

INTRODUCCIÓN

El avance tecnológico alcanzado por la informática está haciendo de este valioso medio de automatización de la información, un factor imprescindible para el desarrollo de la tecnología en todos los campos.

Las tecnologías de la Información y la Comunicación (TIC) son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

Comercial el Carmen es una empresa dedicada a la distribución de productos y accesorios para damas en la ciudad de Tarija, la misma que pretende el liderazgo en este rubro. Por tal razón uno de sus propósitos es contar con tecnología de punta para brindar un mejor servicio y atención a sus Clientes.

En la actualidad el **Comercial el Carmen** no cuenta con un sistema automatizado de administración de la sección de Almacenes para su respectivo control.

Los problemas se dan porque la Sección de Almacén no cumple adecuadamente su rol en la organización, y uno de los motivos al cual va dirigido el proyecto es que el manejo de la información en la Sección de Almacén es ineficiente.

El desarrollo de este proyecto de administración de la sección de almacenes vía web contribuye al cumplimiento adecuado del rol de la sección de almacenes, específicamente el propósito del proyecto es mejorar su administración con el fin de Mejorar la Organización y Control de la Administración de la Sección de Almacenes del **Comercial el Carmen**.

Para poder lograr el propósito del proyecto desarrollé una plataforma Web. La Plataforma consiste en un sistema de administración completa de almacenes.

El Desarrollo del Sistema coadyuva a los empleados de la Sección de Almacén a una alta disponibilidad de la información y un acceso rápido y de manera muy confiable y segura.

Además para complementar el proyecto, se llevó adelante la Elaboración de Manuales y la Capacitación de los Encargados de la Sección de Almacenes sobre el uso y manejo de los componentes producidos por el proyecto, así también se llevó a cabo estrategias de socialización de los mismos, para que este sea conocido y usado por los propietarios y todo el personal.

I. Capítulo 1: El Proyecto

1.1. Presentación del Proyecto

1.1.1. Título del Proyecto

Mejoramiento de la Administración y Control de Almacenes de Productos del “Comercial el Carmen” – CONAL.

1.1.2. Carrera / Unidad

Ingeniería Informática

1.1.3. Facultad

Ciencias y Tecnología.

1.1.4. Duración del Proyecto

Ocho meses.

1.1.5. Área / Línea de Investigación Priorizado

Tecnologías de la Información y Comunicación/Desarrollo de Sistemas y Software.

1.1.6. Responsable del Proyecto

Carrera de Ingeniería Informática – Taller III.

1.1.7. Entidad Asociada (s)

Comercial el Carmen – Tarija.

1.2. Personal Vinculado al Proyecto

1.2.1. Director del Proyecto

Apellido Paterno: Portal	Apellido Materno: Rivera	Nombre: Noemí Ilsen	C.I.: 7148813 tja.
Carrera: Ingeniería Informática		Facultad: Ciencias y Tecnología	

Telf.: Domicilio: Pasaje Municipal entre Ballivián y Hernán Siles, # 0836.	Celular: 71867569	Correo electrónico: noemi_16_48@hotmail.com	Firma:
--------------------------------------------------------------------------------------------	---------------------------------	-------------------------------------------------------	----------------------------

Tabla 1. Director del Proyecto

1.2.2. Participantes del Equipo de Trabajo

Categoría	Nombres y Apellidos	Carrera/Profesión	C.I.	Firma
Director	Noemí I. Portal Rivera	Ingeniería Informática	7148813	
Analista				
Programador				
Ingeniero de Software				
Formador o Capacitador				
Asesor	Lic. Gustavo Succi Aguirre	Lic. en Ingeniería Informática		

Tabla 2. Participantes del Equipo de Trabajo

1.2.3. Equipo de Trabajo de: Empresas/Instituciones/Organizaciones Participantes/Cooperantes

Nombre: “Comercial el Carmen”			
Dirección: Calle D. Campos		Telef. Oficina:66-63204	
Nombre y Apellidos	Cargo	C.I.	Firma
Sr. Pedro Salluca	Propietario	5684521	

Tabla 3. Equipo de Trabajo

1.2.4. Actividades Previstas para los Integrantes del Equipo de Investigación

Responsable *	Actividades
Director	<p>Como Jefe de Proyecto:</p> <p>Organizar el equipo de trabajo.</p> <p>Planificar las actividades y control del cronograma del proyecto.</p> <p>Asignar y gestionar recursos y prioridades a los distintos componentes y actividades del proyecto.</p> <p>Mantener al equipo del proyecto enfocado en los objetivos.</p> <p>Realizar el seguimiento a cada etapa del proyecto.</p> <p>Supervisar el desarrollo del proyecto.</p> <p>Presentación final del Sistema.</p>
Analista	<p>Como Analista de Sistemas:</p> <p>Capturar la especificación y validación de requisitos interactuando con los usuarios mediante entrevistas.</p> <p>Elaborar el Análisis y Diseño del Sistema.</p> <p>Elaborar el Modelo de Datos (Base de Datos del Sistema).</p>
Programador	<p>Como Programador:</p> <p>Realizar la Programación del Sistema Informático.</p> <p>Construcción de prototipos.</p>
Ingeniero de Software	<p>Como Ingeniero de Software:</p> <p>Elaborar las pruebas funcionales del Sistema Informático.</p>

Formador o Capacitador	<p>Como Formador o Capacitador:</p> <p>Formar al personal en el uso de las TIC para el manejo del producto final.</p>
Asesor	<p>Asesoramiento en los aspectos tecnológicos para el desarrollo del proyecto.</p> <p>Asesoramiento en el uso de la Metodología RUP (Utilización UML).</p> <p>Evaluación del documento del proyecto.</p>

Tabla 4. Actividades Previstas para los Integrantes de Equipo de Investigación

1.2.4.1. Unidades de Gestión: Organigrama del Equipo del Proyecto

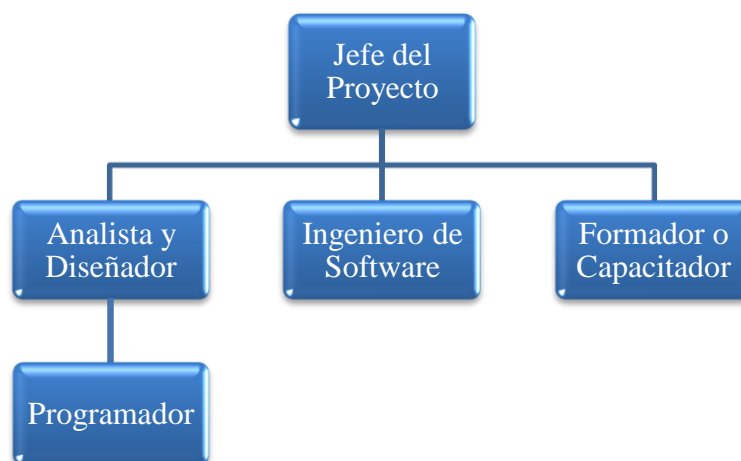


Figura 1. Organigrama del Equipo del Proyecto

1.3. Descripción del Proyecto

1.3.1. Resumen Ejecutivo del Proyecto

El avance tecnológico alcanzado por la informática está haciendo de este valioso medio de automatización de la información, un factor imprescindible para el desarrollo de la tecnología en todos los campos.

La rápida evolución de la tecnología dificulta la predicción de su rumbo, pero no deja de ser menos interesante y retadora. Cuando una organización cuenta dentro de su plataforma con un compromiso firme en el uso de la tecnología, busca maximizar su utilización y esto debe hacerse de manera rápida y decidida, para continuar al paso de la misma.

Las tecnologías de la Información y la Comunicación (TIC) son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

Comercial el Carmen es una empresa dedicada a la distribución de productos y accesorios para damas en la ciudad de Tarija, la misma que pretende el liderazgo en este rubro. Por tal razón uno de sus propósitos es contar con tecnología de punta para brindar un mejor servicio y atención a sus Clientes.

En la actualidad el **Comercial el Carmen** no cuenta con un sistema automatizado de administración de la sección de Almacenes para su respectivo control.

Los problemas se dan porque la Sección de Almacén no cumple adecuadamente su rol en la organización, y uno de los motivos al cual va dirigido el proyecto es que el manejo de la información en la Sección de Almacén es ineficiente.

El desarrollo de este proyecto de administración de la sección de almacenes vía web contribuirá al cumplimiento adecuado del rol de la sección de almacenes, específicamente el propósito del proyecto es mejorar su administración con el fin de

mejorar la organización y Control de la Administración de la Sección de Almacenes del **Comercial el Carmen**.

Para poder lograr el propósito del proyecto desarrollé una plataforma Web. La Plataforma consiste en un sistema de administración completa de almacenes.

El Desarrollo del Sistema coadyuvara a los empleados de la Sección de Almacén una alta disponibilidad de la información y un acceso rápido y de manera muy confiable y segura. Además para complementar el proyecto, se llevó adelante la Elaboración de Manuales y la Capacitación de los Encargados de la Sección de Almacenes sobre el uso y manejo de los componentes producidos por el proyecto, así también se llevó a cabo estrategias de socialización de los mismos, para que este sea conocido y usado por los propietarios y todo el personal.

1.3.2. Descripción, Fundamentación y Justificación del Proyecto

El proyecto **CONAL** tiene como de sus objetivos mejorar la administración en los procesos de Empleados, Almacenes, etc. que se realizan internamente en el **Comercial el Carmen**, por la simple razón de que la mayoría de las empresas presentan pérdida de datos al efectuar el registro de estas operaciones, por lo que se accede a una información errada de los movimientos dentro de las mismas.

El proyecto ofrece mayor rapidez en la toma de decisiones acerca de qué productos necesitan ser solicitados con mayor urgencia, ya que se cuenta con un almacén actualizado en todo momento, ya que actualmente existen demoras al tomar éste tipo de decisiones, esto implica no contar a tiempo con la cantidad de productos requeridos por los Clientes en determinado momento, ocasionando incomodidad en los mismos.

Sistema Web que permita la atención rápida y confiable a los Clientes de **Comercial el Carmen** permitiendo seguridad en la gestión de sus procesos administrativos, ya que el mismo admite demora en la atención a los Clientes debido a los procesos manuales que se efectúan a algunos fallos de registros eventuales. Las dificultades y necesidades hasta aquí mencionadas no se han personalizado por una razón bien simple: la mayoría de las

empresas se encuentran, independientemente de su tamaño, con los mismos problemas. Mientras que las grandes compañías disponen de instrumentos muy refinados para su solución, el resto de las empresas descansan en la responsabilidad de un equipo dirigente unipersonal o muy reducido, cuyas energías no permiten atender eficazmente todas las áreas de la empresa ni pueden encontrar colaboradores rentables.

Por ello es que **CONAL** pretende apoyar a la empresa **Comercial el Carmen** proveyéndole de ese medio que le hace falta para integrarse a un mundo competitivo del siglo XXI, para llevar adelante estrategias de nuevas formas de administrar sus procesos cotidianos por medio del uso de las TIC.

Justificación:

En el Punto de **Vista Social**, es importante tratar el problema existente en la Organización al existir gran dificultad para dar a conocer la información existente debido a que muchas veces no está a disposición de los funcionarios de la organización o de los usuarios en el momento, muchas veces la información no está clasificada, además se teme por la seguridad de la información, de esta manera se ahorrará tiempo para usuarios cuando quieran realizar alguna consulta a la organización.

En el Punto de **Vista Económico**, se justifica económicamente, ya que los costos que tendrá la Organización al implementar el sistema automatizado propuesto son menores que los beneficios que ha de adquirir. Debido a la automatización de sus procesos, lo cual significa que la Organización contará con un fuerte respaldo de seguridad, fiabilidad y oportunidad de la información generada por la misma.

En el Punto de **Vista Académico**, el conocimiento adquirido a lo largo de la carrera se volcará a través del presente proyecto de grado en la busca metodológica y científica de soluciones a problemas que confronta la Sección de Almacén para poder acceder a la información actualizada cuando se requiera.

En el Punto de **Vista tecnológico**, la obtención de un instrumento especializado en la Administración de la Sección de Almacén del **Comercial el Carmen** conlleva un desafío al definir las herramientas de software y hardware más adecuadas que puedan implementar con gran facilidad en términos visuales. La Institución donde se va a utilizar este Sistema cuenta con recursos de Hardware, cuentan con equipos de computación, un servidor, una red, que va a llegar a cubrir las necesidades del usuario del Sistema.

En el Punto de **Vista operativo**, la Sección de Almacén cuenta con recursos humanos calificados para la operación del Sistema Computadorizado Integrado.

Para la Administración de la Sección de Almacén se cuenta con una interfaz de usuario sencillo, flexible y también necesitó de un tiempo mínimo de aprendizaje.

Elaboré un manual de usuario, diseño de interfaces amigables opciones para el usuario para realzar las tareas de mantenimiento del sistema.

1.3.2.1. Análisis de Causas del Problema

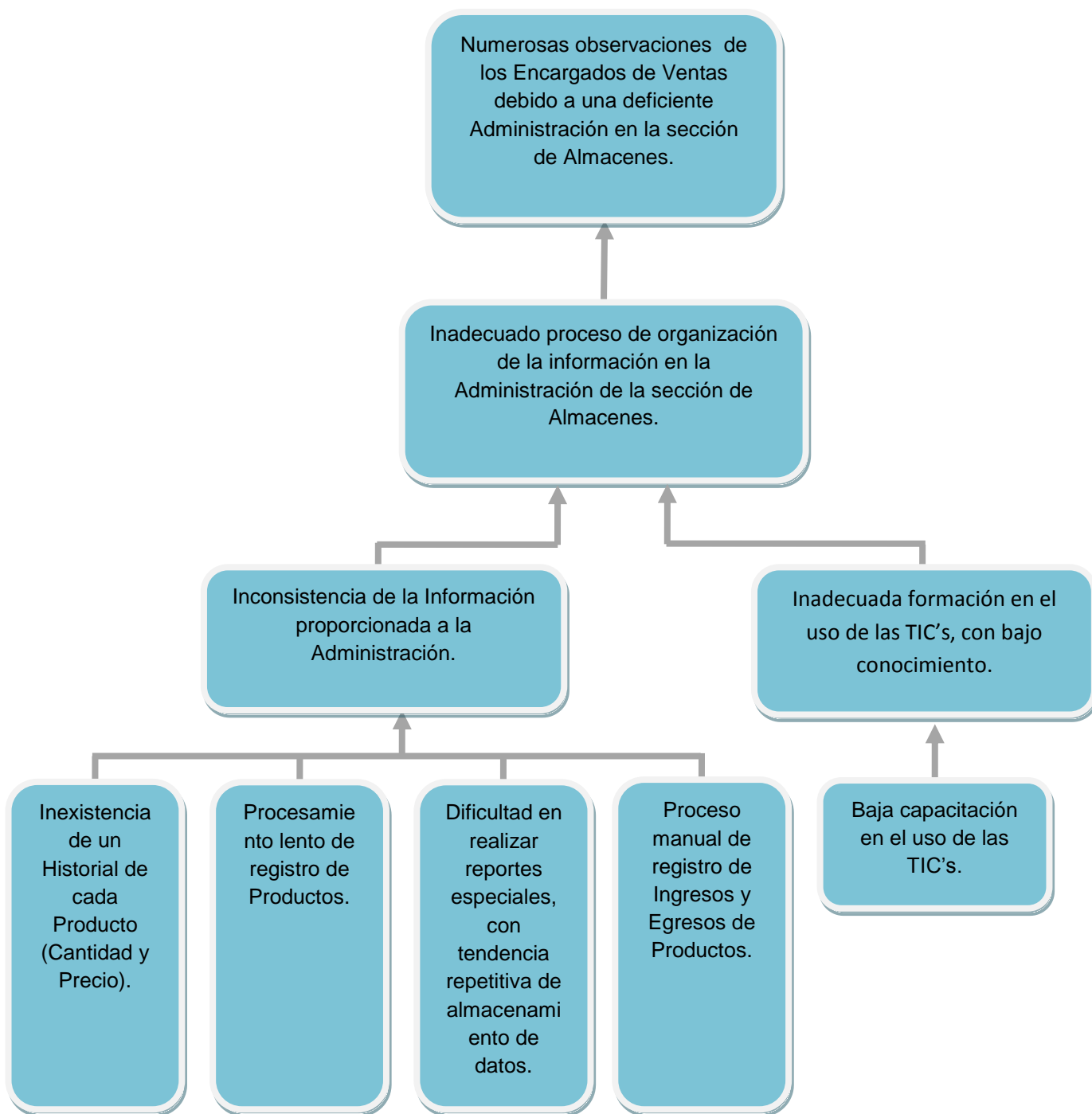


Figura 2. Árbol de Problemas

1.3.2.2. Análisis de Objetivos

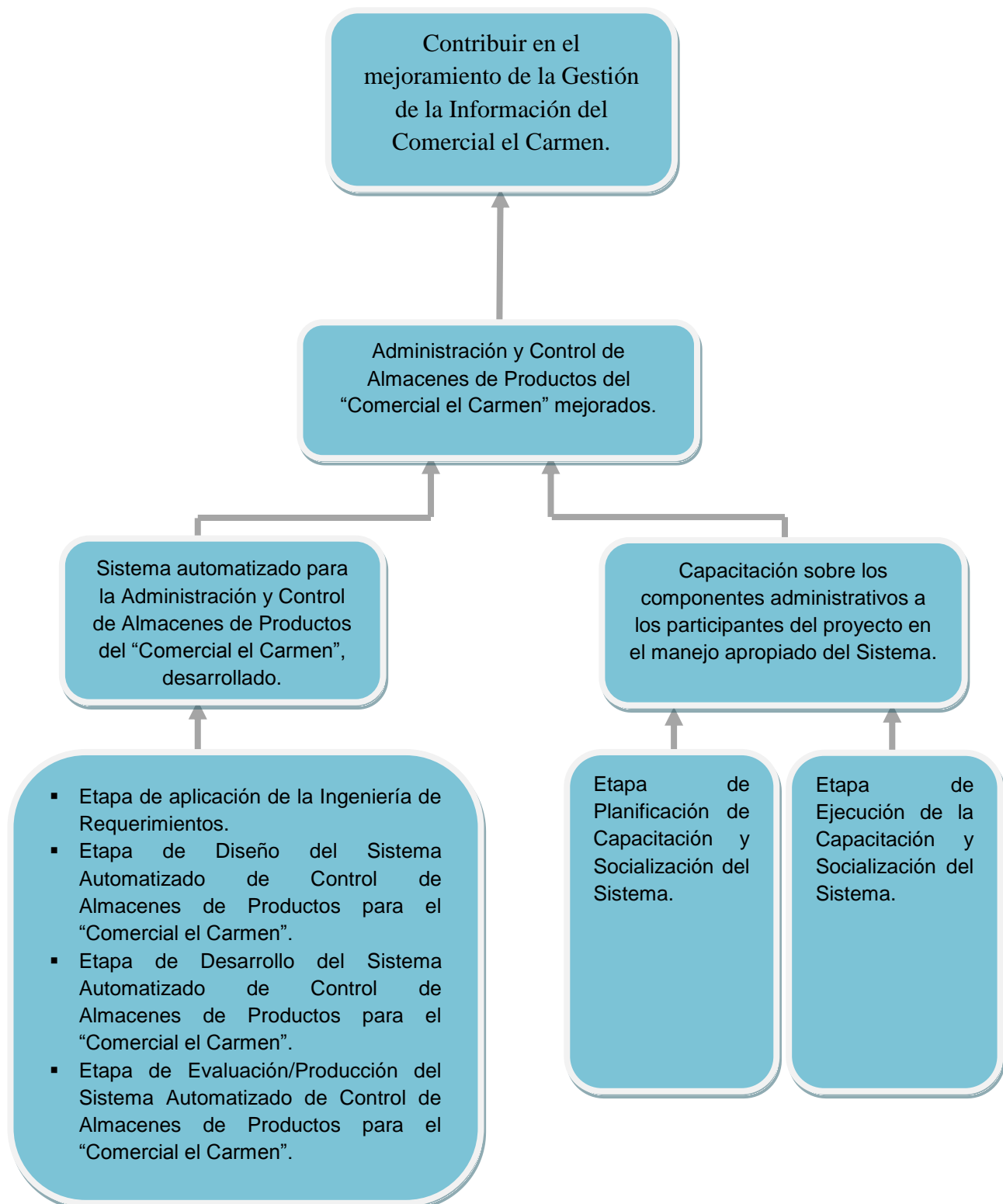


Figura 3. Árbol de Objetivos

1.3.3. Objetivos

1.3.3.1.Objetivo General

- Agilizar la Gestión de la información en la Sección de Almacén del Comercial el Carmen.

1.3.3.2.Objetivos Específicos

- Sistema de Control y Administración de Almacenes del “Comercial el Carmen” desarrollado.
- Capacitación sobre los componentes administrativos a los participantes del proyecto en el manejo apropiado del Sistema, ejecutada.
- Manual de Usuario, desarrollado.

1.3.4. Metodología

La forma de trabajo de este proyecto estará basada en:

1.3.4.1.Metodología para el Desarrollo de las Aplicaciones

La manera de estudio de este proyecto estará basada en una Metodología RUP (Rational Unified Process) en la que se procederá a cumplir con todas las fases que marca la metodología y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos cabe destacar que RUP que se trata de un proceso iterativo incremental.

Es importante destacar esto puesto que utilizaremos la terminología RUP en todo el proceso de desarrollo de software. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se proyectarán las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

Así también UML (Lenguaje de Modelado Unificado) que es una metodología de Ingeniería de Software basado en una notación gráfica la cual permite: especificar,

construir, visualizar y documentar los objetos de un Sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas.

El flujo de trabajo fundamental tiene los siguientes pasos:

- **Requerimientos:** Necesidades del negocio trasladadas a un Sistema automatizado.
- **Análisis y Diseño:** Requerimientos dentro de la arquitectura de software.
- **Programación e Implementación:** Software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** El comportamiento requerido es el correcto y que todo lo solicitado está presente.

Requerimientos: En base a las entrevistas se obtendrán las informaciones que reflejen las necesidades de los involucrados para la determinación de requerimientos.

Análisis y Diseño: En base a la especificación de requerimientos, se estructurará las diferentes vistas (Diagramas, Base de Datos y Pantallas) de la aplicación, tomando en cuenta metodologías de desarrollo de software.

Programación e Implementación: La programación será modular y orientada a objetos, se utilizarán tecnologías de punta, creando la aplicación informática que tenga el comportamiento deseado.

Pruebas y Validación: Antes de desarrollar las pruebas se procederá a la introducción de datos. Incluida esta información al Sistema se dará inicio a la fase de pruebas de desarrollo que serán mediante casos de prueba tomados de cada módulo y se realizarán los ajustes necesarios para una correcta validación.

Este proceso se torna repetitivo si se detectan inconsistencias en el Sistema implicando el retorno de cualquiera de las fases anteriores para su corrección.

Resumen de la Metodología para el Desarrollo

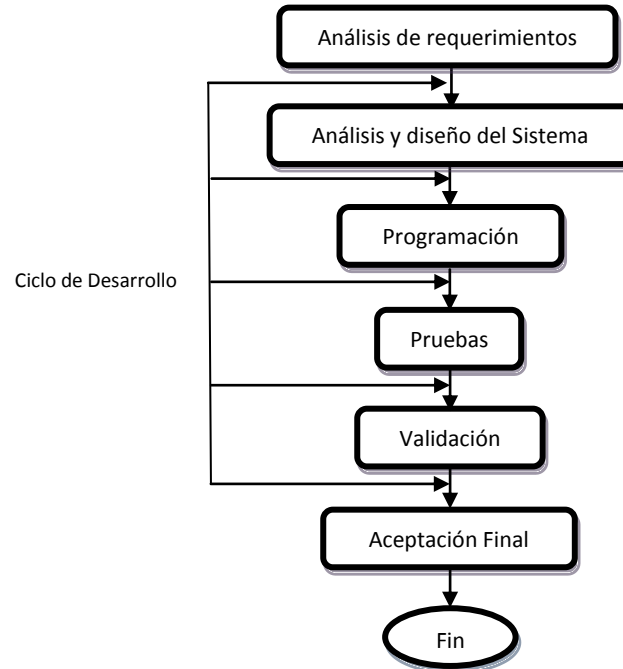


Figura 4. Resumen de la metodología para el Desarrollo

El propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de los usuarios finales.

1.3.4.2. Metodología para la Socialización y Capacitaciones

Para la Socialización y capacitación del personal existen cuestiones específicas que deben ser tomadas en consideración.

Para determinar la estructura y la metodología de la capacitación, existe una cantidad de cuestiones interrelacionadas que requiere solución como pueden ser:

- Cuál será la mejor estructura para el programa de capacitación y cuándo se deberá instrumentar. (Momento Adecuado para Impartir la Capacitación);
- Qué se necesitará para llevar a cabo una presentación exitosa. (Materiales de Referencia con Propósitos de Socialización);
- Qué temas deberán cubrir las sesiones de capacitación y cómo deberán organizarse (Estilo para Impartir la Capacitación);

- Cómo se medirán los resultados de la capacitación. (Evaluación del Conocimiento).

El proyecto plantea llevar a cabo para la Socialización una:

✓ **Metodología de Enseñanza Socializada**

El cuál pretende ejecutar una metodología de enseñanza socializada, dirigida a los gerentes con lo que se busca una integración social.

Para llevarlas a cabo se desarrollará el curso usando las siguientes técnicas:

✓ **Técnica expositiva**

Es una técnica explosiva centrada en el instructor, y consiste en proporcionar información al grupo, a través de la exposición oral, por parte del capacitador, en la que se estimulará la participación del personal en los trabajos que se realicen, este requiere una buena motivación para atraer la atención de los participantes.

✓ **Técnica de la experiencia**

La experiencia es un procedimiento eminentemente activo y que procura:

- Exponer de forma presencial los pasos a seguir para efectuar alguna acción Explicar el funcionamiento de algo mientras es observado.
- Comprobar, con razones lo que va a suceder, partiendo de experiencias.
- Conferir confianza para actuar en el terreno de la realidad de manera lógica.
- Convencer a cerca de la veracidad de la ley de causa y efecto.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

El proyecto plantea llevar a cabo para la Capacitación una:

✓ **Metodología de Enseñanza Centrada en el Estudiante**

Esta metodología permite que los estudiantes construyan e integren los nuevos conocimientos adquiridos.

Los profesores que practican la forma de enseñanza centrada en el estudiante tienen un rol primordial para lograr que los estudiantes integren los nuevos conocimientos, su rol consiste en:

a) El profesor como organizador: Analiza los datos que tiene a su disposición:

- Quiénes son sus estudiantes.
- Conocimientos previos (concepciones, creencias y aprendizajes adquiridos, etc.).
- Aprendizajes que se quieren alcanzar en la clase.

Luego, en función de estos antecedentes, selecciona el objetivo y, por lo tanto, el problema que tienen que resolver los alumnos.

Finalmente organiza el aspecto operacional definiendo:

- El material
- Los grupos
- El tiempo/plazo
- Las especificaciones de la tarea: El sentido y la finalidad de la actividad; las condiciones de realización (duración, procedimientos, ayudas posibles, formato del trabajo y criterios de evaluación, etc.); el ambiente espacial para propiciar el intercambio de ideas.

Todo este aspecto operacional se concibe de manera que se obtenga un intercambio de puntos entre los estudiantes.

b) El profesor como mediador

La mediación del profesor se produce en dos niveles:

- Entre el conocimiento y los estudiantes (ya que el profesor conoce el objetivo conceptual, dirigirá el aprendizaje socio cognitivo en función de dicho objetivo).
- Entre los mismos alumnos (el profesor actúa como moderador, da la palabra, administra el tiempo, etc.).
- A través de esta mediación, el profesor actúa en tres niveles:
- Sobre la percepción que tienen los estudiantes de la tarea encomendada (aporta precisiones, ordena la discusión, etc.).

1.3.4.3.Métodos para la Evaluación del Curso

Cuestionario pre curso: Para evaluar los conocimientos pre curso acerca de computación.

Guías de aprendizaje: Los/as participantes evalúan sus propias habilidades

Verificación de lo aprendido: El/la capacitador/a evalúa las habilidades de los/las participantes en el manejo del Sistema de gestión.

Evaluación del curso de capacitación: Los/as participantes evalúan el curso.

1.3.5. Resultados Esperados

El proyecto tendrá un impacto importante en el desempeño del “Comercial el Carmen” referente a la mayor flexibilidad de los procesos administrativos, el aumento de la calidad de los servicios y del trabajo de los empleados y en el incremento de la eficiencia en la entrega del producto a los clientes, ya que tendrán a su disposición una herramienta que les permita adaptarse a las nuevas técnicas para llevar a cabo el registro de Empleados, Salidas, Ingresos, Productos y otros procesos de gestión, para de ésta manera poder competir mejor en el mercado tarijeño.

El desarrollo de un Sistema Web con el uso de las últimas tecnologías estará destinado a la adopción de las TIC por parte de la empresa “Comercial el Carmen”, de forma individualizada, el sistema permitirá al propietario, crear usuarios del sistema, asignarles roles con restricciones de acceso, en donde podrá realizar la administración de Productos, empleados y sus especialidades.

Se llevará a cabo la planificación y la implementación de estrategias de socialización de los componentes del proyecto la cual estará dirigida a los propietarios, encargados de las diferentes áreas y la población tarijeña, transfiriendo de esta forma los beneficios otorgados por el proyecto a los propietarios.

Se realiza manuales de los procesos más importantes del Sistema que servirán de guía al usuario en la duración del curso y se evaluara el conocimiento adquirido en el curso en cuál se espera obtener la aprobación en la mayoría de los participantes del curso.

1.3.6. Transferencia de Resultados

1.3.6.1. Medios y Estrategias para la Transferencia de Resultados

Entregar los resultados por medio de una presentación a la Gerencia y a las Áreas de Ventas y Almacenes del Comercial el Carmen.

Los cursos de capacitación se darán a conocer a través de los memorandos a los beneficiarios del curso.

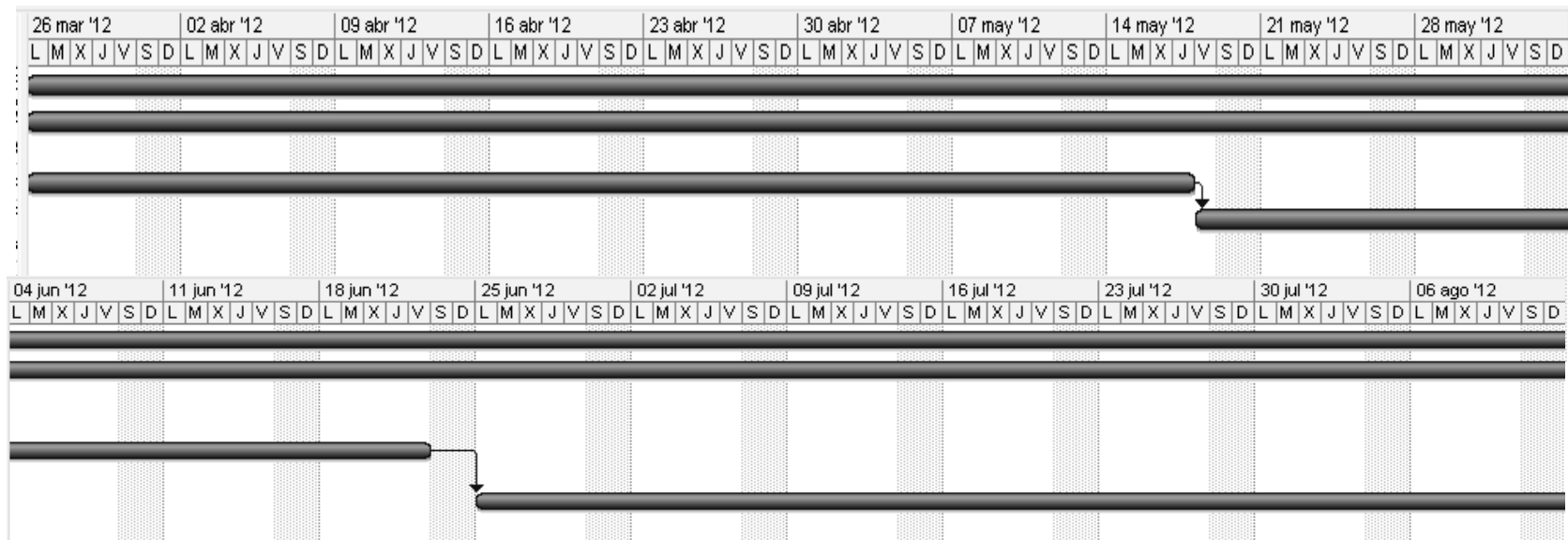
1.3.6.2. Grupo de Beneficiarios de los Resultados

Los beneficiarios del proyecto son los siguientes:

- Gerentes Propietarios de “Comercial el Carmen”.
- Encargados de las áreas de Ventas y Almacenes.
- Clientes en general.

1.3.7. Cronograma del Proyecto

	Nombre de tarea	Duración	Comienzo	Fin	Presupuesto	Predecesoras
1	DURACIÓN DEL PROYECTO	178 días	lun 26/03/12	mié 28/11/12	TOTAL 4150 Bs	
2	Sistema de Administración del Almacenes para el "Comercial el Carmen" desarrollado	164 días	lun 26/03/12	jue 08/11/12	3300 Bs	
3	Etapa de Aplicación de la Ingeniería de Requerimientos	39 días	lun 26/03/12	jue 17/05/12	700	
4	Etapa de Diseño del Sistema de Administración de Almacenes del Comercial el Carmen	26 días	vie 18/05/12	vie 22/06/12	450	3
5	Etapa de Desarrollo del Sistema de Administración de Almacenes del Comercial el Carmen	110 días	lun 25/06/12	vie 23/11/12	1750	4
6	Etapa de Evaluación/Producción del Sistema de Administración de Almacenes del Comercial el Carmen	19 días	lun 26/11/12	jue 20/12/12	400	5
7	Capacitación realizada al personal del Área de Almacenes y Ventas del Comercial el Carmen	9 días	vie 21/12/12	mié 02/01/13	850 Bs	6
8	Etapa de Planificación de Capacitación y Socialización del Sistema	3 días	lun 19/11/12	mié 21/11/12	150	2
9	Etapa de Ejecución de la Capacitación y Socialización del Sistema	8 días	jue 22/11/12	lun 03/12/12	600	8
10	Etapa de Elaboración del Informe Final	3 días	mar 04/12/12	jue 06/12/12	100	9



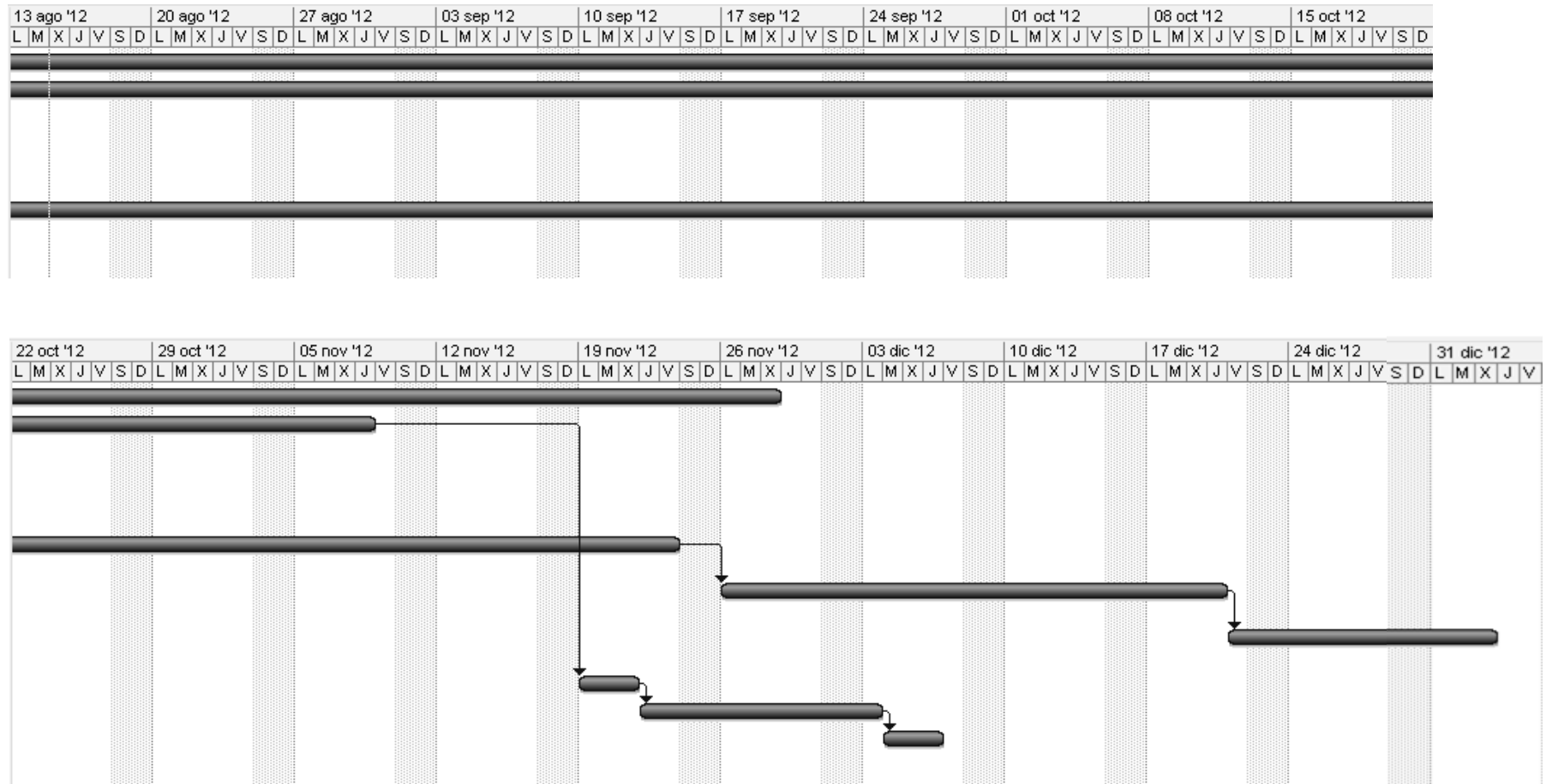


Figura 5. Cronograma de Actividades

1.3.8. Marco Lógico del Proyecto

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
<p><u>Fin</u></p> <p>Contribuir en el mejoramiento de la Gestión de la Información del Comercial el Carmen.</p>	<p>A dos años de finalizado el proyecto los empleados del Comercial el Carmen, expresan su conformidad con la Gestión de la Información.</p>	<p>Encuesta realizada al personal referente al grado de satisfacción con la nueva gestión administrativa de “Comercial el Carmen”.</p>	<p>Estabilidad política en la Administración del “Comercial el Carmen”.</p> <p>Monto presupuestado por el “Comercial el Carmen” para invertir en iniciativas y opciones nuevas.</p>
<p><u>Propósito</u></p> <p>“Administración y Control de los Almacenes de Productos del “Comercial el Carmen”,” mejorada.</p>	<p>Al concluir el proyecto, en la gestión 2013 la institución agiliza la capacidad de responder al pedido de información aumentado en</p>	<p>1. Cuadro comparativo referente al tiempo de obtención de la información de los almacenes, tomando en</p>	<p>a) La Administración de la sección de Almacenes Socializan el Sistema con el Personal.</p>

	un 45% en el control de almacenes respecto a los resultados del año anterior.	cuenta antes y después de la puesta en marcha del proyecto, avalado por el Jefe de Almacén.	b) La administración del sistema es efectiva. c) Voluntad por parte del Área de Administración para el mejoramiento del proceso de administración de información.
--	-------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

<u>Componentes</u>			
<p>1. Sistema automatizado para la Administración y Control de Almacenes de Productos del “Comercial el Carmen” desarrollado.</p>	<p>1.1. Al finalizar el proyecto, se ha desarrollado el Sistema de Administración de acuerdo a la ingeniería de requerimientos, para el Área de Almacenes del Comercial el Carmen.</p>	<p>1.1. Informe del Administrador del Sistemas de “Comercial el Carmen” con referencia al cumplimiento de los requerimientos establecidos al inicio del proyecto y expresados bajo la norma IEEE830 avalado por el Jefe de Almacén.</p>	<p>a) El “Comercial el Carmen” contribuye comprometida y eficazmente a la ejecución del proyecto.</p> <p>b) Los encargados de almacén no se oponen a la iniciativa de poner en operación el Sistema.</p> <p>c) Apoyo a la implementación del Sistema en las fechas sugeridas por el Ingeniero en Sistemas.</p>
	<p>1.2. A finales de Octubre se pone en funcionamiento el Sistema con al menos la</p>	<p>1.2. Informe de conformidad del Gerente del “Comercial el Carmen” con</p>	

<p>2. Capacitación sobre los componentes administrativos a los participantes del proyecto en el manejo apropiado del Sistema, implementada.</p>	<p>implementación de un 70% de los procesos administrativos que se realizan en el área de Almacenes.</p> <p>2.1. Al finalizar el proyecto a menos el 80% de los empleados de la sección de almacenes, ha realizado al menos un taller de capacitación en el uso del nuevo Sistema.</p> <p>2.2. Al finalizar el proyecto al menos el</p>	<p>respecto al Sistema Informático desarrollado.</p> <p>2.2.1. Informe de conformidad del Administrador del Sistema del “Comercial el Carmen”.</p> <p>2.2.2. Informe de conformidad del Gerente Propietario del “Comercial el Carmen”.</p>	<p>a) Se cuenta con recursos humanos suficientes en el área de almacenes.</p> <p>b) Los miembros involucrados en el proyecto reconocen la importancia de la capacitación.</p> <p>c) Usuarios (Personal del área de Almacenes) con disponibilidad e interés en capacitarse en el manejo del Sistema.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	90% de los empleados de la empresa participan en la socialización sobre la importancia del uso de las TIC en las empresas comerciales.	2.2.3. Lista de los participantes de los cursos de capacitación.	

