas y Salidas e Inventario.

#### **Implementar una clase**

En las fases de Inicio y Elaboración se realizo el análisis de la clase proveedor, para la realización de esta actividad se debe implementar esta clase, creando una base de



datos que contenga a la tabla proveedor, esta tabla deberá contener los siguientes campos: Código proveedor, nombre proveedor, dirección proveedor, ciudad proveedor; estos campos representan los atributos de la clase Proveedor, las operaciones de la clase Proveedor son: Registrar(), Modificar(), Eliminar() y Listar(), para representar estas operaciones se utilizaran métodos, los cuales serán implementados en el sistema.

### **Implementar un subsistema**

Siguiendo el plan de construcción realizado en la actividad Integrar el sistema, para esta fase se implementara el subsistema Gestión de registro de Información, para llevar a cabo esta construcción se debe implementar los casos de uso, en el orden siguiente: Registrar proveedor, registrar conductor, registrar transporte, registrar entradas de productos, registrar cliente registrar salidas de productos, cuando estos casos de uso estén implementados, el subsistema Gestión de registro de Información habrá sido implementado.

### **3.3.5 Pruebas**

#### **Planificación de pruebas**

En esta fase de prueba se evaluar el sistema, se utiliza la prueba de la caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que se usa en cada proceso del Sistema de entradas y salidas e inventario para garantizar que cada línea de código es utilizado al menos una vez.

### **3.4. Fase Construcción**

Al finalizar esta fase se obtendrá una primera versión del sistema de entradas salidas e inventario.

### 3.4.1 Recopilar requisitos

#### Actores y Casos de uso

No se han encontrado otros actores ni casos de uso adicionales por ello no sufre modificaciones el modelo de casos de uso.

**Detallar un caso de uso**, la explicación detallada de los casos de uso lo podemos encontrar en el [Anexo D]

**Desarrollar un prototipo de la interfaz de usuario**, se muestra las interfaces de usuario del Sistema entradas salidas e inventario. [Anexo G]

### 3.4.2 Análisis

#### Análisis de la arquitectura

No se encuentra otros casos de uso, por lo tanto no sufre cambios.

#### Analizar un caso de uso

Para esta fase ya se tiene analizado el 100 por ciento de los casos de uso identificados para este sistema se encuentran en el [Anexo E].

#### Analizar una clase

Para esta se continúa con el análisis de clase realizado en la fase de elaboración. A continuación se muestra el análisis de algunas de las clases del Sistema de Entradas y Salidas e Inventario.

*Clase: Cliente*

*Rol:* Esta clase se crea en el caso de uso Registrar cliente. Debe tener un encargado el cual está encargado de registrar cliente.

*Responsabilidades:* Del usuario del sistema

*Registrar:* cliente

*Atributos:* Los atributos de la clase de análisis cliente son:

Código de cliente, nombre del cliente, dirección del cliente, teléfono del cliente, ciudad y país del cliente.

### **Analizar un paquete**

Al no haberse encontrado otros casos de usos ni otros paquetes, las dependencias identificadas en la actividad Análisis de la Arquitectura se mantiene y por lo tanto no es necesario refinarlas.

### **3.4.3 Diseño**

#### **Diseño de arquitectura**

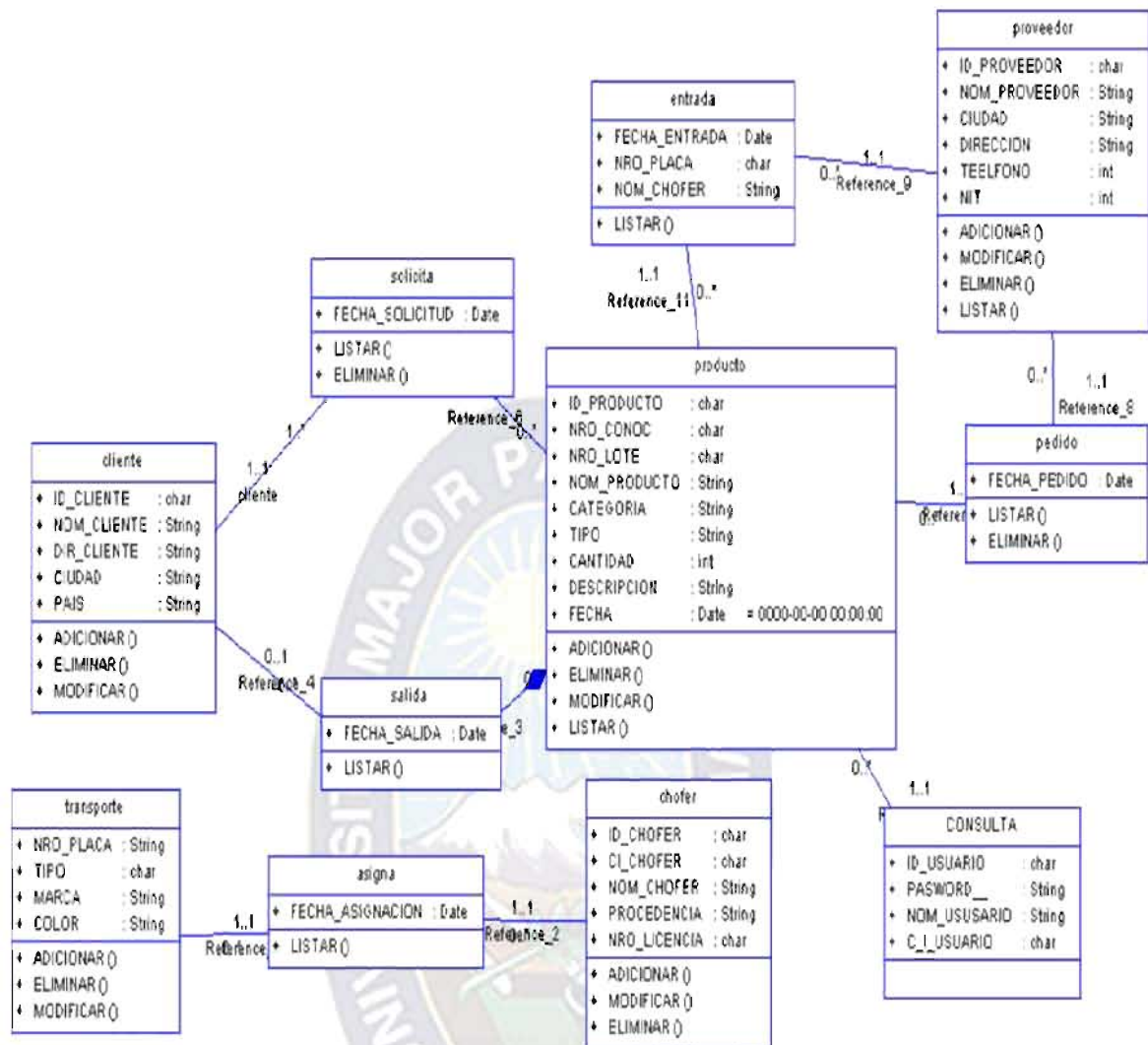
Los elementos hallados en las anteriores fases se mantienen debidos a que el flujo de trabajo no se añadió otros paquetes del análisis.

#### **Diseño de un caso de uso:**

Para esta fase ya se tiene diseñado todos los casos de uso identificados en fases anteriores, el diseño de cada uno de estos casos de uso están en el [Anexo F].

#### **Diseño de una clase**

A continuación se describe los diagramas de clases es el que se muestra tan solo las clase generales de los casos de uso incorporados a la aplicación, de la fase de construcción y de su modelo de datos se describe las entidades que participan en las relaciones definidas en el proyecto. Para esta fase ya se han identificado a todas las clases del Sistema de entradas y salidas e inventario, identificándose así sus atributos y sus operaciones.



**Figura N° 3.17:** Diagrama de Clases: Sistema de Entrada y Salidas e Inventario

**Fuente:** [Elaboración Propia]

### 3.4.4 Implementación

**Implementación de la arquitectura:** La arquitectura del sistema de entradas salidas e inventario no ha sido modificado no necesita ser actualizado.

### Implementar una clase e implementar un subsistema

Para esta fase todas las clases identificadas deben ser implementadas.

## Integrar el sistema

En esta fase, se ha concluido la construcción del Sistema de entradas y salidas e inventario. Se realizara las pruebas respectivas de los procesos del sistema en primer instancia tendremos la prueba del proceso de búsqueda de kardex y las demás pruebas ver [Anexo H].

### 3.4.5 Pruebas

#### Proceso de búsqueda de Kardex

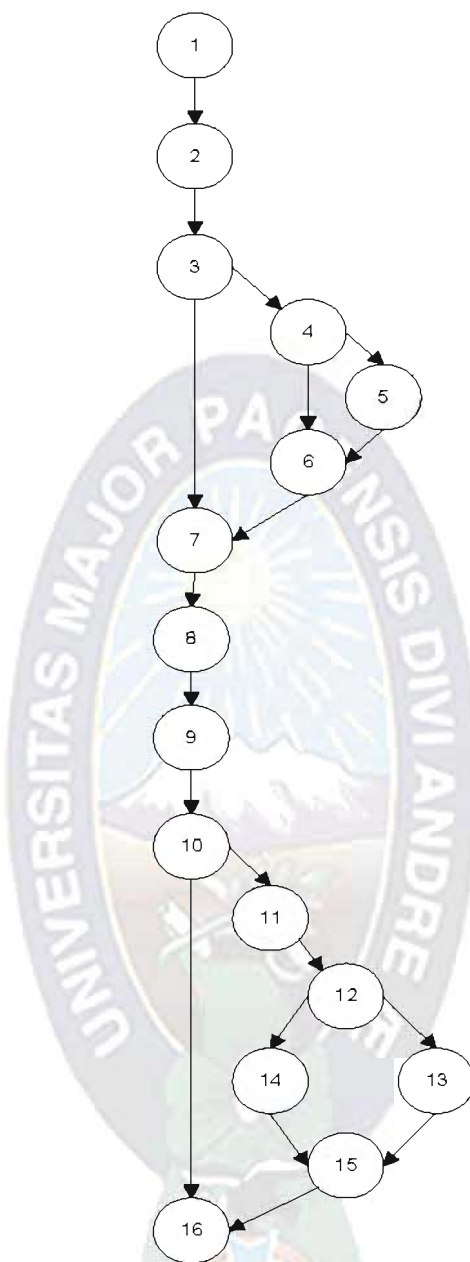
```
<?php
(1) include ("../includes/configurar.php");
(2) include('../includes/conectar.php');
    $id_producto=$_POST['id_producto'];
(3) if($id_producto=="||$id_producto=='---')$id_producto='%';
    $nombre_categoria=$_POST['nombre_categoria'];
(4)     if($nombre_categoria=="||$nombre_categoria=='---')
(5)     $nombre_categoria='%';
(6) $sql="SELECT fecha_reg, 'egreso ' as tipo ,codigo_salida as codigo,
    salidas.id_producto, salidas.nombre_unidad, '' as cantidad_entrada,
    cantidad_salida, costo_unidad,
    importe_total,salidas.nrolote_producto,nombre_categoria,nombre_subcategori
a,total_acumulado, stock_temp FROM salidas, producto WHERE
    salidas.id_producto like '$id_producto' and salidas.id_producto =
    producto.id_producto and nombre_categoria like '$nombre_categoria' unión
    SELECT fecha_reg,'ingreso' as tipo ,codigo_entrada as codigo,
    entradas.id_producto, entradas.nombre_unidad, cantidad_entrada, '' as
    cantidad_salida,costo_unidad,importe_total,entradas.nrolote_producto,nombre
_categoria,nombre_subcategoria,total_acumulado, stock_temp FROM
    entradas, producto
(7) $resul=mysql_query($sql);
    <tr>
    <th class=titulo colspan=100>
    id_producto    <select name="id_producto">
    <?php
(8) $r=mysql_query("select id_producto, nrolote_producto from producto");
    echo printSelect($r, 'id_producto','nrolote_producto',$id_producto);
    ?>
```

```

</select>
nombre_categoria    <select name="nombre_categoria">
<?php
(9) $r=mysql_query("select nombre_categoria from categoria");
    echo printSelect($r,
'nombre_categoria','nombre_categoria',$nombre_categoria);
    ?>
    <?=$nombre_categoria ?>
    </select>
    <input type="submit" value="filtrar" />
    <a href="kardex_sesi(pdf).php?nombre_categoria=<?=$nombre_categoria
?>&id_producto=<?=$id_producto ?>" target="_blank">Exportar PDF</a>
</th></tr><tr>
<th class="titulo" colspan=100>
detalle del kardex
</th></tr><tr>
<th>fecha_reg</th>
<th>detalle</th>
<th>nrolote_producto</th>
<th>entrada</th>
<th>salida</th>
<th>SALDO TEMP</th>
<th>STOCK</th>
<th>unidad</th></tr>
<?php
(10)     while($row = mysql_fetch_assoc($resul))
(11)     {
(12)         if($class == 'even')
(13)             $class = 'odd';
(14)         else $class = 'even?>
(15)         <tr class="<?=$class?>">
            <td><?=$row[fecha_reg] ?></td>
            <td><?=$row[tipo] ?></td>
            <th align="right" style="color:red" ><?=$row[cantidad_entrada] ?></th>
            <th align="right" style="color:red" ><?=$row[cantidad_salida] ?></th>
            <th align="right" style="color:red">
                <?=$row[stock_temp]+$row[cantidad_entrada]-$row[cantidad_salida]
                ?></th>
            <th align="right" style="color:red" ><?=$row[total_acumulado] ?></th>
            <td><?=$row[nombre_unidad] ?></td>
            </tr>
        <?php
(16)     }
    ?>
</table>

```





**Figura N° 3.18:** Grafo del proceso de búsqueda en kardex

**Fuente:** [Elaboración propia]

$$V(G)=A-N+2$$

$$V(G)=20 \text{ aristas} - 16 \text{ nodos} + 2 = 6$$

*Camino 1:* 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 15 – 16

*Camino 2:* 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16

*Camino 3:* 1 – 2 – 3 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16

*Camino 4:* 1 - 2 - 3 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 15 - 16

*Camino 5:* 1 - 2 - 3 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 15 - 16

*Camino 6:* 1 - 2 - 3 - 7 - 8 - 9 - 10 - 16

### **3.5 Fase Transición**

En esta fase ya se tiene la primera versión del sistema de entradas y salidas e inventario para la empresa Bolital. La fase de Transición es cuando el producto de software ha sido completado y se encuentra listo para ser entregado e implementado para los usuarios.



# CAPÍTULO 4

## CALIDAD DEL SOFTWARE

### 4.1 Introducción

La norma la ISO 9126 define un modelo general de calidad basado en características principales para determinar la calidad del software se encuentra la funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad

### 4.2 Funcionalidad

La funcionalidad del software va ser medida utilizando el Punto de Función. Esta relación se deriva con una relación empírica del dominio de información y las evaluaciones de complejidad.

Para medir la funcionalidad del Sistema SESI se deben determinar los siguientes cinco características del ámbito de la información:

*Número de entradas de usuario:* Dentro del Sistema se han hallado diez entradas de usuario.

*Número de salidas de usuario:* Se han encontrado diez salidas de usuario en el Sistema.

*Número de peticiones de usuario:* En el sistema se han encontrado quince peticiones de usuario

*Numero de archivos:* El sistema tiene un archivo maestro lógico, que es la base de datos

*Número de interfaces externas:* El sistema tiene una interfaz externa.

Parámetros de medición	Cuenta		Factor de ponderación				Total
			Simple	Medio	Complejo		
Número de entradas de usuario:	10	x	3	4	6	=	30
Número de salidas de usuario:	10	x	4	5	7	=	40
Número de peticiones de usuario	15	x	3	4	7	=	45
Numero de archivos:	1	x	7	10	15	=	7
Número de interfaces externas	1	x	5	7	10	=	5
Cuenta Total							128

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables? 2
2. ¿Se requiere comunicación de datos? 5
3. ¿Existen funciones de procesamiento distribuido? 1
4. ¿Es crítico el rendimiento? 3
5. ¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado? 5
6. ¿Requiere el sistema entrada de datos interactiva? 4
7. ¿Requiere la entrada de datos interactiva que las transacciones

de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	3
8. ¿Se actualizan los archivos maestros de forma interactiva?	5
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	4
10. ¿Es complejo el procesamiento interno?	4
11. ¿Se ha diseñado el código para ser reutilizable?	5
12. ¿Están incluidas en el diseño la conversión y la instalación?	5
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5

Luego para calcular el punto función (PF) se utiliza la relación siguiente

$$PF = \text{cuenta total} * (X + \text{Min} (Y) * \sum F_i)$$

$$PF = 128 * (0.65 + 0.01) * 55 = 154$$

Si  $\sum F_i = 70$  es considerado como el 100% del valor de ajuste de complejidad, por lo tanto el punto función máximo es:

$$PF = 128 * (0.65 + 0.01 * 70) = 172.8$$

$$PF / PF \text{ máximo} = 154 / 172.8 = 0.8912$$

Por lo tanto, tomando es cuanta el punto función máximo, la funcionalidad del sistema aproximadamente es de 89%.

### 4.3 Confiabilidad

Para el cálculo de la confiabilidad del Sistema de entradas y salidas e inventario se utilizara la siguiente función:

$$F(t) = \begin{cases} PF \times e^{-t/10} & , x > 0 \\ 0 & , e.o.c \end{cases}$$

Donde  $f(t)$  es la probabilidad de falla en un tiempo  $t$  y  $1-F(t)$  es la probabilidad de trabajo sin falla en un tiempo  $t$ . Se determino que el Sistema entradas y salidas e inventario tiene una funcionalidad del 89% por lo tanto, se reemplaza en  $PF$  el valor de 0,89 en la función  $F(t)$ . El cálculo será realizado para una gestión, la cual tiene 12 meses; reemplazando estos valores se tendrá:

$$F(t) = PF \times e^{-t/10}$$

$$F(12) = 0,89 \times e^{-12/10} = 0.27$$

$$1-F(t) = 1-(PF \times e^{-t/10}) = 1-0,27 = 0.73$$

Por lo tanto, la probabilidad de que haya fallas en el sistema en 12 meses es del 27% y la probabilidad de que no haya fallas es del 73% esto indica que el Sistema de entradas y salidas e inventario es confiable.

### 4.4 Eficiencia

El uso de los recursos del sistema determina el grado de eficiencia del mismo. El grado de eficiencia que se espera del sistema SESI está en función a la cantidad de recursos informáticos y del código que fue necesario para su construcción, así como de los siguientes factores de calidad: El tiempo de uso del sistema.

### 4.5 Portabilidad



La portabilidad es el esfuerzo necesario para transferir el programa de un entorno de sistema hardware y/o software a otro. La portabilidad del Sistema de entradas y salidas e inventario la dividiremos en dos secciones:

Portabilidad del lado del Servidor:

A nivel de Software es portable bajo los siguientes sistemas operativos: Windows XP, Windows2003 Server y Windows Vista

Portabilidad del lado del Servidor:

A nivel de Software, el ordenador del cliente funciona bajo los siguientes sistemas operativos: Windows XP, Windows2003 Server y Windows Vista

El sistema Funciona bajo cualquiera de los siguientes navegadores: Internet Explorer, Mozilla Firefox, Opera, Netscape.