<u>Actividades</u>			
<p>1.1. Etapa de aplicación de la Ingeniería de Requerimientos.</p> <p>1.1.1. Recopilación de la información sobre los requerimientos de la Sección de Almacén del “Comercial el Carmen”.</p> <p>1.1.2. Etapa de Planificación</p> <p>1.1.3. Elaboración del Sistema de Marco Lógico.</p> <p>1.1.4. Elaboración del Cronograma de Actividades.</p>	<p>(39 días)</p> <p>1.1.1. Fecha: 26 de Marzo-17 de Mayo.</p> <p>Presupuesto: 700 Bs.</p> <p>(26 días)</p>	<p>1.1.1. Documentos e Información recogidos.</p> <p>1.1.2. Documentación obtenida por la aplicación de la Ingeniería de Requerimientos.</p> <p>1.2.1. Documento de la</p>	<p>a) Los desembolsos se realizan de acuerdo al cronograma.</p> <p>b) Estimaciones presupuestarias estables, adecuadas al alcance del Proyecto.</p>

<p>1.2. Etapa de Diseño del Sistema automatizado de Control de Almacenes de Productos para el “Comercial el Carmen”.</p> <p>1.2.1. Diseñar el Modelo de Objetos del Sistema (Paradigma Orientado a Objetos).</p> <p>1.2.2. Diseñar la Base de Datos.</p> <p>1.2.3. Diseñar la interfaz gráfica del Sistema.</p>	<p>1.2.1. Fecha: 18 de Mayo-22 de Junio. Presupuesto: 450 Bs.</p> <p>(110 días)</p>	<p>Base de Datos del Sistema.</p> <p>1.2.2. Documento del Avance del Prototipo del Sistema.</p>	
<p>1.3. Etapa de Desarrollo del Sistema automatizado de Control de Almacenes de Productos para el “Comercial el Carmen”.</p>	<p>1.3.1. Fecha: 25 de Junio-23 de Noviembre. Presupuesto: 1750 Bs.</p>	<p>1.3.1. Sistema Desarrollado y Documentación del Sistema.</p>	

<p>1.3.1. Etapa de Programación del Sistema.</p>			
<p>1.3.2. Documentación del Código Fuente del Sistema.</p>			
<p>1.4. Etapa de Evaluación/Producción del Sistema automatizado de Control de Almacenes de Productos para el “Comercial el Carmen”.</p>	<p>(19 días)</p> <p>1.4.1. Fecha: 26 de Noviembre-20 de Diciembre. Presupuesto: 400 Bs.</p>	<p>1.4.1. Documentación obtenida del Manual del Usuario del Sistema.</p>	
<p>1.4.1. Elaboración de Manuales.</p>	<p>(3 días)</p>		
<p>2.1. Etapa de Planificación de Capacitación y Socialización del Sistema.</p>	<p>2.1.1. Fecha: 19 de Noviembre-21 de Noviembre. Presupuesto: 150 Bs.</p>		
<p>2.1.1. Diseñar la</p>			

<p>metodología de capacitación para los usuarios (Personal Almacén del “Comercial el Carmen”).</p> <p>2.2. Etapa de Ejecución de la Capacitación y Socialización del Sistema.</p> <p>2.3. Etapa de Elaboración del informe final.</p>	<p>(8 días)</p> <p>2.2.1. Fecha: 22 de Noviembre-03 de Diciembre.</p> <p>Presupuesto: 600 Bs.</p> <p>(3 días)</p> <p>2.3.1. Fecha: 04 de Diciembre-06 de Diciembre.</p> <p>Presupuesto: 100 Bs.</p>	<p>2.2.1. Documentación del Informe de la Capacitación del Director del Proyecto.</p> <p>2.3.1. Documentación del Informe final.</p>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------	--

Tabla 5. Marco Lógico del Proyecto

1.4. Presupuesto / Justificación

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
20000	SERVICIOS NO PERSONALES			
	21000. Servicios Básicos		2920	2920
	22000. Servicios de transporte		120	120
	25000. Servicios Profesionales y Comerciales		720	720
	Sub total rubro			3760
30000	MATERIALES Y SUMINISTROS			
	32000. Productos de Papel, Cartón e Impresos		580	580
	Sub total rubro			580
40000	ACTIVOS REALES			
	49000. Descripción del Gasto de Otros Activos		560 420	560 420

	Sub total rubro			980
	TOTAL			5320
	TOTAL + 40% Incentivo			

Tabla 6. Presupuesto / Justificación

GRUPO 20000. SERVICIOS NO PERSONALES

b) SUB GRUPO 21000. Descripción de los Gastos de Servicios Básicos

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación	270 Bs.	8	2160 Bs.
21200	Energía Eléctrica	60 Bs.	8	480 Bs.
21300	Agua			
21400	Servicios Telefónicos	35 Bs.	8	280 Bs.
Total				2920 Bs.

* Se refiere principalmente a los gastos por servicios; como: servicio de correo, radiogramas, servicio telefónico, fax, Internet.

c) SUB GRUPO 22000. Descripción de los Gastos de Viajes y Transporte de Personal

Partida	Personal	Lugar	Nº de viajes	Costo unitario*	Costo total
22100	Pasajes		40	2 Bs.	60 Bs.
Total					120 Bs.

* En el caso de pasajes debe indicarse el costo de ida y vuelta (costo unitario), indicando el número de viajes.

Partida	Personal	Lugar	Duración (días)	Costo unitario*	Costo total
22200	Viáticos				
22300	Fletes y Almacenamientos				
22600	Transporte de Personal				
Total					
Total sub grupo 22000					

* En el caso de los viáticos, debe considerarse la escala establecida por la UAJMS.

d) SUB GRUPO 23000. Descripción de los Gastos por Concepto de Alquileres de Equipos y Maquinarias

Partida	Alquiler de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
23100	Alquiler de Edificios			
23200	Alquiler de Equipos y Maquinaria			
23300	Alquiler de Tierras y Terrenos			
Total				

* Se refiere principalmente a los Gastos por el uso de Edificios y Equipos y Maquinaria en General

e) SUB GRUPO 24000. Descripción Mantenimiento y Reparación

Partida	Mantenimiento y reparación de	Costo	Tiempo	Costo total
----------------	--------------------------------------	--------------	---------------	--------------------

	equipo y maquinaria	unitario	mes	
24100	Mantenimiento y Reparación de Edificios y Equipos			
24300	Otros Gastos por Mantenimiento y Reparación			
Total				

* Se refiere principalmente a los gastos por el mantenimiento y reparación de edificios y equipos y maquinaria en general.

f) SUB GRUPO 25000. Descripción de los Gastos en Servicios Profesionales y Comerciales

Partida	Tipo de servicio profesional y comercial *	Cantidad	Costo unitario	Tiempo mes	Costo total
25200	Estudios e Investigaciones				
25500	Publicidad				
25600	Imprenta		120	8	120 Bs.
25700	Capacitación de Personal	3 personas.	200	1	600 Bs.
25800	Estudios e Investigaciones Para Proyectos de Inversión				
Total					720 Bs

* Se refiere a gastos por servicios profesionales de asesoramiento especializado, se incluyen, estudios, investigaciones, publicidad, imprenta, fotocopias, capacitación de personal y otros ejecutados por terceros.

GRUPO 30000. MATERIALES Y SUMINISTROS

g) SUB GRUPO 31000. Descripción de los Gastos Alimentos y Productos**Agroforestales**

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
31110	Refrigerios y Gastos Administrativos Capacitación: Bebida o Refrigerio Aperitivos			
31200	Alimento para Animales			
31300	Productos Agroforestales y Pecuarios			
Total				

* Se refiere a la adquisición de materiales y bienes como: alimentos y productos agroforestales, alimentos y bebidas para personas (indicar el total de refrigerios), alimentos para animales, productos pecuarios.

h) SUB GRUPO 32000. Descripción del Gasto de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio	580		580
32200	Productos de Artes Gráficas, Papel y Cartón			
32300	Libros y Revistas			

32400	Textos de Enseñanza			
32500	Periódicos			
Total				580 Bs

* Se refiere a la adquisición de papel y cartón en sus diversas formas y clases, impresos y publicaciones, periódicos, revistas, libros, fotocopias, etc.

i) SUB GRUPO 33000. Descripción del Gasto en Textiles y Vestuario

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
33100	Hilados y Telas			
33200	Confecciones Textiles			
33300	Prendas de Vestir			
33400	Calzados			
Total				

* Se refiere principalmente a los gastos por vestuario uniformes, ropa de trabajo

j) SUB GRUPO 34000. Combustibles, Productos Químicos, Farmacéuticos y Otros

Partida	Combustibles, Productos Químicos, Farmacéuticos y Otros	Cantidad	Costo/Unitario	Total
34110	Combustibles y Lubricantes para Consumo			
34200	Productos Químicos y Farmacéuticos			
34400	Productos de Cuero y Caucho			

34500	Productos de Minerales no Metálicos y Plásticos			
34600	Productos Metálicos			
34700	Minerales			
34800	Herramientas Menores			
Total				

* Se refiere a gastos de combustibles, químicos, productos farmacéuticos, llantas etc.

k) SUB GRUPO 39000. Descripción del Gasto en Productos Varios

Partida	Productos de cuero y caucho	Cantidad	Costo/Unitario	Total
39100	Material de Limpieza			
39400	Instrumental Menor Médico - Quirúrgico			
39500	Útiles de Escritorio y de Oficina			
	Agenda Personal	1	25	25
	Memorias Flash	2	150	300
	Bolígrafos, Lápices	6	3	18
	CD, DVD	20	3	60
	Tinta Negra	10	35	350
	Tinta a Colores	6	30	180
39700	Útiles y Materiales Eléctricos			
39800	Otros Repuestos y Accesorios			
Total				933

*Se refiere principalmente a los gastos por productos de limpieza, todo lo referente al funcionamiento de la oficina en material de escritorio.

GRUPO 40000. ACTIVOS REALES

l) SUB GRUPO 43000. Descripción del Gasto de Maquinaria y Equipo

Partida	Tipos de productos	Cantidad	Costo/Unitario	Total
43100	Equipo de Oficina y Muebles Impresoras			
43200	Maquinaria y Equipo de Producción			
43300	Equipos de Transporte, Tracción y Elevación			
43400	Equipo Médico y de Laboratorio			
43700	Otra Maquinaria y Equipo			
Total				

* Se refiere principalmente a los gastos por muebles y enseres, equipo de oficina, comunicación, equipamiento.

m) SUB GRUPO 46000. Descripción de Estudios y Proyectos para Inversión

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
46100	Para Construcción de Bienes de Dominio Privado			
Total				

* Se refiere principalmente a los gastos por servicios de terceros para la realización de investigaciones y otras actividades técnico – profesionales necesarias para la construcción y mejoramiento de bienes.

n) SUB GRUPO 49000. Descripción del Gasto de Otros Activos

Partida	Tipos de productos *	Cantidad	Costo/Unitario	Total
49100	Ext js	1	560	560
	Dhtmlx	1	420	420
Total				980

* Se refiere a los gastos en la compra de software, licencias.

1.4.1. Currículum Vitae

1.4.1.1. Antecedentes Personales

Portal	Rivera	Noemí Ilsen	7148813
Apellido Paterno	Apellido Materno	Nombre	C.I.
26-08-1987	M..... F...X....	Barrio 12 de Octubre Pasaje Municipal y Ballivián	
Fecha de nacimiento	Sexo	Dirección	
Tarija	-	71867569	noemi_16_48@hotmail.com
	Teléfono		

Ciudad	Domicilio	Celular	Correo electrónico
---------------	------------------	----------------	---------------------------

Tabla 7. Antecedentes Personales

1.4.1.2.Cursos Realizados

Curso	Institución	Duración en Hrs.	Año
Operador de Computadoras	Cruz Roja Boliviana – Filial Tarija	120	2005

Tabla 8. Cursos Realizados

II. Capítulo 2: Contexto

1.1. Componente 1: Sistema de Administración de Almacenes del Comercial el Carmen

1.1.1. Marco Teórico

1.1.1.1. Antecedentes

Desde el surgimiento de la teoría de la organización, la función esencial de la información en las organizaciones se ha acentuado. Una organización es un Sistema compuesto por tres elementos: Personas, materiales e información. Los sistemas de información, por su parte, surgen como sistemas complejos y abiertos que interactúan con otros sistemas y subsistemas como parte de su actuación. Por los años 90, una de las concepciones más defendidas por la gestión de la información fue que las organizaciones son sistemas de información.

El uso de ciertos conceptos tomados de la teoría de sistemas y del campo de la informática llevó a un alto grado de desarrollo entre los sistemas de información. Aunque existen diversas definiciones, hechas desde diferentes enfoques, sobre los sistemas de información, en su gran mayoría tienen puntos en común. El análisis realizado sobre las definiciones más frecuentes efectuadas en la década de los años 90 revela que constituyen un conjunto integrado de procesos, elementos o componentes que –según las estrategias y necesidades de una organización– recopilan, elaboran y distribuyen la información necesaria.

Un Sistema moderno de gestión de información exige la aplicación de nuevas tecnologías de información; sin embargo, la tecnología por sí sola no es suficiente para lograr una buena gestión de información. Son diversos los procesos que conforman los sistemas de gestión de información; ellos generan las entradas y salidas del Sistema o de otros procesos relacionados; también pueden identificarse, controlarse, corregirse o actualizarse en la medida en que se producen las transformaciones del entorno y evoluciona la organización, como vía incuestionable para garantizar su calidad, eficiencia y mejora continua.

Como resumen de este antecedente, puede decirse que los sistemas de gestión de información, en su definición más general, se refieren al conjunto de todos los

componentes necesarios que se interrelacionan, con el objetivo de tramitar y facilitar la información sobre el tema de interés para su consumo en cualquier medio, momento y lugar.

1.1.1.2. Estudio de la Variable Independiente: Gestión de la Información

1.1.1.2.1. Comercial el Carmen

Comercial el Carmen es una empresa dedicada a la distribución de productos y accesorios para damas en la ciudad de Tarija, la misma que pretende el liderazgo en este rubro. Por tal razón uno de sus propósitos es contar con tecnología de punta para brindar un mejor servicio y atención a sus Clientes.

La empresa hace prestación de sus servicios en la ciudad de Tarija provincia cercado del departamento de Tarija contando en la actualidad con una agencia en la ciudad de Yacuiba provincia Gran Chaco del mismo departamento, siendo su ámbito de acción dentro del departamento de Tarija.

1.1.1.2.2. Misión y Visión

La MISIÓN del Comercial el Carmen:

Somos una empresa vanguardista orientada en todo momento a lograr que nuestros productos cuenten con la elegancia, moda, tendencia y calidad que nuestros clientes más exigentes merecen tener.

La VISIÓN del Comercial el Carmen:

Ser una empresa reconocida, distinguida, renombrada y demandante, en el mundo comercial enfocado a la elegancia y vanguardia; gracias a la buena reputación y distinción adquirida por nuestros productos de gran calidad siempre en contacto con la tendencia y moda, logrando así enfrentar el mercado nacional.

1.1.1.2.3. Objetivo Institucional

Objetivo de la empresa:

Establecernos oficialmente como una empresa reconocida y competente a nivel local, ser una empresa en la cual nuestros clientes ya estén totalmente identificados con nuestras tiendas; tengan fe y gusto por nuestros productos adquiriendo favoritismo por éstos

1.1.1.2.4. Organigrama – Comercial el Carmen

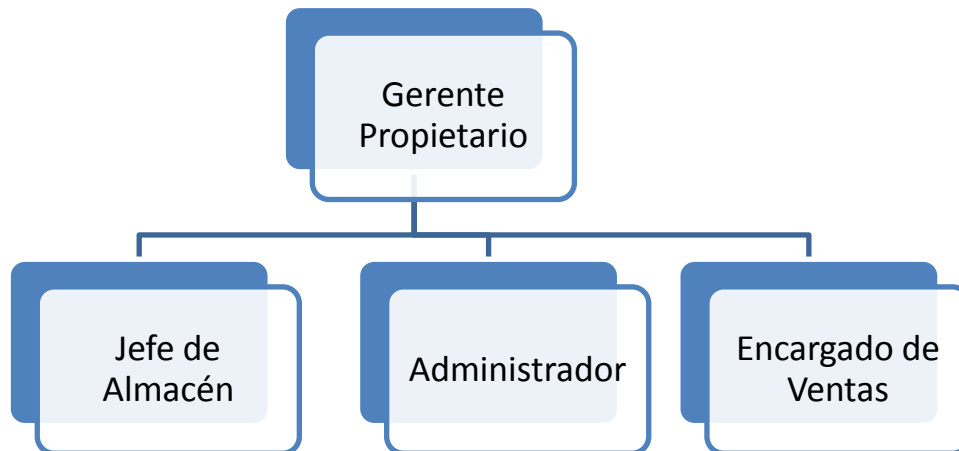


Figura 6. Organigrama Comercial el Carmen

1.1.1.3. Estudio de la Variable Dependiente: Sistema de Gestión de la Información

Actualmente la información manejada por el “Comercial el Carmen” sobre el Control de Almacenes son realizadas en forma manual lo cual produce un proceso lento y tedioso cuando se quiere obtener información sobre algún Producto o cuando se quiere saber las Entradas y Salidas de los mismos, así también para la realización de Reportes. Los problemas se dan porque el Área de Almacenes no cuenta con un software informático que le permita desarrollar su actividad en forma ágil y confiable en la administración de los Productos.

Es por eso que se propone como solución el desarrollo de un Sistema de información que ayudará al mejoramiento del Control y Administración de Almacenes.

1.1.1.4. Metodología de Desarrollo

1.1.1.4.1. Metodología RUP (Rational Unified Process)

a) Definición.- RUP es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. Es un proceso de Ingeniería de Software que captura las mejores prácticas del conocimiento de líderes en Ingeniería de Software y que provee un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo.

Los **procesos de RUP** estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos por RUP se enfocan fuertemente sobre arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme.

Nos permite elaborar un levantamiento exhaustivo de requerimientos.

Las actividades de RUP se centran en crear y mantener modelos, utilizando UML, en forma efectiva.

- Busca detectar defectos en las fases iniciales.
- Intenta reducir al número de cambios tanto como sea posible.
- Realiza el Análisis y Diseño, tan completo como sea posible.
- Diseño genérico, intenta anticiparse a futuras necesidades.
- Las necesidades de clientes no son fáciles de discernir.
- Existe un contrato prefijado con los clientes.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.

b) Características Esenciales.- Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: Está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

- **Está dirigido por los Casos de Uso:** Los Casos de Uso es 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.

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

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del Sistema. También guían su diseño, implementación y prueba.

- **Está Centrado en su 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.

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.

- **Es Iterativo e Incremental:** según 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. 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 efectúa una integración de los resultados con lo conseguido de las iteraciones anteriores.

c) Fases en el Ciclo de Desarrollo.- Este proceso de desarrollo considera que cualquier progreso de un Sistema software debe pasar por cuatro fases que se describirán a continuación, la figura muestra las fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica, donde se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

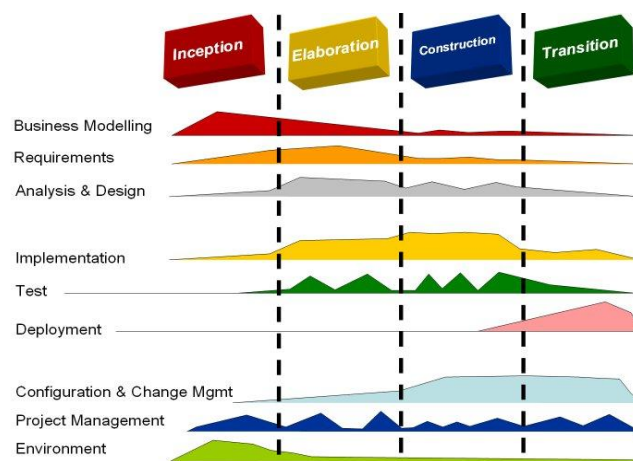


Figura 7. Fases en el Ciclo de Desarrollo de R.U.P.

- **Fase 1: Preparación Inicial (“Inicio”)**

Su objetivo primordial es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de definir el alcance del Sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

El caso de negocio incluye criterios de éxito, la evaluación de riesgos, y la estimación de los recursos necesarios, y un plan de la fase que muestren las fechas previstas e hitos importantes.

- **Fase 2: Preparación Detallada (“Elaboración”)**

Su finalidad principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del Sistema, acumulando la información necesaria para el plan de construcción y adquiriendo suficiente información para hacer realizable el caso del negocio.

El resultado de la fase de elaboración es:

Un modelo de Caso de Uso (por lo menos 80% completo) - todos los Casos de Uso y actores deben haber sido identificados-, y se han desarrollado la mayoría de las descripciones de Casos de Uso.

Requerimientos suplementarios que capturan los requerimientos no funcionales o cualquier requerimiento que no se asocie a un caso de uso específico.

- **Fase 3: Construcción (“Construcción”)**

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

- **Fase 4: Transición (“Transición”)**

Su objetivo principal es hacer la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

Éste incluye:

- Operación en paralelo con un Sistema anterior que el nuevo Sistema esté sustituyendo.
- La conversión de las bases de datos operacionales.
- Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.

1.1.1.4.2. UML (Lenguaje Unificado de Modelado)

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un Sistema.

Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

a) Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML posibilita expresar de una forma gráfica un Sistema de modo que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un Sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del Sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

b) Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
- Relaciones: Relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

c) UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y admite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

d) UML ofrece notación y semántica estándar:

UML prescribe una notación estándar y semánticas esenciales para el modelado de un Sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de las metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí.

UML cuenta con varios tipos de diagramas, los cuáles muestran diferentes aspectos de las entidades representadas.

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, es útil categorizarlos jerárquicamente.

Los *Diagramas de Estructura* enfatizan en los elementos que deben existir en el Sistema modelado:

- Diagrama de Clases, representan la estructura estática en términos de clases y relaciones.
- Diagrama de Componentes, representan los componentes físicos de una aplicación.
- Diagrama de Objetos, representan los objetos y sus relaciones, corresponden a diagramas de colaboración simplificados sin la representación del envío de mensajes.
- Diagrama de Estructura compuesta (UML 2.0)
- Diagrama de Despliegue, representan el despliegue de los componentes sobre los dispositivos físicos.
- Diagrama de Paquetes, muestra como un Sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

Los *Diagramas de Comportamiento* enfatizan en lo que debe suceder en el Sistema modelado:

- Diagrama de Actividades, representan el comportamiento del Sistema en términos de acciones.
- Diagrama de Casos de Uso, representan funcionalidad del Sistema desde el punto de vista del usuario.
- Diagrama de Estados, representan el comportamiento de una clase en término de estados.

Los *Diagramas de Interacción* son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del Sistema modelado:

- Diagrama de Secuencia, son una representación temporal de los objetos y sus interacciones.
- Diagrama de Comunicación, que es una versión simplificada del Diagrama de Colaboración (UML 1.x), son una representación espacial de los objetos, enlaces e interacciones entre ellos.

- Diagrama de Tiempos (UML 2.0).

Diagrama Global de Interacciones o Diagrama de Vista de Interacción (UML 2.0).

1.1.1.4.2.1. Tipos de Diagramas Utilizados

1.1.1.4.2.1.1. Diagramas de Clases

Es un tipo de diagrama estático que describe la estructura de un Sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el Sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

1.1.1.4.2.1.2. Diagramas de Casos de Uso

Que capturan las interacciones de los Casos de Uso y los actores. Describe los requisitos funcionales del Sistema, la forma en la que las cosas externas (actores) interactúan a través del límite del Sistema y la respuesta del Sistema.

1.1.1.4.2.1.3. Diagramas de Secuencias

Que es una representación estructurada del comportamiento como una serie de pasos secuenciales a lo largo del tiempo. Se usa para representar el flujo de trabajo, el paso de mensajes y cómo los elementos en general cooperan a lo largo del tiempo para lograr un resultado.

1.1.1.5. Herramientas de Construcción de Software

1.1.1.5.1. Eclipse Juno

Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.¹

Esta plataforma, típicamente ha sido usada para fomentar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados

¹ http://es.wikipedia.org/wiki/Eclipse_%28software%29

también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas.

Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

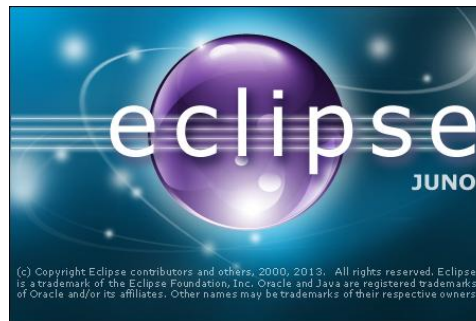


Figura 8. Eclipse Juno

1.1.1.5.2. Rational Rose

Rational Rose es una herramienta para “**modelado visual**”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida de desarrollo del Software.

Rational Rose permite completar una gran parte de las **disciplinas** (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- Modelado del negocio.
- Captura de requisitos (parcial).
- Análisis y Diseño (completo).
- Implementación (como ayuda).
- Control de Cambios y Gestión de Configuración (parte).

Las vistas de Rose son las siguientes:

- a) **La Vista de Casos de Uso, Use Case View**, que es la vista en la que se presenta el comportamiento deseado del sistema: en ella se encontrarían los modelos

relacionados con la captura de requisitos. Según el proceso que hemos visto en clase, en esta vista se ubicarían el modelo del negocio, el modelo conceptual y el modelo de casos de uso del sistema.

- b) **La Vista Lógica, Logical View**, en la que encontraríamos los modelos que muestran el vocabulario y la funcionalidad (estructura y comportamiento) del sistema, a través de un conjunto de colaboraciones que realizan los casos de uso de la vista de casos de uso (colaboraciones que se modelan mediante diagramas de clases y diagramas de interacción: secuencia y colaboración).
- c) **La Vista de Componentes, Component View**, en la que se representa la implementación del sistema mediante componentes, la organización modular del Software. Esta vista está relacionada con la gestión de la configuración del software. Los paquetes en esta vista se organizan en niveles.
Un componente está relacionado con un archivo de software y un lenguaje de programación. Las clases de la vista lógica se asignarían a los componentes de la vista de componentes.
- d) **La Vista de Despliegue, Deployment View**, en la que se modela la distribución o despliegue de los componentes a los nodos de procesamiento del sistema. Muestra la topología, distribución e instalación del sistema.

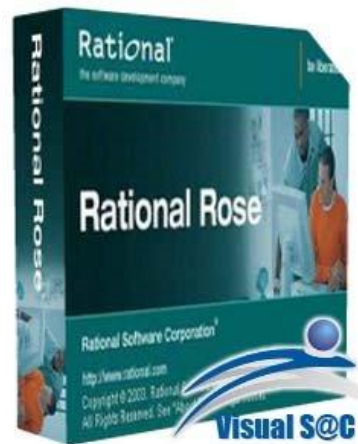


Figura 9. Rational Rose

1.1.1.5.3. Tomcat

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software

Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.²

Podemos dividir los contenedores de Servlets en:

Contenedores de Servlets Stand-alone (Independientes)

Estos son una parte integral del servidor web. Este es el caso cuando usando un servidor web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer (actualmente sustituido por iPlanet). Éste el modo por defecto usado por Tomcat.

Sin embargo, la mayoría de los servidores, no están basados en Java, lo que nos lleva a los dos siguientes tipos de contenedores:

Contenedores de Servales dentro-de-Proceso

El contenedor Servales es una combinación de un plumín para el servidor web y una implementación de contenedor Java. El plumín del servidor web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un serles, el plumín toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad.

Contenedores de Servlets fuera-de-proceso

El contenedor Servlets es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un servlets, el plugin toma el control sobre la petición y lo pasa al contenedor Java. El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente se soportan los

² <http://es.wikipedia.org/wiki/Tomcat>

servidores Apache, IIS y Netscape). Esto significa que siempre que desplaguemos Tomcat tendremos que decidir cómo usarlo, y, si seleccionamos las opciones 2 ó 3, también necesitaremos instalar un adaptador de servidor web.

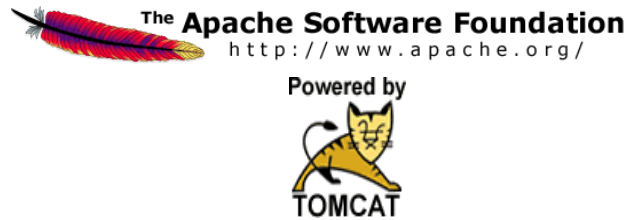


Figura 10. Tomcat

1.1.1.5.4. pgAdmin III PostgreSQL Tools

PgAdmin es un proyecto de software libre publicado bajo la licencia de PostgreSQL. El software está disponible en fuente y el formato binario de la red de servidores espejos de PostgreSQL.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo.

pgAdmin 3 es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (EnterpriseDB Postgres Plus Advanced Server y Greenplum Database).³ Incluye:

- Interfaz administrativa gráfica
- Herramienta de consulta [SQL](#) (con un EXPLAIN gráfico)
- Editor de código procedural
- Agente de planificación [SQL](#)/shell/batch
- Administración de Slony-I

pgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas [SQL](#) hasta desarrollar bases de datos complejas.

³ <http://www.arpug.com.ar/trac/wiki/PgAdmin>

La interface gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris.

pgAdmin III soporta versiones de servidores 7.3 y superiores. Versiones anteriores a 7.3 deben usar pgAdmin II.

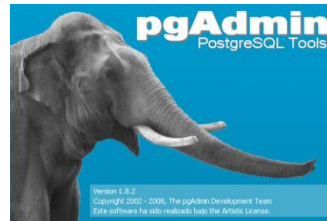


Figura 11. PgAdmin III PostgreSQL Tools

1.1.1.5.5. iReport

iReport es una herramienta visual que sirve para generar ficheros XML (plantillas de informe) que se puedan utilizar con la herramienta de generación de informes JasperReports.

Jasper Reports es una herramienta open source para la generación de reportes basado en la tecnología JAVA, con capacidad de EXPORTACIÓN en formato PDF, XML, HTML, CSV y XLS.⁴



Figura 12. iReport

1.1.1.6. Técnica

1.1.1.6.1. HTML (*HyperText Markup Language*)

⁴ <http://www.buenastareas.com/ensayos/Configuracion/1272199.html>

HTML (Lenguaje de Marcas de Hipertexto) es un lenguaje estático para el desarrollo de sitios web (Lenguaje de Marcas Hipertextuales). Es un lenguaje de fácil aprendizaje siendo el despliegue del mismo rápido. Así, el HTML es una aplicación del estándar ISO 8879:1986(SGML), que se formalizó en 1990 con la aparición de la Word Wide Web.

1.1.1.6.2. XML (*Markup Language*)

Es un metalenguaje para la descripción y estructuración de datos utilizando marcas (Markup Language). Metalenguaje significa un lenguaje para definir otros lenguajes: XHTML, WML, etc.

El Objetivo de XML es separar de un documento o información.

Facilidad de administración (separar contenido, lógica y presentación).

Muchas aplicaciones Web que extraen información de BD, la convierten en tablas, perdiendo información de los campos.

1.1.1.6.3. CCS (*Cascading Style Sheets*)

Una hoja de estilo es un conjunto de instrucciones que definen la apariencia de diversos elementos de un documento HTML. En otras palabras una hoja de estilo nos permite indicar por ejemplo el tamaño de la fuente, color y estilo de cierto párrafo que nosotros indiquemos, mediante la definición de estilos y aplicación de los mismos.

Las hojas de estilo se usan porque tienen muchas ventajas sobre los tags tradicionales, ya que por ejemplo es posible crear una sola hoja de estilo que compartan muchos documentos, y al hacer un cambio en la hoja de estilo todos los documentos que la usan tendrán la apariencia deseada.

También se puede tener control sobre ciertos aspectos que antes no se tenía, por ejemplo se pueden definir los márgenes de un documento o párrafo, o definir el espacio entre caracteres.

Las hojas de estilo son la innovación más importante al HTML (también se usan en otros lenguajes como XML y SGML), ya que le dan capacidades que nunca tuvo.

El uso de diversas unidades de medición pixeles, puntos, picas, mm, en los principales elementos del HTML, como son tablas, fluentes, bordes y en general los elementos que tenían atributos como “size “height” width”.

El posicionamiento de bloques de texto en cualquier parte del documento HTML, ya que es posible definir en diversas unidades la posición de un bloque de texto.

Mejor control sobre las fluentes que es necesario para otras tecnologías relacionadas como las fluentes dinámicas.

El poder cambiar las características de una hoja de estilo mediante el uso de lenguajes de programación Web como “Javascript” o “VB script”.

Las hojas de estilo son la base de la implementación estándar del HTML dinámico o DHTML.

1.1.1.6.4. ExtJs

ExtJS es una biblioteca de JavaScript para el desarrollo de Aplicaciones Ricas en Internet (RIA) usando tecnologías como AJAX, DHTML y DOM.

Originalmente construida como una extensión de la biblioteca YUI, en la actualidad puede usarse como extensión para las bibliotecas jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente.

ExtJS incluye widgets personalizables, buenos diseños, intuitivo, API extensa y fácil de utilizar, licencia Libre y Comercial disponibles.⁵



Figura 13. ExtJs

1.1.1.6.5. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.⁶

⁵ http://www.ecured.cu/index.php/Sencha_Ext_JS

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.

Características:

- Javascript es un lenguaje interpretado.
- Desarrollo en web.
- Ejecución de los programas en el navegador cliente: Es el navegador el que interpreta las instrucciones.
- No hay intervención por parte del servidor.

1.1.1.6.6. DHTMLX

DHTMLX es una serie de librerías AJAX con una gran biblioteca de componentes javascript DHTMLX nos permiten crear aplicaciones RIA bastante interesantes con muy poco esfuerzo. DHTMLX incluye buenos diseños, intuitivo, API extensa y fácil de emplear, licencia Libre y Comercial disponibles.



Figura 14. DHX

⁶<http://www.librosweb.es/javascript/capitulo1.html>

1.1.1.6.7. Java

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para Java script. Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet).

⁷

Java es un lenguaje de objetos, independiente de la plataforma.

Algunas características notables:

- Robusto.
- Gestiona la memoria automáticamente.
- No permite el uso de técnicas de programación inadecuadas.
- Multithreading.
- Cliente-Servidor.
- Mecanismos de seguridad incorporados.
- Herramientas de documentación incorporadas.

Java posee ciertas características que hoy día se consideran estándares en los lenguajes OO:

- Objetos.
- Clases.
- Métodos.
- Subclases.
- Herencia simple.
- Enlace dinámico.
- Encapsulamiento.

⁷ <http://informatica-full2.blogspot.com/2009/06/que-es-java.html>

Java es un lenguaje que ha sido diseñado para producir software:

- **Confiable:** minimiza los errores que se escapan a la fase de prueba.
- **Multiplataforma:** los mismos binarios funcionan correctamente en Windows/95/NT/XP/VISTA, Linux, Unix/Motif y Power/Mac.
- **Seguro:** Applets recuperados por medio de la red no pueden causar daño a los usuarios.
- **Orientado a objetos:** Beneficioso tanto para el proveedor de bibliotecas de clases como para el programador de aplicaciones.
- **Robusto:** Los errores se detectan en el momento de producirse, lo que facilita la depuración.