#### **4.6 Mantenibilidad**

La mantenibilidad se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el software, para esto se va a hallar el índice de madures del software que proporciona una indicación de la estabilidad del producto software. Se debe determinar los siguientes datos:

$Mt = \text{numero de módulos en la versión actual} = 8$

$Fc = \text{numero de módulos en la versión actual que se han cambiado} = 0$

$Fa = \text{numero de módulos en la versión actual que se han añadido} = 2$

$Fd = \text{numero de módulos de la versión anterior que se han borrado en la versión actual en la versión actual} = 0$

Luego de haber determinado los datos necesarios para hallar IMS, ahora se remplazara los datos en la siguiente ecuación:

$$IMS = [Mt - (Fc + Fa + Fd)] / Mt$$

$$IMS = [8 - (0 + 0 + 2)] / 8$$

$$IMS = 0.75 = 75\%$$

Por lo tanto el sistema SESI es estable en un 75%, que es la facilidad de mantenimiento en la corrección de fallas o errores del sistema, y su adaptación a nuevas necesidades.



## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

## 5.1 Conclusiones

Al concluir el presente proyecto se observa que se han realizado todas las actividades propuestas al inicio.

Se desarrollo el Sistema de Entradas y Salidas e Inventario para la empresa Bolital S.R.L., empleando la metodología RUP. Con lo que se logro automatizar el manejo de todas las actividades primordiales que desarrolla la empresa como es el inventario el cual ayuda de gran medida a la empresa.

Se logro desarrollar los siguientes módulos:

Modulo de control de entradas de productos.

Modulo de control de salidas de productos.

Modulo de control de inventarios

Se logro un modulo que genere reportes de los productos (Kardex).

Se logro construir una base de datos para almacenar toda la información de los productos, para esto se hizo uso del gestor de Base de Datos MySQL.

Se establecieron mecanismos de seguridad para precautelar el acceso a la información que brinda el sistema, estos mecanismos son: La autenticación utilizando nombre y contraseña.

## 5.2 Recomendaciones

Luego de haber desarrollado el Sistema de Entradas y Salidas e Inventario, se hace las siguientes recomendaciones:

El sistema desarrollado para la empresa, no se podrá integrar con algún sistema contable.

Se recomienda realizar copias de seguridad de la base de datos periódicamente para evitar pérdida de información.

Se recomienda estar protegido de algún antivirus, porque puede haber pérdida de información.



## BIBLIOGRAFÍA

### REFERENCIAS BIBLIOGRÁFICAS

[PRES02] Roger S. Pressman R. S. 2002, "Ingeniería de Software Un enfoque práctico" Quinta edición. Interamericana de España /McGraw-Hill: Concepción Fernández Madrid.

[LARM00] Craig Larman, UML y Patrones, Introducción al análisis y diseño orientado a objetos.

[ALAR00] Raúl Alarcón 2000, Diseño Orientado a Objetos con UML Grupo Eidos.

[TCR03] Ernest Teniente López, Dolors Costal Costa, M. Ribera S. 2003 Especificación de sistemas software en UML.

[TAHA96] Handy -Taha "Investigación Operativa (TAHA) Una Introducción" Modelos de Inventarios Determinísticos.

[WLW92] Wayne L. Winston, Investigación de operaciones Modelos determinísticas de inventario 1992.

[OFJ06] Orellana Funes Juan, "El ABC de la Contabilidad" Edición 2006 Editorial Sabiduría Cochabamba – Bolivia.

[SCHMU] Joseph Schmuller Aprendiendo UML en 24 horas

[UML00] UML Resource Center. Rational Software. <http://www.rational.com/uml/>

[JACOBSON00] Jacobson, I., Booch, G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, 2000 Addison Wesley

[KRU00] Kruchten, P., The Rational Unified Process: An Introduction, 2000 Addison Wesley

[KRU95] Kruchten, P. Architectural BlueprintsView Model of Software Architecture.

IEEE Software 12 (6), November 1995,

[RSC02] Rational Software Corporation, Product: Rational Software Corporation, 2002

[RSC98] Rational Software Corporation, Rational Unified Process. Best Practices for Software Development Teams, 1998





# ANEXOS

## ANEXO A Aspectos legales de la empresa



### CERTIFICACIONES BOLITAL ALMENDRAS

Almendras BOLITAL cuenta con los Siguients Documentos:

- Número de Identificación Tributaria (NIT)
- Padrón Municipal
- Certificado de Registro Sanitario
- Licencia Superintendencia Forestal
- Registro Único de Exportadores
- Certificado de aprobación de la Alcaldía de Riberalta
- Certificación para exportar productos orgánicos por la certificadora "IMO"
- Manifiesto Ambiental

**CÁMARA NACIONAL DE EXPORTADORES DE BOLIVIA (CANEX)**

**REGISTRO DEL EXPORTADOR**

N° 0500000010

**SE CERTIFICA**

Que la Empresa: **ALMENDRAS BOLITAL S.R.L.**

Cumple con los Requisitos Sanitarios según sistema Fitosanitario.

Quantitativo registrado con el R.S. No:

**01 02 03 04 05 06 07 08 09 10**

**SE CERTIFICA**

Que la Empresa: **ALMENDRAS BOLITAL S.R.L.**

Cumple con los Requisitos Sanitarios según sistema Fitosanitario.

Quantitativo registrado con el R.S. No:

**01 02 03 04 05 06 07 08 09 10**

**SE CERTIFICA**

Que la Empresa: **ALMENDRAS BOLITAL S.R.L.**

Cumple con los Requisitos Sanitarios según sistema Fitosanitario.

Quantitativo registrado con el R.S. No:

**01 02 03 04 05 06 07 08 09 10**

**Registro del Exportador**

**REPÚBLICA DE BOLIVIA**

**Ministerio de Agricultura, Ganadería y Fomento (MAGF)**

**Servicio Nacional de Sanidad Agropecuaria e Inspección Alimentaria (SENASAG)**

**CERTIFICADO DE REGISTRO SANITARIO**

N° 50029

**SE CERTIFICA**

Que la Empresa: **ALMENDRAS BOLITAL S.R.L.**

Cumple con los Requisitos Sanitarios según sistema Fitosanitario.

Quantitativo registrado con el R.S. No:

**01 02 03 04 05 06 07 08 09 10**

**SE CERTIFICA**

Que la Empresa: **ALMENDRAS BOLITAL S.R.L.**

Cumple con los Requisitos Sanitarios según sistema Fitosanitario.

Quantitativo registrado con el R.S. No:

**01 02 03 04 05 06 07 08 09 10**

**SE CERTIFICA**

Que la Empresa: **ALMENDRAS BOLITAL S.R.L.**

Cumple con los Requisitos Sanitarios según sistema Fitosanitario.

Quantitativo registrado con el R.S. No:

**01 02 03 04 05 06 07 08 09 10**

**Certificado del SENASAG**

**REPUBLICA DE BOLIVIA**  
**SUPERINTENDENCIA FORESTAL DE LA NACION**  
**LEY FORESTAL 1706, 12 DE JULIO 1996**

**LICENCIA ADMINISTRATIVA DE  
 FUNCIONAMIENTO  
 DE ACTIVIDAD FORESTAL**

Denominación o Jurisdicción: **ALMENDRAS BOUTAL S.R.L.**  
 Domicilio en: **AV. EJERCITO Nº 851 - RIVERA**  
 Teléfono: **802-3173** Cód.: **2303** Búsqueda: **8430413**  
 Se encuentra sujeta al Registro de Exportación por parte del Registro  
 Forestal bajo el Código: **2304-411**  
 Categoría: **EXPORTADOR "A"**

Habiendo cumplido con los requisitos exigidos en los Art. 72º, Inc. c) y 76º del  
 Decreto Supremo Nº 24459 Derogando de la ley forestal 1706 y la Decisión  
 sobre la Inscripción Anual de Empresas matricadas en el Registro Forestal de  
 la Nación, aprobada por la Superintendencia Forestal, queda autorizada para  
 desarrollar sus respectivas actividades durante la gestión señalada.  
 A partir de la fecha la empresa está obligada a cumplir con los deberes forestales  
 sobre el establecimiento, procesamiento y comercialización de madera prima en  
 estricta conformidad a la Norma Técnica D.N. Nº 134/99.

Esta licencia tiene validez hasta: **31 DE DICIEMBRE DE 2004**

Lugar: **RIVERA** Fecha: **18** de **JULIO** de **2004**

*[Firmas y sellos de autoridades]*

**ESTA LICENCIA DEBE SER COLOCADA EN LUGAR VISIBLE**

Superintendencia Forestal

**CERTIFICATE**  
 N.º 238  
**ALMENDRAS BOUTAL S.R.L.**  
 Av. Del Ejercito Final S/N, Rivera, Serbi Bolivia

The Institute for Market Ecology (IMO) hereby certifies that the above mentioned operation  
 was inspected and certified according to the control procedures as outlined in Regulation  
 (EC) No. 1831/2003 and that the given plant material and products were found to be in  
 compliance with the production rules defined in said Regulation. It is the certificate holder's  
 duty to report to the government compliance with the specified requirements.

Por medio del presente documento, el Instituto de Ecología de Mercado (IMO) certifica que el  
 Operador antes mencionado ha sido verificado y certificado conforme a los procedimientos  
 de control establecidos según el Reglamento Europeo (2003/18) y que los productos y  
 productos de origen vegetal fueron certificados con los niveles de producción establecidos  
 en dicho Reglamento. El Operador que aparece ante nosotros tiene plena responsabilidad  
 de cumplir permanentemente con los requisitos correspondientes.

Identidad Actual: **Organic Agricultural Production**  
 Controlado: **Processing and Marketing of Organic Products**  
 (Producción y Comercialización de Productos Orgánicos)

Fecha: **2007** Cumplimiento Obligatorio de exportación: **2007**

Organismo:  
 Alimento:  
 Dirección de certificación: **Alimentación Orgánica**  
 Lugar de origen: **Chile**  
 Lugar de destino: **Chile**

Elaborado: **27.08.2007**

*[Firma y sello de IMO]*

Certificado IMO



## ANEXO B

### Productos Bolital S.R.L.

#### PRODUCTOS

##### BOLITAL ALMENDRAS



El producto a comercializar es la Almendra (castaña) mayormente conocida como BRAZIL NUTS.