Figura 15. Java

1.1.1.6.8. Arquitectura y Diseño: Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El patrón MVC se ve frecuentemente en *aplicaciones web*, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.⁸

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño

⁸ http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

ofuscado, es decir, que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

A todo esto, se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Los elementos del patrón son tres:

- Modelo: Datos y reglas de negocio.
- Vista: Muestra la información del modelo al usuario.
- Controlador: Gestiona las entradas del usuario.

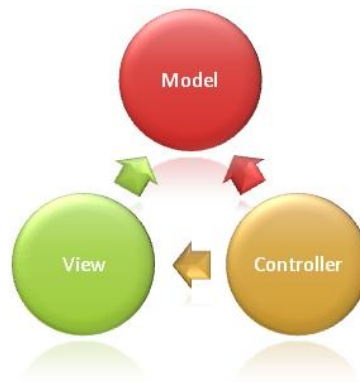


Figura 16. Modelo Vista Controlador (MVC)

1.1.1.6.8.1. Concepto: Patrón MVC

A alto nivel, el patrón MVC permite separar los distintos aspectos de una aplicación (Inputs Lógicos, Lógica de Negocio, Lógica de UI), otorgando un bajo acoplamiento entre estos elementos. Esto permite que al momento de construir una aplicación, nos focalicemos sólo en un aspecto, como por ejemplo en una vista, sin tener dependencia de la lógica del negocio.

Además el desacoplamiento de sus componentes, permiten que los distintos aspectos se puedan desarrollar en paralelo. Así, un desarrollador se puede abocar a la vista, un segundo desarrollador a la lógica del controlador y otro focalizarse en la lógica de negocio en el modelo.

1.1.1.6.8.2.Ventajas de Usar MVC

- Proporciona tener completo control sobre el comportamiento de una aplicación
- Testear una aplicación se vuelve más fácil, al no tener que instanciar la clase de la página individual, sus controles anidados y clases adicionales que dependan de su uso para el testeo.
- Soporta muy bien aplicaciones que requieran de un gran equipo de desarrollo, donde se necesita un gran nivel de profundidad y modularización, otorgando un completo control sobre el módulo asignado, independiente del resto.

1.1.1.6.8.3.Componentes Patrón MVC

- **Modelo:** Corresponde a la parte de la aplicación que implementa la lógica del Dominio de Datos de la aplicación, además de incorporar la persistencia de datos. Frecuentemente, los objetos del modelo recuperan y almacenan los estados del modelo en la base de datos. Aquí se hace el levantamiento de los objetos que el Sistema deba utilizar y es el proveedor de los recursos al Controlador
- **Vista:** Es el componente que despliega la interfaz de usuario de aplicación. Normalmente (no siempre), esta UI es creada a partir de los datos del modelo (Un ejemplo puede ser una vista de edición de una tabla de Productos que muestran cuadros de texto, listas desplegadas y casillas de verificación basado en el estado actual de los objetos Producto). Esta puede ser una web HTML, un XML, un archivo binario, etc.
- **Controlador:** El controlador maneja la interacción del usuario, trabaja con el modelo, y por último selecciona la vista a renderizar en la interfaz de usuario. En una aplicación MVC, la vista sólo despliega información; el controlador maneja y responde los inputs e interacción del usuario. Éste escucha los cambios a la vista y se los envía al modelo.

Un ciclo a alto nivel se desarrolla de la siguiente forma: el usuario envía peticiones (a la vista) las cuáles son recibidas por el lado servidor con el controlador, éste sabe cómo y quién se debe comunicar con el Modelo para responder ante el requerimiento. La respuesta del controlador finalmente es una vista, o una vista parcial.

Muchos sistemas informáticos que utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: En líneas generales del **MVC** que corresponde al modelo. La unión

entre *capa de presentación* y *capa de negocio* conocido en el paradigma de la Programación por capas representaría la integración entre **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la *capa visual gráfica* de su correspondiente *programación y acceso a datos*, algo que mejora el desarrollo y mantenimiento de la *Vista* y el *Controlador* en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

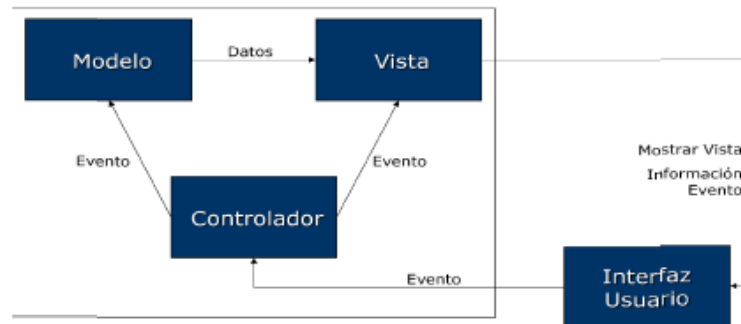


Figura 17. Patrón MVC

1.1.1.6.9. Servlet

Los servlets son objetos que corren dentro del contexto de un contenedor de servlets (ej.: Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones (ej.: OC4J Oracle), que, además de contenedor para servlet, tendrá contenedor para objetos más avanzados, como son los EJB (Tomcat sólo es un contenedor de servlets).⁹

La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor.

El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

Un servlet implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ej.: `javax.servlet.HttpServlet`). Al

⁹<http://www.vagos.es/showthread.php?t=62737>

implementar esta interfaz el servlet es capaz de interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` quienes contienen la información de la página que invocó al servlet.

1.1.1.6.10. Spring

Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia de OpenSource.

Lo primero que hay que entender de Spring, es que es un conjunto de módulos, de los cuáles podemos utilizar los que queramos.¹⁰

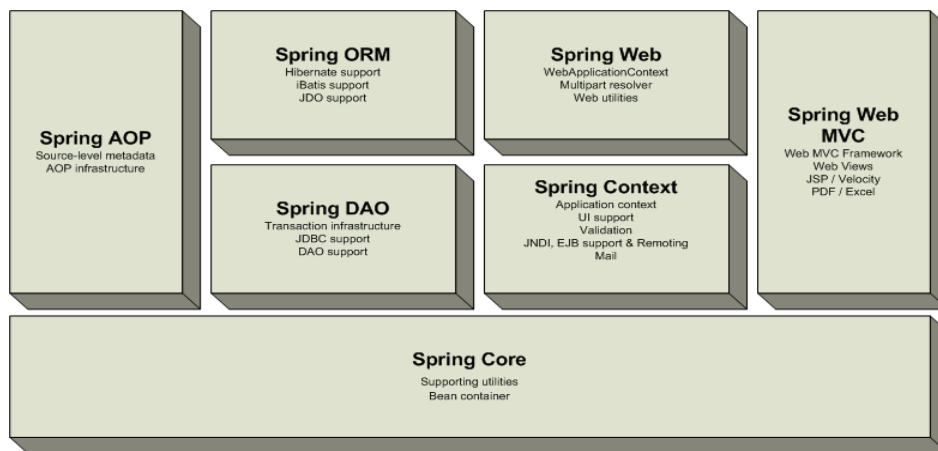


Figura 18. Arquitectura de Spring

En el proyecto utilizaremos el módulo Spring Web MVC para el desarrollo de aplicaciones Java basadas en Web construido sobre el núcleo de Spring. Dos de los objetivos más importantes de Spring MVC es permitir que el desarrollo se concentre en la lógica del negocio y que se haga empleando buenos principios de diseño orientado a objetos.

Para lograrlo se utiliza un concepto muy interesante llamado Inversión del Control (IoC), también conocido como el principio Hollywood: “No nos llames, nosotros te llamaremos.” Esto permite que el código escrito por los desarrolladores para la lógica principal del Sistema no tenga dependencias sobre las clases del *framework*; lo cual

¹⁰ <http://www.consultoriajava.com/tools/spring.shtml>

redunda en un código mucho más limpio y con la posibilidad de utilizar todas las ventajas de la programación orientada a objetos (específicamente la herencia).

1.1.1.6.11. Hibernate

Hibernate ofrece la *Persistencia Relacional para Java*, que para los no iniciados, proporciona unas muy buenas maneras para la persistencia de sus objetos de Java a y desde una base de datos subyacente. Más que ensuciar con SQL tus objetos y convertir consultas a y desde los objetos de primera magnitud, Hibernate puede preocuparse de todo ese maremágnum por ti. Tú utilizas solamente a los objetos, Hibernate se preocupa del SQL y de que las cosas terminan en la tabla correcta.¹¹

Permite trabajar con software orientado a objetos y bases de datos relacionales puede hacernos invertir mucho tiempo en los entornos actuales.

Hibernate es una herramienta que realiza el *mapping* entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Hibernate no sólo realiza esta transformación sino que nos proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo.



Figura 19. Arquitectura de Hibernate

¹¹ http://www.javahispano.org/contenidos/archivo/75/hibernate_1.pdf

Hibernate funciona asociando a cada tabla de la base de datos un Plain Old Java Object (POJO, a veces llamado Plain Ordinary Java Object). Un POJO es similar a una Java Bean, con propiedades accesibles mediante métodos setter y getter.

Algunas de las características se mencionan a continuación:

- Opensource (LGPL).
- Una tecnología muy madura.
- Un API personalizable.
- Persistencia utilizando JavaBeans.
- Consultas flexibles y poderosas.
- Trabaja los objetos persistentes en transacciones separadas.
- Provee el HQL que es un lenguaje de Consultas orientadas a Objetos muy flexible.
- Provee un mecanismo de peticiones a través de criterios.
- Soporta queries de SQL nativas.
- Soporta las operaciones relacionales.
- Inner/outer/full joins.
- Agregaciones (max, avg) y agrupamientos (Group).
- Ordenamientos (Order By).
- Consultas anidadas.

De una manera muy rápida y optimizada podremos generar BBDD en cualquiera de los entornos soportados: Oracle, DB2, MySql, PostgreSQL, etc.

Motores de persistencia:

Un motor de persistencia es un componente software encargado de traducir entre objetos (de un programa orientado a objetos) y registros (de la base de datos relacional).

Es decir, es el encargado de que el programa y la base de datos se “entiendan”.

Esto nos da dos ventajas importantes:

- Podemos programar con orientación a objetos, aprovechando las ventajas de flexibilidad, mantenimiento y reusabilidad.
- Y además podemos usar una base de datos relacional, aprovechándonos de su madurez y su estandarización así como de las herramientas relacionales que hay para ella.

Tipos de relaciones (Componentes y Colecciones):

En todo diseño relacional los objetos se referencian unos a otros a través de relaciones, las típicas son:

- Uno a Uno (One-To-One) 1 – 1
- Uno a Muchos (One-To-Many) 1 – N
- Muchos a Muchos (Many-To-Many) N – M

El Proceso del Desarrollo:

Hay varias maneras de acercarse al desarrollo con Hibérnate. Aquí está el que estamos utilizando nosotros hoy, porque es probablemente el más simple de entender:

- Crea tu tabla del SQL para guardar tus objetos persistentes.
- Crea un JavaBean que represente ese objeto en código.
- Crea un archivo de mapeo de manera que Hibérnate sepa qué características del bean se mapean a que campos del SQL.
- Crea un archivo de propiedades de manera que Hibérnate conozca la configuración JDBC para acceder a la base de datos.
- Comenzar a usar el Hibérnate API.



Figura 20. Hibernate

1.1.1.7.Base de Datos

1.1.1.7.1. Definición de Base de Datos

Una serie de datos organizados y relacionados entre sí, los cuáles son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Entre las principales características de los sistemas de Base de Datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la emplean. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.¹²

1.1.1.7.2. Herramienta PostgreSQL 9.0 para el manejo de la BD

PostgreSQL es un Sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley.

PostgreSQL es una derivación libre (OpenSource), y utiliza el lenguaje SQL92/SQL99, así como otras características.

PostgreSQL es un Sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones,

¹² <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un Sistema de gestión de bases de datos puramente orientado a objetos.

Algunas de las características para hacer uso de este gestor se mencionan a continuación:

- *Alta concurrencia.* Mediante un Sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés), PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- *Soporte para una amplia variedad de tipos nativos.* PostgreSQL provee nativamente soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC, Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Consta con un lenguaje propio llamado [PL/PgSQL (similar al PL/SQL de oracle), pero también soporta los lenguajes C, C++, Java PL/Java web, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme.

PostgreSQL es un magnífico gestor de bases de datos. Tiene prácticamente todo lo que tienen los gestores comerciales, haciendo de él una muy buena alternativa GPL para el presente proyecto.

Postgres cuenta con Herramientas de Administración como es PgAdmin3 y PgAccess: Entorno de escritorio visual, PhpPgAdmin y psql que es un Cliente de consola.

1.1.1.8.Sistema de Información Automatizado

1.1.1.8.1. La Internet

Algunos definen Internet como "La Red de Redes", y otros como "La Autopista de la Información".

Efectivamente, Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores, o sea de unos pocos ordenadores en un mismo edificio o empresa.

Por la Red Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información. Hay 50 millones de "Internautas", es decir, de personas que "Navegan" por Internet en todo el mundo. Se dice "Navegar" porque es normal el ver información que proviene de muchas partes distintas del mundo en una sola sesión.

Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras. Incluso podemos ver conectados a la red cámaras de vídeo, robots, y máquinas de refrescos, etc.

1.1.1.8.2. La World Wide Web o la Web

La World Wide Web consiste en brindar una interface simple y consistente para acceder a la inmensidad de los recursos de Internet. Es la forma más moderna de ofrecer información. El medio más potente. La información se ofrece en forma de páginas electrónicas.

El World Wide Web o WWW o W3 o simplemente Web, permite saltar de un lugar a otro en pos de lo que no interesa. Lo más interesante es que con unas pocas órdenes se puede mover por toda la Internet.

La World Wide Web permite una manera más organizada de acceder a la información disponible en Internet, presentando una interfaz amigable con el usuario mediante navegadores como Netscape, Mosaic y Microsoft Internet Explorer, Mozilla FireFox.

El surgimiento de la World Wide Web ha ayudado a un crecimiento considerable de Internet en la actualidad. Compañías pequeñas, empresas grandes, ayuntamientos,

estados, gobiernos de distintos países, universidades, bibliotecas, están presentes en Internet.

Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un Sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

1.1.1.9.Sistemas de Información vía Web

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

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

Pudiésemos definir entonces como Sistema de información al conjunto de elementos relacionados y ordenados, según ciertas reglas que aporta al Sistema objeto, es decir, a la organización a la que sirve y que marca sus directrices de funcionamiento, la información necesaria para el cumplimiento de sus fines; para ello, debe recoger, procesar y almacenar datos, procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. Actualmente, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con los sistemas de información para satisfacer sus necesidades de información.

1.1.2. Plan de Desarrollo del Software

1.1.2.1.Introducción

El presente documento es un Plan de Desarrollo del Software que sentará las bases para el desarrollo del proyecto, es una versión preparada para ser incluida en la propuesta elaborada en respuesta al proyecto de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma “Juan Misael Saracho”. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido propuesto por la universitaria Noemí Ilsen Portal Rivera, basado en la Metodología RUP (Rational Unified Process), en la que se procederá a cumplir con las cuatro fases que marca la metodología. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

El enfoque desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

1.1.2.1.1. Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El Director del proyecto, que lo utiliza para organizar la agenda y necesidades de recursos y para realizar su seguimiento.
- El Docente para evaluar el cumplimiento del proyecto.

1.1.2.1.2. Alcance

Aplicando el Plan de Desarrollo Software obtenemos una herramienta importante para realizar nuestro plan de trabajo el cuál coadyuvará al cumplimiento de nuestros objetivos en el tiempo propuesto gracias al cronograma de actividades establecido.

1.1.2.1.3. Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de Aplicación — proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

1.1.2.2.Vista General del Proyecto

La información que a continuación se incluye ha sido extraída de las diferentes reuniones, entrevistas y charlas que se han realizado con los usuarios destino del Comercial el Carmen – Tienda Central desde el inicio del proyecto.

1.1.2.2.1. Propósito, Alcance y Objetivos

1.1.2.2.1.1.Propósito

Con el presente proyecto se pretende mejorar “El Control y Administración de Almacenes de el Comercial el Carmen”.

1.1.2.2.1.2.Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del Sistema Automatizado de Control de Almacenes **CONAL**, un Sistema Informático que está comprendido por los siguientes módulos:

- **Módulo Personal.**

- **Módulo Almacén**
- **Módulo Reportes**
- **Módulo Backups**

1.1.2.2.1.3.Objetivos

1.1.2.2.1.3.1. Objetivo General

Agilizar la Gestión de la información en la Sección de Almacén del Comercial el Carmen.

1.1.2.2.1.3.2. Objetivos Específicos

- Llegar a brindar un Sistema que obtenga información oportuna de forma ágil y confiable para la Gestión de los Productos existentes en los Almacenes.
- Obtener un Sistema que permita generar reportes de forma rápida y sencilla.
- Otorgar a los usuarios total seguridad en el resguardo de su información, permitiendo el acceso sólo a usuarios autorizados.
- Conceder a los usuarios un Sistema que contribuirá en la Gestión eficiente de Almacenes en sus diferentes etapas, como: Entrada de Productos, Salida de Productos, Proveedores, etc.
- Crear una base de datos de acuerdo a las necesidades de la institución.
- Diseñar una interfaz de usuario de fácil manejo.

1.1.2.2.2. Suposiciones y Restricciones

Acontecimientos que deben ocurrir para que el proyecto sea ejecutado con éxito pero que están totalmente fuera del ámbito del control del equipo de proyecto.

1.1.2.2.2.1.Suposiciones

Suponemos que el Comercial el Carmen, Agencia Tarija cuenta con:

- Equipo de computación. En el que incluye un computador, impresora.
- Un ambiente adecuado para la implementación del Sistema.

- La disponibilidad de otorgar toda la información necesaria para la creación del software.
- Personal con conocimientos básicos de computación.
- Cuenta con equipos en los cuáles se podría implementar el Sistema.
- Presupuesto suficiente para la implementación del Sistema.
- Interés de la institución para utilizar el Sistema.
- Disponibilidad de tiempo de los trabajadores de la Empresa para asistir a la capacitación del Sistema así también de socialización a la Empresa de la importancia del uso de las TICs.

1.1.2.2.2. Restricciones

Limitaciones generalmente fuera del ámbito de control del equipo de proyecto que pueden afectar negativamente a su alcance.

- El Sistema será restringido, sólo usuarios privilegiados podrán acceder al Sistema.
- Para la manipulación de la base de datos, sólo podrá acceder el personal autorizado.
- Deberá contar como mínimo con un Sistema operativo Windows XP/Vista/Seven.
- El administrador deberá contar con el conocimiento necesario para el manejo del Sistema.
- El sistema no abarca el control de las ventas que realiza el Comercial.
- Los Almacenes serán administrados por los encargados de cada Agencia y Sucursales
- No contar con los fondos suficientes para llevar a cabo la capacitación a los usuarios del Sistema.

1.1.2.2.3. Entregables del Proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es necesario destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se expongan los objetivos de cada iteración.

1.1.2.2.3.1. Plan de Desarrollo del Software

Es el presente documento.

1.1.2.2.3.2. Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales y otros sistemas etc.) permite situar al Sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso empleando estereotipos específicos para este modelo.

La descripción del conjunto de procesos del negocio es una tarea crucial, ya que determinan los límites del proceso de modelado posterior, consideramos los objetivos estratégicos de la organización, teniendo en cuenta que esos objetivos serán descompuestos en un conjunto de subobjetivos más concretos, para la determinación de procesos de negocio. Se presentan los modelos definidos en RUP como modelo del negocio (modelo de casos de Uso del Negocio y de Objetos del Negocio).

1.1.2.2.3.2.1. Modelo de Casos de Uso del Negocio Encargado de Ventas

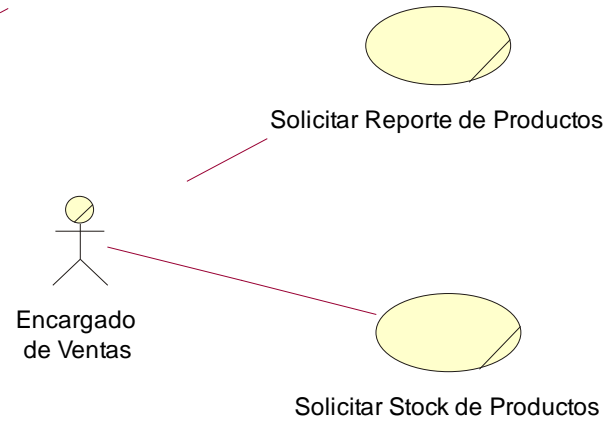


Figura 21. Modelo de Casos de Uso del Negocio Encargado de Ventas

1.1.2.2.3.2.2. Modelo de Casos de Uso del Negocio Administrador

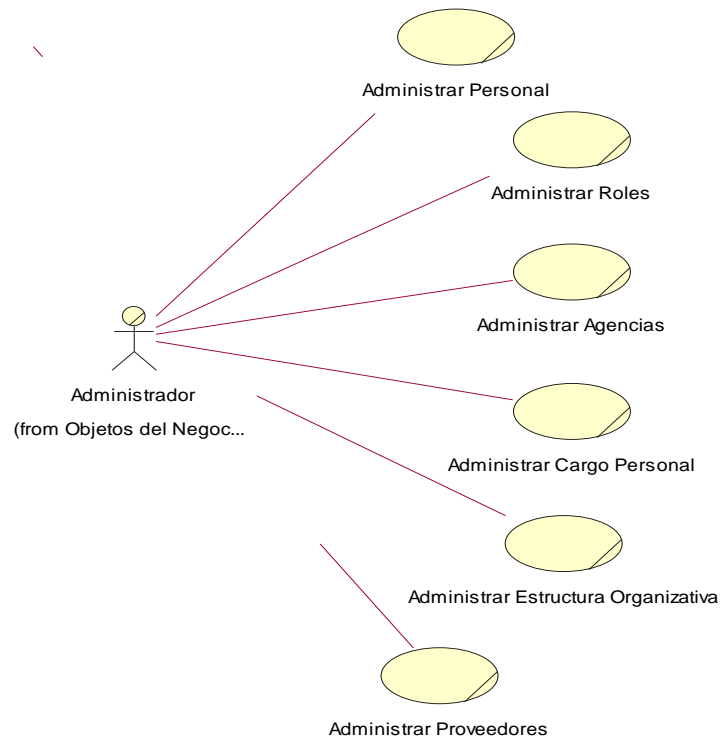


Figura 22. Modelo de Casos de Uso del Negocio Administrador

1.1.2.2.3.2.3. Modelo de Casos de Uso del Negocio Jefe de Almacén

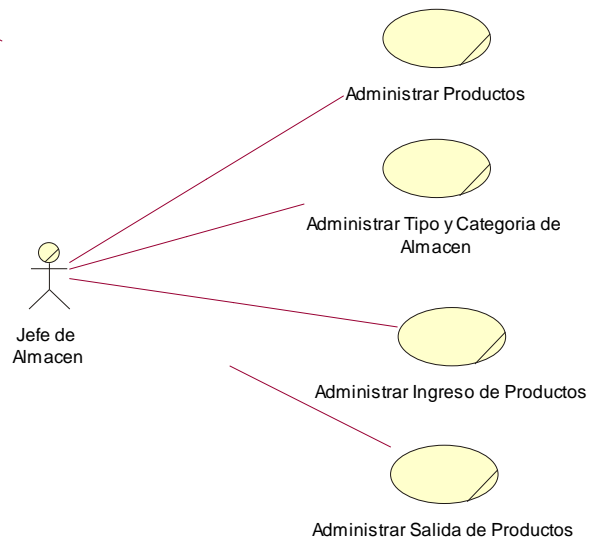


Figura 23. Modelo de Casos de Uso del Negocio Jefe de Almacén

1.1.2.2.3.2.4. Modelo de Casos de Uso del Negocio Gerente

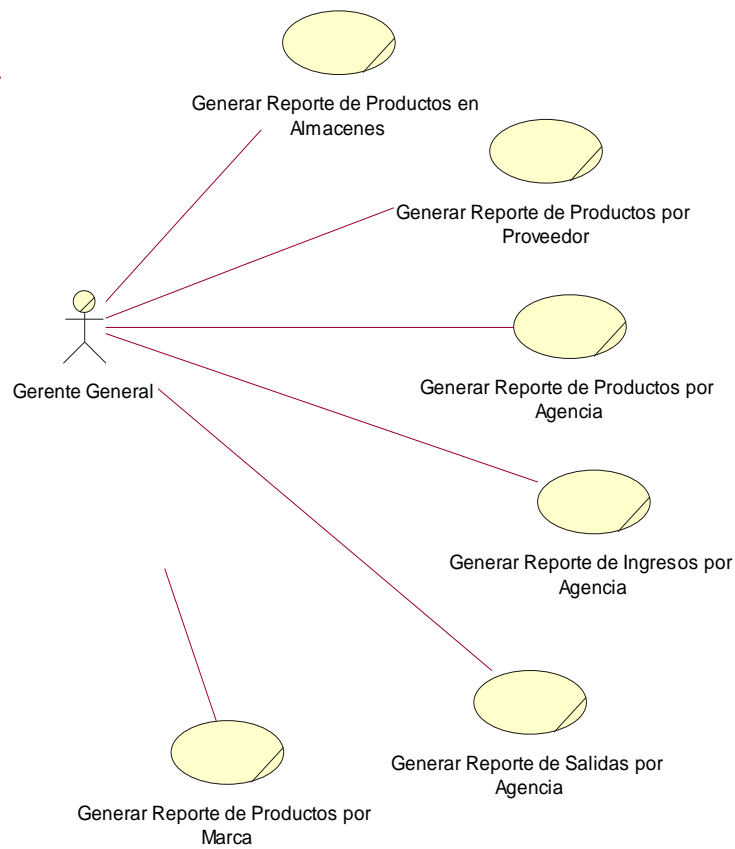


Figura 24. Modelo de Casos de Uso del Negocio Gerente

1.1.2.2.3.3. Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada Caso de Uso del Negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del Sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

1.1.2.2.3.3.1. Generar Reportes de los Productos Existentes en los Almacenes

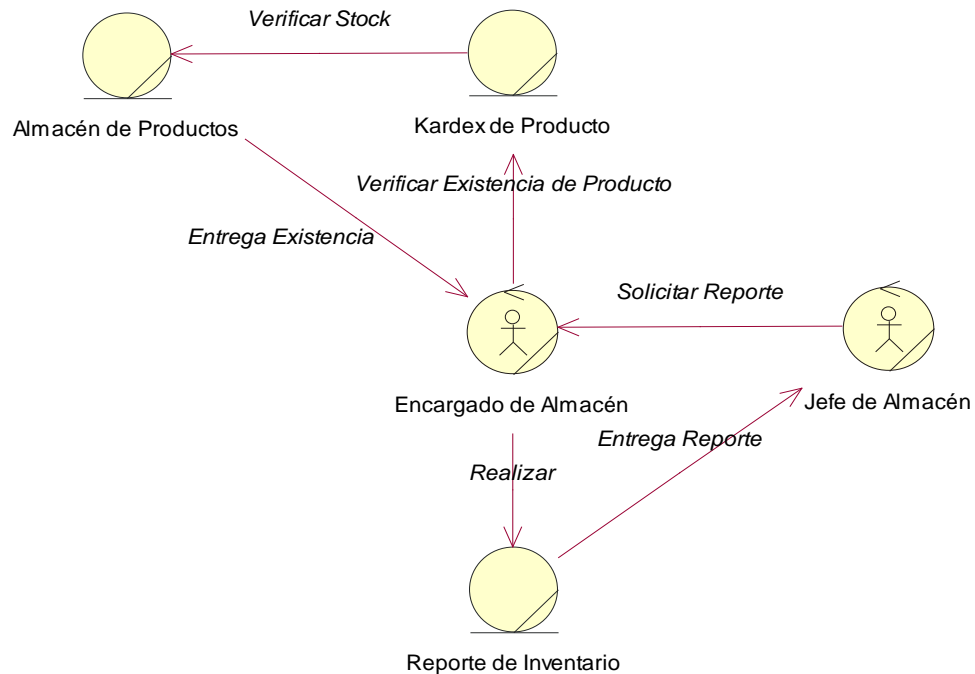


Figura 25. Generar Reportes de los Productos Existentes en los Almacenes

1.1.2.2.3.3.2. Controlar el Inventario de Productos en los Almacenes

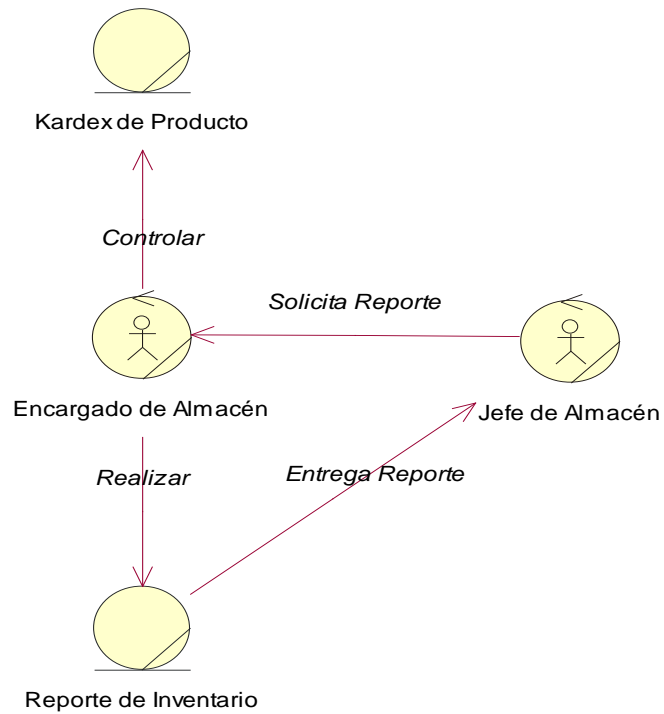


Figura 26. Controlar el Inventario de Productos en los Almacenes

1.1.2.2.3.3.3. Controlar Kardex de Productos

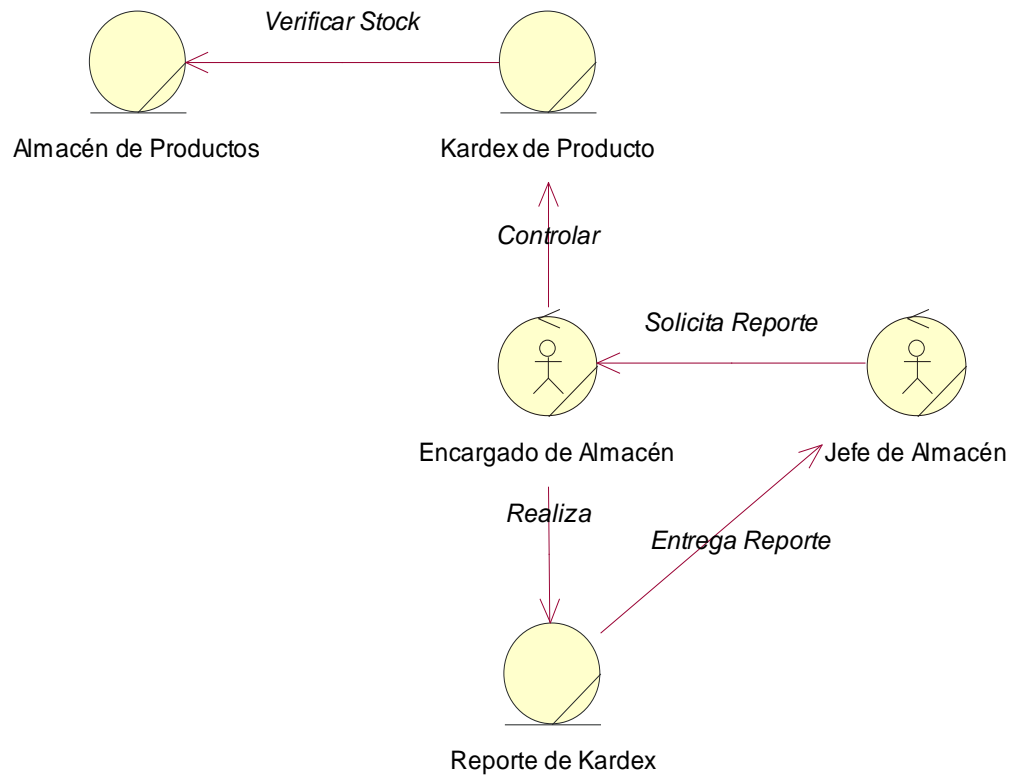


Figura 27. Controlar Kardex de Productos

1.1.2.2.3.3.4. Controlar Ingresos de Productos

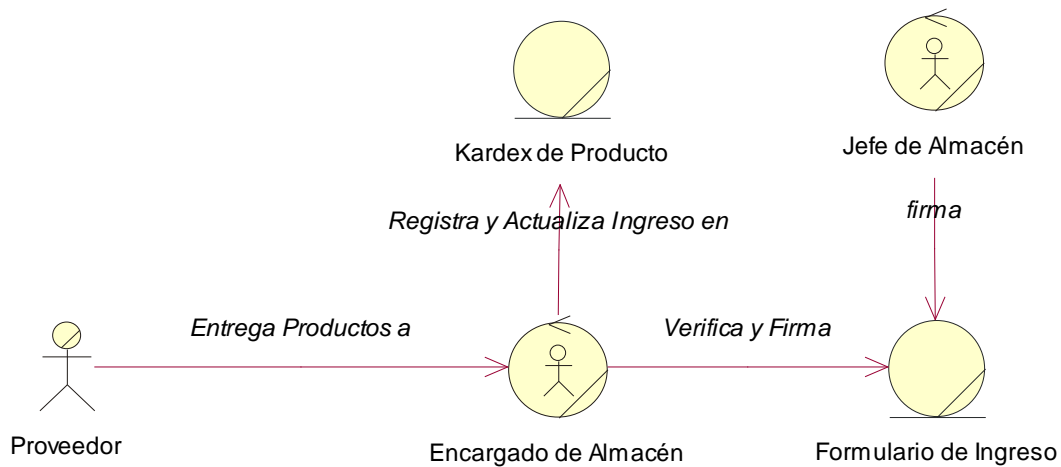


Figura 28. Controlar Ingresos de Productos

1.1.2.2.3.3.5. Controlar Salidas de Productos

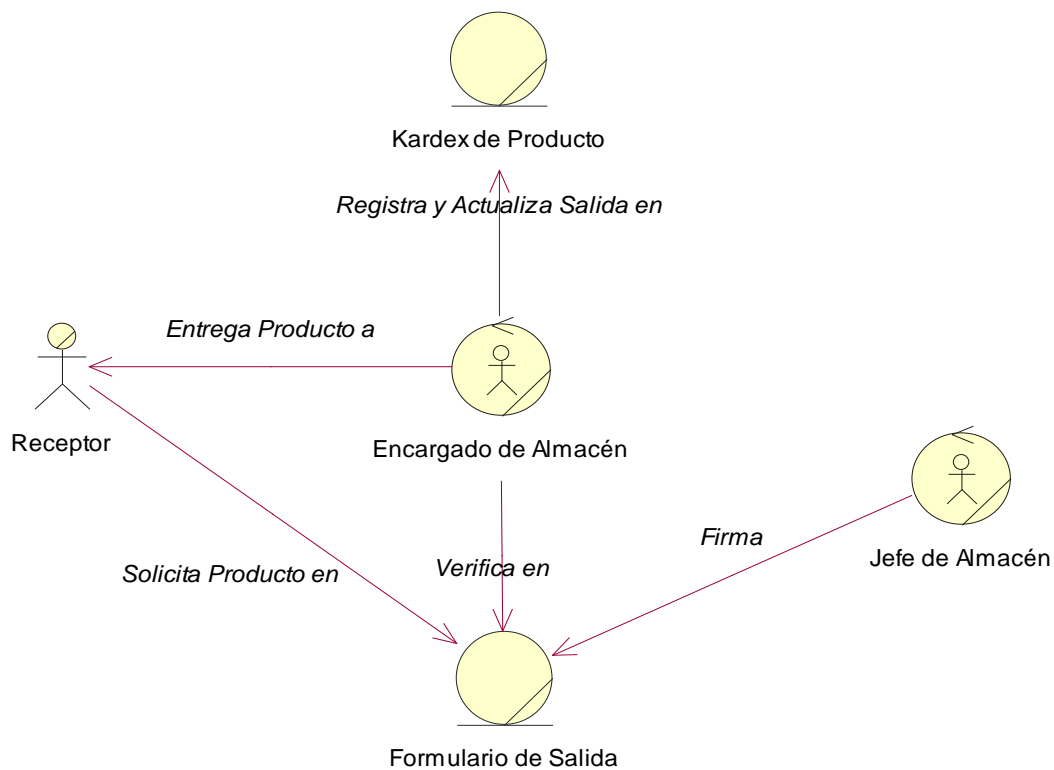


Figura 29. Controlar Salidas de Productos

1.1.2.2.3.3.6. Generar Reportes para el Jefe de Almacén

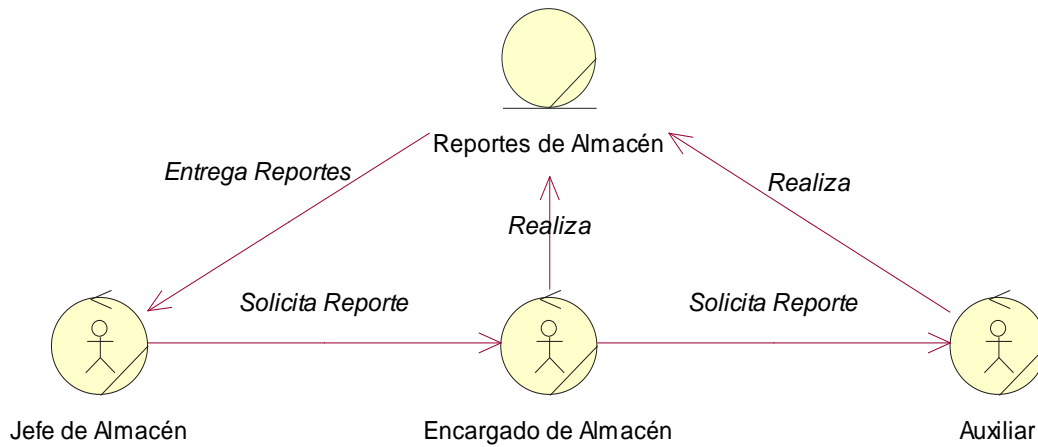


Figura 30. Generar Reportes para el Jefe de Almacén

1.1.2.2.3.3.7. Generar Reportes para la Administración General

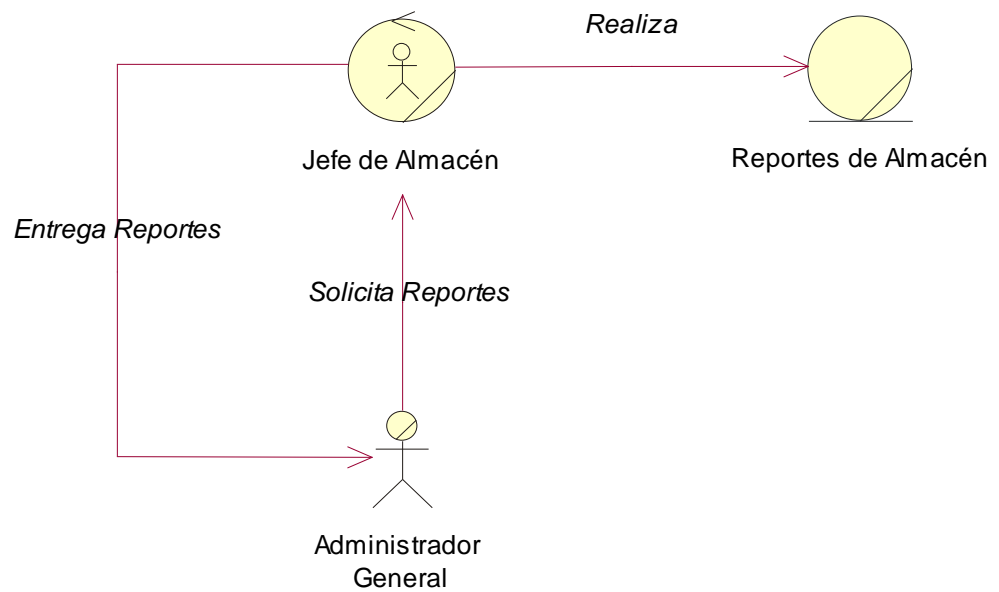


Figura 31. Generar Reportes para la Administración General

1.1.2.2.3.4. Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del Sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

1.1.2.2.3.4.1. Introducción

El presente documento es un artefacto de la disciplina Requisitos en la metodología RUP la cual estamos implementando.

1.1.2.2.3.4.2. Propósito

Comprender la estructura y la dinámica del Sistema desarrollado.

Identificar el nivel de complejidad del Sistema.

Establecer posibles mejoras.

1.1.2.2.3.4.3. Alcance

Identificar y definir procesos del Sistema según los objetivos de la organización.

Definir un Caso de Uso para cada proceso del Sistema (el diagrama de Caso de Uso nos detalla el contexto y los límites de la organización).

1.1.2.2.3.4.4. Diagramas de Casos de Uso

1.1.2.2.3.4.4.1. Casos de Uso del Sistema General

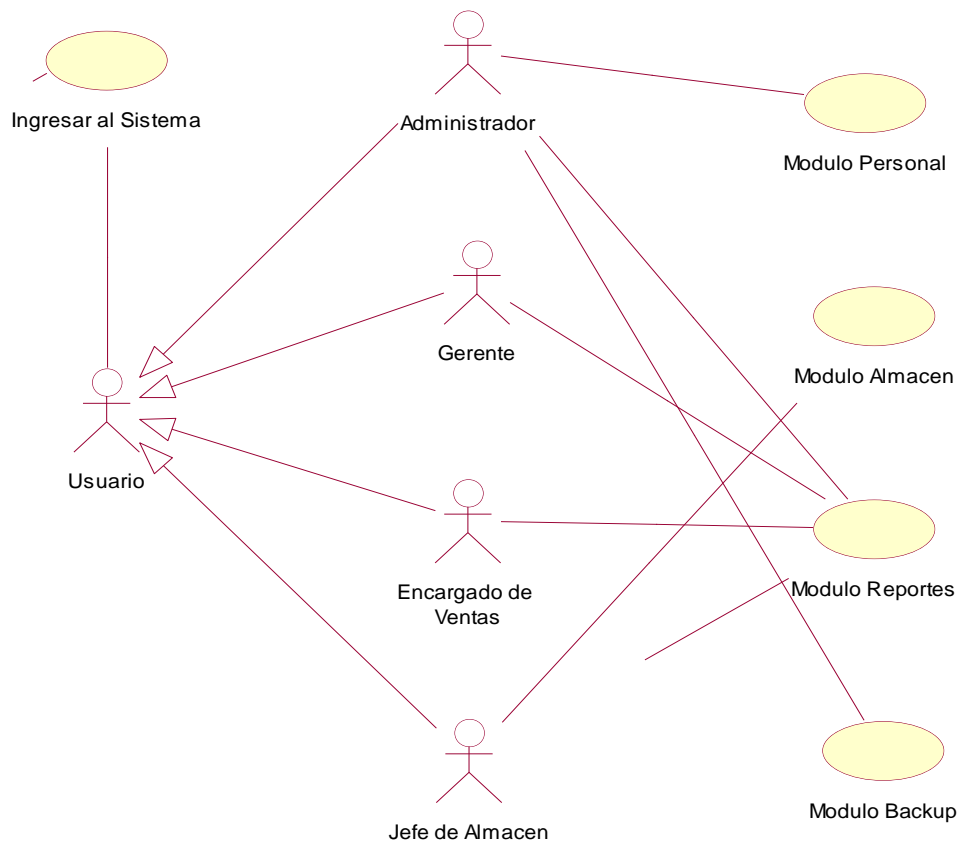


Figura 32. Modelo Casos de Uso del Sistema General

1.1.2.2.3.4.4.1.1. Modelo Casos de Uso Módulo Personal General

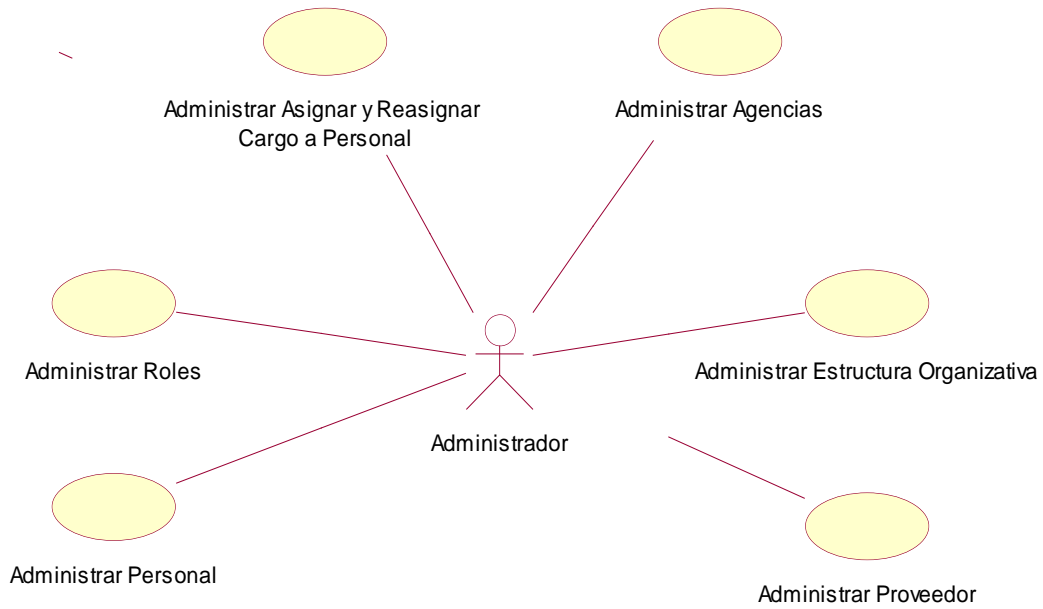


Figura 33. Modelo Casos de Uso Módulo Personal General

1.1.2.2.3.4.4.1.2. Modelo Casos de Uso Módulo Almacén General

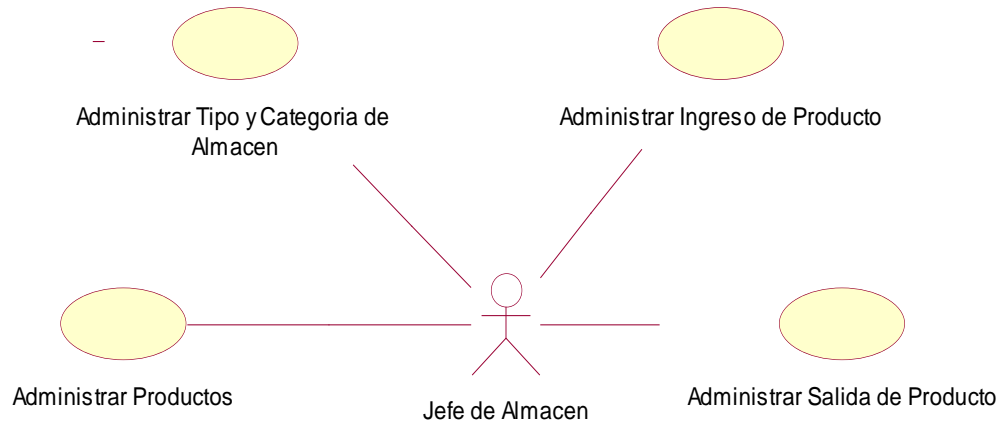


Figura 34. Modelo Casos de Uso Módulo Almacén General

1.1.2.2.3.4.4.1.3. Modelo Casos de Uso Módulo Reportes General

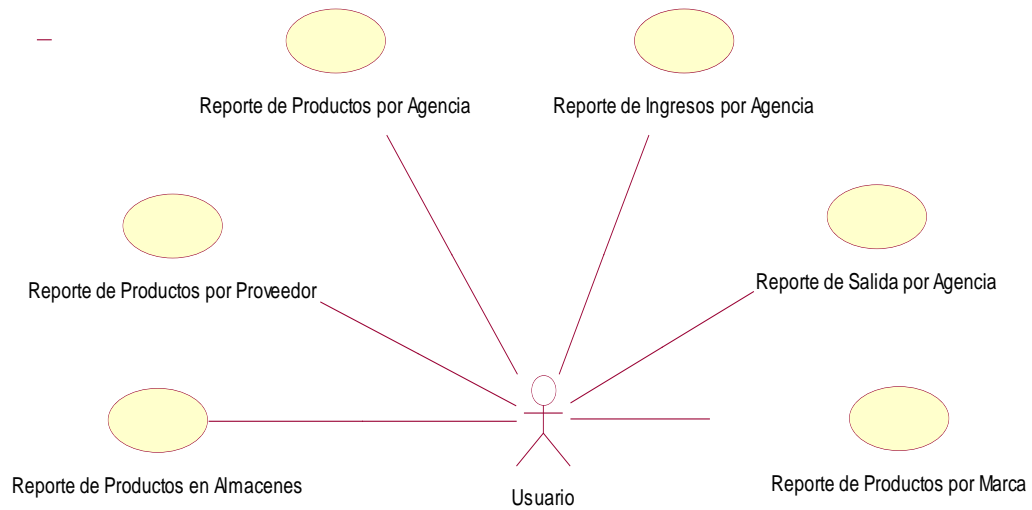


Figura 35. Modelo Casos de Uso Módulo Reportes General

1.1.2.2.3.4.4.1.4. Modelo Casos de Uso Módulo Backup

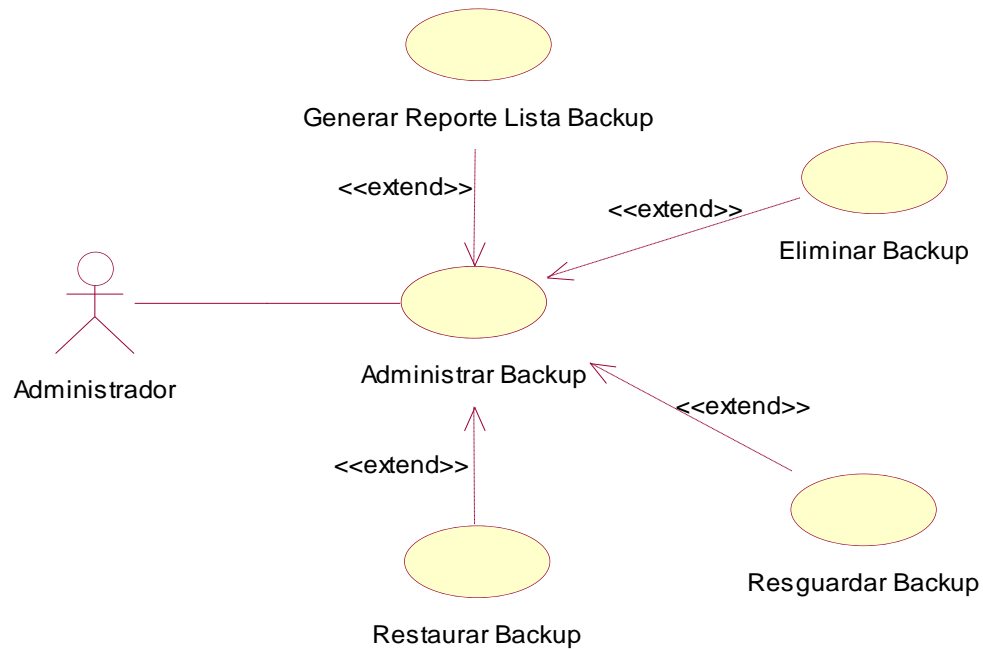


Figura 36. Modelo Casos de Uso Módulo Backup

1.1.2.2.3.4.5. Casos de Usos del Sistema Específicos

1.1.2.2.3.4.5.1. Modelo Casos de Uso Ingresar al Sistema



Figura 37. Modelo Casos de Uso Ingresar al Sistema

1.1.2.2.3.4.5.2. Modelo Casos de Uso Administrar Personal

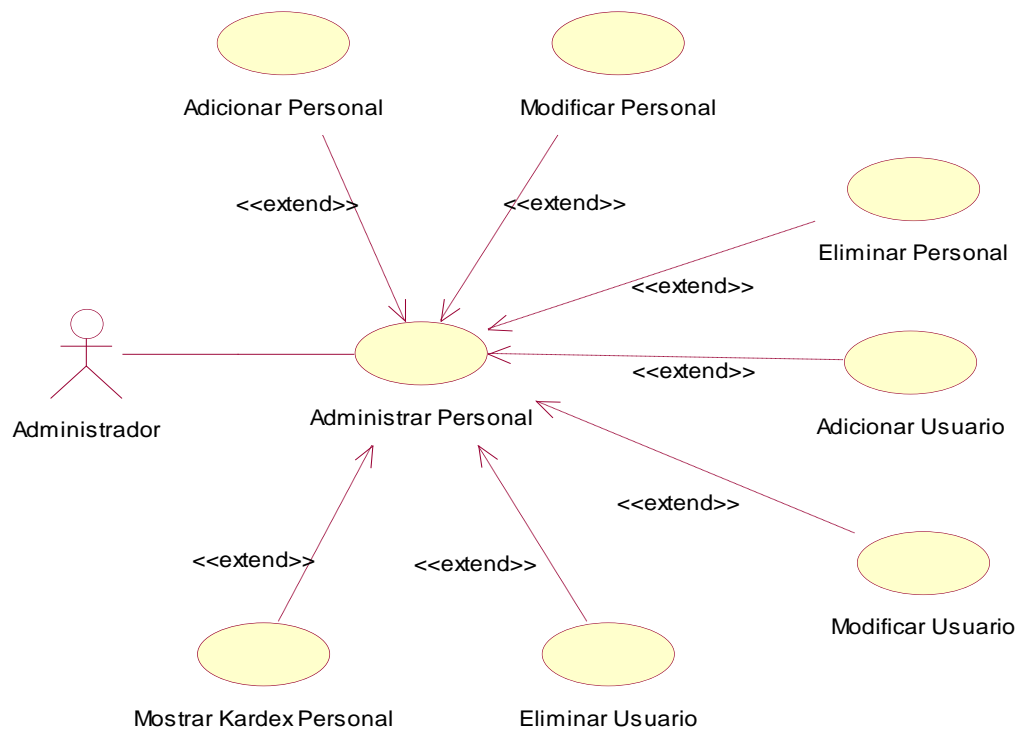


Figura 38. Modelo Casos de Uso Administrar Personal

1.1.2.2.3.4.5.3. Modelo Casos de Uso Administrar Roles

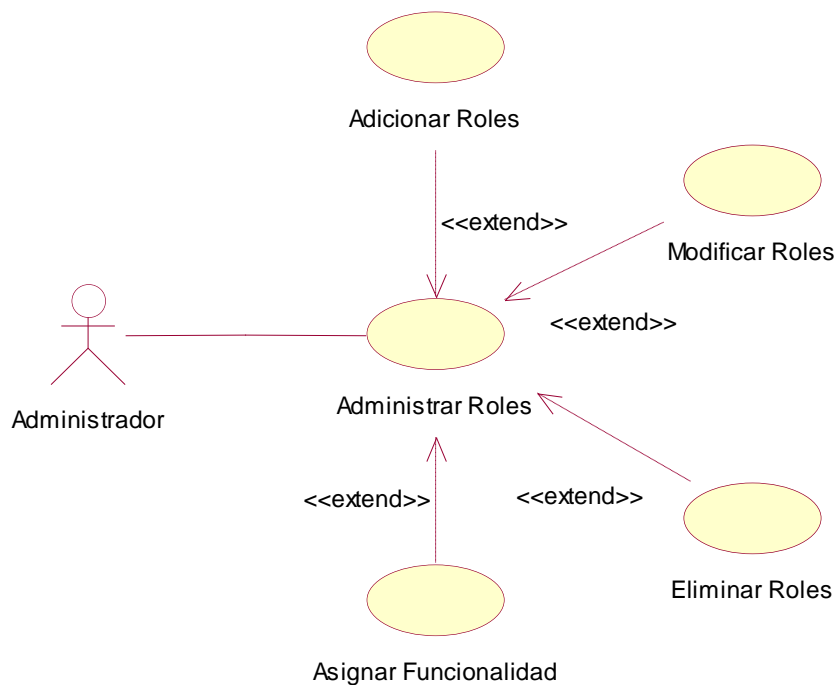


Figura 39. Modelo Casos de Uso Administrar Roles

1.1.2.2.3.4.5.4. Modelo Casos de Uso Administrar Asignar y Reasignar Cargo a Personal

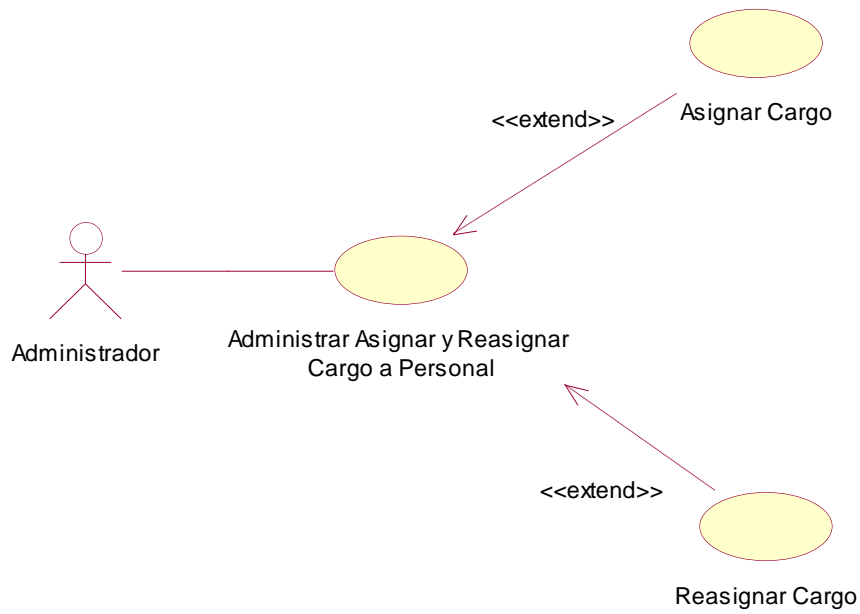


Figura 40. Modelo de Casos de Uso Administrar Asignar y Reasignar Cargo a Personal

1.1.2.2.3.4.5.5. Modelo Casos de Uso Administrar Agencias

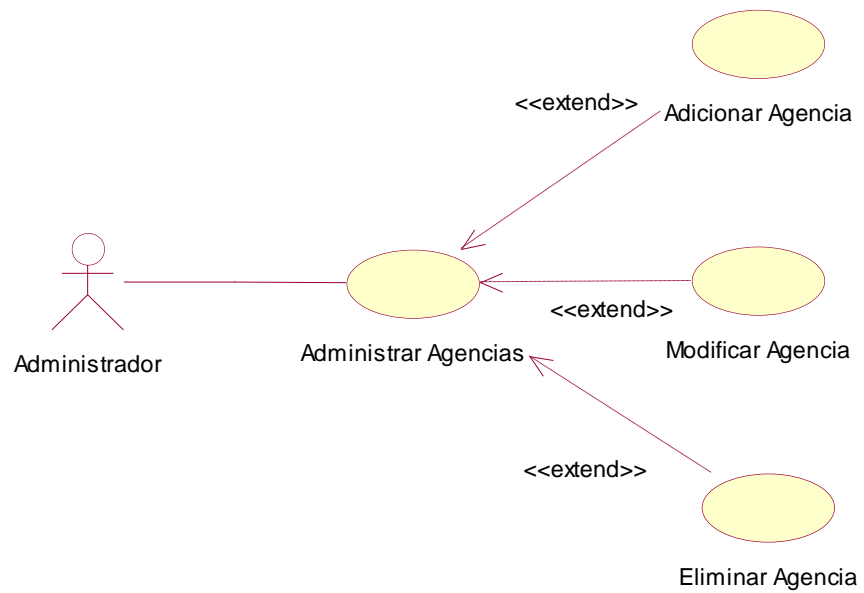


Figura 41. Modelo Casos de Uso Administrar Agencias

1.1.2.2.3.4.5.6. Modelo Casos de Uso Administrar Estructura Organizativa

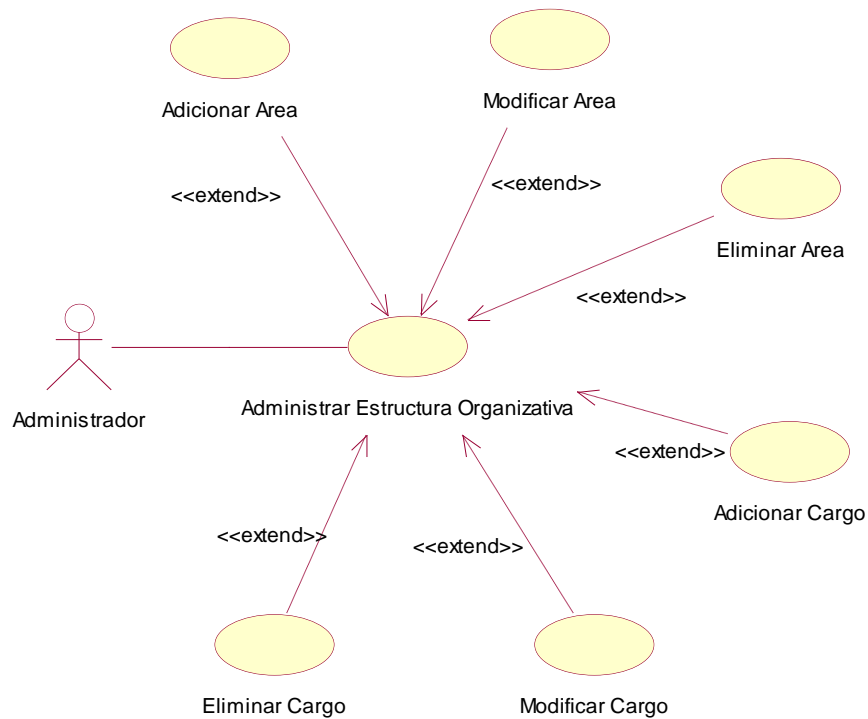


Figura 42. Modelo Casos de Uso Administrar Estructura Organizativa

1.1.2.2.3.4.5.7. Modelo Casos de Uso Administrar Proveedor

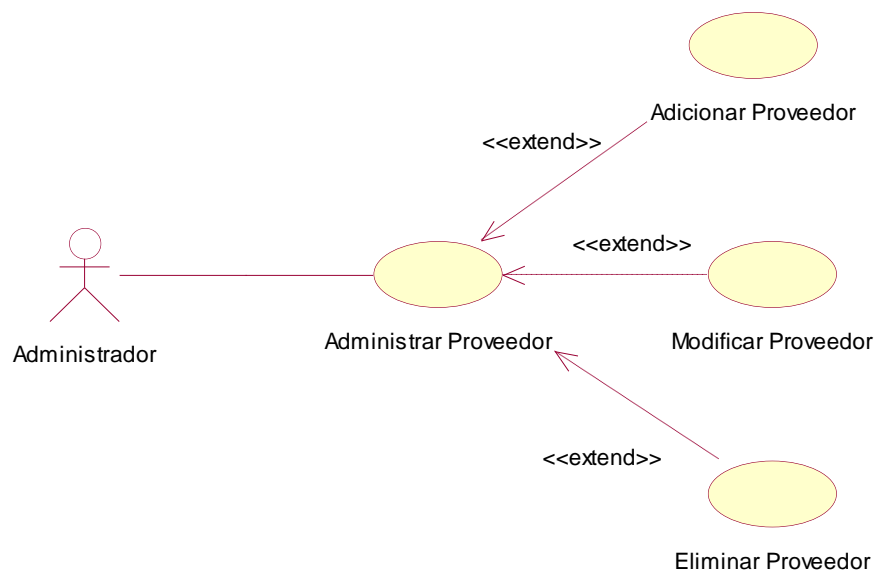


Figura 43. Modelo Casos de Uso Administrar Proveedor

1.1.2.2.3.4.5.8. Modelo Casos de Uso Administrar Productos

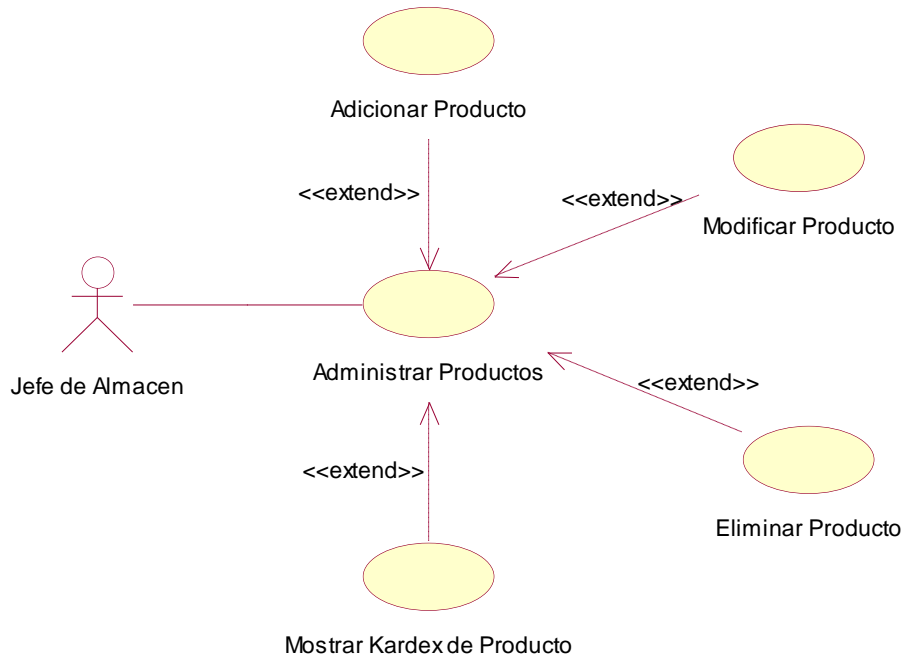


Figura 44. Modelo Casos de Uso Administrar Productos

1.1.2.2.3.4.5.9. Modelo Casos de Uso Administrar Tipo y Categoría de Almacén

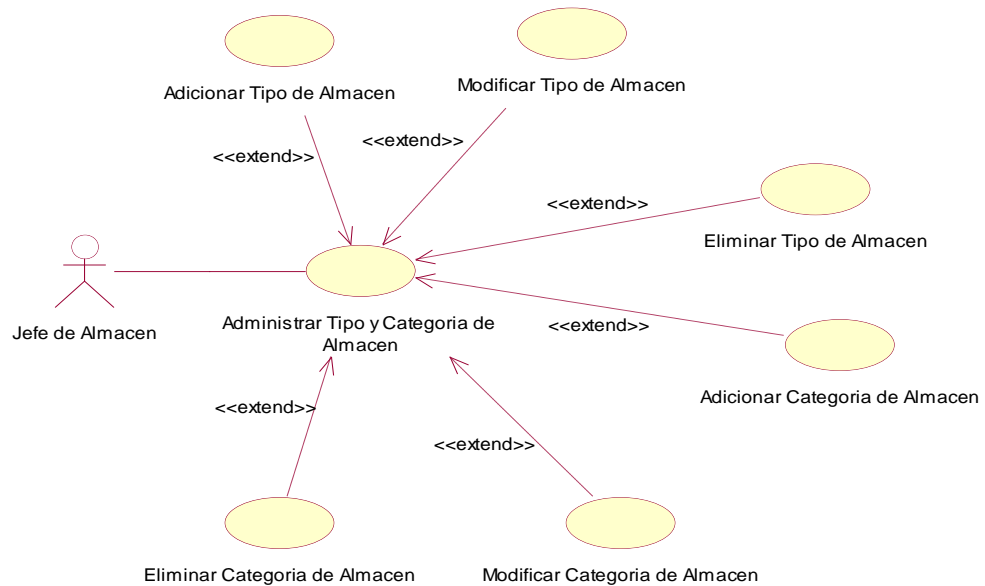


Figura 45. Modelo Casos de Uso Administrar Tipo y Categoría de Almacén

1.1.2.2.3.4.5.10. Modelo Casos de Uso Administrar Ingreso de Producto

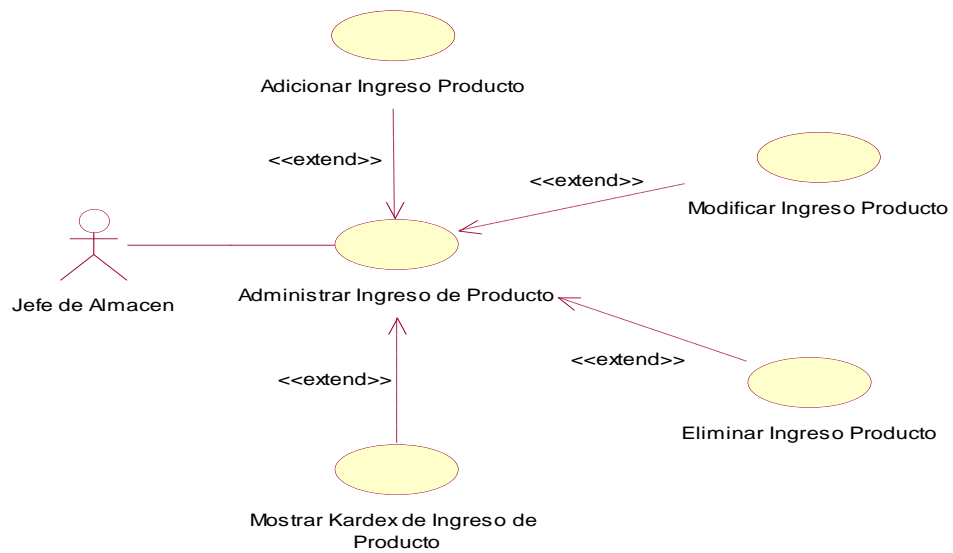


Figura 46. Modelo Casos de Uso Administrar Ingreso de Productos

1.1.2.2.3.4.5.11. Modelo Casos de Uso Administrar Salida de Producto

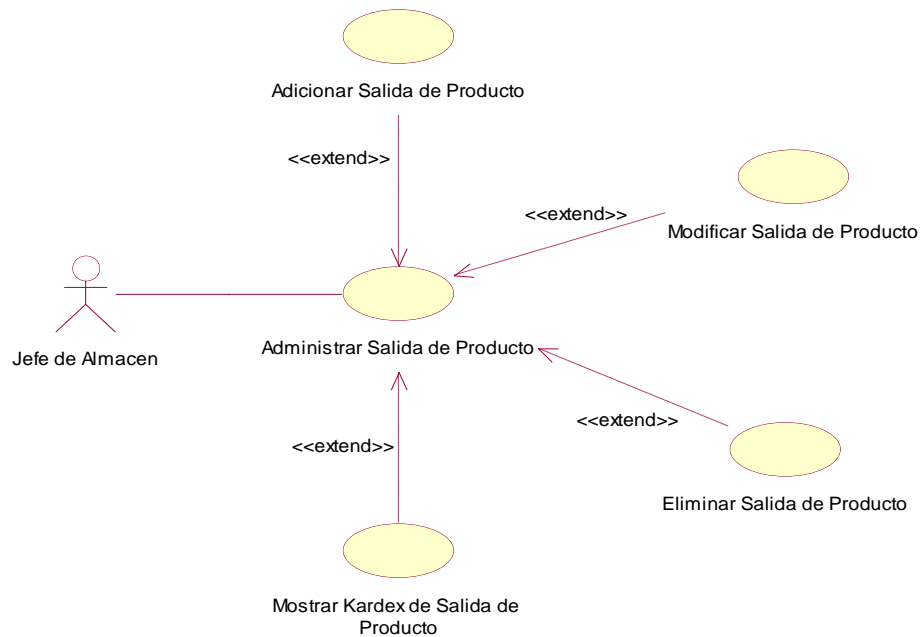


Figura 47. Modelo Casos de Uso Administrar Salida de Producto

1.1.2.2.3.4.5.12. Modelo Casos de Uso Administrar Reportes

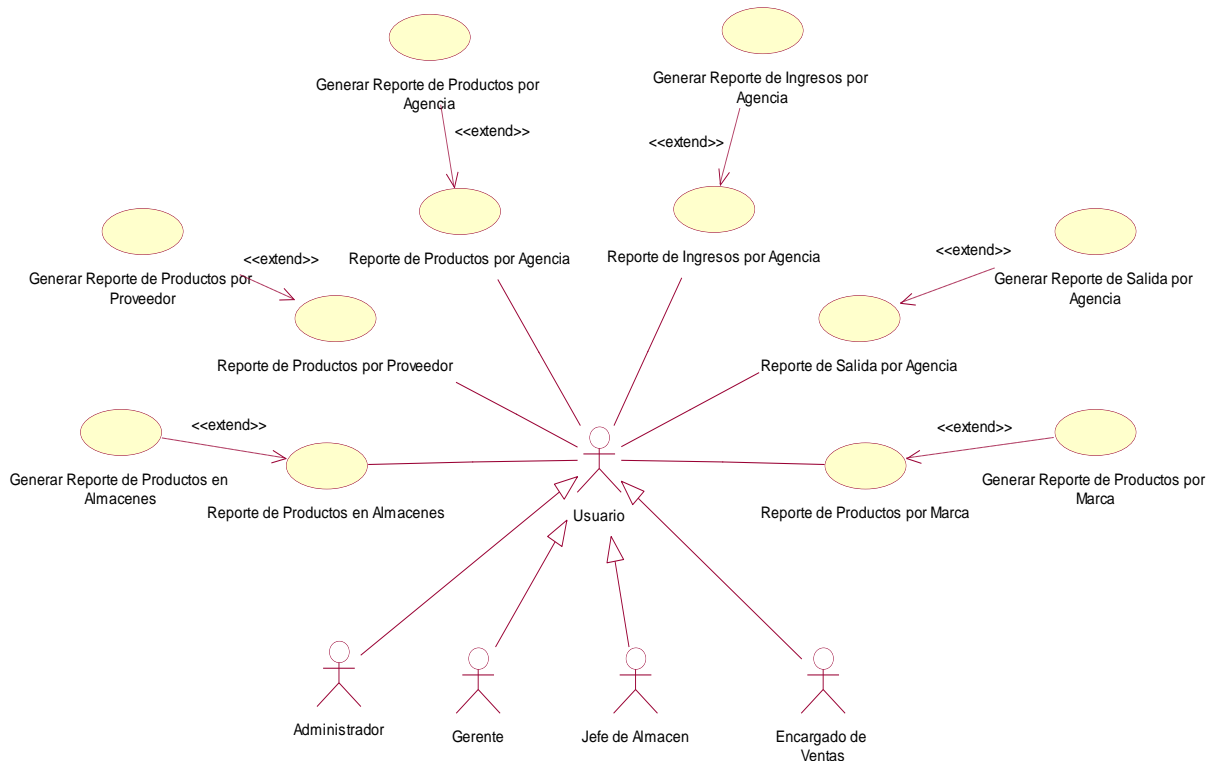


Figura 48. Modelo Casos de Uso Administrar Reportes

1.1.2.2.3.5. Perfil de Usuario

1.1.2.2.3.5.1. Administrador

Representante	Administrador
Descripción	Administrador de los recursos del Sistema y de Gestionar Personal
Tipo	SuperUsuario.
Responsabilidades	Responsable de Administrar Personal, control de Roles y acceso a usuarios al Sistema, Administrar la Estructura Organizativa.
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 9. Administrador

1.1.2.2.3.5.2. Jefe de Almacén

Representante	Operador.
Descripción	Usuario independiente y directo, con ciertos atributos- permisos de ejecución de la aplicación.
Tipo	Usuario1.
Responsabilidades	Hacer uso adecuado del Sistema velar por la integridad del mismo, respetando las funcionalidades del mismo.
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 10. Jefe de Almacén

1.1.2.2.3.5.3. Encargado de Ventas

Representante	Operador.
Descripción	Usuario independiente y directo, con ciertos atributos- permisos de ejecución de la aplicación.
Tipo	Usuario1.
Responsabilidades	Hacer uso adecuado del Sistema velar por la integridad del mismo, respetando las funcionalidades del mismo.
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 11. Revisor

1.1.2.2.3.5.4. Gerente General

Representante	Operador.
Descripción	Usuario independiente y directo, con ciertos atributos- permisos de ejecución de la aplicación.
Tipo	Usuario2.
Responsabilidades	Hacer uso adecuado del Sistema velar por la integridad del mismo, respetando las funcionalidades del mismo.
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 12. Gerente General

1.1.2.2.3.5.4.1. Entorno del Usuario

El Comercial el Carmen debe acceder al Sitio Web del Sistema donde podrá hacer uso de la Información del mismo, en el cual el Usuario debe estar registrado con el Rol de Usuario1 o Usuario2, para poder obtener algunas características adicionales-atributos- permisos. Si es un Administrador debe acceder al programa instalado donde debe estar registrado como SuperUsuario para poder realizar la Gestión de todo el Sistema.