Existen también Castaña de tipo Orgánica, que son las que carecen de una certificación y provienen de bosques manejados inspeccionados por los ingenieros de la Certificadora desde el acopio hasta el transporte de la misma.



#### EL ÁRBOL

Su nombre científico es *Bertholletia Excelsa*, conocido como Almendro o Castaño, puede tener hasta 50 mts. de altura y un diámetro de 2 mts. en la base y crece en la zona amazónica. El árbol de castaño se encuentra en mayor cantidad en el dpto. de Pando y la provincia Vaca Díez del dpto. del Beni, ha sido llamado durante décadas el árbol sagrado de la selva amazónica, el periodo de zafra se realiza durante los meses de Diciembre, Enero, Febrero y Marzo.



#### EL FRUTO

Coco leñoso y esférico de tamaño variable, su interior está dividido en 4 compartimientos que contienen 4 a 5 semillas, o sea un total de 16 a 20 castañas en cada coco.



#### LA SEMILLA

Coco leñoso y esférico de tamaño variable, su interior está dividido en 4 compartimientos que contienen 4 a 5 semillas, o sea un total de 16 a 20 castañas en cada coco.

#### TIPOS

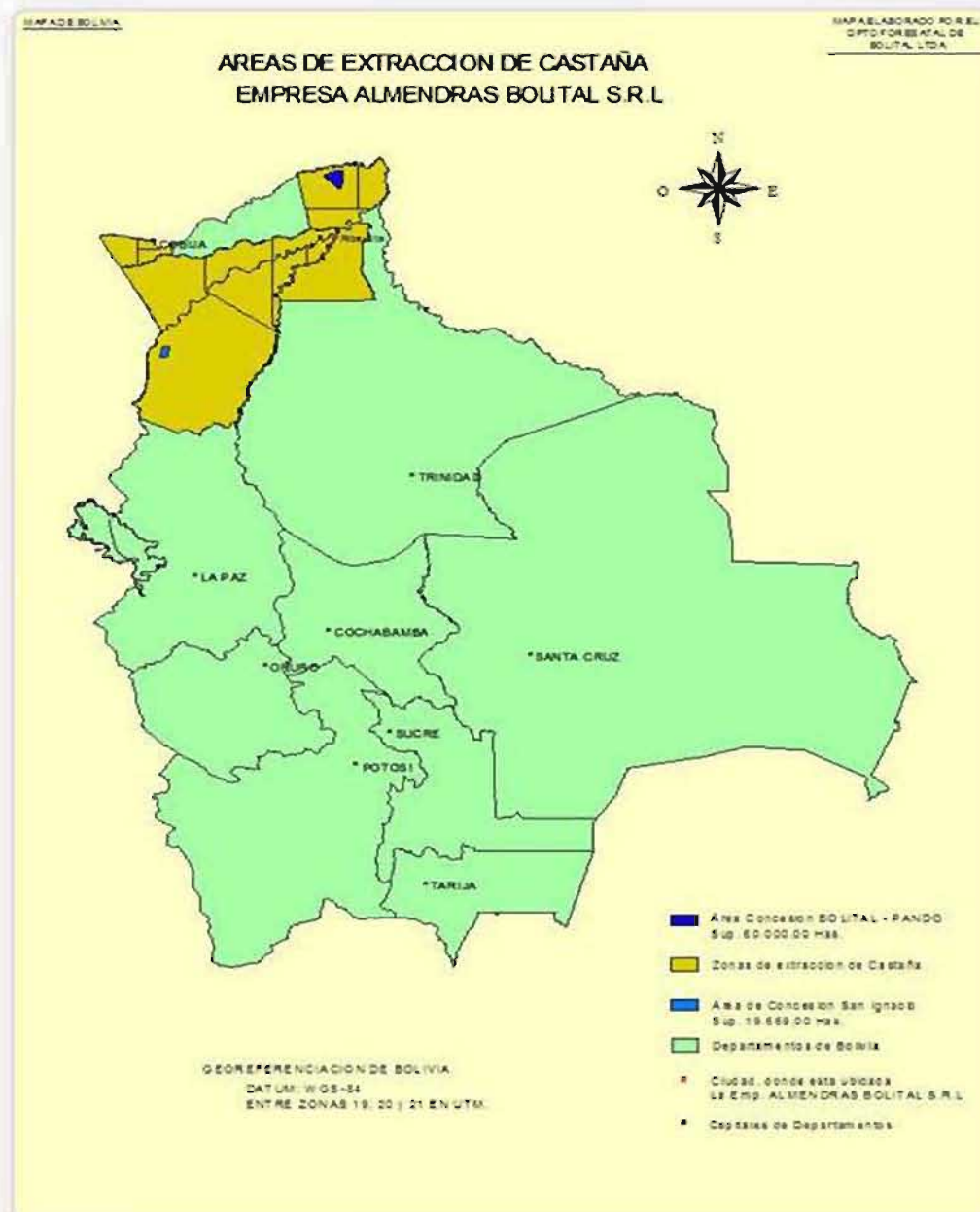
##### DE ALMENDRAS

## ANEXO C

### Proveedores de la empresa

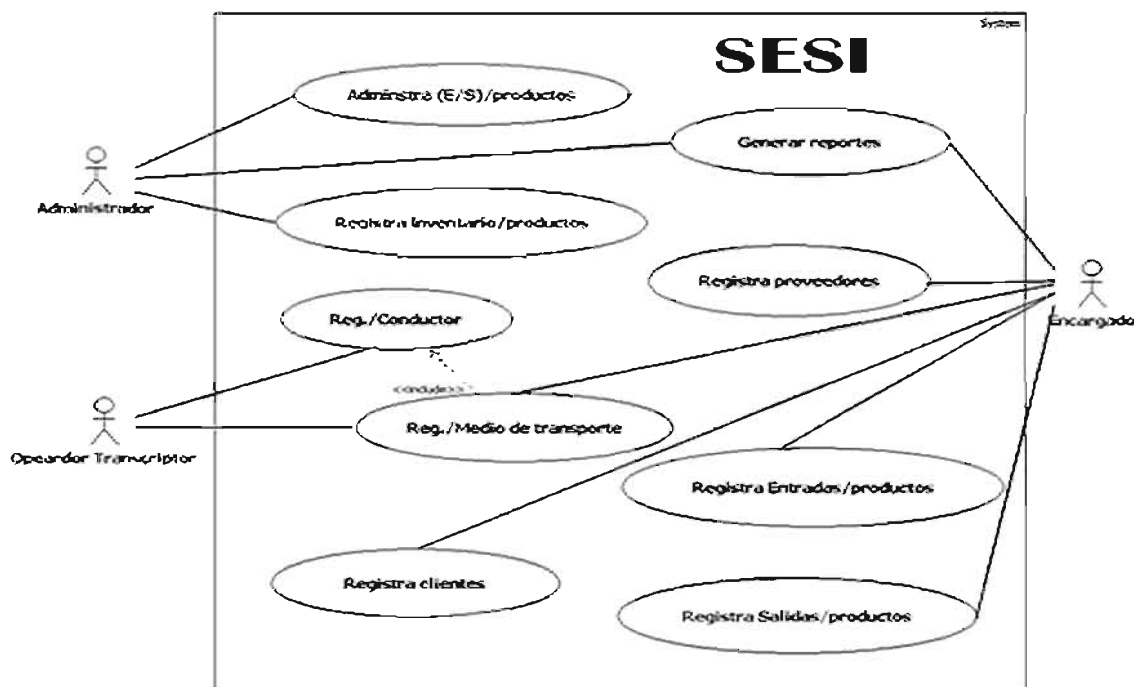
Almendras BOLITAL" cuenta con los siguientes proveedores:

- Áreas Forestales
- TCO
- Comunidades Campesinas



## ANEXO D

### Casos de uso detallados



### Descripción de Casos de Uso

Caso de Uso N° 1:	Administra (Entradas/Salidas)de Productos
Actores:	Administrador
Propósito	Capturar datos de entradas, salidas de productos
Resumen	El Administrador se encarga de administrar todas las actividades.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R2,R12
CURSO NORMAL DE EVENTOS	



Acción del Actor	Respuesta del sistema
1. Este caso de uso comienza el Administrador se identifica  3. Selecciona del menú que desea  5. Verifica datos	2. Verifica autenticación del Administrador y muestra un entorno del sistema de acuerdo al perfil que tiene Administrador  4. Muestra módulos del sistema: <ul style="list-style-type: none"> <li>• Inventario</li> <li>• Entradas</li> <li>• Salidas</li> </ul> 6. Genera reportes 7. Almacena en la Base de Datos.

**Tabla D.1.** Caso de uso expandido: Administra (Entradas/Salidas) de Productos

**Fuente:** [Elaboración Propia]

Caso de Uso N° 2:	Registro de Inventario de productos
Actores:	Administrador
Propósito	Permite el registro de los inventario
Resumen	El Administrador de la empresa realizara el registro de los datos generales del inventario.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R2,R3
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1. Este caso de uso comienza con el Administrador. se Identifica  3. Realiza petición de acuerdo a la necesidad que tenga: <ul style="list-style-type: none"> <li>• Registro de inventario productos</li> </ul> 5. Registra datos necesarios.	2. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este  4. Muestra: <ul style="list-style-type: none"> <li>Listado inventario de productos</li> <li>Formulario de Registro de inventario de productos</li> </ul> 6. Valida y registra datos

	<p>ingresados</p> <p>7. Almacena registro en la Base de Datos</p>
--	---

**Tabla D.2.** Caso de uso expandido: Registro de Inventario de productos

**Fuente:** [Elaboración Propia]

Caso de Uso N° 3:	Registro de Salidas de Productos
Actores:	Encargado
Propósito	El encargado realiza el registro de las salidas de productos
Resumen	El actor registra datos de salidas de productos que salen de la empresa.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R4,R11,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
<p>1. Este caso de uso comienza con el Encargado. se Identifica</p> <p>3. Realiza petición de acuerdo a la necesidad que tenga:</p> <ul style="list-style-type: none"> <li>Registrar de salidas de productos</li> </ul> <p>5. Registra datos necesarios.</p>	<p>2. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este</p> <p>Muestra:</p> <p>Listado de las salidas de productos</p> <p>Formulario de Registro de salidas de productos</p> <p>Listado de clientes</p> <p>Listado de medio de transporte</p> <p>Listado de conductor</p> <p>Listado de productos</p> <p>6. Valida y registra datos ingresados</p>

	7. Almacena registro en la Base de Datos
--	--

**Tabla D.3.** Caso de uso expandido: Registro de Salidas de Productos

**Fuente:** [Elaboración Propia]

Caso de Uso N° 4:	Registro de Entradas de Productos
Actores:	Encargado
Propósito	Registrar el Entradas de productos de la Empresa
Resumen	Encargado registra datos de productos que entran a la empresa.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R4,R11,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1. Este caso de uso comienza un Encargado de la Empresa se Identifica	2. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este
3. Realiza petición de acuerdo a la necesidad que tenga: <ul style="list-style-type: none"><li>• Registrar Entradas de Productos</li></ul>	
5. Registra datos necesarios.	4. Muestra: <ul style="list-style-type: none"><li>• Listado de Entrada de Productos</li><li>• Formulario de Registro de entrada de productos</li></ul> Listado de proveedores Listado de medio de transporte Listado de conductor Listado de productos
	6. Valida y registra datos ingresados
	7. Almacena registro en la Base de Datos.

**Tabla D.4.** Caso de uso expandido: Registro de Entradas de Productos

**Fuente:** [Elaboración Propia]

Caso de Uso N° 5:	Registro de Clientes
Actores:	Encargado
Propósito	Capturar datos de los Clientes que tiene la empresa
Resumen	Actor registra clientes que tiene la empresa.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R7,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1 Comienza cuando el actor requiere transcribir al cliente 2. Este caso de uso comienza un actor se Identifica  4. Realiza petición de acuerdo a la necesidad que tenga: <ul style="list-style-type: none"> <li>• Registrar Cliente</li> </ul> 5. Registra datos necesarios.	3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este 5. Muestra: <ul style="list-style-type: none"> <li>• Listado de Cliente</li> <li>• Formulario de Registro de Clientes</li> </ul> 6. Valida y registra datos ingresados 7. Almacena registro en la Base de Datos. 8. Almacena registro en la Base de Datos.

**Tabla D.5.** Caso de uso expandido: Registro de Clientes  
**Fuente:** [Elaboración Propia]

Caso de Uso N° 6:	Registro de Proveedores
Actores:	Encargado
Propósito	Capturar datos de los Proveedores
Resumen	Actor registra proveedores que tiene la empresa.
Tipo	Primario y Esencial
Referencias	R1, R6,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema

1. Comienza cuando el actor requiere transcribir proveedores de la empresa. 2. Este caso de uso comienza un actor se identifica  4. Realiza petición de acuerdo a la necesidad que tenga: <ul style="list-style-type: none"> <li>• Registrar Proveedor</li> </ul> 6. Registra datos necesarios.	3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este  5. Muestra: <ul style="list-style-type: none"> <li>• Listado de Proveedores</li> <li>• Formulario de Registro de Proveedores</li> </ul> 7. Valida y registra datos ingresados  8. Almacena registro en la Base de Datos.
--	--

**Tabla D.6.** Caso de uso expandido: Registro de Proveedores  
**Fuente:** [Elaboración Propia]

Caso de Uso N° 7:	Registro de medios de transporte
Actores:	Encargado, Transcriptor
Propósito	Registrar medio de transporte
Resumen	Actor registra medio de transporte que se requiere para las salidas y entradas de productos
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R8,R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1. Comienza cuando el actor requiere transcribir el medio de transporte 2. Actor se identifica en el sistema  4. Realiza petición de acuerdo a la necesidad que tenga: <ul style="list-style-type: none"> <li>• Registrar Medios de transporte.</li> </ul>	3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este  5. Muestra: <ul style="list-style-type: none"> <li>• Listado de Medio de transporte</li> </ul>

6. Registra datos necesarios.	<ul style="list-style-type: none"> <li>• Formulario de Registro medio de transporte</li> </ul> 7. Valida y registra datos ingresados 8. Almacena registro en la Base de Datos.
-------------------------------	---

**Tabla D.7.** Caso de uso expandido: Registro de medios de transporte

**Fuente:** [Elaboración Propia]

Caso de Uso N° 8:	Registro de Conductor
Actores:	Encargado, Transcriptor
Propósito	Capturar datos de los
Conductores	Actor registra conductores que tiene la empresa.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R19, R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1. Comienza cuando el actor requiere transcribir los conductores de la empresa. 2. Este caso de uso comienza un actor se Identifica  4. Realiza petición de acuerdo a la necesidad que tenga: <ul style="list-style-type: none"> <li>• Registrar Conductor</li> </ul> 6. Registra datos necesarios.	3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este  5. Muestra: <ul style="list-style-type: none"> <li>• Listado de Conductor</li> <li>• Formulario de Registro de Conductor</li> </ul> 7. Valida y registra datos ingresados 8. Almacena registro en la Base de Datos.

**Tabla D.8.** Caso de uso expandido: Registro de Conductor

**Fuente:** [Elaboración Propia]



Caso de Uso N° 9:	Generar Reportes
Actores:	Administrador, Encargado
Propósito	Generar Reportes referentes a las entradas y salidas e inventario de productos
Resumen	Se requiere del sistema que este genere reportes a medida para ser entregado a la Administración para la toma de decisiones.
Tipo	Primario y Esencial
Referencias Cruzadas:	R1, R12
CURSO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del sistema
1. Este caso de uso comienza cuando un actor requiere información sobre las entradas y salidas e inventario de productos. 2. Se Identifica en el Sistema  4. Seleccionando la opción Reportes   6. Elige Opción de Reporte que necesita.  8. Imprime Reporte.	3. Verifica autenticación del usuario y muestra un entorno del sistema de acuerdo al perfil que tiene este 5. Muestra las siguientes opciones de reportes: <ul style="list-style-type: none"> <li>• Entradas de productos Proveedores Medio de transporte Conductor</li> <li>• Salidas de productos Clientes Medio de transporte Conductor</li> <li>• Inventarios de productos</li> </ul> 7. Genera reporte Seleccionado

**Tabla D.9.** Caso de uso expandido: Generar Reportes.

**Fuente:** [Elaboración Propia]

### Funciones del Sistema

En la siguiente tabla puede apreciar las funciones básicas del sistema:

REF.	FUNCIÓN	CATEGORÍA
R1	Realiza autenticación de usuario y contraseña para ingresar al sistema	Evidente
R2	Administra todos los productos que se tiene la empresa	Evidente
R3	Registra, lista y actualiza el inventario	Evidente
R4	Registra, lista y actualiza entradas de productos.	Evidente
R5	Registra, lista y actualiza salida de productos	Evidente
R6	Registra, lista y actualiza datos de Proveedores	Evidente
R7	Registra, lista y actualiza datos de clientes	Evidente
R8	Registra, lista y actualiza datos del medio de transporte	Evidente
R9	Registra, lista y actualiza datos de conductores	Evidente
R10	Genera reporte sobre depreciaciones de la gestión actual y de gestiones anteriores	Evidente
R11	Genera Reportes: Inventario Entradas de productos Salidas de productos	Evidente
R12	Genera Consultas Listado de productos Listado de clientes Listado de proveedores Listado de medio de transportes Listado de Conductores	Evidente

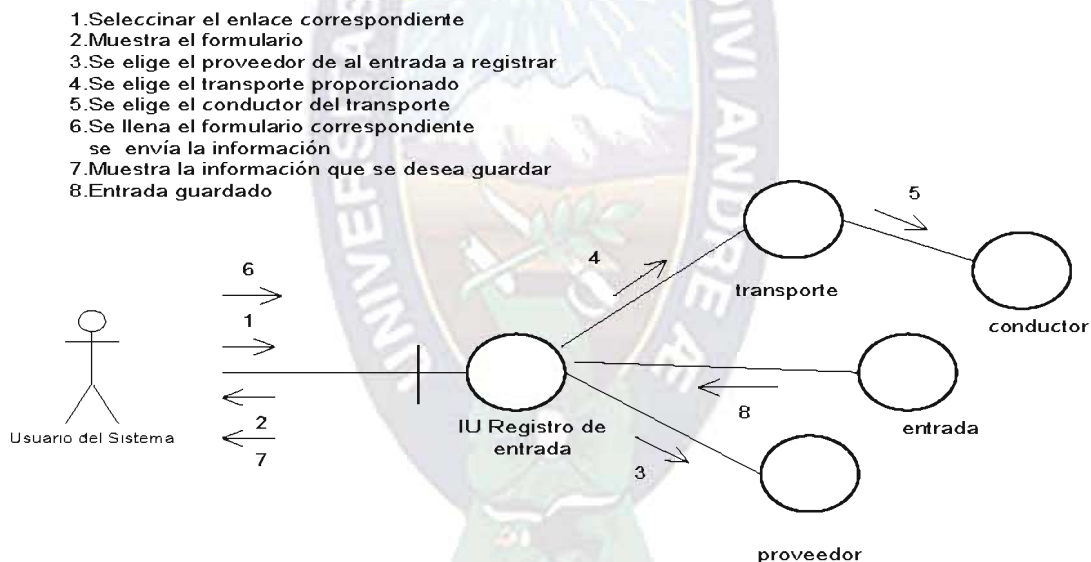
**Tabla D.10.** Funciones del Sistema  
**Fuente:** [Elaboración Propia]

## ANEXO E

### Análisis de Casos de uso

#### Caso de uso: Registrar Entradas

- Clase de entidad: Entradas
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de entradas.