1.1.2.2.3.5.5. Descripción Global del Producto

El impacto tecnológico en las Empresas Comerciales debido a la necesidad de contar con información rápida, oportuna y confiable ocasiona la demanda de las nuevas tecnologías de información y comunicación. Por lo cual elaboramos el proyecto CONAL Sistema Automatizado de Control de Almacenes.

Los beneficios de la utilización de nuevas tecnologías de comunicación como ser el Internet nos aseguran la integridad, veracidad y la organización de nuestra información.

1.1.2.2.3.6. Especificaciones de Casos de Uso

Para los Casos de Uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se ejecuta una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, postcondiciones, flujo de eventos, requisitos no-funcionales asociados. También, para

casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

1.1.2.2.3.6.1. Introducción

Las Especificaciones de los Casos de Uso son una descripción detallada de los Casos de Uso del Sistema.

1.1.2.2.3.6.2. Propósito

Interpretar y describir los Casos de Uso.

1.1.2.2.3.6.3. Alcance

Describe los procesos internos de los Casos de Uso.

Detalla los flujos de los Casos de Uso según lo establecido por la organización.

1.1.2.2.3.6.4. Especificación de los Casos de Uso

1.1.2.2.3.6.4.1. Especificación de Caso de Uso Ingresar al Sistema

Caso de Uso:	Ingresar al Sistema.
Descripción: Permite ingresar al Sistema, este caso tiene como función controlar el acceso y al mismo tiempo recuperar los permisos correspondientes al momento que el Usuario introduzca su login y clave en el Sistema.	
Actores: SuperUsuario (Administrador Sistemas), Usuario1 (Jefe de Almacén, Encargado de Ventas) Usuario2 (Gerente General).	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	

<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor introduce datos de Usuario y clave en el Sistema. 2. El actor presiona botón Ingresar. 3. El Sistema valida sus datos. 4. El actor espera su validación. 5. Si los datos son correctos muestra la Pantalla Principal. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 3. Si el actor no llena el formulario de acceso, se muestra un mensaje donde se informa que los campos son requeridos. 5. Si el usuario no existe se muestra mensaje de error y vuelve a mostrar la pantalla índex.
<p>Pos condiciones: Ninguno</p>	

Tabla13. Especificación de Caso de Uso Ingresar al Sistema

1.1.2.2.3.6.4.2. Especificación de Casos de Uso Módulo Personal

1.1.2.2.3.6.4.2.1. Especificación Caso de Uso Administrar Personal

Caso de Uso:	Administrar Personal.
Descripción: Permite Listar, Adicionar, Modificar, Eliminar, Mostrar Kardex del Personal, así también Adicionar, Modificar, Eliminar Usuario.	
Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El Sistema muestra Listado del 	<p>Flujo Alternativo:</p>

Personal registrado en el Sistema. 2. El actor realiza la Búsqueda de Personal en el listado. 3. El Sistema muestra el Personal que cumple con los Criterios de Búsqueda.	
Pos condiciones: Ninguno	

Tabla 14. Especificación Caso de Uso Administrar Personal

1.1.2.2.3.6.4.2.2. Especificación Caso de Uso Adicionar Personal

Caso de Uso:	Adicionar Personal
Descripción: Permite registrar un nuevo Personal en el Sistema.	
Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal: 1. El actor presiona botón Adicionar. 2. El Sistema le muestra formulario. 3. El actor introduce datos del Personal en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que	Flujo Alternativo: 4. Selecciona botón Cancelar, se regresa a la pantalla Administrar Personal. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

son obligatorios estén introducidos.	
6. El Sistema guarda datos del formulario.	
7. El Sistema regresa a la pantalla de Administrar Personal.	
Pos condiciones: Ninguno	

Tabla 15. Especificación Caso de Uso Adicionar Personal

1.1.2.2.3.6.4.2.3. Especificación Caso de Uso Modificar Personal

Caso de Uso:	Modificar Personal.
Descripción: Permite modificar datos del Personal.	
Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Personal.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor selecciona al Personal y luego presiona el botón Modificar. 2. El Sistema le muestra el formulario con datos del Personal. 3. El actor realiza los cambios de datos del Personal en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 	Flujo Alternativo: <ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la pantalla Administrar Personal. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

<p>6. El Sistema actualiza los datos del formulario.</p> <p>7. El Sistema regresa a la pantalla de Administrar Personal.</p>	
<p>Pos condiciones: Ninguno</p>	

Tabla 16. Especificación Caso de Uso Modificar Personal

1.1.2.2.3.6.4.2.4. Especificación Caso de Uso Eliminar Personal

<p>Caso de Uso:</p>	<p>Eliminar Personal.</p>	
<p>Descripción: Permite eliminar del Sistema un Personal.</p>		
<p>Actores: SuperUsuario (Administrador Sistemas).</p>		
<p>Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Personal.</p>		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona al Personal que desea eliminar. 2. Selecciona botón Eliminar 3. El Sistema muestra el mensaje de confirmación. 4. Por SI, El Sistema realiza la eliminación del Personal. 5. El Sistema regresa a la pantalla de Administrar Personal. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 5. Por NO, El Sistema no realiza ninguna operación. 	

Pos condiciones: Ninguno.

Tabla 17. Especificación Caso de Uso Eliminar Personal

1.1.2.2.3.6.4.2.5. Especificación Caso de Uso Adicionar Usuario

Caso de Uso:	Adicionar Usuario	
Descripción: Permite darle privilegios al Personal de Usuario registrando un nuevo Usuario en el Sistema		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Personal		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor presiona ADDUSU en la fila del Personal. 2. El Sistema le muestra formulario de Usuario. 3. El actor introduce datos del Usuario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El Sistema guarda datos del formulario. 6. El Sistema regresa a la pantalla de Administrar Personal. 	<ol style="list-style-type: none"> 5. Selecciona botón Volver, se regresa a la pantalla Administrar Personal. 6. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta. 	

Pos condiciones: Ninguno.

Tabla 18. Especificación Caso de Uso Adicionar Usuario

1.1.2.2.3.6.4.2.6. Especificación Caso de Uso Modificar Usuario

Caso de Uso:	Modificar Usuario.	
Descripción: Permite modificar datos de Usuario.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Personal; Estar Registrado como Usuario.		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor presiona en MODUSU en la fila del Personal. 2. El Sistema le muestra formulario con datos del Usuario. 3. El actor realiza los cambios de datos del Usuario en el formulario. 4. Selecciona botón Modificar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El Sistema actualiza los datos del formulario. 7. El Sistema regresa a la pantalla de 	<ol style="list-style-type: none"> 3. Selecciona botón Volver, se regresa a la pantalla Administrar Personal. 4. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta. 	

Administrar Personal.	
Pos condiciones: Ninguno	

Tabla 19. Especificación Caso de Uso Modificar Usuario

1.1.2.2.3.6.4.2.7. Especificación Caso de Uso Eliminar Usuario

Caso de Uso:	Eliminar Usuario.	
Descripción: Permite eliminar del Sistema datos de Usuario.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Personal; Estar registrado como Usuario		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor presiona MODUSU en la fila del Personal. 2. El Sistema le muestra formulario con datos del Usuario. 3. Selecciona botón Eliminar 4. El Sistema regresa a la pantalla de Administrar Personal. 		
Pos condiciones: Ninguno		

Tabla 20. Especificación Caso de Uso Eliminar Usuario

1.1.2.2.3.6.4.2.8. Especificación Caso de Uso Mostrar Kardex Personal

Caso de Uso:	Mostrar Kardex Personal	
Descripción: Permite visualizar Kardex del Personal.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor selecciona el personal y luego presiona el botón Reporte. 2. El Sistema muestra el Kardex del Personal Seleccionado. 		
Pos condiciones:		

Tabla 21. Especificación Caso de Uso Mostrar Kardex Personal

1.1.2.2.3.6.4.2.9. Especificación Caso de Uso Administrar Roles

Caso de Uso:	Administrar Roles.	
Descripción: Permite Listar, Adicionar, Modificar, Eliminar de Roles.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal:	Flujo Alternativo:	

<ol style="list-style-type: none"> 1. El Sistema muestra Listado de Roles registrados en el Sistema. 2. El actor realiza la Búsqueda de Roles en el Listado. 3. El Sistema muestra los Roles que cumplen con los Criterios de Búsqueda. 	
<p>Pos condiciones: Ninguno</p>	

Tabla 22. Especificación Caso de Uso Administrar Roles

1.1.2.2.3.6.4.2.10. Especificación Caso de Uso Adicionar Roles

Caso de Uso:	Adicionar Roles.	
Descripción: Permite registrar nuevos Roles en el Sistema.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión.		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona botón Adicionar. 2. El Sistema le muestra formulario. 3. El actor introduce datos del Rol en el formulario y selecciona al menos un módulo para el mismo. 4. Selecciona botón Guardar. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la pantalla Administrar Roles. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta. 	

<p>5. Se valida datos del formulario que son obligatorios estén introducidos.</p> <p>6. El Sistema guarda datos del formulario.</p> <p>7. El Sistema regresa a la pantalla de Administrar Roles.</p>	
<p>Pos condiciones: Ninguno</p>	

Tabla 23. Especificación Caso de Uso Adicionar Roles

1.1.2.2.3.6.4.2.11. Especificación Caso de Uso Modificar Roles

Caso de Uso:	Modificar Roles.	
Descripción: Permite modificar datos del Rol en el Sistema.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Estar registrado el Rol en el Sistema.		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona el Rol y luego presiona Modificar. 2. El Sistema le muestra formulario con datos del Rol. 3. El actor realiza los cambios de datos del Rol en el formulario. 4. Selecciona botón Guardar. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la pantalla Administrar Roles. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta. 	

<p>5. Se valida datos del formulario que son obligatorios estén introducidos.</p> <p>6. El Sistema actualiza los datos del formulario.</p> <p>7. El Sistema regresa a la pantalla Administrar Roles</p>	
Pos condiciones: Ninguno	

Tabla 24. Especificación Caso de Uso Modificar Roles

1.1.2.2.3.6.4.2.12. Especificación Caso de Uso Eliminar Roles

Caso de Uso:	Eliminar Roles.	
Descripción: Permite eliminar del Sistema un Rol.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Rol.		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona el Rol que desea Eliminar. 2. Selecciona botón Eliminar 3. El Sistema muestra el mensaje de confirmación. 4. Por SI, El Sistema realiza la eliminación del Rol. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 3. Por NO. El Sistema no realiza ninguna operación. 	

5. El Sistema regresa a la pantalla de Administrar Roles.	
Pos condiciones: Ninguno	

Tabla 25. Especificación Caso de Uso Eliminar Roles

1.1.2.2.3.6.4.2.13. Especificación Caso de Uso Administrar Asignación y Reasignar de Cargo a Personal

Caso de Uso:	Administrar Asignación y Reasignar de Cargo a Personal	
Descripción: Permite Asignar y Reasignar Cargo al Personal en la Estructura Organizativa		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal: <ol style="list-style-type: none"> 1. El Sistema muestra una sub Pantalla con un Select para que seleccione la agencia donde realizará la Asignación y Reasignación de Cargos. 2. Presiona Botón Cargar 3. Se valida el campo select del formulario. 4. El Sistema muestra pantalla con el 	Flujo Alternativo: <ol style="list-style-type: none"> 3. Muestra Mensaje: Seleccione una Agencia 	

árbol de la Estructura Organizativa.	
Pos condiciones: Ninguno	

Tabla 26. Especificación Caso de Uso Administrar Asignación y Reasignar de Cargo a Personal

1.1.2.2.3.6.4.2.14. Especificación Caso de Uso Asignar Cargo

Caso de Uso:	Administrar Asignar Cargo
Descripción: Permite Asignar al Personal un Cargo dentro la Estructura Organizativa	
Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión, estar registro el Personal en el Sistema.	
Flujo Normal:	Flujo Alternativo:
<ol style="list-style-type: none"> 1. Selecciona en el Cargo el link Asignar 2. El Sistema muestra la Lista del Personal a ser Asignados 3. El actor realiza la Búsqueda del Personal en el Listado. 4. El actor selecciona el Personal que quiere asignar a ese Cargo 5. Presiona Botón Asignar 6. El Sistema carga el Árbol de la Estructura Organizativa de la 	

Agencia en con el Personal recién asignados en el Cargo	
Pos condiciones: Ninguno	

Tabla 27. Especificación Caso de Uso Asignar Cargo

1.1.2.2.3.6.4.2.15. Especificación Caso de Uso Reasignar Cargo

Caso de Uso:	Administrar Reasignar Cargo
Descripción: Permite Reasignar al Personal Cargos	
Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal: <ol style="list-style-type: none"> 1. Selecciona en el Cargo el Link Reasignar 2. El Sistema muestra formulario de Reasignar Personal 3. Presiona Botón Aceptar 4. El Sistema carga el Árbol de la Estructura Organizativa con el Personal Reasignado a otro Cargo de la Agencia 	Flujo Alternativo: <ol style="list-style-type: none"> 3. Presiona Botón Cancelar: El Sistema no realiza ningún cambio.
Pos condiciones: Ninguno	

Tabla 28. Especificación Caso de Uso Reasignar Cargo

1.1.2.2.3.6.4.2.16. Especificación Caso de Uso Administrar Agencia

Caso de Uso:	Administrar Agencia.	
Descripción: Permite Listar, Adicionar, Modificar, Eliminar Agencias.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El Sistema muestra Listado de las Agencias registradas en el Sistema. 2. El actor realiza la Búsqueda de Agencia en el listado. 3. El Sistema muestra el Personal que cumple con los Criterios de Búsqueda. 	<ul style="list-style-type: none"> • 	
Pos condiciones: Ninguno		

Tabla 29. Especificación Caso de Uso Administrar Agencia

1.1.2.2.3.6.4.2.17. Especificación Caso de Uso Adicionar Agencia

Caso de Uso:	Adicionar Agencia.	
Descripción: Permite registrar nuevas Agencias en el Sistema.		

Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona botón Adicionar. 2. El Sistema le muestra formulario. 3. El actor introduce datos de la Agencia en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El Sistema guarda datos del formulario. 7. El Sistema regresa a la pantalla de Administrar Agencia. 	Flujo Alternativo: <ol style="list-style-type: none"> 4. Selecciona botón Volver, se regresa a la pantalla Administrar Agencia. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones:	

Tabla 30. Especificación Caso de Uso Adicionar Agencia

1.1.2.2.3.6.4.2.18. Especificación Caso de Uso Modificar Agencia

Caso de Uso:	Modificar Agencia.
Descripción: Permite modificar datos de la Agencia en el Sistema.	
Actores: SuperUsuario (Administrador Sistemas).	

Precondiciones: Actor haber iniciado sesión; Estar registrada la Agencia en el Sistema..	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona en Editar en la fila de la Agencia que quiere Modificar. 2. El Sistema le muestra formulario con datos de la Agencia. 3. El actor realiza los cambios de datos de la Agencia en el formulario. 4. Selecciona botón Modificar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El Sistema actualiza los datos del formulario. 7. El Sistema regresa a la pantalla de Administrar Agencia. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. Selecciona botón Volver, se regresa a la pantalla Administrar Agencia. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: Ninguno	

Tabla 31. Especificación Caso de Uso Modificar Agencia

1.1.2.2.3.6.4.2.19. Especificación Caso de Uso Eliminar Agencia

Caso de Uso:	Eliminar Agencia.
Descripción: Permite eliminar del Sistema una Agencia.	

Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión; Estar registrada en el Sistema la Agencia.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona Editar en la fila de la Agencia que quiere Eliminar. 2. El Sistema le muestra formulario con datos de la Agencia. 3. Selecciona botón Eliminar 4. El Sistema muestra el mensaje de confirmación. 5. Por Aceptar, El Sistema realiza la eliminación de la Agencia. <ol style="list-style-type: none"> 1. El Sistema regresa a la pantalla de Administrar Agencia. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 3. Por Cancelar. El Sistema no realiza ninguna operación.
Pos condiciones: Ninguno.	

Tabla 32. Especificación Caso de Uso Eliminar Agencia

1.1.2.2.3.6.4.2.20. Especificación Caso de Uso Administrar Estructura Organizativa

Caso de Uso:	Administrar Estructura Organizativa.
Descripción: Lista un árbol de la Estructura Organizativa de Agencia con sus Áreas y Cargos, Adicionar, Modificar, Eliminar Área, Adicionar, Modificar, Eliminar Cargo.	

Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal: 1. El actor presiona en el Árbol de Departamentos la Agencia. 2. El Sistema carga la Agencia con su Estructura Organizativa.	Flujo Alternativo:
Pos condiciones:	

Tabla 33. Especificación Caso de Uso Administrar Estructura Organizativa

1.1.2.2.3.6.4.2.21. Especificación Caso de Uso Adicionar Área

Caso de Uso:	Adicionar Área.
Descripción: Permite registrar nueva Área en la Estructura Organizativa de la Agencia.	
Actores: SuperUsuario (Administrador Sistemas).	
Precondiciones: Actor haber iniciado sesión, Seleccionar la Agencia.	
Flujo Normal: 1. El actor presiona en ADD en la Agencia. 2. El Sistema le muestra formulario. 3. El actor introduce datos del Área	Flujo Alternativo: 4. El selecciona Botón Cancelar entonces Elimina Formulario 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la

<p>en el formulario.</p> <p>4. Selecciona botón Guardar.</p> <p>5. Se valida datos del formulario que son obligatorios estén introducidos.</p> <p>6. El Sistema guarda datos del formulario.</p> <p>7. El Sistema Carga la Agencia Con la Nueva Área</p>	<p>información que falta.</p>
<p>Pos condiciones: Ninguno</p>	

Tabla 34. Especificación Caso de Uso Adicionar Área

1.1.2.2.3.6.4.2.22. Especificación Caso de Uso Modificar Área

<p>Caso de Uso:</p>	<p>Modificar Área</p>	
<p>Descripción: Permite modificar datos de la Área en la Estructura Organizativa.</p>		
<p>Actores: SuperUsuario (Administrador Sistemas).</p>		
<p>Precondiciones: Actor haber iniciado sesión; Selecciona la Agencia; Estar registrado en el Sistema el Área.</p>		
<p>Flujo Normal:</p> <p>1. El actor presiona en MOD en el Área.</p> <p>2. El Sistema le muestra formulario con datos del Área.</p>	<p>Flujo Alternativo:</p> <p>4. Selecciona botón Cancelar entonces el Sistema no realiza ninguna operación y elimina el formulario.</p>	

<ol style="list-style-type: none"> 3. El actor introduce datos del Área en el formulario. 4. Selecciona botón Modificar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El Sistema actualiza los datos del formulario. 7. El Sistema Carga la Agencia Con los datos Modificados de la Área. 	
<p>Pos condiciones: Ninguno.</p>	

Tabla 35. Especificación Caso de Uso Modificar Área

1.1.2.2.3.6.4.2.23. Especificación Caso de Uso Eliminar Área

Caso de Uso:	Eliminar Área.	
Descripción: Permite Eliminar del Sistema un Área.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Selecciona la Agencia; Estar registrado en el Sistema el Área.		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona en MOD en el Área. 2. El Sistema le muestra formulario con 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 5. Por Cancelar, El Sistema no Realiza ninguna Operación 	

<p>datos del Área.</p> <ol style="list-style-type: none"> 3. Selecciona botón Eliminar. 4. El Sistema muestra el mensaje de confirmación. 5. Por Aceptar, El Sistema realiza la eliminación del Área. 6. El Sistema carga la Agencia sin el área eliminada. 	
<p>Pos condiciones:</p>	

Tabla 36. Especificación Caso de Uso Eliminar Área

1.1.2.2.3.6.4.2.24. Especificación Caso de Uso Adicionar Cargo

Caso de Uso:	Adicionar Cargo.	
Descripción: Permite registrar un nuevo Cargo en la Área de la Agencia.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión, Seleccionar una Área.		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona en ADD en la Área. 2. El Sistema le muestra formulario. 3. El actor introduce datos del Cargo 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. El selecciona Botón Cancelar entonces Elimina Formulario 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la 	

<p>en el formulario.</p> <p>4. Selecciona botón Guardar.</p> <p>5. Se valida datos del formulario que son obligatorios estén introducidos.</p> <p>6. El Sistema guarda datos del formulario.</p> <p>7. El Sistema Carga la Área con el nuevo Cargo</p>	<p>información que falta.</p>
<p>Pos condiciones: Ninguno</p>	

Tabla 37. Especificación Caso de Uso Adicionar Cargo

1.1.2.2.3.6.4.2.25. Especificación Caso de Uso Modificar Cargo

Caso de Uso:	Modificar Cargo	
Descripción: Permite modificar datos del Cargo en la Área.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Selecciona el Área; Estar registrado en el Sistema el Cargo.		
<p>Flujo Normal:</p> <p>1. El actor presiona en MOD en el Cargo.</p> <p>2. El Sistema le muestra formulario con datos del Cargo.</p> <p>3. El actor introduce datos del Cargo en</p>	<p>Flujo Alternativo:</p> <p>4. Selecciona botón Cancelar entonces el Sistema no realiza ninguna operación y elimina el formulario.</p>	

<p>el formulario.</p> <p>4. Selecciona botón Modificar.</p> <p>5. Se valida datos del formulario que son obligatorios estén introducidos.</p> <p>6. El Sistema actualiza los datos del formulario.</p> <p>7. El Sistema Carga la Área Con los datos Modificados del Cargo.</p>	
<p>Pos condiciones: Ninguno.</p>	

Tabla 38. Especificación Caso de Uso Modificar Cargo

1.1.2.2.3.6.4.2.26. Especificación Caso de Uso Eliminar Cargo

Caso de Uso:	Eliminar Cargo.	
Descripción: Permite Eliminar del Sistema un Cargo.		
Actores: SuperUsuario (Administrador Sistemas).		
Precondiciones: Actor haber iniciado sesión; Selecciona la Área; Estar registrado en el Sistema el Cargo.		
Flujo Normal:		Flujo Alternativo:
<p>1. El actor presiona en MOD en el Cargo.</p> <p>2. El Sistema le muestra formulario con datos del Cargo.</p>		<p>3. Por Cancelar, El Sistema no Realiza ninguna Operación</p>

<ol style="list-style-type: none"> 3. Selecciona botón Eliminar. 4. El Sistema muestra el mensaje de confirmación. 5. Por Aceptar, El Sistema realiza la eliminación del Cargo. 6. El Sistema carga la Área sin el Cargo. 	
<p>Pos condiciones:</p>	

Tabla 39. Especificación Caso de Uso Eliminar Cargo

1.1.2.2.3.6.4.2.27. Especificación Caso de Uso Administrar Proveedor

Caso de Uso:	Administrar Proveedor.	
Descripción: Permite Listar, Adicionar, Modificar, Eliminar Proveedor.		
Actores: SuperUsuario (Administrador Sistemas)		
Precondiciones: Actor haber iniciado sesión.		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El Sistema muestra Listado de los Proveedores de los Productos, registrados en el Sistema. 2. El actor realiza la Búsqueda de Proveedor en el listado. 3. El Sistema muestra el Proveedor 	<p>Flujo Alternativo:</p>	

que cumple con los Criterios de Búsqueda.	
Pos condiciones: Ninguno	

Tabla 40. Especificación Caso de Uso Administrar Proveedor

1.1.2.2.3.6.4.2.28. Especificación Caso de Uso Adicionar Proveedor

Caso de Uso:	Adicionar Proveedor
Descripción: Permite registrar un nuevo Proveedor en el Sistema.	
Actores: SuperUsuario (Administrador Sistemas)	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona botón Adicionar. 2. El Sistema le muestra formulario. 3. El actor introduce datos del Proveedor en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El Sistema guarda datos del formulario. 7. El Sistema regresa a la pantalla de 	Flujo Alternativo: <ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la pantalla Administrar Proveedor. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

Administrar Proveedor.	
Pos condiciones: Ninguno	

Tabla 41. Especificación Caso de Uso Adicionar Proveedor

1.1.2.2.3.6.4.2.29. Especificación Caso de Uso Modificar Proveedor

Caso de Uso:	Modificar Proveedor.
Descripción: Permite modificar datos del Proveedor.	
Actores: SuperUsuario (Administrador Sistemas)	
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Proveedor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor selecciona el Proveedor y luego presiona el botón Modificar. 2. El Sistema le muestra formulario con datos del Proveedor. 3. El actor realiza los cambios de datos del Proveedor en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que 	Flujo Alternativo: <ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la pantalla Administrar Proveedor. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

<p>son obligatorios estén introducidos.</p> <p>6. El Sistema actualiza los datos del formulario.</p> <p>7. El Sistema regresa a la pantalla de Administrar Proveedor.</p>	
Pos condiciones: Ninguno	

Tabla 42. Especificación Caso de Uso Modificar Proveedor

1.1.2.2.3.6.4.2.30. Especificación Caso de Uso Eliminar Proveedor

Caso de Uso:	Eliminar Proveedor.
Descripción: Permite eliminar del Sistema un Proveedor.	
Actores: SuperUsuario (Administrador Sistemas)	
Precondiciones: Actor haber iniciado sesión; Estar registrado en el Sistema el Proveedor.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona el Proveedor. 2. Selecciona botón Eliminar 3. El Sistema muestra el mensaje de confirmación. 4. Por SI, El Sistema realiza la eliminación del Proveedor. 5. El Sistema regresa a la pantalla de Administrar Proveedor. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. Por NO. El Sistema no realiza ninguna operación.

Pos condiciones: Ninguno.

Tabla 43. Especificación Caso de Uso Eliminar Proveedor

1.1.2.2.3.6.4.3.Especificación de Casos de Uso Módulo Almacén

1.1.2.2.3.6.4.3.1. Especificación de Caso de Uso Administrar Productos

Caso de Uso:	Administrar Productos.
Descripción: Permite Listar Productos, Adicionar Producto, Modificar Producto, Eliminar Producto y Mostrar Kardex del Producto.	
Actores: Usuario1 (Jefe de Almacén)	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:
<ol style="list-style-type: none"> 1. El Sistema muestra el árbol de Clases de Almacén. 2. El actor selecciona una clase de Almacén. 3. El Sistema muestra el Tipo de Almacén que existe dentro de cada Clase de Almacén. 4. El actor selecciona un Tipo de Almacén. 5. El Sistema muestra un Listado de la Categoría de Productos que pertenecen al Tipo de Almacén. 	

<p>6. El actor realiza la Búsqueda de Productos en el listado.</p> <p>7. El sistema muestra el Producto que cumple con los criterios de Búsqueda.</p>	
<p>Pos condiciones: Ninguno.</p>	

Tabla 44. Especificación de Caso de Uso Administrar Productos

1.1.2.2.3.6.4.3.2. Especificación de Caso de Uso Adicionar Producto

<p>Caso de Uso:</p>	<p>Adicionar Producto.</p>	
<p>Descripción: Permite Registrar un nuevo Producto en el Sistema.</p>		
<p>Actores: Usuario1 (Jefe de Almacén)</p>		
<p>Precondiciones: Actor haber iniciado sesión.</p>		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona la Categoría. 2. El actor presiona el botón Adicionar. 3. El sistema le muestra el formulario. 4. El actor introduce datos del Producto en el formulario. 5. Selecciona botón Guardar. 6. Se valida que datos del formulario que son obligatorios estén introducidos. 7. El sistema guarda datos del 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 5. Selecciona botón Cancelar y regresa a la pantalla Administrar Productos. 6. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta. 	

formulario.	
8. El Sistema regresa a la Pantalla de Administrar Productos.	
Pos condiciones: Ninguno.	

Tabla 45. Especificación de Caso de Uso Adicionar Producto

1.1.2.2.3.6.4.3.3. Especificación de Caso de Uso Modificar Producto

Caso de Uso:	Modificar Producto.
Descripción: Permite Modificar datos del Producto.	
Actores: Usuario1 (Jefe de Almacén)	
Precondiciones: Actor haber iniciado sesión. Producto estar registrado en el sistema.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona el Producto que desea modificar. 2. El actor presiona el botón Modificar. 3. El sistema le muestra el formulario con datos del Producto. 4. El actor realiza los cambios en los datos del Producto en el formulario. 5. Selecciona botón Guardar. 6. Se valida que datos del formulario que son obligatorios estén introducidos. 7. El sistema actualiza los datos del 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 5. Selecciona botón Cancelar y regresa a la pantalla Administrar Productos. 6. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

formulario.	
8. El Sistema regresa a la Pantalla de Administrar Productos.	
Pos condiciones: Ninguno.	

Tabla 46. Especificación de Caso de Uso Modificar Producto

1.1.2.2.3.6.4.3.4. Especificación de Caso de Uso Eliminar Producto

Caso de Uso:	Eliminar Producto.	
Descripción: Permite Eliminar un Producto del sistema.		
Actores: Usuario1 (Jefe de Almacén)		
Precondiciones: Actor haber iniciado sesión. Producto estar registrado en el sistema.		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor selecciona el Producto que desea eliminar. 2. El actor presiona el botón Eliminar. 3. El sistema muestra el mensaje de confirmación. 4. Por SI, el sistema realiza la eliminación del Producto. 5. El sistema regresa a la pantalla de Administrar Productos. 	<ol style="list-style-type: none"> 4. Por NO, el sistema no realiza ninguna operación. 	
Pos condiciones: Ninguno.		

Tabla 47. Especificación de Caso de Uso Eliminar Producto

1.1.2.2.3.6.4.3.5. Especificación de Caso Uso Administrar Tipo y Categoría de Almacén

Caso de Uso:	Administrar Tipo y Categoría de Almacén.	
<p>Descripción: Permite Listar los Tipos y Categorías de Almacenes asi también permite, Adicionar Tipo de Almacén, Modificar y Eliminar Tipo de Almacén. Asi mismo permite Adicionar, Modificar y Eliminar Categoría de Almacén.</p>		
<p>Actores: Usuario1 (Jefe de Almacén)</p>		
<p>Precondiciones: Actor haber iniciado sesión.</p>		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El Sistema muestra el árbol de Clases de Almacén. 2. El actor selecciona una clase de Almacén. 3. El Sistema muestra el Tipo de Almacén que existe dentro de cada Clase de Almacén. 4. El actor selecciona un Tipo de Almacén. 5. El Sistema muestra un Listado de la Categoría de Productos que pertenecen al Tipo de Almacén. 6. El actor realiza la Búsqueda de Productos en el listado. 7. El sistema muestra el Producto que cumple con los criterios de 	<p>Flujo Alternativo:</p>	

Búsqueda.	
Pos condiciones: Ninguno.	

Tabla 48. Especificación de Caso de Uso Administrar Tipo y Categoría de Almacén

1.1.2.2.3.6.4.3.6. Especificación de Caso de Uso Administrar Ingreso de Producto

Caso de Uso:	Administrar Ingreso de Producto.
Descripción: Permite Listar los Ingresos de Productos, así también permite Adicionar y Eliminar un Ingreso de Producto	
Actores: Usuario1 (Jefe de Almacén)	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:
<ol style="list-style-type: none"> 1. El usuario selecciona la opción Administrar Ingreso de Producto. 2. El Sistema muestra un Listado con todos los Ingresos de Productos. 3. El actor realiza la Búsqueda de Ingresos de Productos en el listado. 4. El sistema muestra el Ingreso que cumple con los criterios de Búsqueda. 	

Pos condiciones: Ninguno.

Tabla 49. Especificación de Caso de Uso Administrar Ingresos de Productos

1.1.2.2.3.6.4.3.7. Especificación de Caso de Uso Administrar Salida de Producto

Caso de Uso:	Administrar Salida de Producto.
Descripción: Permite Listar las Salidas de Productos, así también permite Adicionar y Eliminar una Salida de Producto	
Actores: Usuario1 (Jefe de Almacén)	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:
<ol style="list-style-type: none"> 1. El usuario selecciona la opción Administrar Salida de Producto. 2. El Sistema muestra un Listado con todas las Salidas de Productos. 3. El actor realiza la Búsqueda de Ingresos de Productos en el listado. 4. El sistema muestra el Ingreso que cumple con los criterios de Búsqueda. 	
Pos condiciones: Ninguno.	

Tabla 50. Especificación de Caso de Uso Administrar Salida de Productos

1.1.2.2.3.6.4.4. Especificación de Casos de Uso Módulo

Reportes

1.1.2.2.3.6.4.4.1. Especificación de Caso de Uso Reporte de Productos en Almacenes

Caso de Uso:	Generar Reporte de Productos en Almacenes.	
Descripción: El Reporte de Productos en Almacenes es un listado de todos los Productos de la Empresa con su respectivo Kardex incluido su inventario, este reporte esta en formato .pdf.		
Actores: SuperUsuario(Encargado de Sistemas),Usuario1 (Jefe de Almacén, Encargado de Ventas), Usuario2 (Gerente General).		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor selecciona Reporte de Productos en Almacenes del menú principal del Módulo Reportes. 2. El Sistema muestra el reporte. 	<ul style="list-style-type: none"> • 	
Pos condiciones: Ninguno		

Tabla 51. Especificación de Caso de Uso Reporte de Productos en Almacenes

1.1.2.2.3.6.4.4.2. Especificación de Caso de Uso Reporte de Productos por Proveedor

Caso de Uso:	Generar Reporte de Productos por Proveedor.	
Descripción: El Reporte de Productos por Proveedor es un listado de todos los Productos que ofrece un determinado Proveedor con sus respectivos datos, este reporte esta en		

formato .pdf.	
Actores: SuperUsuario(Encargado de Sistemas),Usuario1 (Jefe de Almacén, Encargado de Ventas), Usuario2 (Gerente General).	
Precondiciones: Actor haber iniciado sesión.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona Reporte de Productos por Proveedor del menú principal del Módulo Reportes. 2. El Sistema muestra el Listado de Proveedores 3. El actor selecciona un Proveedor 4. El Sistema muestra el reporte. 	<p>Flujo Alternativo:</p> <ul style="list-style-type: none"> •
Pos condiciones: Ninguno	

Tabla 52. Especificación de Caso de Uso Reporte de Productos por Proveedor

1.1.2.2.3.6.4.4.3. Especificación de Caso de Uso Reporte de Productos por Agencia

Caso de Uso:	Generar Reporte de Productos por Agencia.
Descripción: El Reporte de Productos por Agencia es un listado de todos los Productos que tiene en Almacén una Agencia, este reporte esta en formato .pdf.	
Actores: SuperUsuario(Encargado de Sistemas),Usuario1 (Jefe de Almacén, Encargado de Ventas), Usuario2 (Gerente General).	

Precondiciones: Actor haber iniciado sesión.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor selecciona Reporte de Productos por Agencia del menú principal del Módulo Reportes. 2. El Sistema muestra el Listado de las Agencias. 3. El actor selecciona una Agencia 4. El Sistema muestra el reporte. 	<p>Flujo Alternativo:</p> <ul style="list-style-type: none"> •
Pos condiciones: Ninguno	

Tabla 53. Especificación de Caso de Uso Reporte de Productos por Agencia

1.1.2.2.3.6.4.4.4. Especificación de Caso de Uso Reporte de Ingresos por Agencia

Caso de Uso:	Generar Reporte de Ingresos por Agencia.
<p>Descripción: El Reporte de Ingresos por Agencia es un listado de todos los Ingresos de Productos de Productos que entran en Almacenes con sus respectivos detalles, este reporte esta en formato .pdf.</p>	
<p>Actores: SuperUsuario(Encargado de Sistemas), Usuario1 (Jefe de Almacén, Encargado de Ventas), Usuario2 (Gerente General).</p>	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:

<ol style="list-style-type: none"> 1. El actor selecciona Reporte de Ingresos por Agencia del menú principal del Módulo Reportes. 2. El Sistema muestra el Listado de las Agencias. 3. El actor selecciona una Agencia 4. El Sistema muestra el reporte. 	<ul style="list-style-type: none"> •
Pos condiciones: Ninguno	

Tabla 54. Especificación de Caso de Uso Reporte de Ingresos por Agencia

1.1.2.2.3.6.4.4.5. Especificación de Caso de Uso Reporte de Salidas por Agencia

Caso de Uso:	Generar Reporte de Salidas por Agencia.	
Descripción: El Reporte de Salidas por Agencia es un listado de todas las Salidas de Productos de Productos hacia una Agencia con sus respectivos detalles, este reporte esta en formato .pdf.		
Actores: SuperUsuario(Encargado de Sistemas), Usuario1 (Jefe de Almacén, Encargado de Ventas), Usuario2 (Gerente General).		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal: <ol style="list-style-type: none"> 1. El actor selecciona Reporte de Salidas por Agencia del menú principal del Módulo Reportes. 2. El Sistema muestra el Listado de 	Flujo Alternativo: <ul style="list-style-type: none"> • 	

las Agencias. 3. El actor selecciona una Agencia 4. El Sistema muestra el reporte.	
Pos condiciones: Ninguno	

Tabla 55. Especificación de Caso de Uso Reporte de Salidas por Agencia

1.1.2.2.3.6.4.4.6. Especificación de Caso de Uso Reporte de Productos por Marca

Caso de Uso:	Generar Reporte de Productos por Marca.
Descripción: El Reporte de Productos por Marca es un listado de todos los Productos de una Marca específica con sus respectivos datos, este reporte esta en formato .pdf.	
Actores: SuperUsuario(Encargado de Sistemas), Usuario1 (Jefe de Almacén, Encargado de Ventas), Usuario2 (Gerente General).	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal: 5. El actor selecciona Reporte de Productos por Marca del menú principal del Módulo Reportes. 6. El Sistema muestra el Listado de las Marcas. 7. El actor selecciona una Marca 8. El Sistema muestra el reporte.	Flujo Alternativo: <ul style="list-style-type: none"> •

Pos condiciones: Ninguno

Tabla 56. Especificación de Caso de Uso Reporte de Productos por Marca

1.1.2.2.3.6.4.5. Especificación de Casos de Uso Módulo Backup

1.1.2.2.3.6.4.5.1. Especificación de Caso de Uso Administrar Backup

Caso de Uso:	Administrar Backup.
Descripción: Permite Listar, Resguardar, Restaurar, Eliminar y Generar Reporte PDF de Backup.	
Actores: Super Usuario(Administrador)	
Precondiciones: Actor haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:
<ol style="list-style-type: none"> 1. El Sistema muestra Listado de Backup's registrados en el sistema. 2. El actor realiza la búsqueda de Backup en el Listado. 3. El sistema muestra el (los) Backup(s) que cumple(n) los criterios de búsqueda. 	
Pos condiciones: Ninguno.	