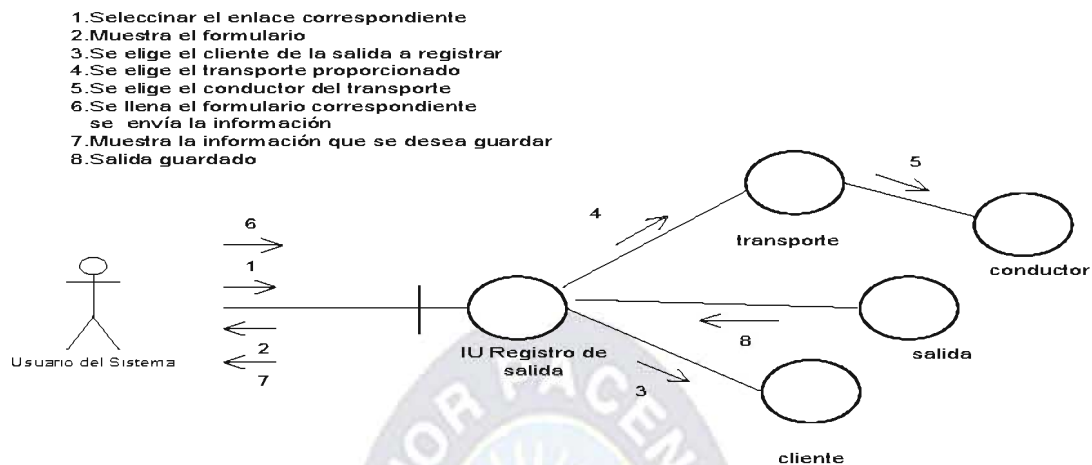


**Figura E.1.** Diagrama de Colaboración del caso de uso: Registrar entradas

**Fuente:** [Elaboración Propia]

#### Caso de uso: Registrar Salidas

- Clase de entidad: Salidas
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de salidas.



**Figura E.2.** Diagrama de Colaboración del caso de uso: Registrar salidas

**Fuente:** [Elaboración Propia]

### Caso de uso: Registrar conductor

- Clase de entidad: conductor
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de conductor



**Figura E.3.** Diagrama de Colaboración del caso de uso: Registrar conductor

**Fuente:** [Elaboración Propia]

### Caso de uso: Registrar Transporte

- Clase de entidad: Transporte
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de Transporte



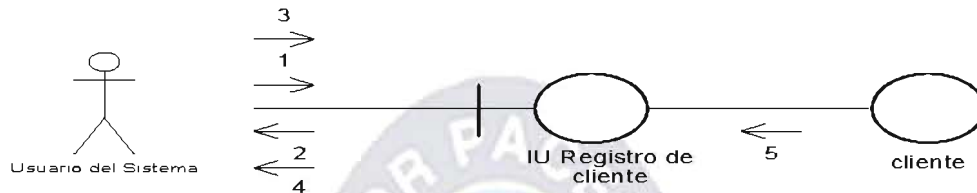
**Figura E.4.** Diagrama de Colaboración del caso de uso: Registrar transporte

**Fuente:** [Elaboración Propia]

### Caso de uso: Registrar Clientes

- Clase de entidad: Clientes
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de Clientes

1. Seleccionar el enlace correspondiente
2. Muestra el formulario
3. Llenado de formulario
4. Muestra la información que se desea guardar
5. Cliente guardado

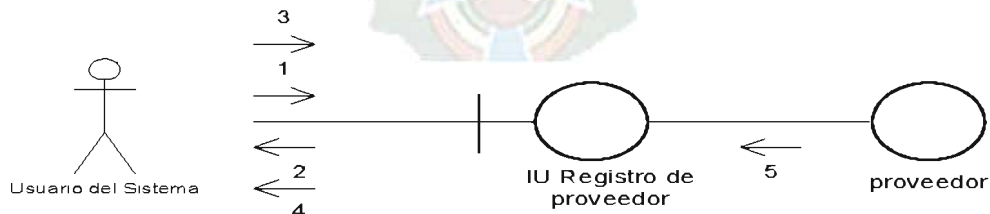


**Figura E.5.** Diagrama de Colaboración del caso de uso: Registrar Clientes **Fuente:**  
[Elaboración Propia]

### Caso de uso: Registrar Proveedor

- Clase de entidad: Proveedor
- Clase de interfaz central: Interfaz de usuario del sistema
- Clase Interfaz primitiva: Interfaz de usuario de registro de Proveedor.

1. Seleccionar el enlace correspondiente
2. Muestra el formulario
3. Llenado de formulario
4. Muestra la información que se desea guardar
5. Proveedor guardado



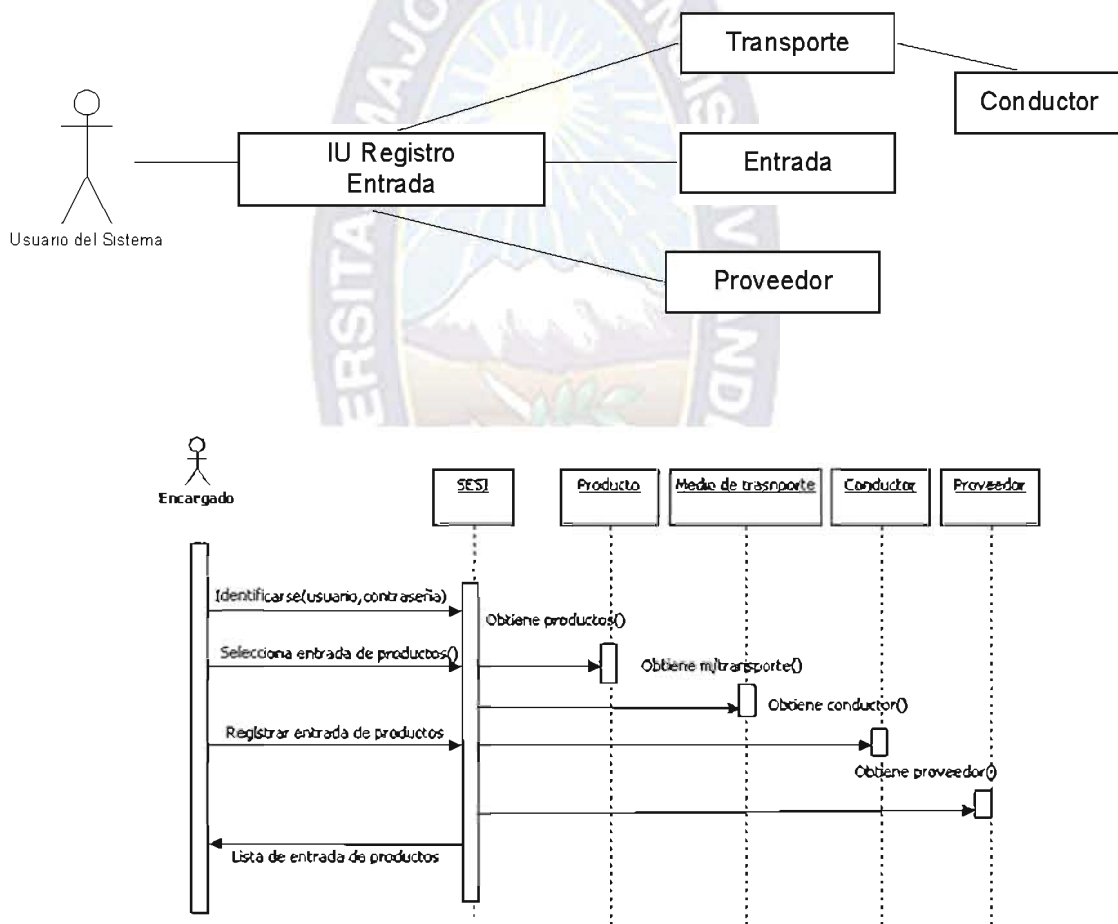
**FiguraE.6.** Diagrama de Colaboración del caso de uso: Registrar Proveedor **Fuente:**  
[Elaboración Propia]

## ANEXO F

### Diseño de casos de uso

Se van describir eventos de los casos de usos, los actores que interactúan directamente con el sistema.

*Caso de uso:* Registro de entradas de productos, donde el sistema interactúa con el actor que es el encargado de registrar las entradas.

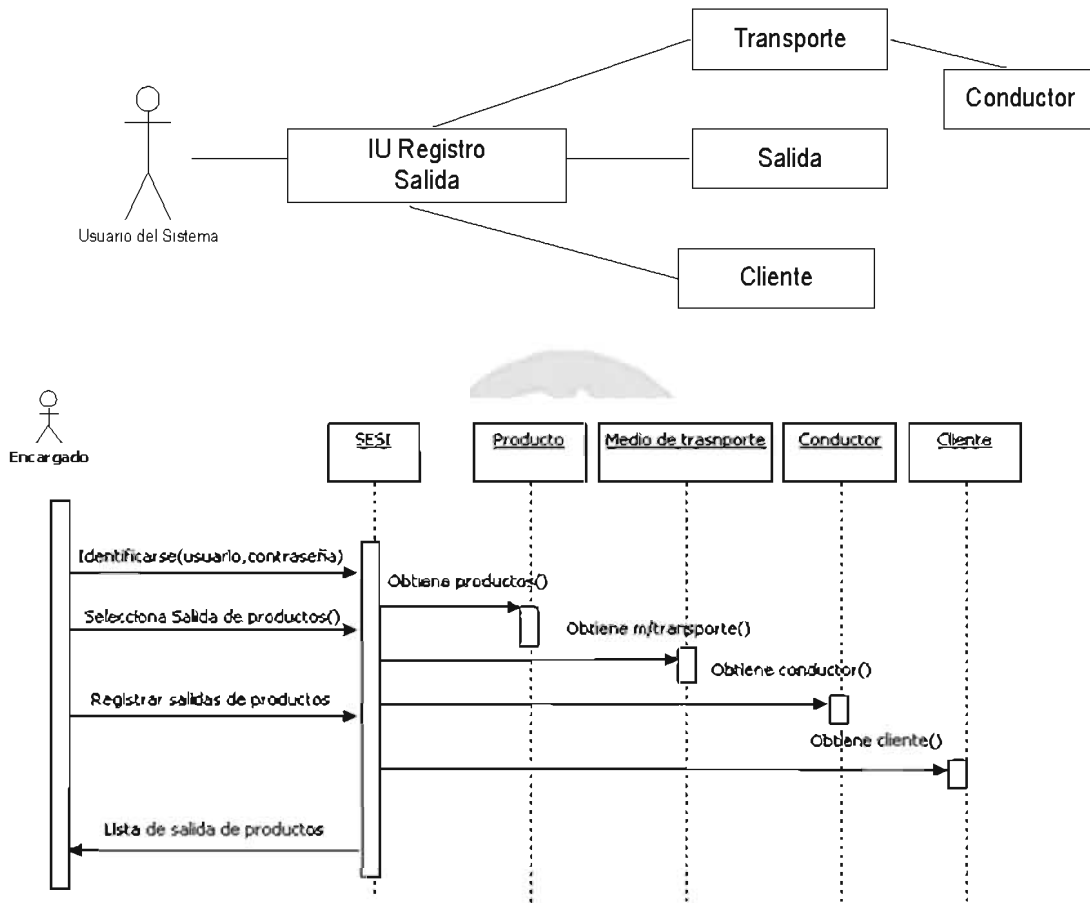


**Figura F.1.** Diagrama de secuencia: Registro de entradas de productos

**Fuente:** [Elaboración Propia]

*Caso de uso:* Registro de salidas de productos, donde el sistema interactúa con el actor que es el encargado de registrar las salidas.



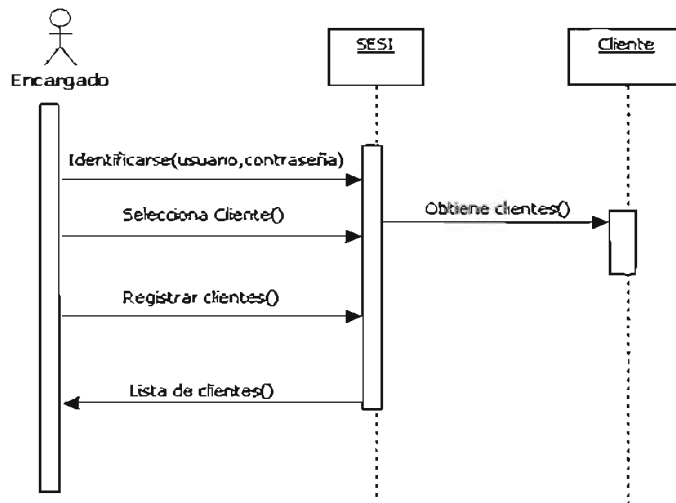


**Figura F.2.** Diagrama de secuencia: Registro de salidas de productos

**Fuente:** [Elaboración Propia]

*Caso de uso:* Registro de clientes donde el actor hace el respectivo registro.

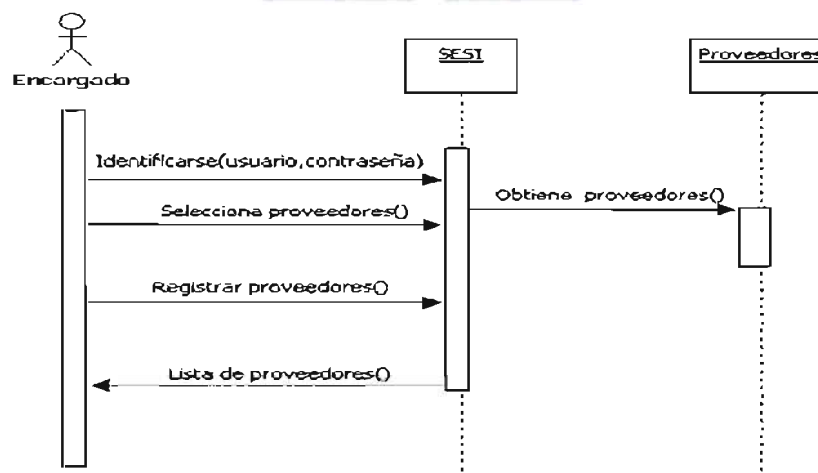




**Figura F.3.** Diagrama de secuencia: Registro de clientes

**Fuente:** [Elaboración Propia]

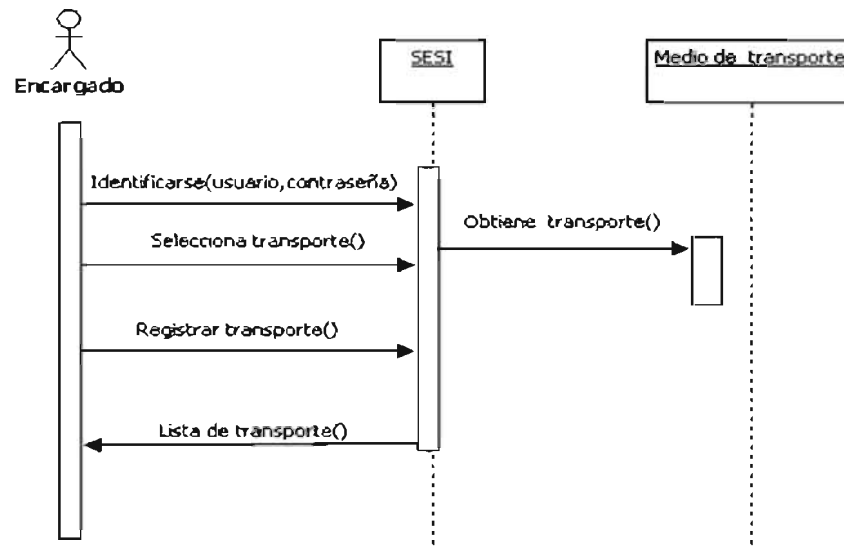
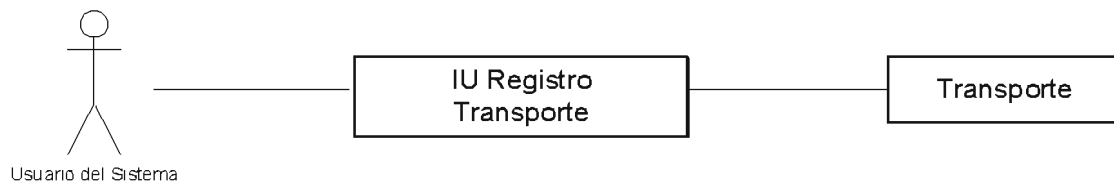
*Caso de uso:* Registro de proveedores



**Figura F.4.** Diagrama de secuencia: Registro de proveedores

**Fuente:** [Elaboración Propia]

### Caso de uso: Registro de medios de transporte

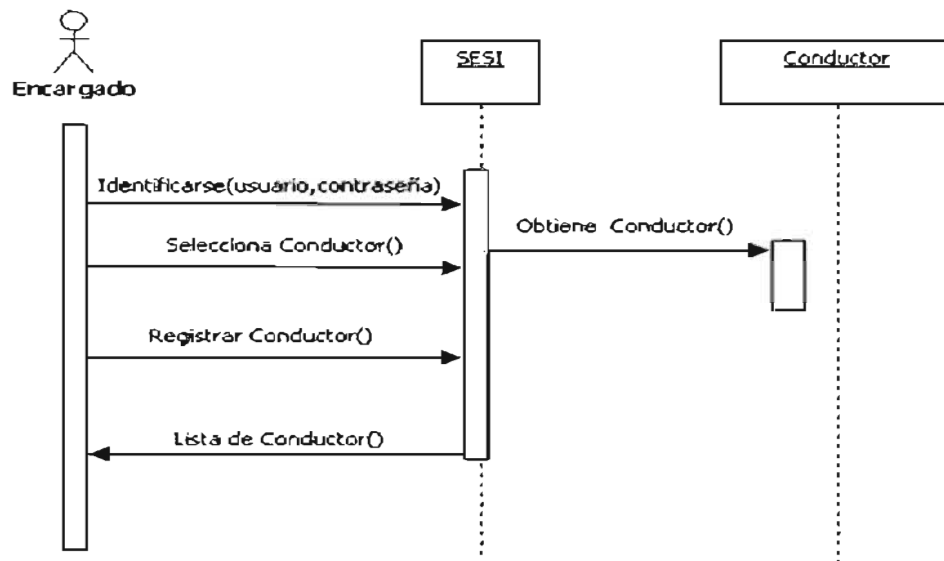


**Figura F.5.** Diagrama de secuencia: Registro de medios de transporte

**Fuente:** [Elaboración Propia]

### Caso de uso: Registro de conductor

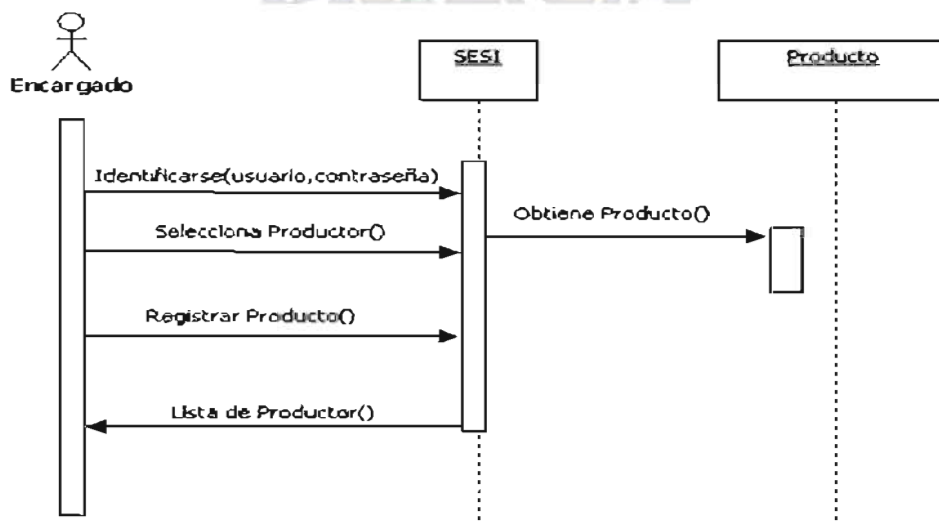




**Figura F.6.** Diagrama de secuencia: Registro de conductor

**Fuente:** [Elaboración Propia]

*Caso de uso:* Registro de producto



**Figura F.7.** Diagrama de secuencia: Registro de producto

**Fuente:** [Elaboración Propia]

## ANEXO G

### Prototipo de interfaz de usuario

**Figura G.1.**  
bienvenida



Interfaz de  
Sistema SESI

**Fuente:**  
[Elaboración  
Propia]

**Figura G.2.** Menú principal  
**Fuente:**



del sistema  
[Elaboración Propia]