Tabla 57. Especificación de Caso de Uso Administrar Backup

1.1.2.2.3.6.4.5.2. Especificación de Caso de Uso Resguardar Backup

Caso de Uso:	Resguardar Backup.	
Descripción: Permite Resguardar una nueva Backup en el sistema.		
Actores: Super Usuario(Administrador)		
Precondiciones: Actor haber iniciado sesión.		
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor presiona el botón Resguardar. 2. El sistema muestra el formulario. 3. El actor introduce datos del Backup en el formulario. 4. Selecciona botón Resguardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la Pantalla Administrar Backup. 	<ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la Pantalla Administrar Backup. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta. 	
Pos condiciones: Ninguno.		

Tabla 58. Especificación de Caso de Uso Resguardar Backup

1.1.2.2.3.6.4.5.3. Especificación de Caso de Uso Restaurar Backup

Caso de Uso:	Restaurar Backup.
---------------------	-------------------

Descripción: Permite Restaurar un Backup en el sistema.	
Actores: Super Usuario(Administrador)	
Precondiciones: Actor haber iniciado sesión. Estar registrado en el sistema el Backup.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Restaurar. 2. El sistema muestra el formulario con datos del Backup. 3. El actor realiza la introducción de datos del Backup en el formulario. 4. Selecciona botón Restaurar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la Pantalla Administrar Backup. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. Selecciona botón Cancelar, se regresa a la Pantalla Administrar Backup. 5. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: Ninguno.	

Tabla 59. Especificación de Caso de Uso Restaurar Backup

1.1.2.2.3.6.4.5.4. Especificación de Caso de Uso Eliminar Backup

Caso de Uso:	Eliminar Backup.
Descripción: Permite Eliminar del sistema una Backup.	

Actores: Super Usuario(Administrador)	
Precondiciones: Actor haber iniciado sesión. Estar registrada en el sistema una Backup.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por SI, el Sistema realiza la eliminación del Backup. 4. El Sistema Regresa a la Pantalla Administrar Backup. 	Flujo Alternativo: <ol style="list-style-type: none"> 3. Por NO. El sistema no realiza ninguna operación.
Pos condiciones: Ninguno.	

Tabla 60. Especificación de Caso de Uso Eliminar Backup

1.1.2.2.3.7.Especificaciones Adicionales

Este documento capturará todos los requisitos que no han sido incluidos como parte de los Casos de Uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: Sistema operativo, requisitos de compatibilidad, etc. **Ver Anexo I: Especificación de Requisitos de Software Norma IEEE835**

1.1.2.2.3.8.Prototipos de Interfaces de Usuario

1.1.2.2.3.8.1. Introducción

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el Sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del Sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos,

siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la Fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la Fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

1.1.2.2.3.8.2. Propósito

Presentar los prototipos de pantallas para que el usuario tenga una idea de la interfaz que se presentaran en el Sistema.

1.1.2.2.3.8.3. Alcance

Mostrar los Prototipos de Pantallas, sujeto a modificaciones a lo largo del desarrollo del Sistema.

1.1.2.2.3.8.4. Prototipo de Pantallas

1.1.2.2.3.8.4.1. Pantalla Ingresar al Sistema

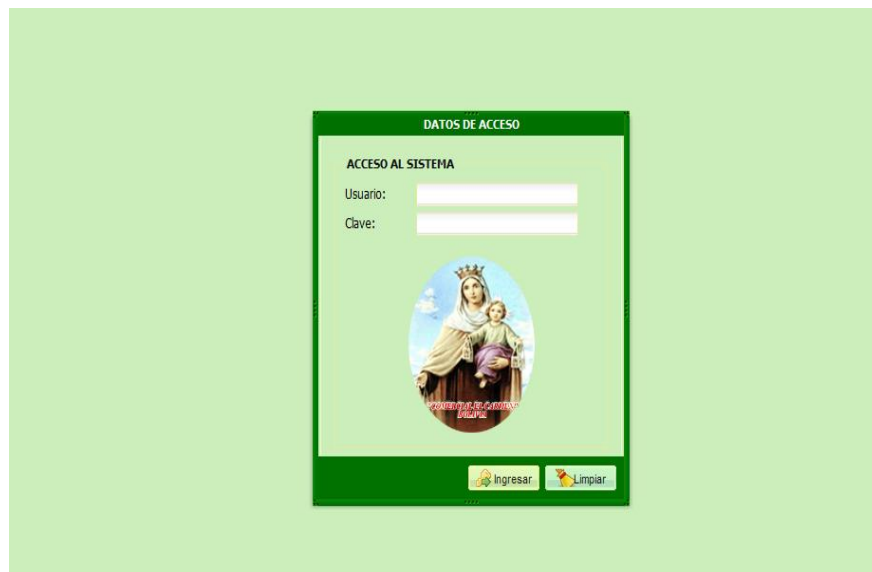


Figura 49. Pantalla Ingresar al Sistema

1.1.2.2.3.8.4.2. Pantalla Principal

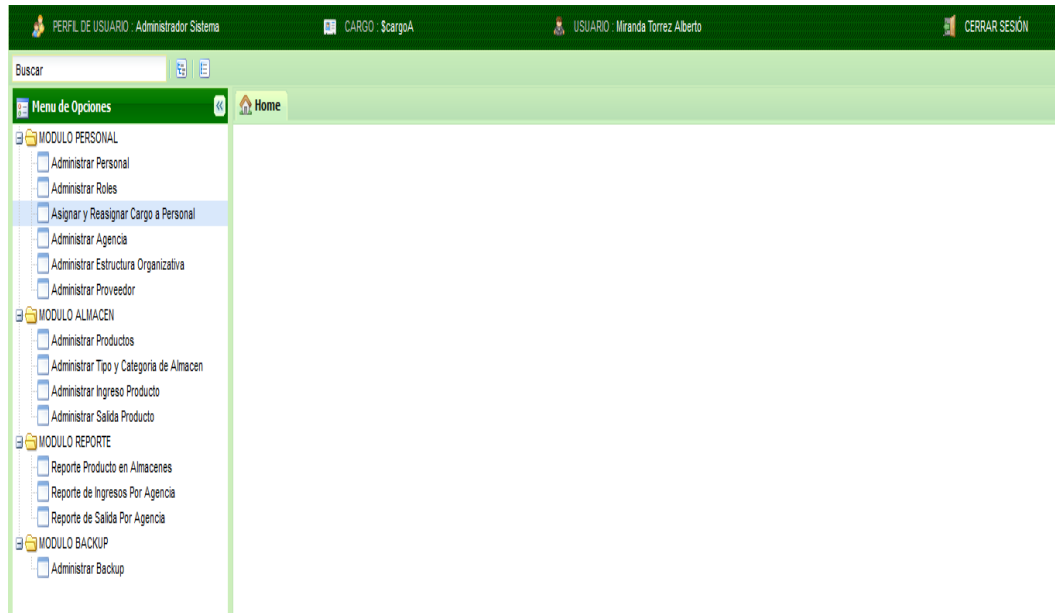


Figura 50. Pantalla Principal

1.1.2.2.3.8.4.3. Pantalla Administrar Personal

The screenshot shows the 'Administrar Personal' screen. At the top, there is a header with 'Home' and 'Administrar Personal'. Below the header is a green bar with the title 'LISTA DE PERSONAL'. Underneath, there are buttons for 'ADICIONAR', 'MODIFICAR', 'ELIMINAR', and 'KARDEX', along with a search bar labeled 'BUSCAR'. The main content is a table with the following data:

CODIGO	CEDULA	NOMBRE	DIRECCIÓN	TELEFONOS	ASIGNAR USUARIO	
1	7	23456	ertyui fghjk bnm	Z/ 2345 C/ rtyu N° 4567	Telf: 234567 - Cel: 3456789	ADDUSU <input type="checkbox"/>
2	6	761313	noemi portal rivera	Z/ sdfhjs C/ hjhj N° 4567	Telf: 23456 - Cel: 23456789	ADDUSU <input type="checkbox"/>
3	2	4154778	Fabio Barrios Calizaya	Z/ Chapacos C/ Heroes del Independencia N° 12	Telf: 6674875 - Cel: 71847854	MODUSU <input type="checkbox"/>
4	1	6623322	Alberto Miranda Torrez	Z/ Tejar C/ Heriberto Trigo N° 7845245	Telf: 6943635 - Cel: 71812886	MODUSU <input type="checkbox"/>



At the bottom of the screen, there is a footer with navigation icons and the text 'Página 1 de 1' and 'Mostrando 1 - 4 de 4'.


Figura 51. Pantalla Administrar Personal

1.1.2.2.3.8.4.4. Pantalla Formulario (ABM) Personal


ADICIONAR PERSONAL

DATOS PERSONALES

Nro. CI:  Fecha de Nac.: 

Nombres: 

Ap. Paterno:

Ap. Materno: 

DATOS DE UBICACION

Zona: Nro Casa:

Calle:

DATOS DE CONTACTO

Telefono Celular: Telefono Fijo:



 

Figura 52. Pantalla Formulario (ABM) Personal

1.1.2.2.3.8.4.5. Pantalla (ABM) Usuario

Asignar Contraseña y Rol

Formulario Usuario

Usuario :

Clave :

Clave Verificar :

ID	Rol
2	<input type="checkbox"/> Encargado de Almacen
1	<input type="checkbox"/> Administrador Sistema


 

Figura 53. Pantalla (ABM) Usuario

1.1.2.2.3.8.4.6. Pantalla Administrar Roles

CODIGO	NOMBRE	DESCRIPCIÓN
2	Jefe de Almacen	Todos los modulos para administrar el almacen
1	Administrador Sistema	Encargado de Administrar todo el Sistema

Figura 54. Pantalla Administrar Roles

1.1.2.2.3.8.4.7. Pantalla (ABM) Roles

DATOS DEL ROL

Nombre: Descripción:

DATOS DE SELECCION ACCESO A LOS MODULOS DEL ROL

ID	NOMBRE DEL MODULO	DESCRIPCIÓN
1 24	\$modulo.nombre	\$modulo.descripcion
2 23	fdsf	\$modulo.descripcion
3 22	sdfds	\$modulo.descripcion
4 21	sdf	\$modulo.descripcion
5 20	fsdfds	\$modulo.descripcion
6 19	dsf	\$modulo.descripcion
7 18	dsf	\$modulo.descripcion
8 17	dsfds	\$modulo.descripcion

Página 1 de 1 Mostrando 1 - 24 de 24

Guardar Cancelar

Figura 55. Pantalla (ABM) Roles

1.1.2.2.3.8.4.8. Pantalla Administrar Asignación y Reasignar Cargo a Personal

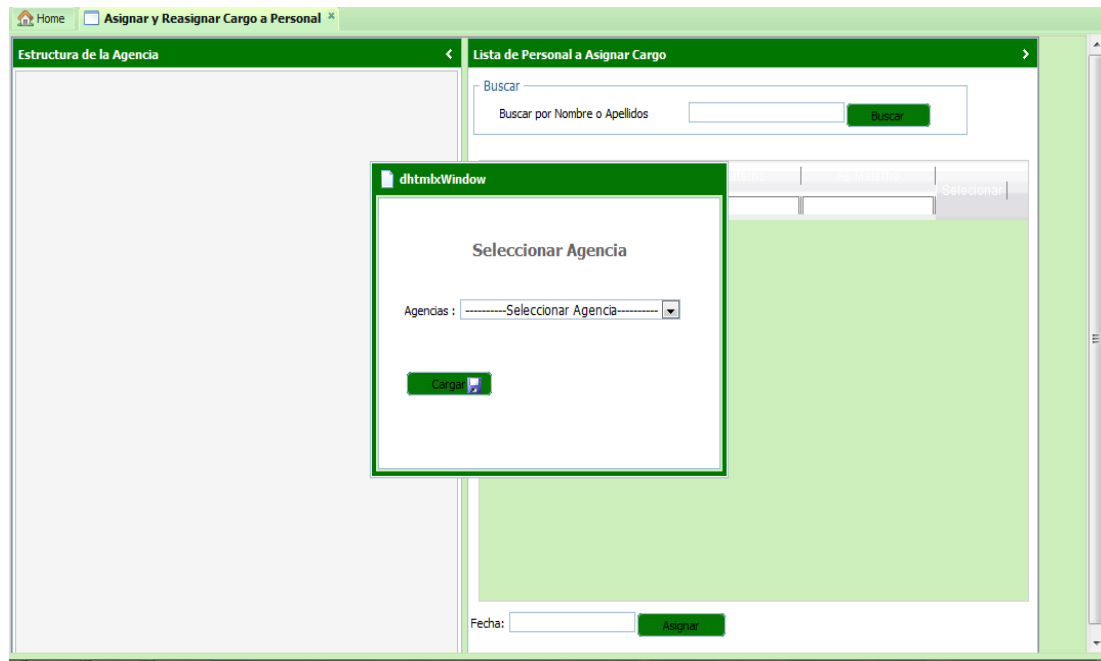


Figura 56. Pantalla Administrar Asignación y Reasignar Cargo a Personal

1.1.2.2.3.8.4.9. Pantalla Asignar Cargo

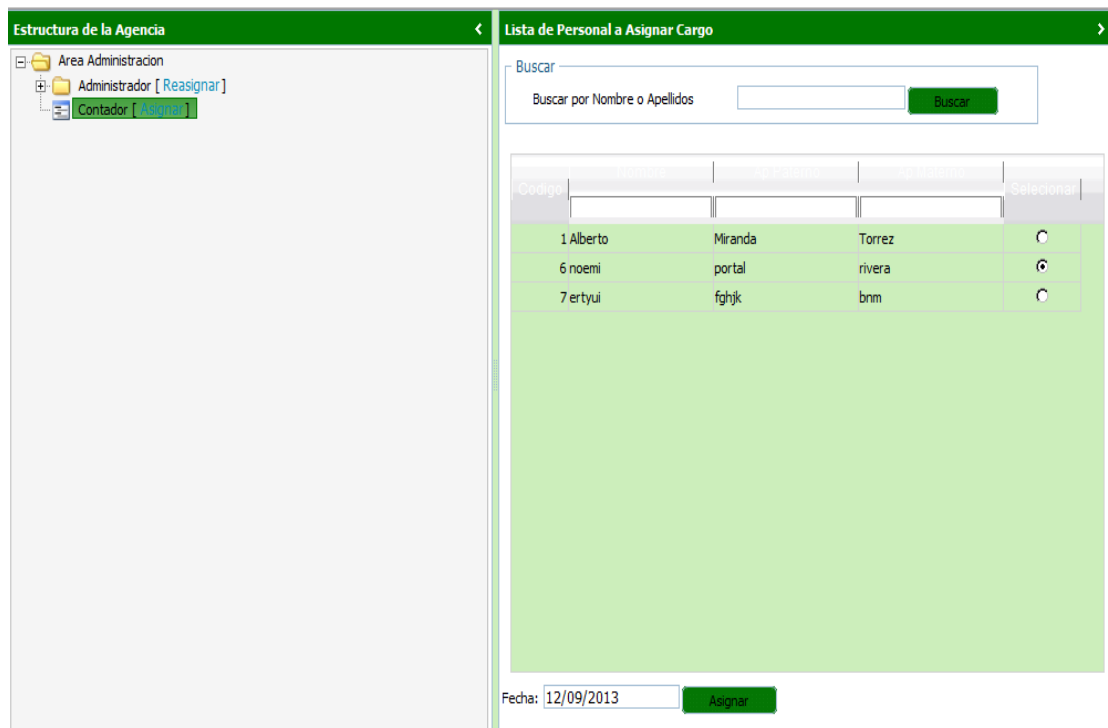


Figura 57. Pantalla Asignar Cargo

1.1.2.2.3.8.4.10. Pantalla Reasignar Cargo

Reasignar Personal

Formulario Reasignar Personal

Cargo Actual : Contador

Nombre del Personal : portal rivera noemi

Nueva Area : -----Seleccionar Area-----

Nuevo Cargo : -----Seleccionar Cargo-----

Fecha de Reasignación del Personal : 12/09/2013

Aceptar Cancelar

Figura 58. Pantalla Reasignar Cargo

1.1.2.2.3.8.4.11. Pantalla Administrar Agencias

Home Administrar Agencia *

LISTA DE AGENCIAS

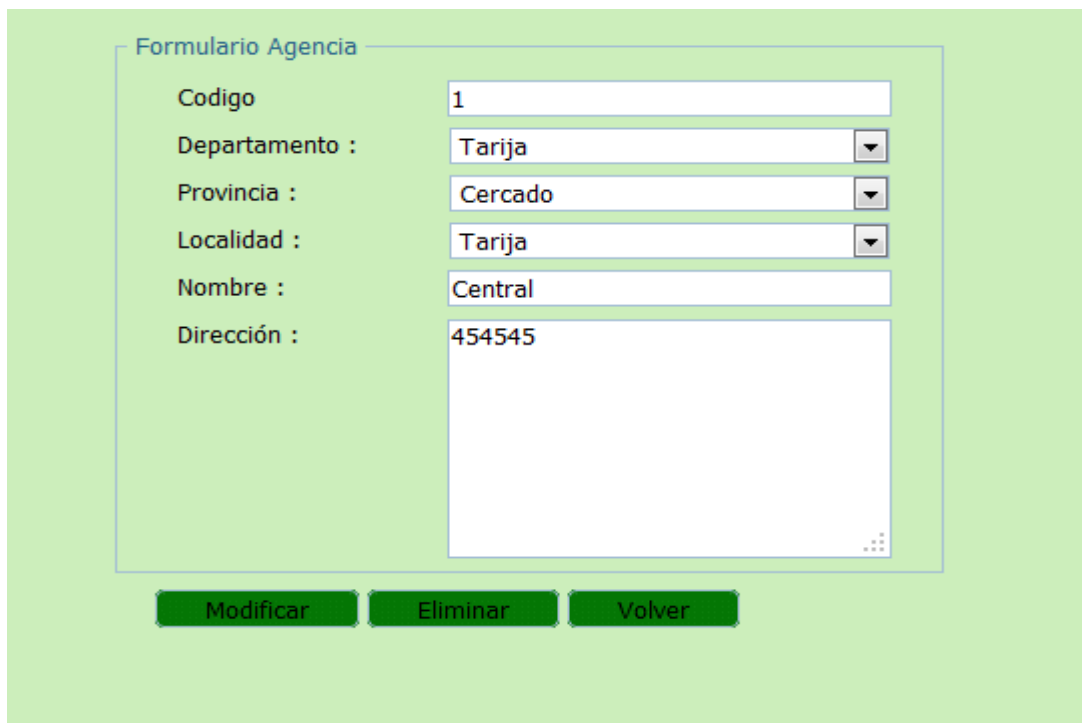
ADICIONAR MODIFICAR ELIMINAR KARDEX BUSCAR

CODIGO	NOMBRE	DIRECCIÓN	LOCALIDAD	PROVINCIA	DEPARTAMENTO
1 2	Sucursal 1	Z/ Terminal	San Lorenzo	Cercado	Tarja
2 1	Central	454545	Tarja	Cercado	Tarja

Página 1 de 1 Mostrando 1 - 2 de 2

Figura 59. Pantalla Administrar Agencias

1.1.2.2.3.8.4.12. Pantalla (ABM) Agencia



Formulario Agencia

Codigo	1
Departamento :	Tarija
Provincia :	Cercado
Localidad :	Tarija
Nombre :	Central
Dirección :	454545

Modificar Eliminar Volver

Figura 60. Pantalla (ABM) Agencia

1.1.2.2.3.8.4.13. Pantalla Administrar Estructura Organizativa

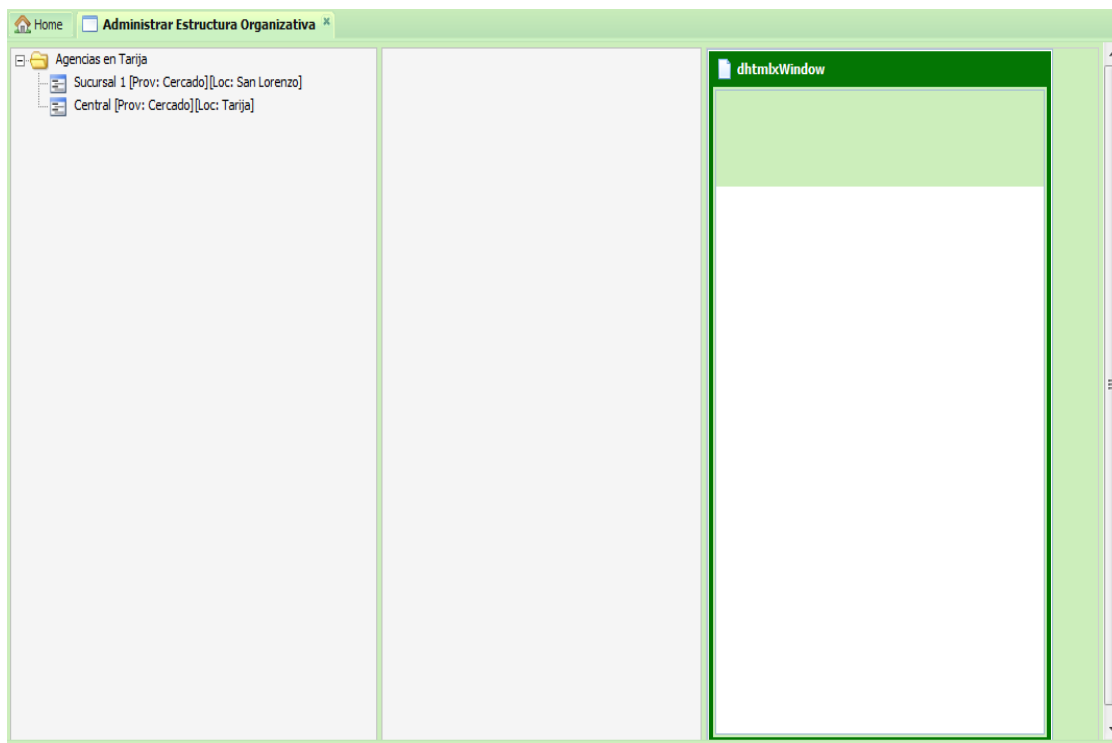


Figura 61. Pantalla Administrar Estructura Organizativa

1.1.2.2.3.8.4.14. Pantalla (ABM) Área

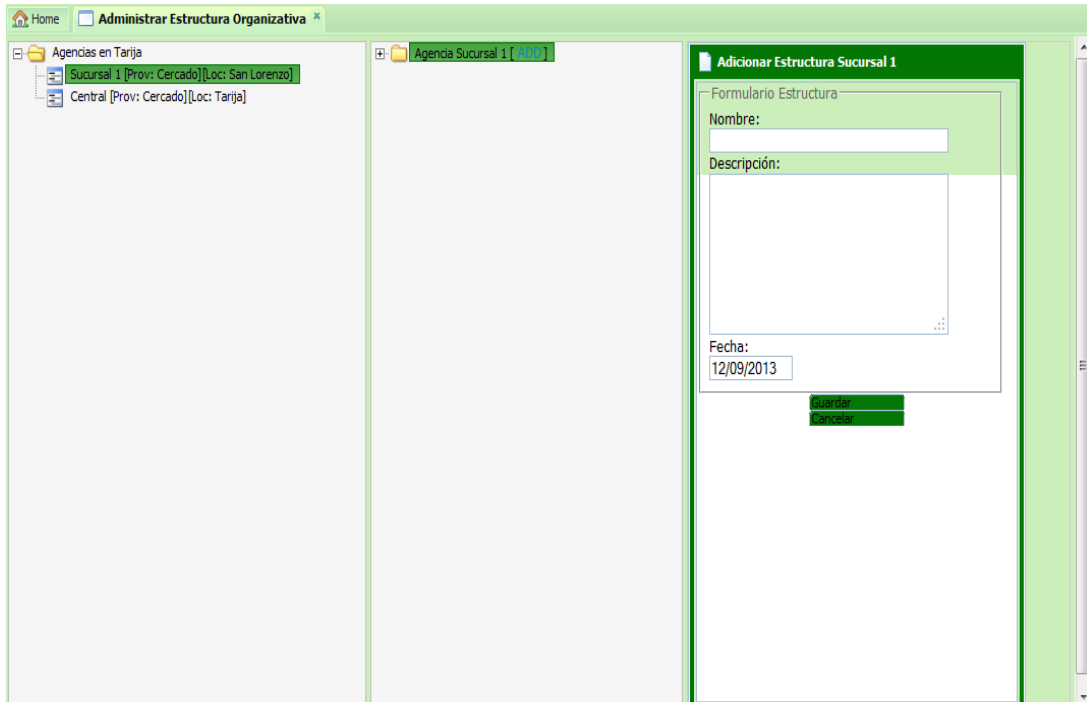


Figura 62. Pantalla (ABM) Área

1.1.2.2.3.8.4.15. Pantalla (ABM) Cargo

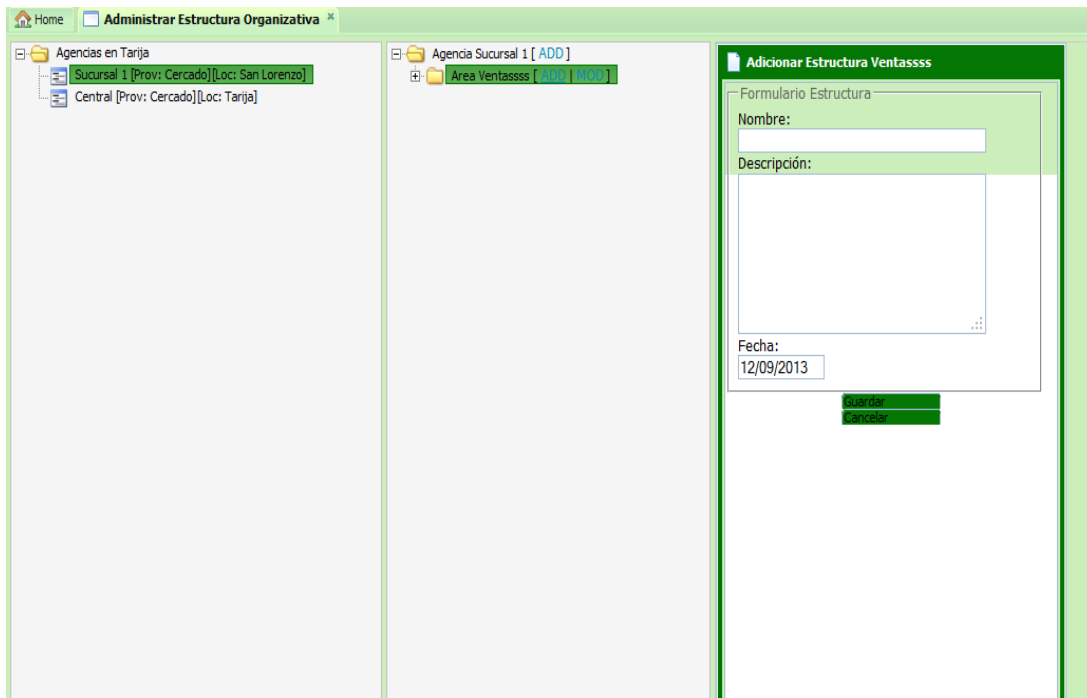
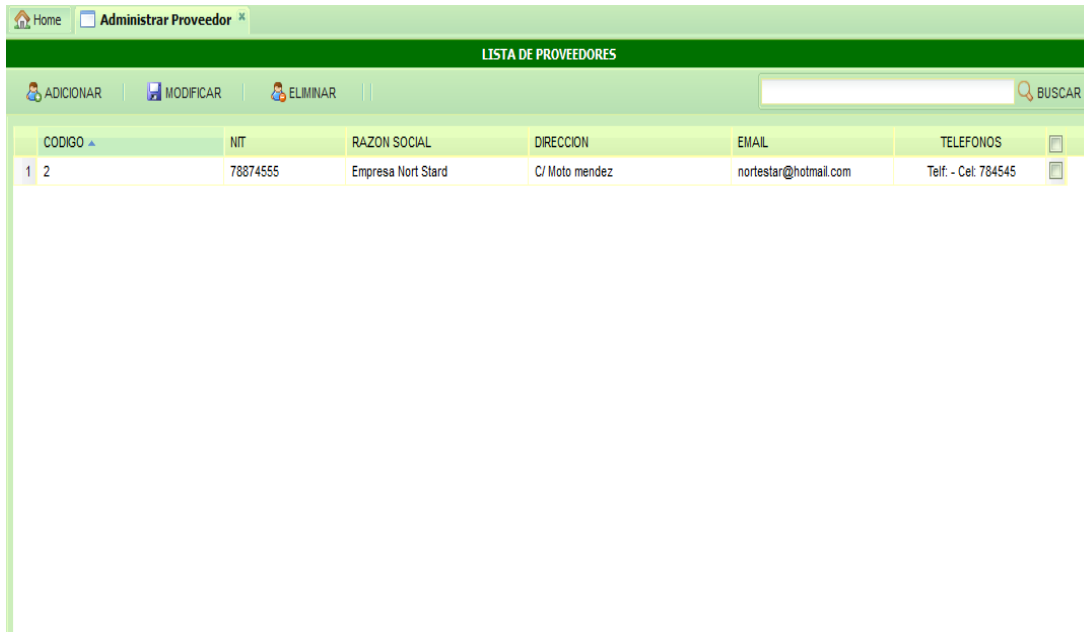


Figura 63. Pantalla (ABM) Cargo

1.1.2.2.3.8.4.16. Pantalla Administrar Proveedor

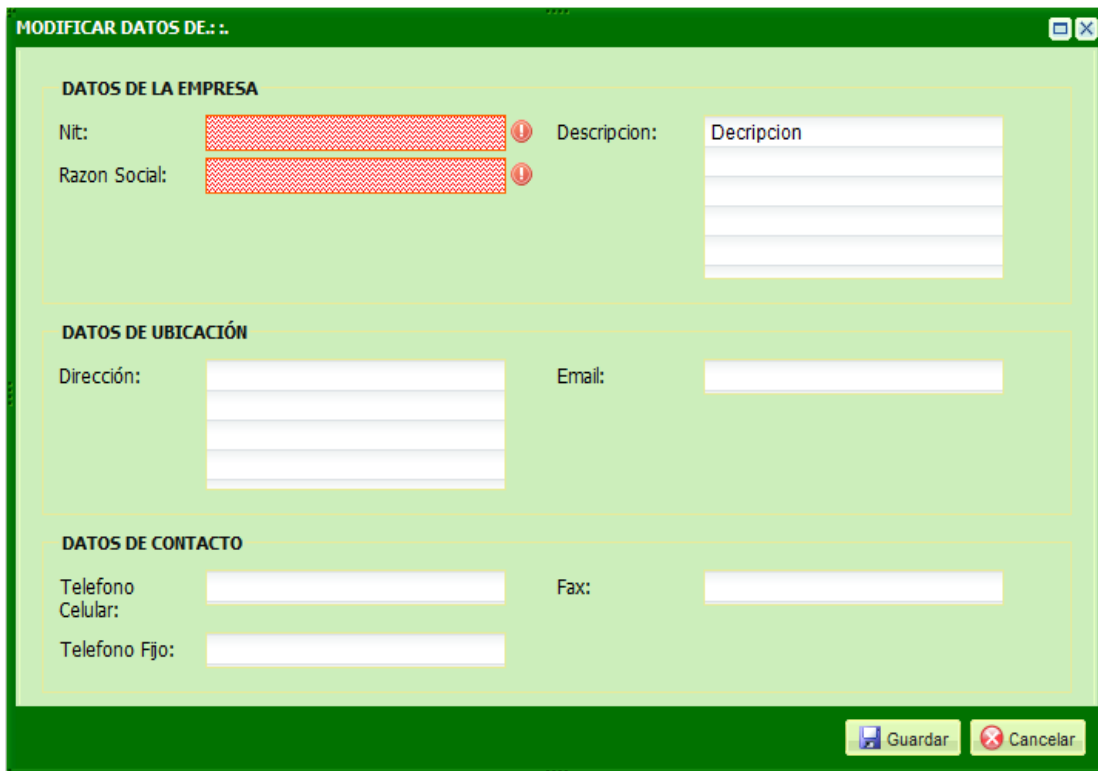


The screenshot shows a web application interface for managing providers. At the top, there is a navigation bar with 'Home' and 'Administrar Proveedor'. Below this is a header 'LISTA DE PROVEEDORES' and a toolbar with buttons for 'ADICIONAR', 'MODIFICAR', and 'ELIMINAR', along with a search box labeled 'BUSCAR'. The main content is a table with the following data:

CODIGO	NIT	RAZON SOCIAL	DIRECCION	EMAIL	TELEFONOS
1 2	78874555	Empresa Nort Stard	C/ Moto mendez	nortestar@hotmail.com	Telf: - Cel: 784545

Figura 64. Pantalla Administrar Proveedor

1.1.2.2.3.8.4.17. Pantalla (ABM) Proveedor



The screenshot shows a form titled 'MODIFICAR DATOS DE :.' for editing provider information. The form is divided into three sections: 'DATOS DE LA EMPRESA', 'DATOS DE UBICACIÓN', and 'DATOS DE CONTACTO'. The 'DATOS DE LA EMPRESA' section includes fields for 'Nit:', 'Razon Social:', and 'Descripcion:'. The 'DATOS DE UBICACIÓN' section includes fields for 'Dirección:' and 'Email:'. The 'DATOS DE CONTACTO' section includes fields for 'Telefono Celular:', 'Telefono Fijo:', and 'Fax:'. The form has a green background and a dark green border. At the bottom right, there are buttons for 'Guardar' and 'Cancelar'.

Figura 65. Pantalla (ABM) Proveedor

1.1.2.2.3.8.4.18. Pantalla Administrar Productos

The screenshot shows a web application interface for managing products. On the left, there is a navigation tree under 'ALMACEN EL CARMEN' with categories for 'Almacen Zapatos' (further divided into 'Mujeres' and 'Masculinios') and 'Categoria Almacen Zapatos para Ninas'. The main area is titled 'LISTA DE PRODUCTOS' and contains a table with the following data:

ID	COD. PRODUCTO	NOMBRE	DESCRIPCION	MARCA	
1	3	74845	Brasiera	Zapatillas toper para ninas de 10 años	Tooper
2	1	4744	Munequita	Zapatos con evilla	Tooper
3	2	74874	Chinela	zapatillas para ninas # 44	Addidas

At the bottom of the screen, it indicates 'Página 1 de 1' and 'Mostrando 1 - 3 de 3'.

Figura 66. Pantalla Administrar Productos

1.1.2.2.3.8.4.19. Pantalla (ABM) Producto

The screenshot shows a form titled 'PRODUCTOS' for adding or modifying a product. The form is divided into 'DATOS DEL PRODUCTO' and contains the following fields:

- Cod. Producto:** A text input field with a red wavy pattern and a red exclamation mark icon.
- Nombre:** A text input field with a red wavy pattern and a red exclamation mark icon.
- Unidad:** A dropdown menu with the text '--- Seleccionar Unidad ---'.
- Marca:** A dropdown menu with the text '--- Seleccionar Marca ---'.
- Descripcion:** A text area with the text 'Decripcion' and several empty lines below it.

At the bottom of the form, there are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel).

Figura 67. Pantalla (ABM) Producto

1.1.2.2.3.8.4.20. Pantalla Administrar Tipo y Categoría de Almacén

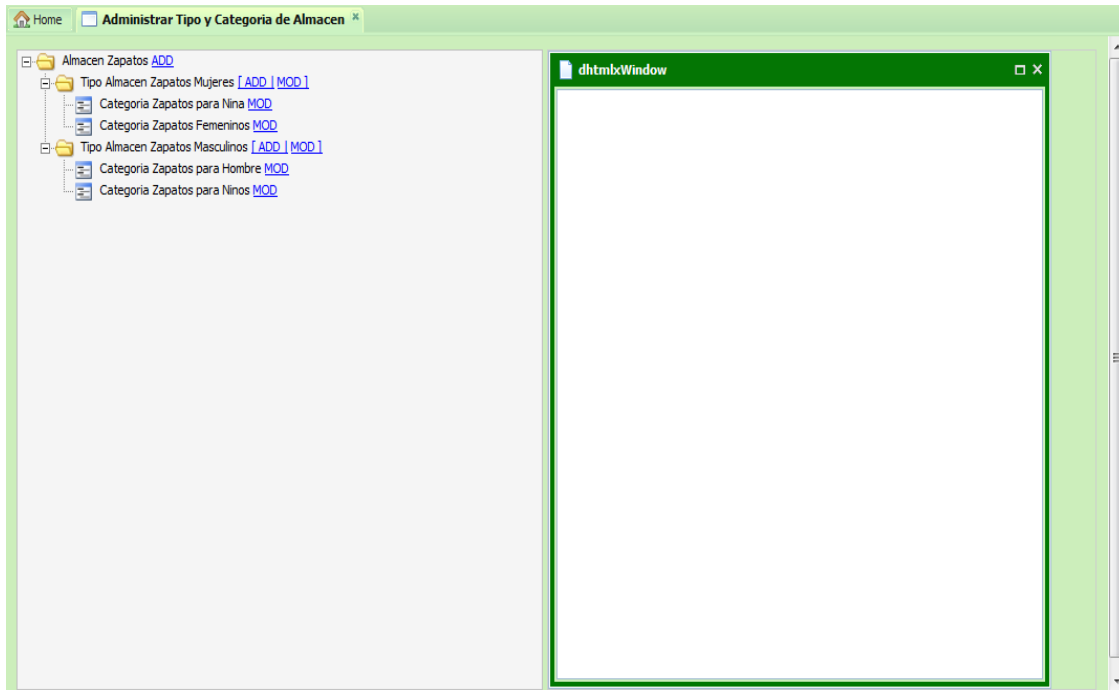


Figura 68. Pantalla Administrar Tipo y Categoría de Almacén

1.1.2.2.3.8.4.21. Pantalla (ABM) Tipo de Almacén

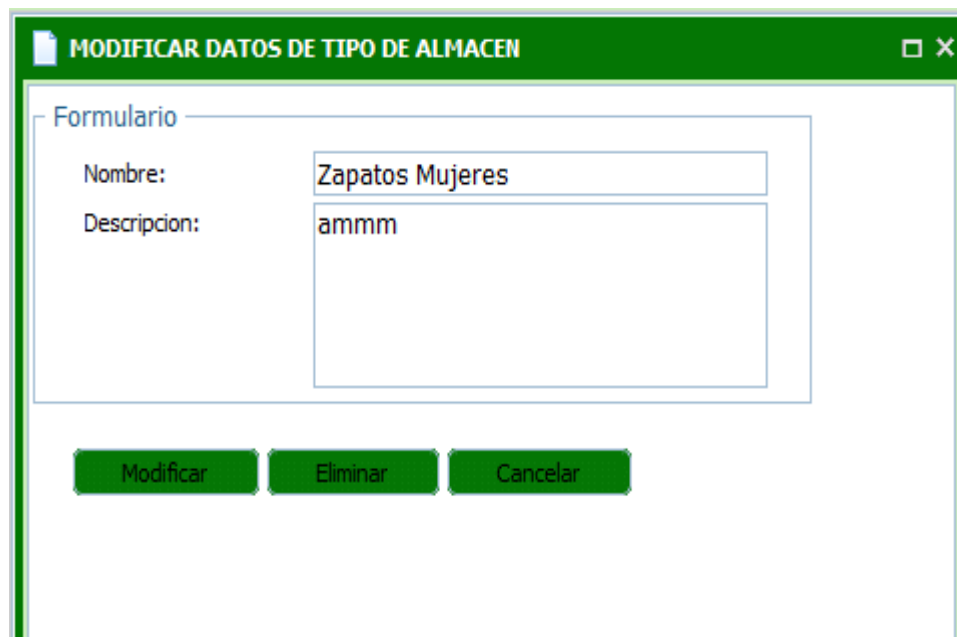
The image shows a form window titled 'MODIFICAR DATOS DE TIPO DE ALMACEN'. The form is titled 'Formulario' and contains two input fields: 'Nombre:' with the value 'Zapatos Mujeres' and 'Descripcion:' with the value 'ammm'. Below the form, there are three buttons: 'Modificar', 'Eliminar', and 'Cancelar'.