CLIENTE

[Adicionar](#) [Modificar](#) [Eliminar](#)

CODIGO:	<input type="text"/>
NOMBRE:	<input type="text"/>
DIRECCION:	<input type="text"/>
TELEFONO:	<input type="text"/>
CIUDAD:	<input type="text"/>
PAIS:	<input type="text"/>
<input type="button" value="INSERTAR REGISTRO"/>	

**Figura G.3.** Formulario para registra clientes  
**Fuente:** [Elaboración Propia]

**FiguraG.4.**  
modificar datos  
**Fuente:**

CLIENTE

[Adicionar](#) [Modificar](#) [Eliminar](#) [Listar](#) [Listar](#)

CODIGO:	CL0002
NOMBRE:	PRO. BOL
DIRECCION:	VIA RORTE CAVALLEGGERI 30
TELEFONO:	12-454-4545
CIUDAD:	LIVORNO
PAIS:	ITALIA
<input type="button" value="ACTUALIZAR REGISTRO"/>	

Formulario para  
de clientes  
[Elaboración

Propia]

**Figura G.5.**

CLIENTE

[Adicionar](#) [Modificar](#) [Eliminar](#) [Listar](#) [Reporte](#)

CÓDIGO	NOMBRE	DIRECCION	TELEFONO	CIUDAD	PAIS	
CLI0001	RED RIBER FOODS	P.O. BOX 35612	23543	RICHMOND	ESTADOS UNIDOS	<a href="#">ELIMINAR</a>
CLI0002	PRO. BOL	VIA RORTE CAVALLEGGERI 30	0	LIVORNO	ITALIA	<a href="#">ELIMINAR</a>
CLI0005	MARCOA MANEDECZ		22518020	LA PAZ	BOLIVIA	<a href="#">ELIMINAR</a>
CLI0003	RIBER FOND	P.O. BOX 35612	23543	RICHMOND	ESTADOS UNIDOS	<a href="#">ELIMINAR</a>

Formulario

pare eliminar clientes

**Fuente:** [Elaboración Propia]

CONDUCTOR

[Adicionar](#) [Modificar](#) [Eliminar](#) [Listar](#) [Reporte](#)

CODIGO	NOMBRE	DIRECCION	TELEFONO
CON0001	5479005 LP	FREDDY EQUISE K.	5479005 LP
CON0003	5479005	DANIEL YUCRA	2264458

**Figura G.6.** Formulario de referencia de conductor

**Fuente:** [Elaboración Propia]



ENTRADA DE DATOS									
CODIGO_ENTRADA: EN0001					NROCONOC: 123				
NROTRAMITE_CFO: 123456					NOM_PROVEEDOR: ALMENDRAS BOLITAL S.R.L.				
CIUDAD_PROVEEDOR: FILADELFIA					NOM_CONDUCTOR: DANIEL YUCRA				
NRO_LICENCIA: 5479005 LP					NRO_PLACA: 1706-EKA				
FECHA_ENTRADA 2008-09-01					TIPO_ENTRADA ENTRADA DE PRODUCTOS				
<a href="#">LISTADO</a> <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>									

PRODUCTOS ADICIONADOS AL LOTE									
ID	ACCION	CODIGO	PRODUCTO	CATEGORIA	UNIDAD	DESCRIPCION	CANTIDAD	COSTO	IMPORTE
61	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	PR0001	MEDIUN	CONVENCIONAL	CAJAS		200	1	200
62	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	PR0002	BROKEN	CONVENCIONAL	CAJAS		120	1	120
63	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	PR0003	TINY	CONVENCIONAL	CAJAS		52	1	52

Figura G.7. Formulario para el registro de entrada de datos

Fuente: [Elaboración Propia]

CODIGO:	EN0001
NROCONOC:	123
NROTRAMITE CFO:	123456
PROVEEDOR:	Almendras Bolital S.R.L. ▼
CIUDAD:	FILADELFIA ▼
CONDUCTOR:	DANIEL YUCRA ▼
NRO LICENCIA:	5479005 LP ▼
NRO PLACA:	1706-EKA ▼
FECHA:	2008-09-01
TIPO ENTRADA:	Entrada de productos ▼
<a href="#">LISTADO</a> <a href="#">GUARDAR DATOS</a>	

Figura G.8. Formulario para el registro de entrada de productos

Fuente: [Elaboración Propia]

ENTRADA DE DATOS									
CODIGO_ENTRADA: EN0004					NROCONOC: 13123				
NROTRAMITE_CFO: 45436					NOM_PROVEEDOR: ALMENDRAS BOLITAL				
CIUDAD_PROVEEDOR: FILADELFIA					NOM_CONDUCTOR: DANIEL YUCRA				
NRO_LICENCIA: 5479005 LP					NRO_PLACA: 1474-KR				
FECHA_ENTRADA 2010-01-05					TIPO_ENTRADA ENTRADA DE PRODUCTOS				
<a href="#">LISTADO</a> <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>									

PRODUCTOS ADICIONADOS AL LOTE									
ID	ACCION	CODIGO	PRODUCTO	CATEGORIA	UNIDAD	DESCRIPCION	CANTIDAD	COSTO	IMPORTE
66	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	PR0001	MEDIUN	CONVENCIONAL	CAJAS		2142	1	2142

LISTA DE PRODUCTOS DISPONIBLES PARA ADICIONAR							
ACCION	CODIGO	PRODUCTO	CATEGORIA	SUB CATEGORIA	UNIDAD	DESCRIPCION	CANT TOTAL
<a href="#">ADICIONAR</a>	PR0002	BROKEN	CONVENCIONAL	BROKEN	CAJAS		0
<a href="#">ADICIONAR</a>	PR0003	TINY	CONVENCIONAL	TINY	CAJAS		100

Figura G.9. Formulario de detalle de todas las entradas de productos

Fuente: [Elaboración Propia]



LISTA DE SALIDAS						
ACCION	CODIGO_SALIDA	NROREGISTRO_SALIDA	FECHA_SALIDA	TIPO_SALIDA	ID_CLIENTE	ID_PRODUCTO
<a href="#">DETALLES</a> <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	SA0001	45612	2010-01-11	VENTA	CLI0003	PR0003
<a href="#">DETALLES</a> <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	SA0002	33456	2010-01-22	VENTA	CLI0002	PR0002

Registros 1 a 2 de 2  
[Crear](#)

**Figura G.10.** Formulario de listado de las salidas de productos

Fuente: [Elaboración Propia]

**Figura G.11.**  
salida de

CODIGO:	SA0001
NRO REGISTRO:	45612
FECHA	2010-01-11
TIPO SALIDA	venta
CLIENTE	RIBER FOND
PRODUCTO	Red Riber Foods
NOMBRE UNIDAD	Pro. Bol
IMPORTE TOTAL	RIBER FOND
<a href="#">LISTADO</a>	<a href="#">GUARDAR DATOS</a>

Formulario de registro de  
productos

Fuente: [Elaboración Propia]

ID_PRODUCTO	---	NOMBRE_CATEGORIA	---	FILTRAR	EXPORTAR PDF		
DETALLE DEL KARDEX							
FECHA_REG	DETALLE	NROLOTE_PRODUCTO	ENTRADA	SALIDA	SALDO TEMP	STOCK	UNIDAD
2010-01-08 16:28:12	INGRESO	MEDIUN	200		200	2294	CAJAS
2010-01-08 16:28:29	INGRESO	BROKEN	120		120	0	CAJAS
2010-01-08 16:28:42	INGRESO	TINY	52		52	100	CAJAS
2010-01-08 16:31:43	EGRESO	MEDIUN		100	100	2294	CAJAS
2010-01-08 16:32:02	EGRESO	BROKEN		70	50	0	CAJAS
2010-01-08 16:35:28	INGRESO	MEDIUN	52		152	2294	CAJAS
2010-01-08 16:35:41	INGRESO	TINY	48		100	100	CAJAS
2010-01-08 16:36:30	EGRESO	BROKEN		50	0	0	CAJAS
2010-01-08 17:30:26	INGRESO	MEDIUN	2142		2294	2294	CAJAS

**Figura G.12.** Formulario de búsqueda de producto

Fuente: [Elaboración Propia]

ID_PRODUCTO	BROKEN	NOMBRE_CATEGORIA		FILTRAR	EXPORTAR PDF		
DETALLE DEL KARDEX							
FECHA_REG	DETALLE	NROLOTE_PRODUCTO	ENTRADA	SALIDA	SALDO TEMP	STOCK	UNIDAD
2010-01-08 16:28:29	INGRESO	BROKEN	120		120	0	CAJAS
2010-01-08 16:32:02	EGRESO	BROKEN		70	50	0	CAJAS
2010-01-08 16:36:30	EGRESO	BROKEN		50	0	0	CAJAS

Figura G.13. Formulario de búsqueda de producto por categoría

Fuente: [Elaboración Propia]

**Reporte de Entradas y Salidas**

DESDE	<input type="text"/>
HASTA	<input type="text"/>
<input type="button" value="ACEPTAR"/> <input type="button" value="LIMPIAR"/>	

Figura G.14. Formulario de búsquedas de entradas y salidas de productos

Fuente: [Elaboración Propia]