Figura 69. Pantalla (ABM) Tipo de Almacén

1.1.2.2.3.8.4.22. Pantalla (ABM) Categoría de Almacén

MODIFICAR DATOS DE CATEGORIA DE ALMACEN

Formulario

Nombre: Zapatos para Nina

Descripcion: zapatos para ninas

Modificar Eliminar Cancelar

Figura 70. Pantalla (ABM) Categoría de Almacén

1.1.2.2.3.8.4.23. Pantalla Administrar Ingreso de Productos

Home Administrar Ingreso Producto

LISTA DE INGRESOS DE PRODUCTOS

ADICIONAR ELIMINAR KARDEX BUSCAR

	CODIGO	Nº PEDIDO	PROVEEDOR	INGRESO DE AGENCIA	RESPONSABLE DE INGRESO	FECHA DE INGRESO
1	2	2222	Empresa Nort Stard	Sucursal 1	Encargado de Ventas : Marcela Perales Gonzales	2013-08-05
2	1	1231	Empresa Manaco	Central	Administrador : Alberto Miranda Torrez	2013-08-05

Página 1 de 1 Mostrando 1 - 2 de 2

Figura 71. Pantalla Administrar Ingreso de Productos

1.1.2.2.3.8.4.24. Pantalla (ABM) Ingreso de Productos

DATOS PARA INGRESO DE PRODUCTOS

Nº Pedido:

Nº Factura:

Ingresar Productos A:

Proveedor:

Fecha de Ingreso:

Figura 72. Pantalla (ABM) Ingreso de Productos

1.1.2.2.3.8.4.25. Pantalla Detalle de Ingreso de Producto

ADICIONANDO DETALLE DE INGRESO

DATOS DE INGRESO DE PRODUCTOS

Nº FACTURA: 567
 PROVEEDOR: Grendha
 FECHA: 01/10/2013
 Nº PEDIDO: 567

FORMULARIO DE INGRESO DE PRODUCTO

Encargado:

Ingreso:

Producto:

Color/Talla:

Cantidad:

Precio Unitario:

ID PERTENCE: 0
 ID AGENCIA: 1
 CODPRODUCTO:
 IDCOLOR TALLA:
 OPERACION:

Producto Producto

DETALLE DE INGRESOS DE PRODUCTOS

CODIGO K.	COD. PRODUCTO	NOMBRE PRODUCTO	CANTIDAD	PRECIO UNI	COSTO TOTAL
CATEGORIA:			0	\$0.00	\$0.00
					\$0.00

Figura 73. Pantalla Detalle de Ingreso de Producto

1.1.2.2.3.8.4.26. Pantalla Administrar Salida de Productos

	CODIGO	N° FORMULARIO	SALIDA DE AGENCIA	RESPONSABLE DE SALIDA	FECHA DE SALIDA
1	8	22	Central	Administrador: Fabio Barrios Calzaya	2013-09-10
2	6	233	Central	Contador: Fabio Barrios Calzaya	2013-08-05
3	7	2000	Sucursal 1	Encargado de Ventas: Marcela Perales Gonzales	2013-08-05

Figura 74. Pantalla Administrar Salida de Productos

1.1.2.2.3.8.4.27. Pantalla (ABM) Salida de Productos

DATOS PARA SALIDA DE PRODUCTOS

N° Formulario:
45

Salida Productos De:
Central

Fecha de Salida:
12/09/2013

Continuar Cancelar

Figura 75. Pantalla (ABM) Salida de Productos

1.1.2.2.3.8.4.28. Pantalla Detalle de Salida de Productos

Home Administrar Salida Producto

FORMULARIO DE SALIDA DE PRODUCTO

DATOS DE SALIDA DE PRODUCTOS

Nº FORMULARIO: 34567

SALIDA DE PRODUCTOS DE: Agencia Principal--COMERCIAL EL C.

FECHA: 06/10/2013

ENGARGADO SALIDA: ----- Seleccionar Empleado -----

FORMULARIO DE SALIDA DE PRODUCTO

Producto: ----- Seleccionar Producto -----

Color/Talla: ----- Seleccionar Color/Talla -----

Cantidad Sol: [Redacted]

Producto Producto

DETALLE DE SALIDA DE PRODUCTOS

CODIGO KARDEX	COD. PRODUCTO	NOMBRE	CANTIDAD
CATEGORIA:			0

Guardar Cancelar

Figura 76. Pantalla Detalle de Salida de Productos

1.1.2.2.3.8.4.29. Pantalla Reporte de Productos por Agencia

Home Reporte Producto en Almacenes

DATOS DE REPORTE

Producto: 11-1 Agencia: Agencia Principal--COMERCIAL EL

Mostrar

Figura 77. Pantalla Reporte de Productos por Agencia

1.1.2.2.3.8.4.30. Pantalla Reporte de Ingresos por Agencia

Home Reporte de Ingresos Por Agencia

DATOS DE REPORTE

Agencia: Agencia Principal--COMERCIAL E

Mostrar

REPORTE DE INGRESOS

AGENCIA : Agencia Principal-- AGENCIA : Tarija
DIRECCION : C/ Daniel Campos entre Bolivar e Ingavi

USUARIO : RENE GUTIERREZ PEREZ FECHA : viernes 27 septiembre 2013

FECHA INGRESO	NRO. PEDIDO	NRO. FACTURA	RESPONSABLE DE INGRESO	PROVEEDOR
04/10/2013	222	2	Rene Gutierrez Perez	Grendha

DETALLE DEL INGRESO:

CODIGO	DESCRIPCION	CANTIDAD	PRECIO	SUB TOTAL
		1 / 1		

TOTAL Bs.: null

Figura 78. Pantalla Reporte De Ingresos por Agencia

1.1.2.2.3.8.4.31. Pantalla Reporte de Salidas por Agencia

Home Reporte de Salida Por Agencia

DATOS DE REPORTE

Agencia: Agencia Principal--COMERCIAL E

Mostrar

Figura 79. Pantalla Reporte de Salidas por Agencia

1.1.2.2.3.8.4.32. Pantalla Reporte de Productos Por Proveedor



The screenshot shows a web browser window with the title 'Reporte de Productos por Proveedor'. At the top, there is a search filter for 'DATOS DE REPORTE' with 'Grendha' selected in the 'Proveedor' dropdown and a 'Mostrar' button. Below this is a header section with a religious image on the left and the title 'REPORTE DE PRODUCTOS POR PROVEEDOR' in the center. A horizontal line separates the header from the user information: 'USUARIO: RENE GUTIERREZ PEREZ' on the left and 'FECHA: viernes 27 septiembre 2013' on the right. The main content area contains contact details for 'Grendha':

EMPRESA:	Grendha	NIT:	54647443
EMAIL :	grendha@hotmail.com	FAX :	74874
TELEFONO :	443325	CELULAR :	7454774
DIRECCION :	La Paz C/ Sopocachi # 12		
DESCRIPCION :	Proveedor de chinelas		

Below the contact information is the section 'PRODUCTOS POR PROVEEDOR' which contains a table with columns for 'NRO', 'CODIGO PRODUCTO', 'MARCA', and 'CATEGORIA'. The table lists two products:

NRO	CODIGO PRODUCTO	MARCA	CATEGORIA
1	11-1	Brasilera	Zapatos Ninas
2	54	Adidas	Zapatos Ninas

Figura 80. Pantalla Reporte de Productos por Proveedor

1.1.2.2.3.9. Modelo de Análisis y Diseño

1.1.2.2.3.9.1. Modelado de Diagrama de Actividades

1.1.2.2.3.9.1.1. Introducción

El diagrama de Actividades es un artefacto de la disciplina Análisis de Sistemas en la metodología RUP la cual estamos implementando.

Los Diagramas de Actividades se utilizan para modelar aspectos dinámicos de un Sistema, esto implica modelar los pasos secuenciales de un proceso.

1.1.2.2.3.9.1.2. Propósito

Comprender la estructura y dinámica del Sistema deseado.

Identificar posibles mejoras en el Sistema.

1.1.2.2.3.9.1.3. Alcance

Describir los procesos del Sistema y los Clientes.

Identificar y definir los procesos de los casos de uso según los objetivos de la organización.

Definir un Diagrama de Actividad para cada Caso de Uso del Sistema.

1.1.2.2.3.9.1.4. Diagramas de Actividades

1.1.2.2.3.9.1.4.1. Diagrama de Actividad: Caso de Uso Ingresar al Sistema

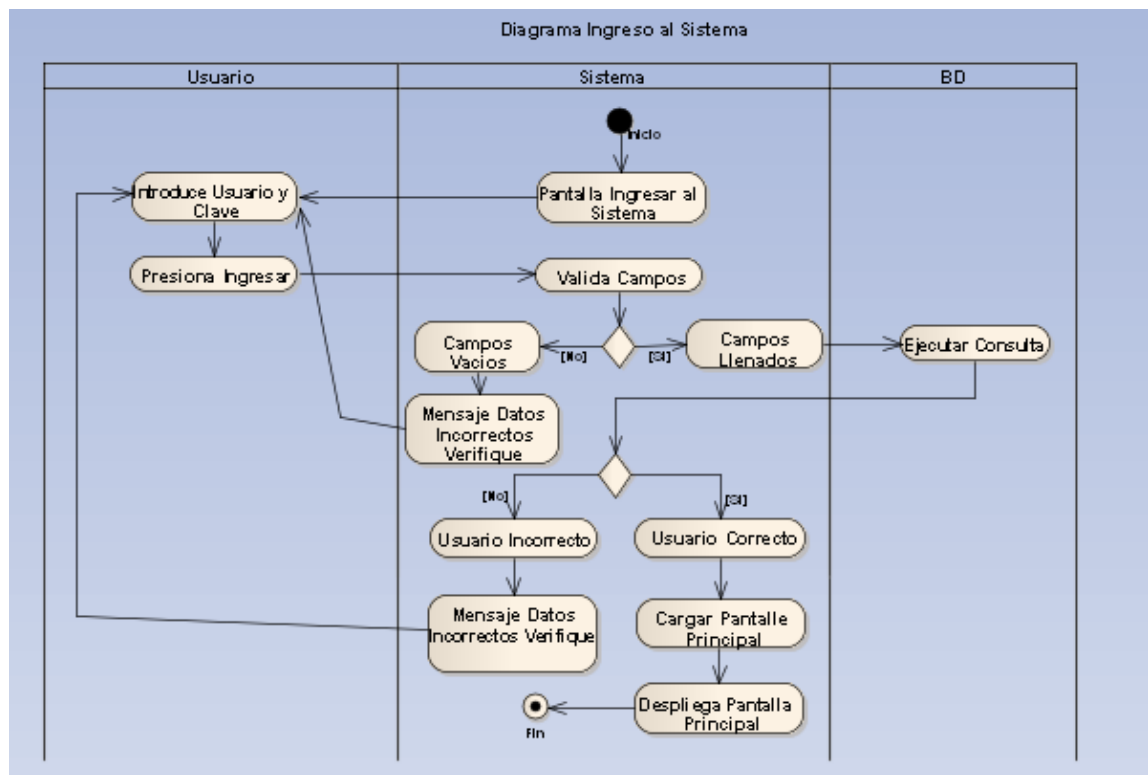


Figura 81. Diagrama de Actividad: Ingresar al Sistema

1.1.2.2.3.9.1.4.2. Diagrama de Actividad: Caso de Uso Administrar Personal

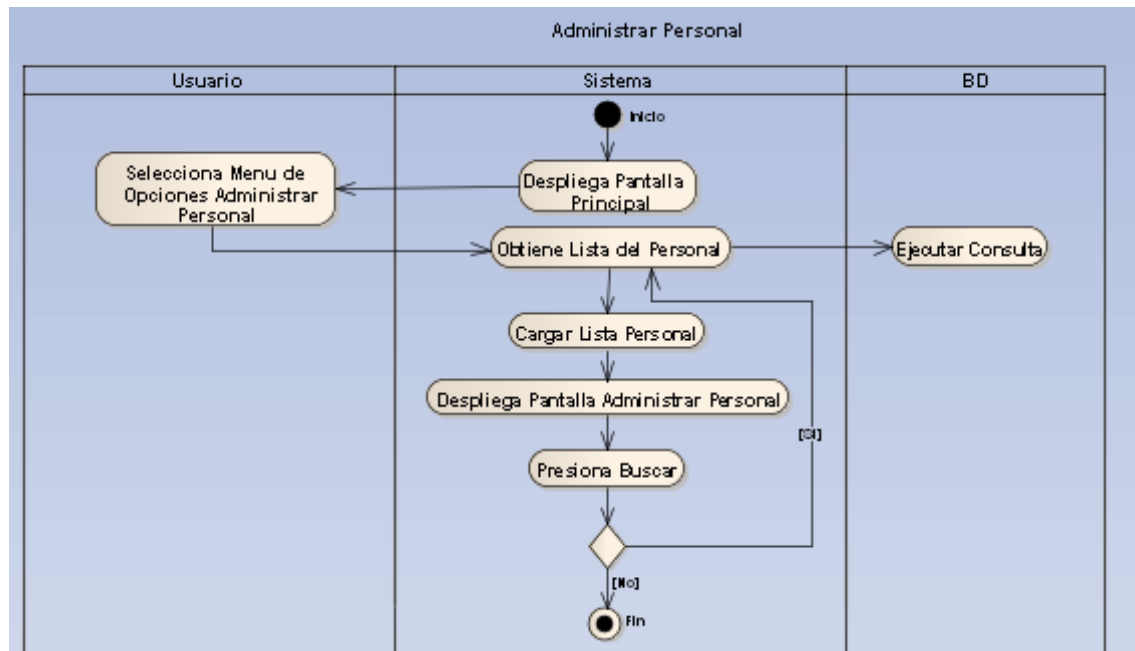


Figura 82. Diagrama de Actividad: Administrar Personal

1.1.2.2.3.9.1.4.3. Diagrama de Actividad: Caso de Uso Adicionar Personal

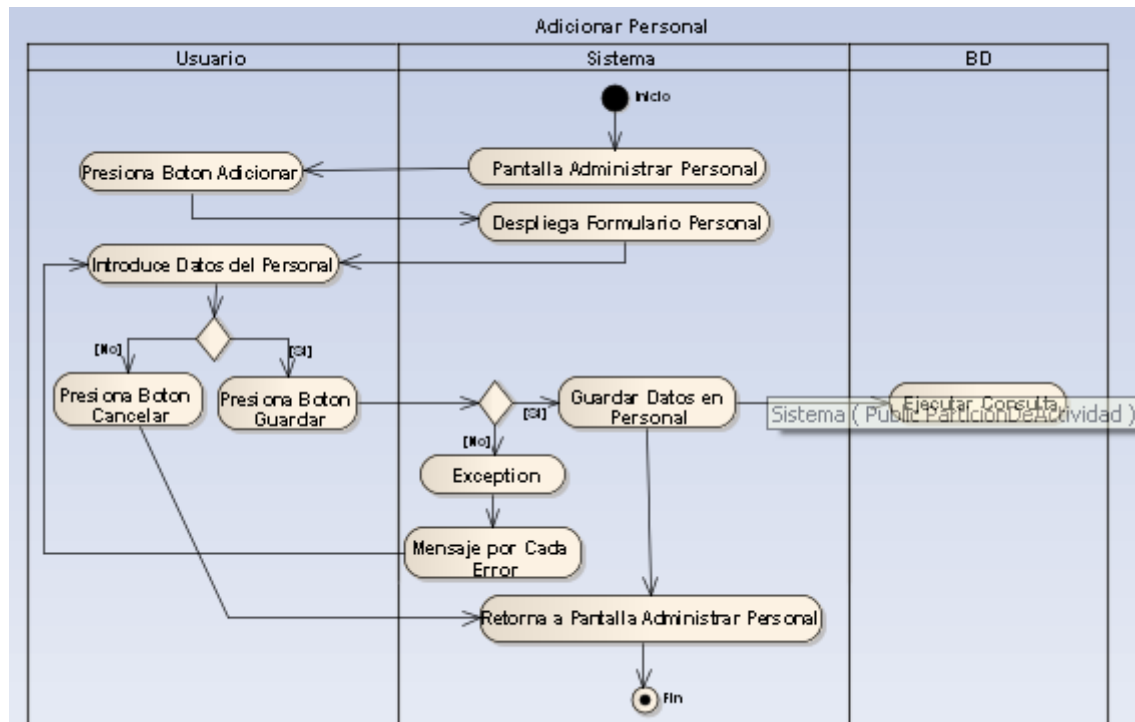


Figura 83. Diagrama de Actividad: Adicionar Personal

1.1.2.2.3.9.1.4.4. Diagrama de Actividad: Caso de Uso Modificar Personal

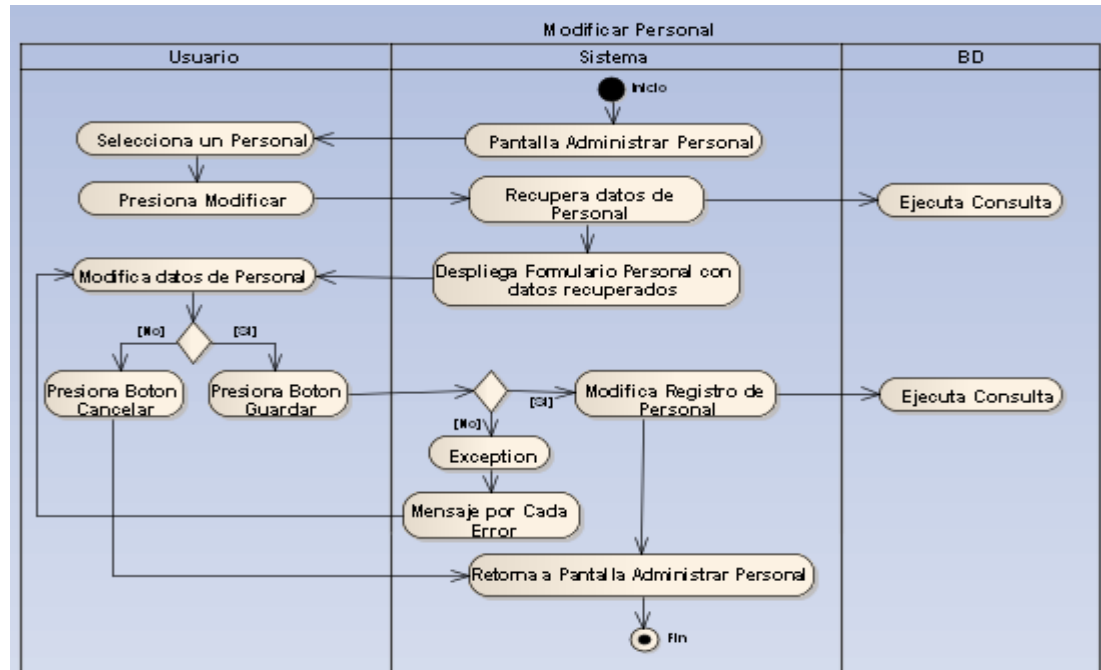


Figura 84. Diagrama de Actividad: Modificar Personal

1.1.2.2.3.9.1.4.5. Diagrama de Actividad: Caso de Uso Eliminar Personal

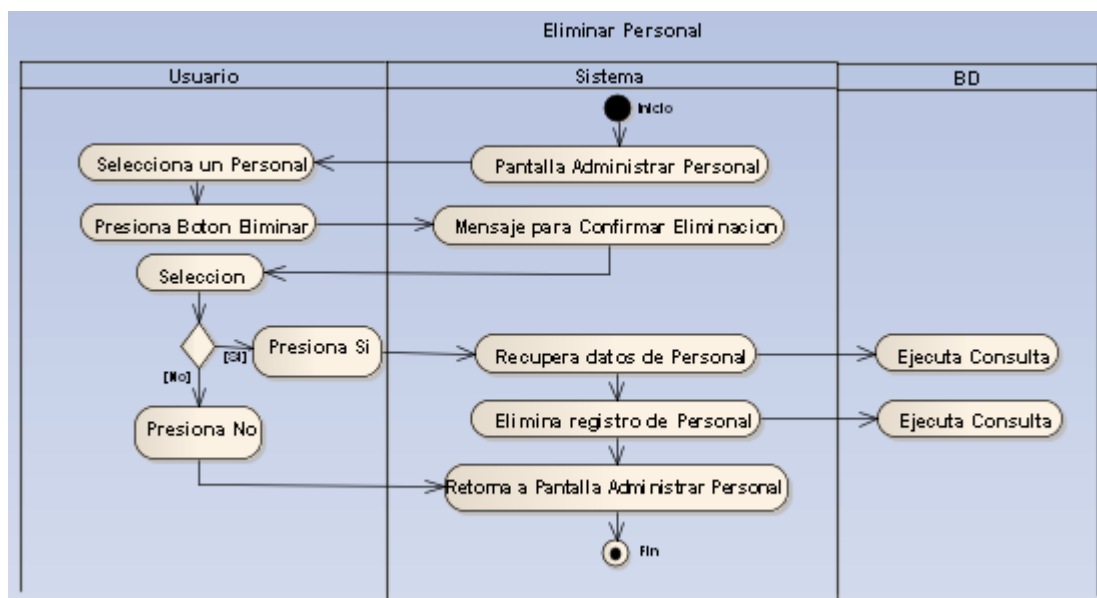


Figura 85. Diagrama de Actividad: Eliminar Personal

1.1.2.2.3.9.1.4.6. Diagrama de Actividad: Caso de Uso Adicionar Usuario

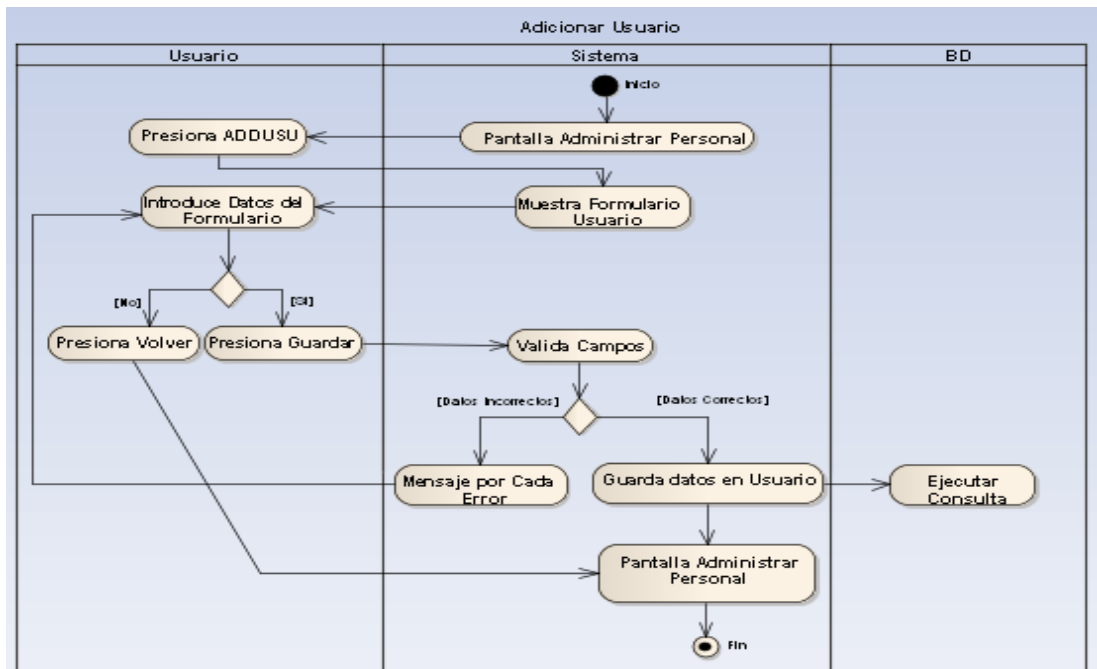


Figura 86. Diagrama de Actividad: Adicionar Usuario

1.1.2.2.3.9.1.4.7. Diagrama de Actividad: Caso de Uso Modificar Usuario

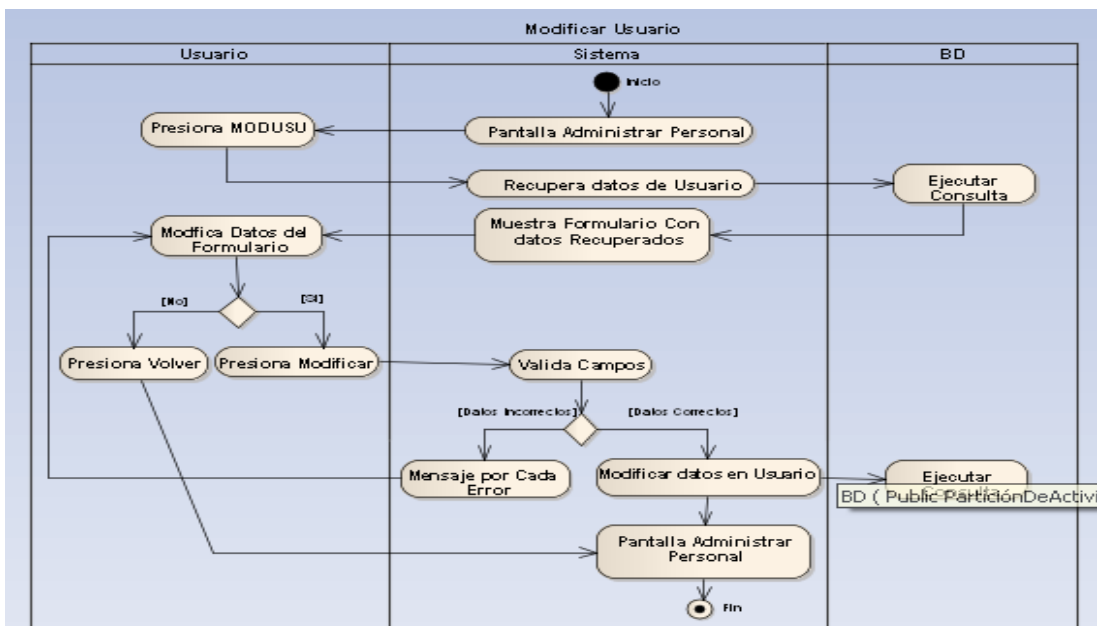


Figura 87. Diagrama de Actividad: Modificar Usuario

1.1.2.2.3.9.1.4.8. Diagrama de Actividad: Caso de Uso Eliminar Usuario

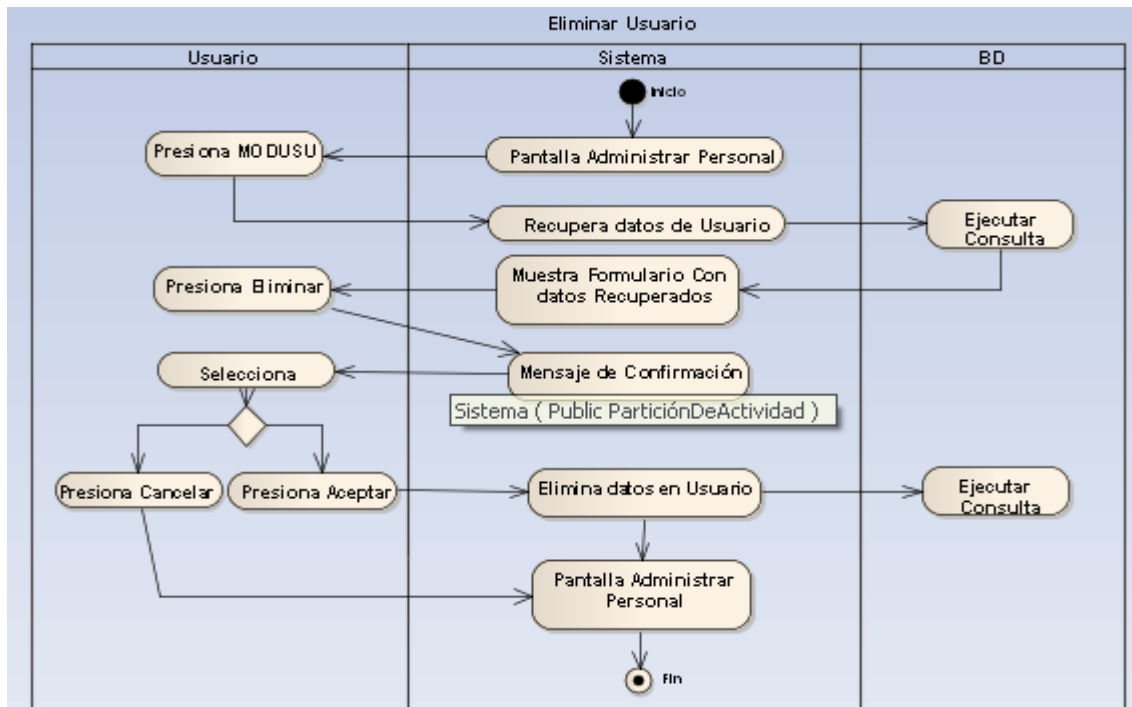


Figura 88. Diagrama de Actividad: Eliminar Usuario

1.1.2.2.3.9.1.4.9. Diagrama de Actividad: Caso de Uso Mostrar Kardex Personal

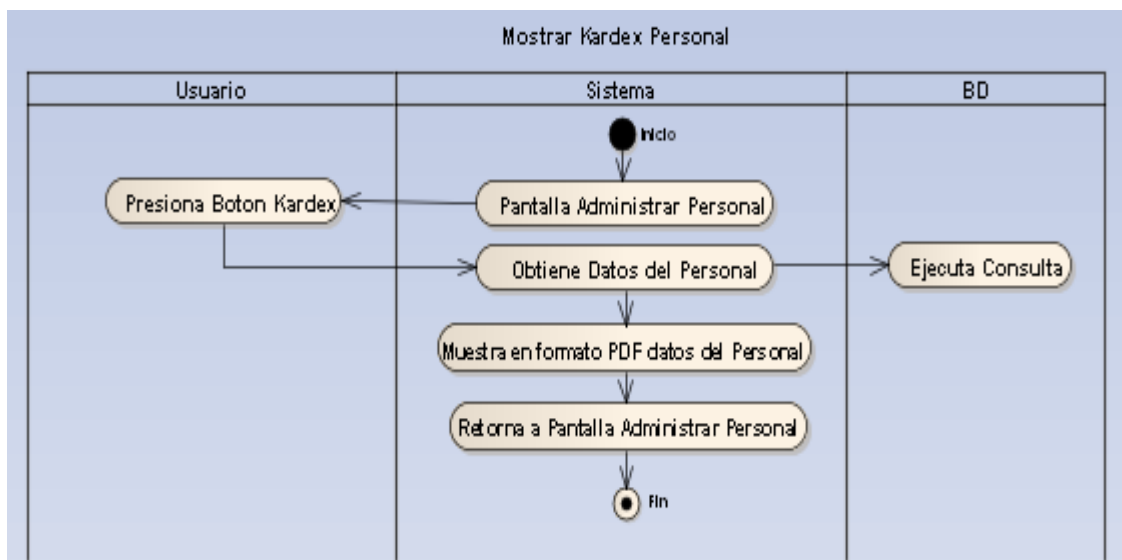


Figura 89. Diagrama de Actividad: Mostrar Kardex Personal

1.1.2.2.3.9.1.4.10. Diagrama de Actividad: Caso de Uso Administrar Roles

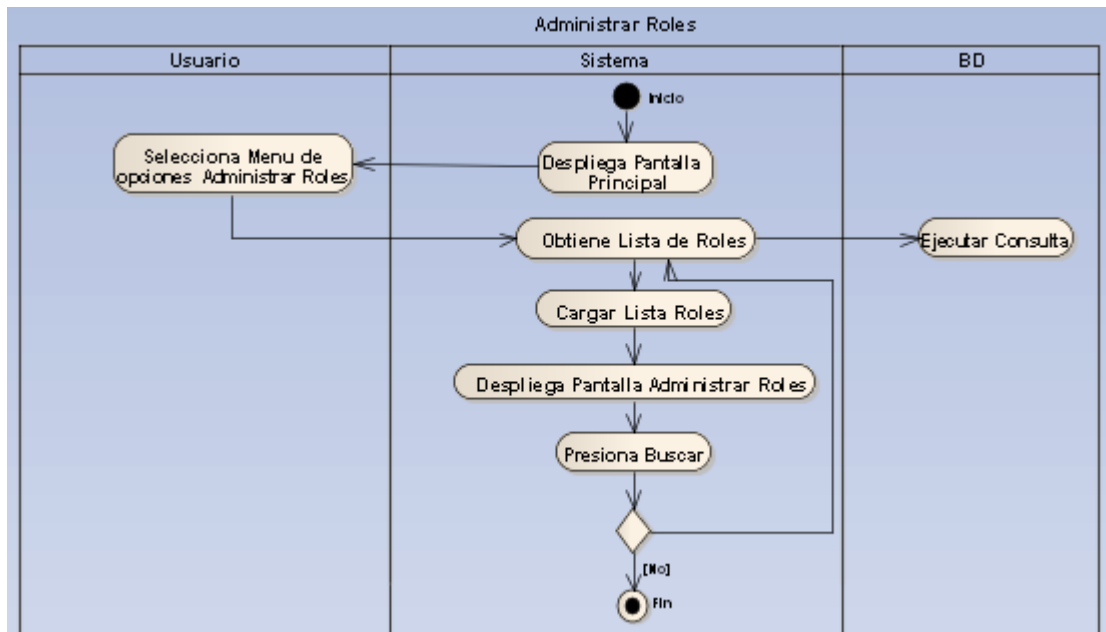


Figura 90. Diagrama de Actividad: Administrar Roles

1.1.2.2.3.9.1.4.11. Diagrama de Actividad: Caso de Uso Adicionar Rol

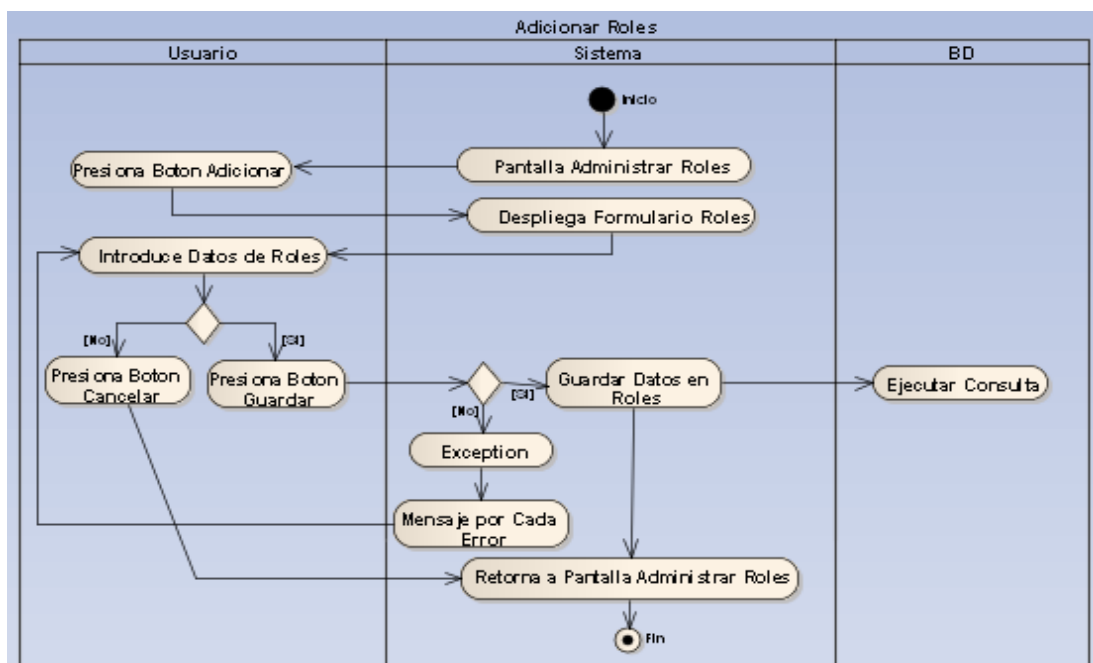


Figura 91. Diagrama de Actividad: Adicionar Rol

1.1.2.2.3.9.1.4.12. Diagrama de Actividad: Caso de Uso Modificar Rol

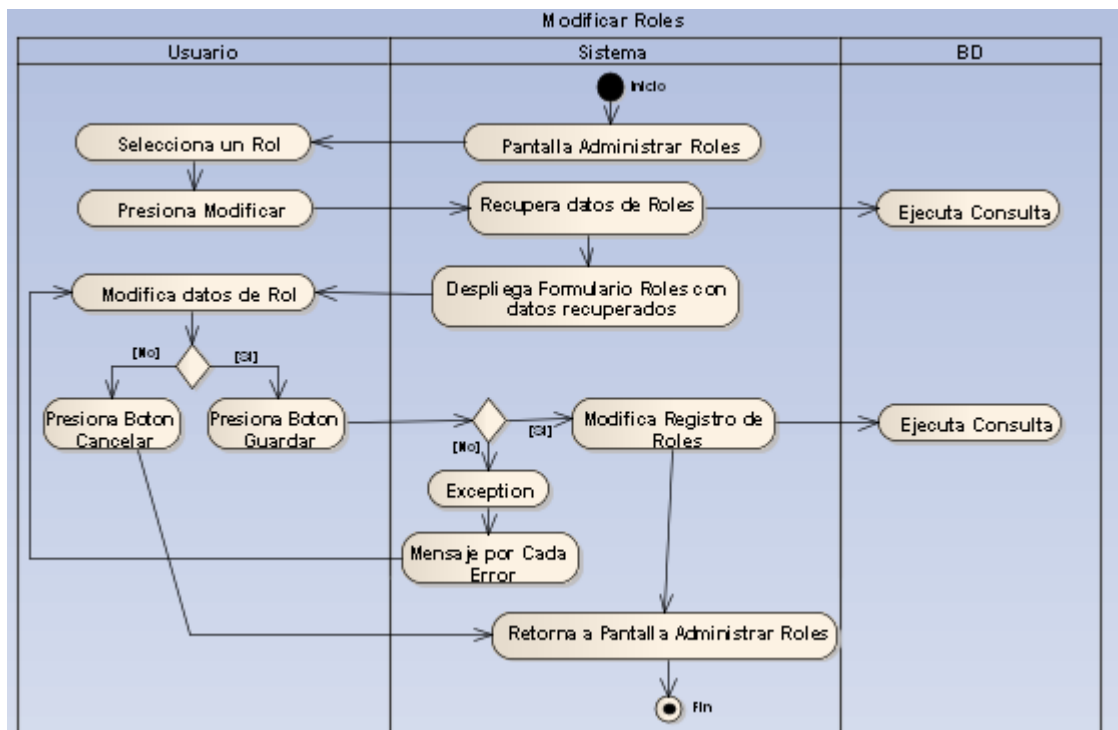


Figura 92. Diagrama de Actividad: Modificar Rol

1.1.2.2.3.9.1.4.13. Diagrama de Actividad: Caso de Uso Eliminar Rol

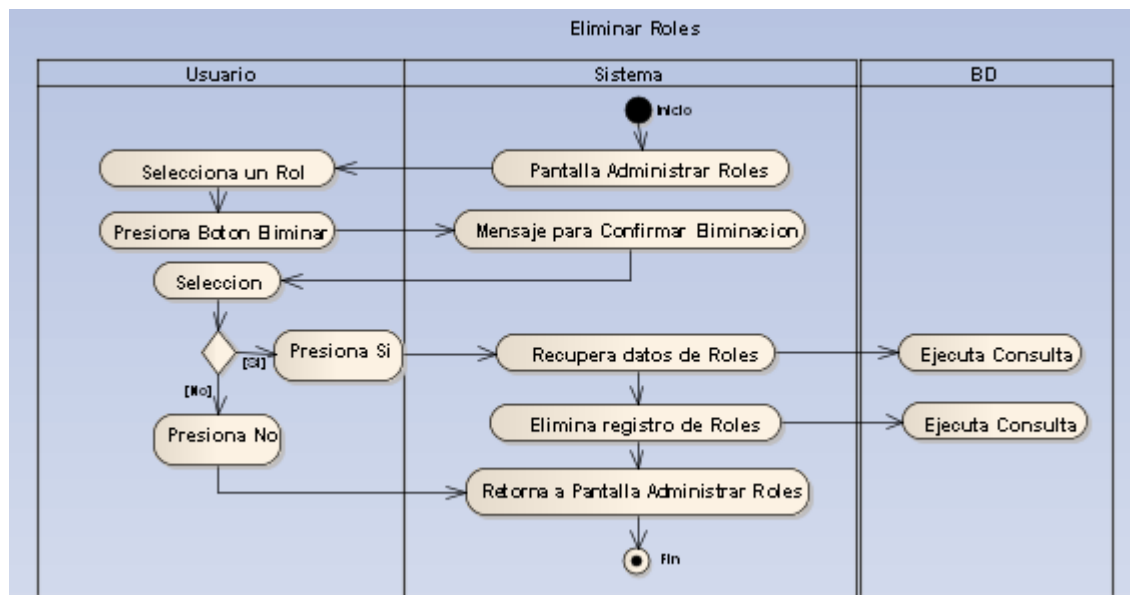


Figura 93. Diagrama de Actividad: Eliminar Rol

**1.1.2.2.3.9.1.4.14. Diagrama de Actividad: Caso de
Uso Administrar Asignar y
Reasignar Cargo a Personal**

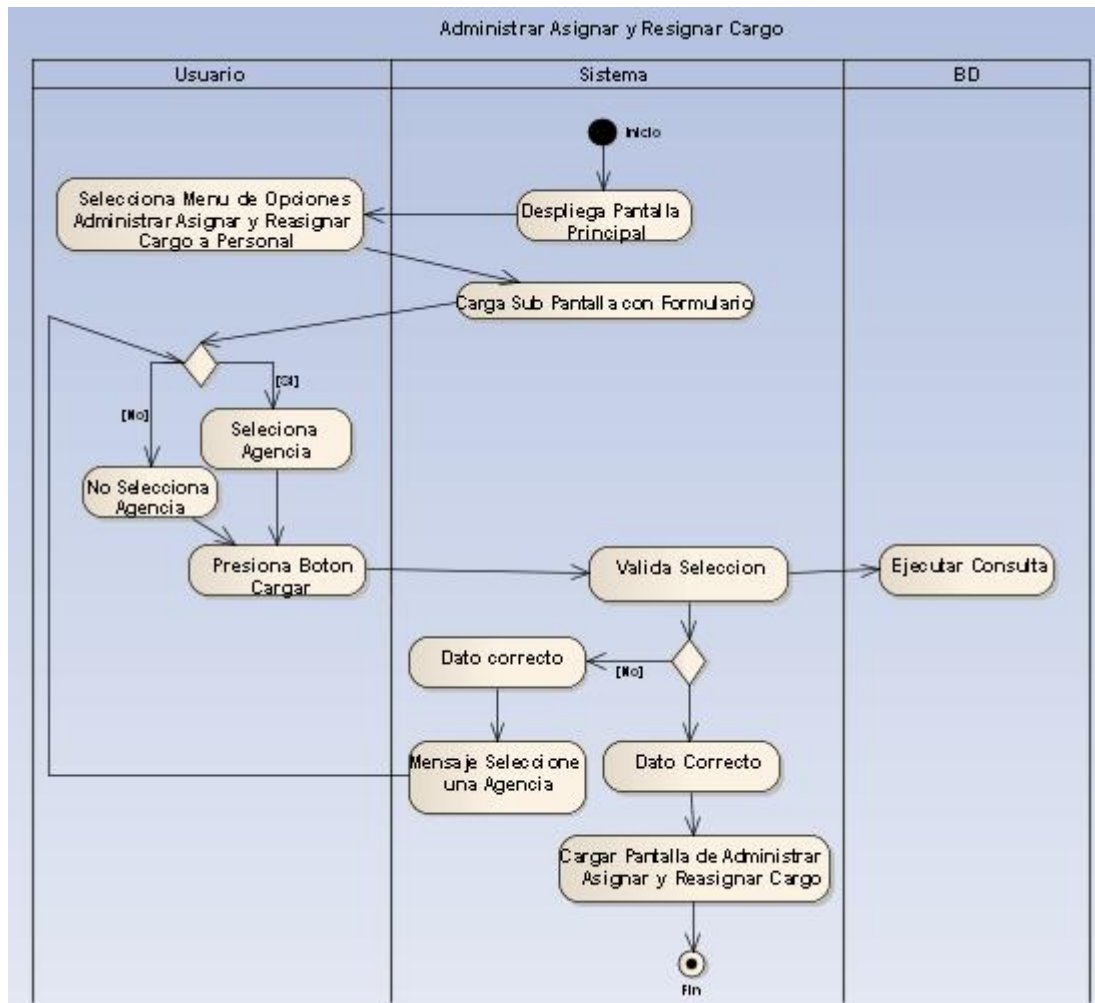


Figura 94. Diagrama de Actividad: Administrar Asignar y Reasignar Cargo a Personal

1.1.2.2.3.9.1.4.15. Diagrama de Actividad: Caso de Uso Asignar Cargo

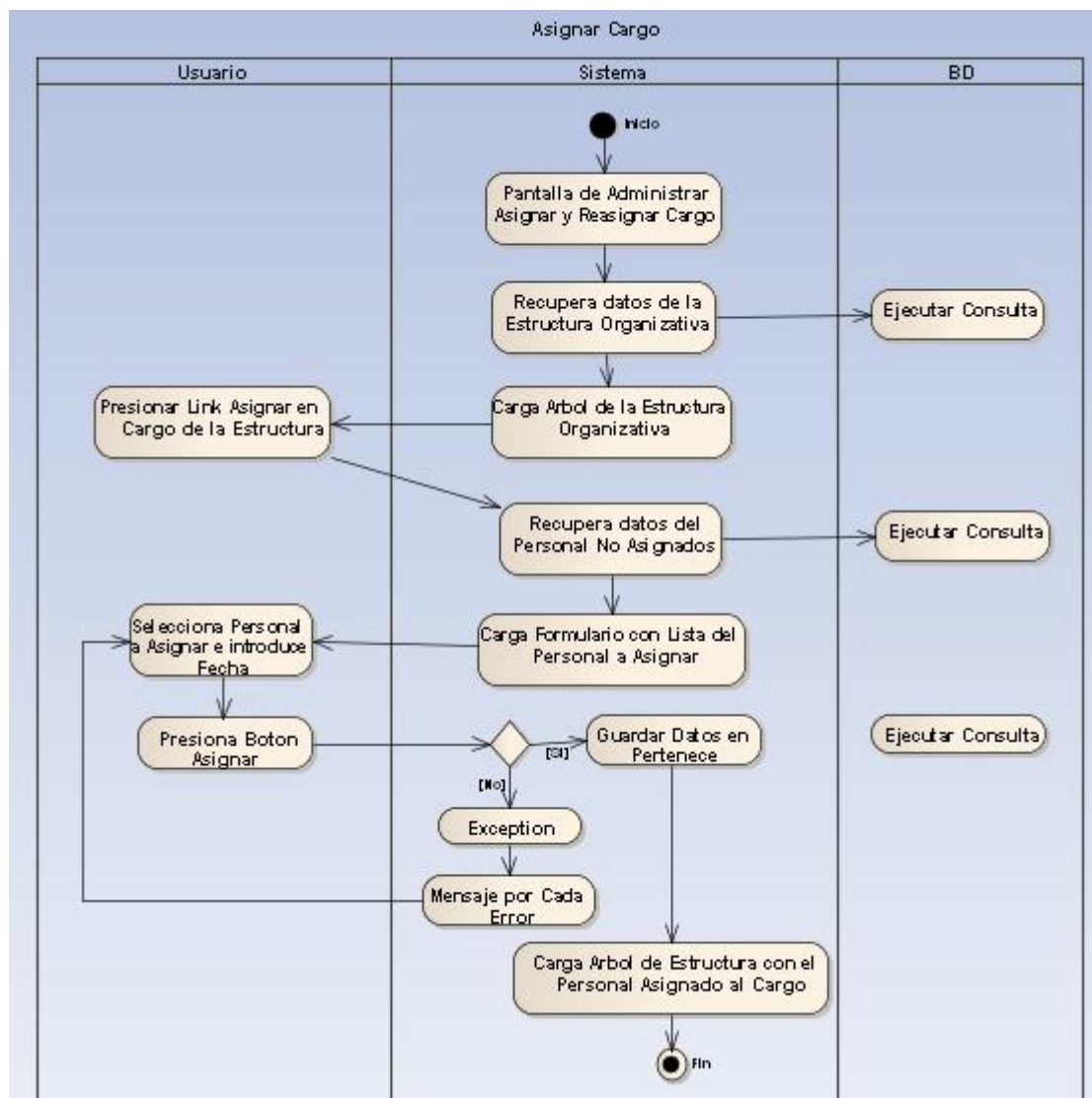


Figura 95. Diagrama de Actividad: Asignar Cargo a Personal

1.1.2.2.3.9.1.4.16. Diagrama de Actividad: Caso de Uso Reasignar Cargo

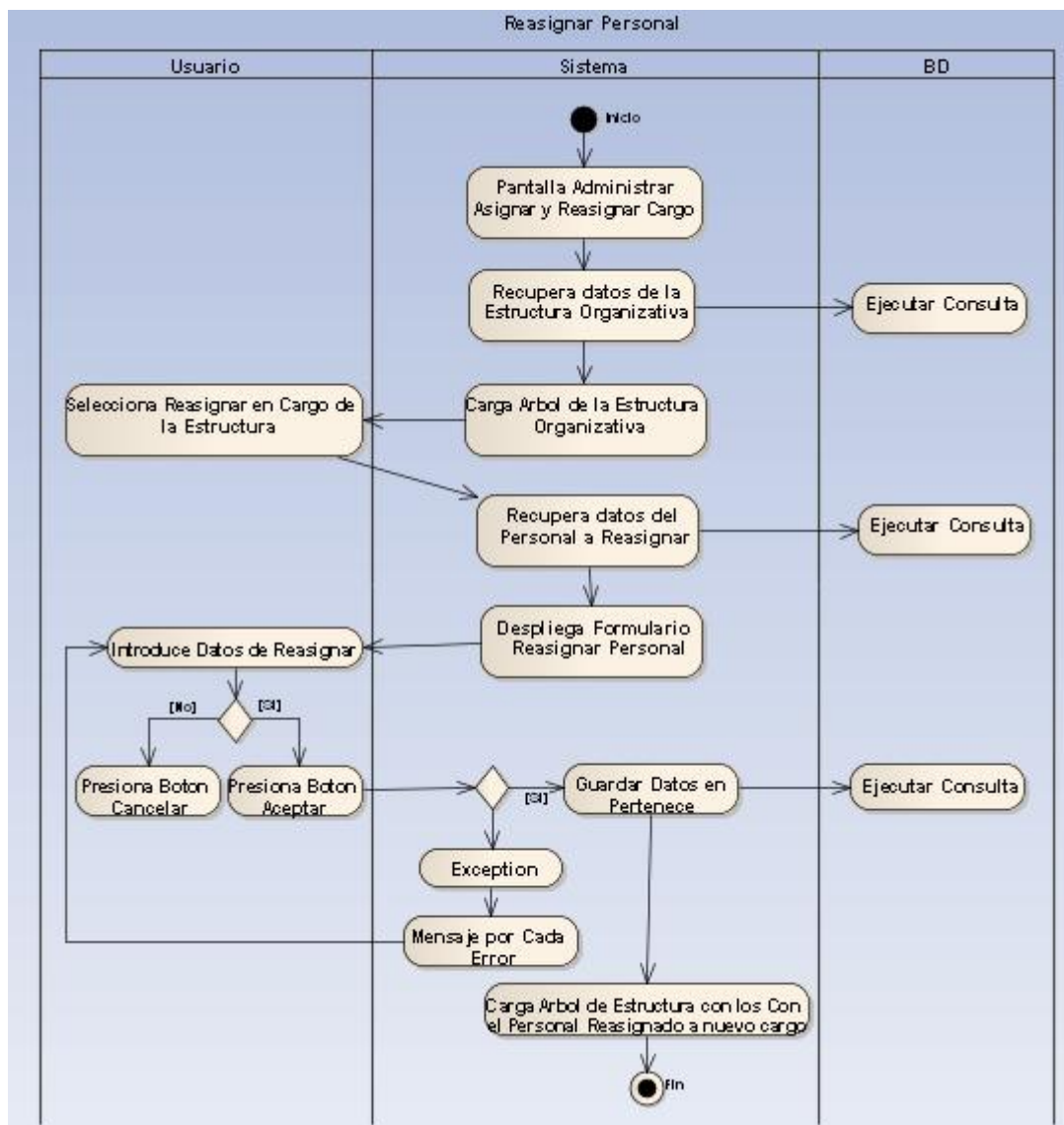


Figura 96. Diagrama de Actividad: Reasignar Cargo a Personal

1.1.2.2.3.9.1.4.17. Diagrama de Actividad: Caso de Uso Administrar Agencia

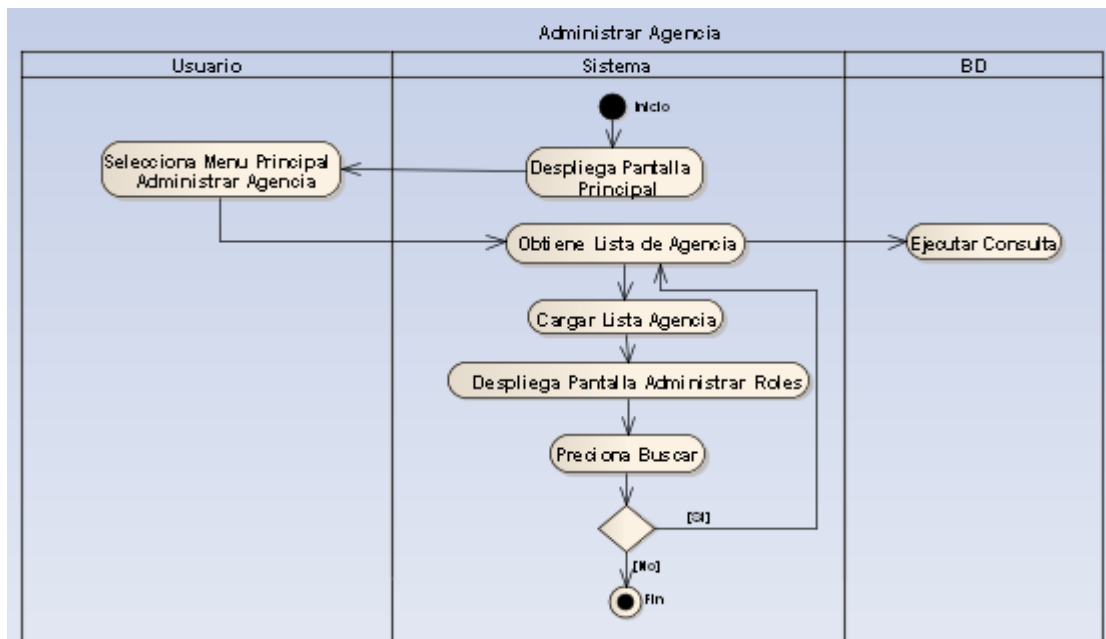


Figura 97. Diagrama de Actividad: Administrar Agencia

1.1.2.2.3.9.1.4.18. Diagrama de Actividad: Caso de Uso Adicionar Agencia

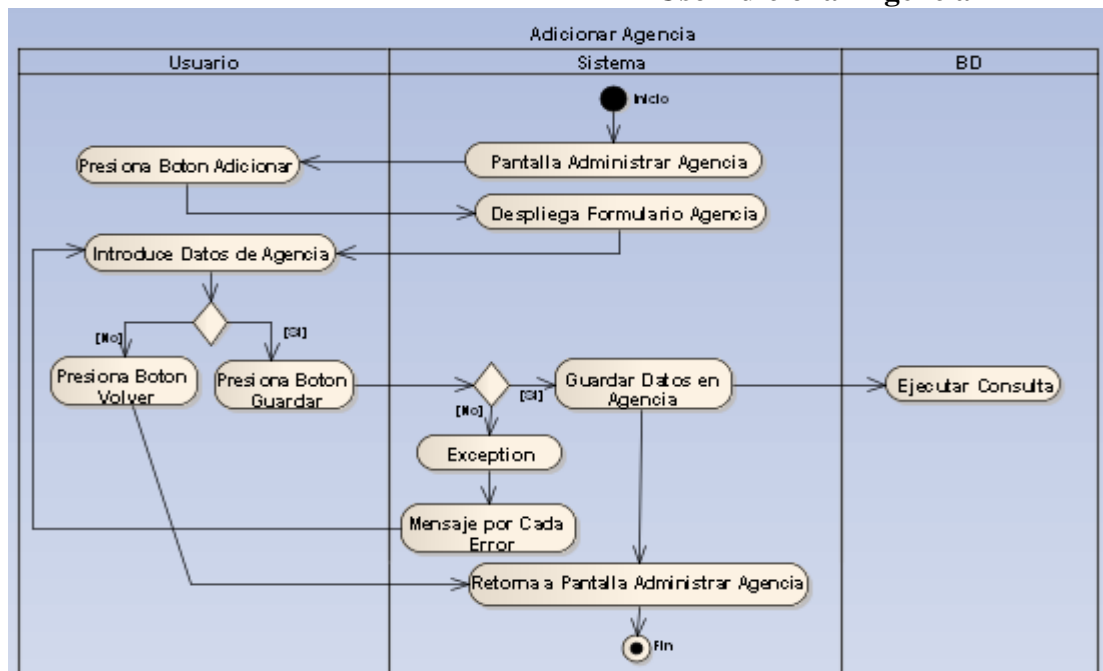


Figura 98. Diagrama de Actividad: Adicionar Agencia

1.1.2.2.3.9.1.4.19. Diagrama de Actividad: Caso de Uso Modificar Agencia

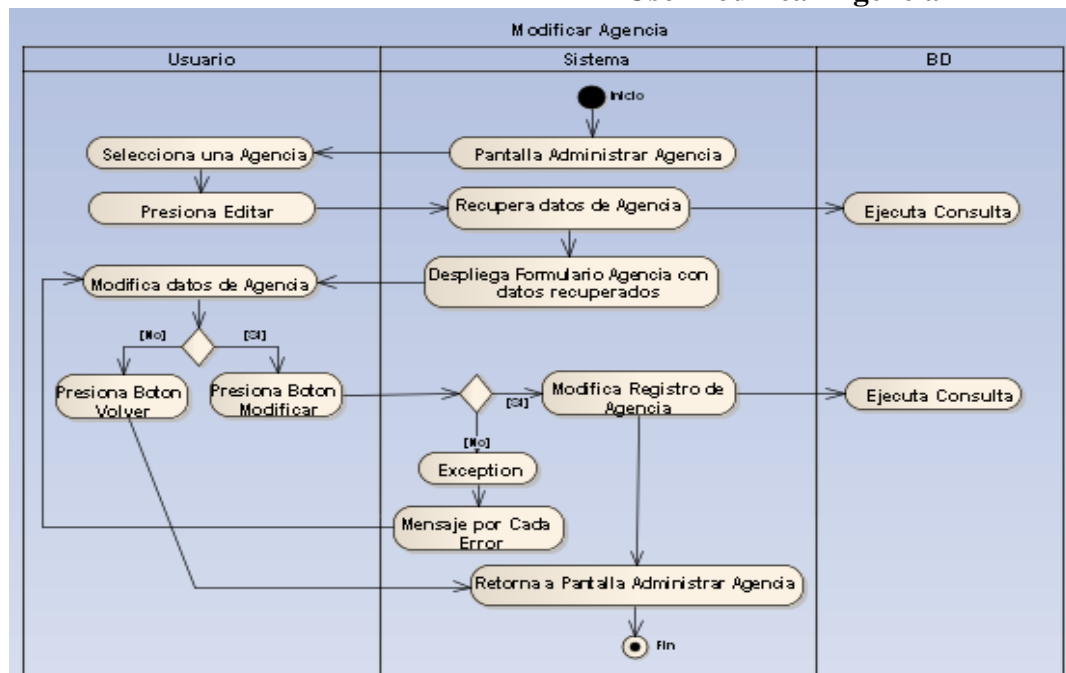


Figura 99. Diagrama de Actividad: Modificar Agencia

1.1.2.2.3.9.1.4.20. Diagrama de Actividad: Caso de Uso Eliminar Agencia

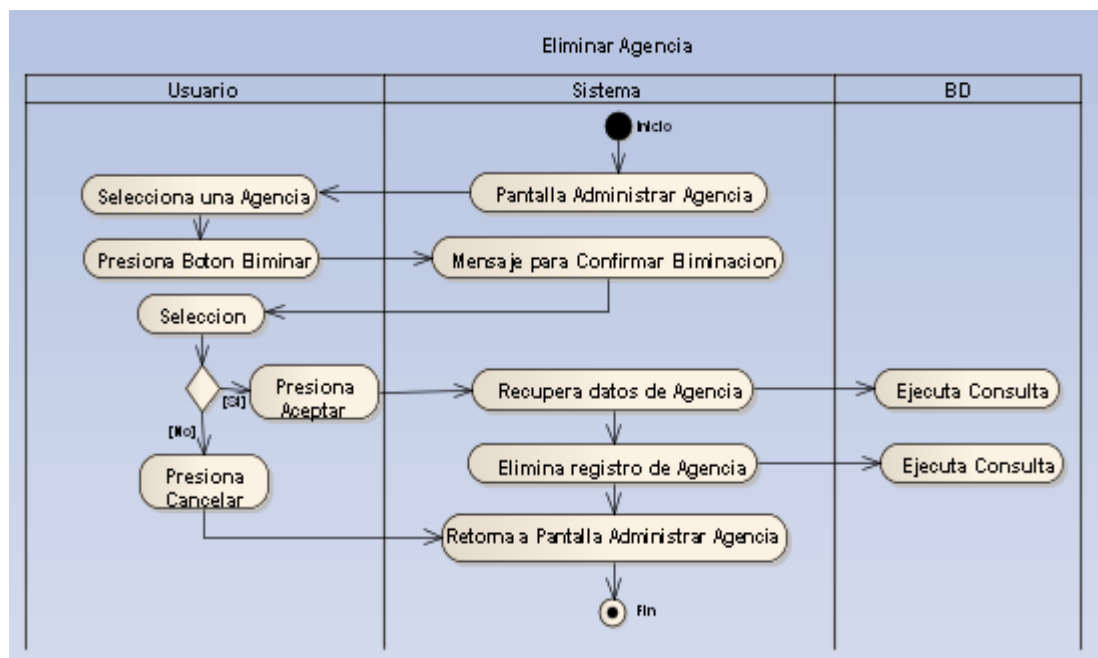


Figura 100. Diagrama de Actividad: Eliminar Agencia

1.1.2.2.3.9.1.4.21. Diagrama de Actividad: Caso de Uso Administrar Estructura Organizativa

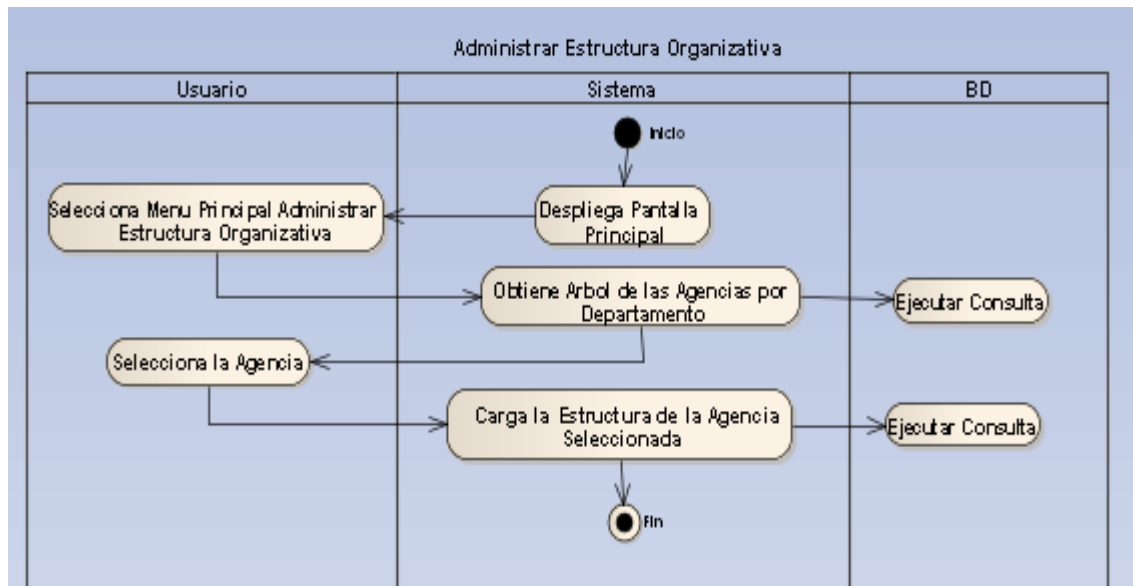


Figura 101. Diagrama de Actividad: Administrar Estructura Organizativa

1.1.2.2.3.9.1.4.22. Diagrama de Actividad: Caso de Uso Adicionar Área

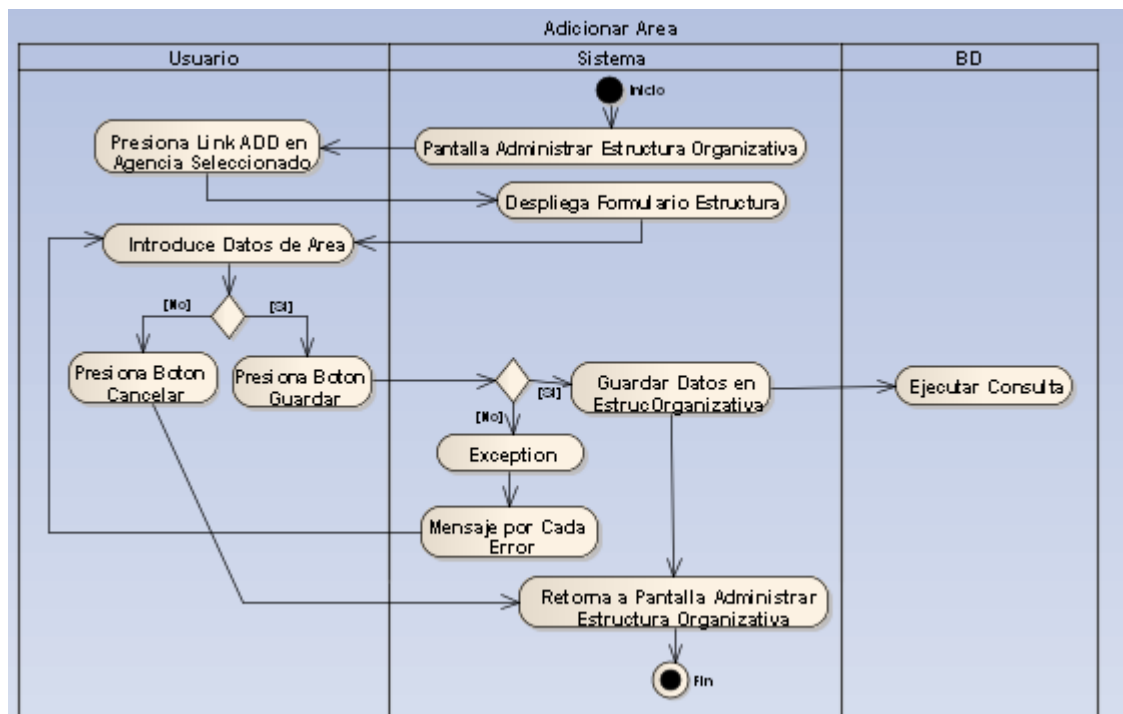


Figura 102. Diagrama de Actividad: Adicionar Área

1.1.2.2.3.9.1.4.23. Diagrama de Actividad: Caso de Uso Modificar Área

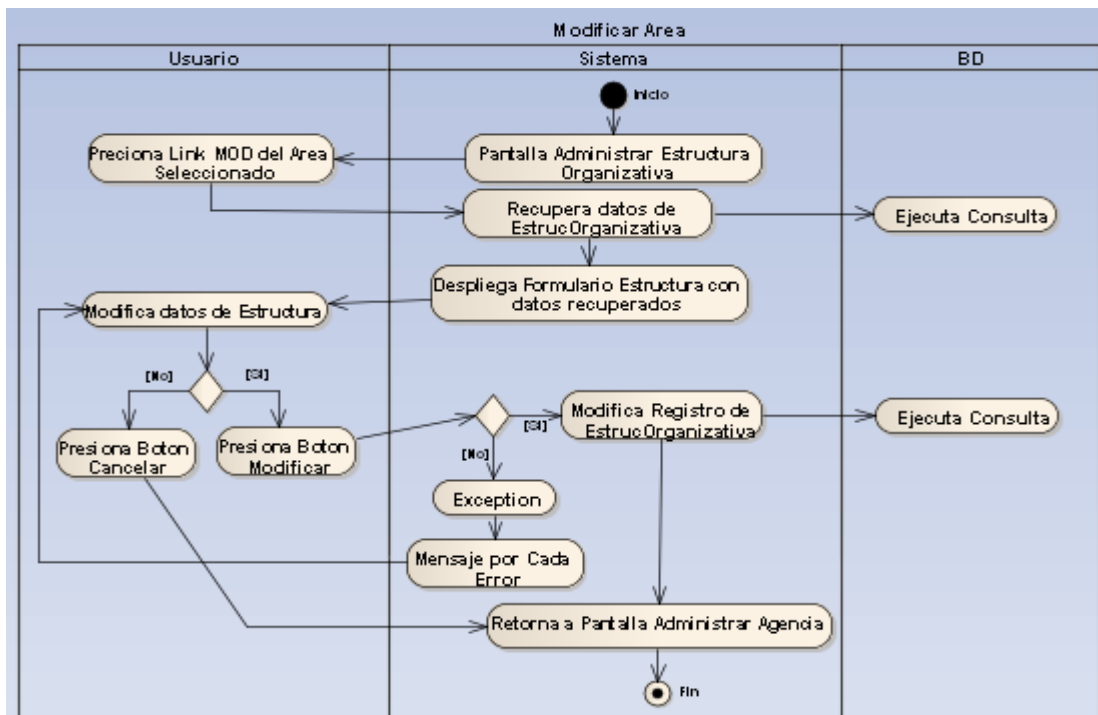


Figura 103. Diagrama de Actividad: Modificar Área

1.1.2.2.3.9.1.4.24. Diagrama de Actividad: Caso de Uso Eliminar Área

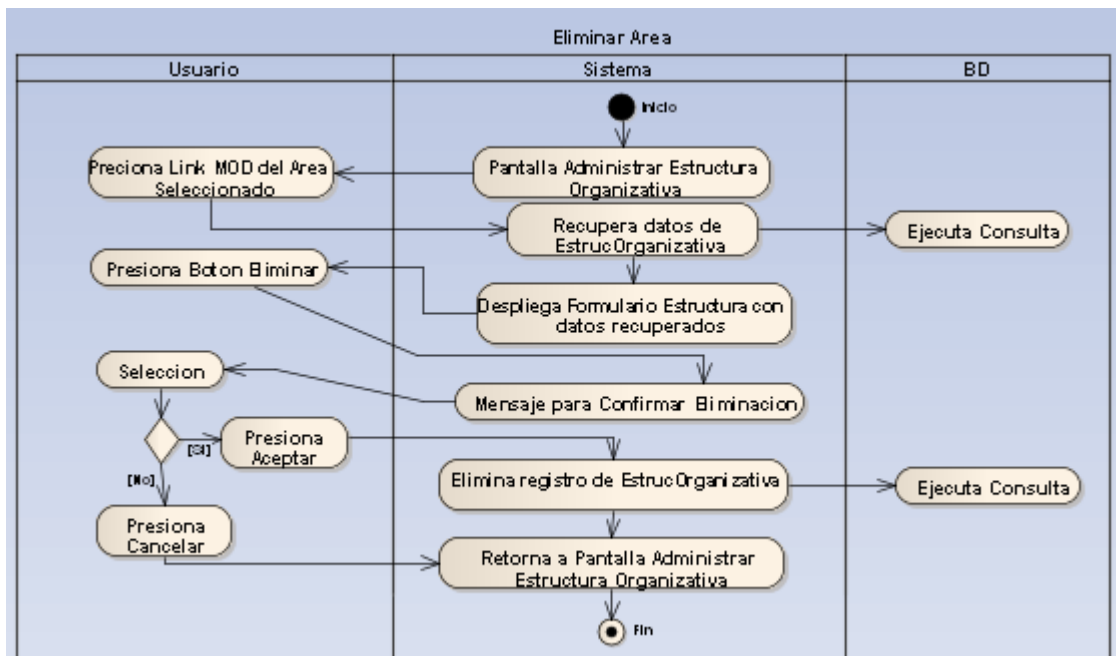


Figura 104. Diagrama de Actividad: Eliminar Área

1.1.2.2.3.9.1.4.25. Diagrama de Actividad: Caso de Uso Adicionar Cargo

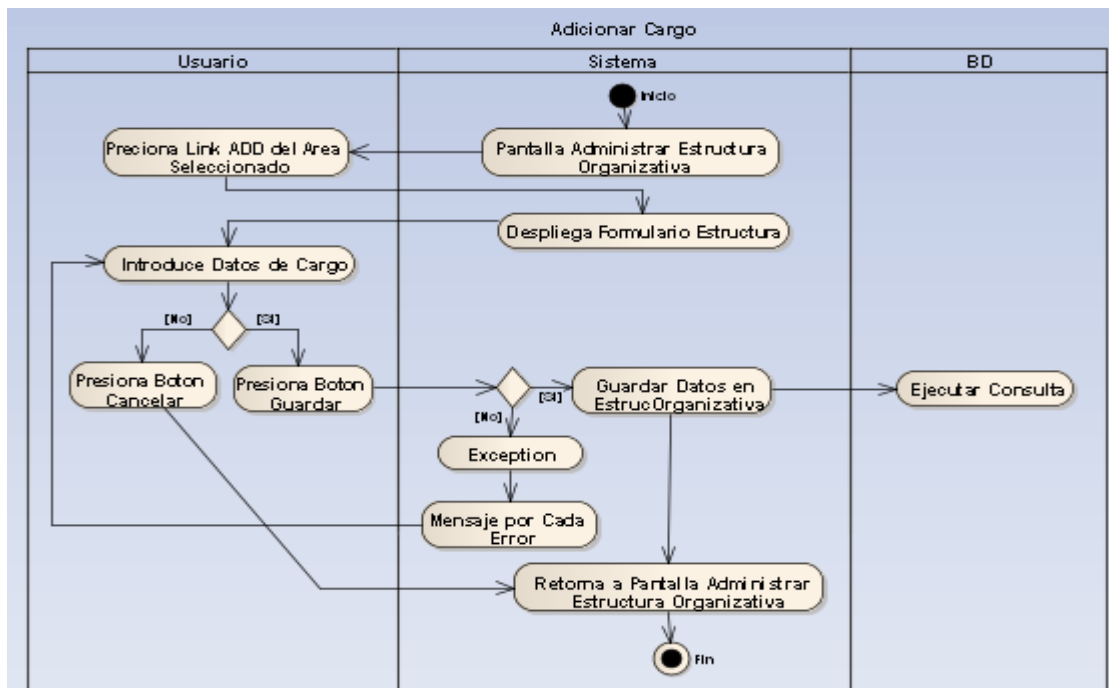


Figura 105. Diagrama de Actividad: Adicionar Cargo

1.1.2.2.3.9.1.4.26. Diagrama de Actividad: Caso de Uso Modificar Cargo

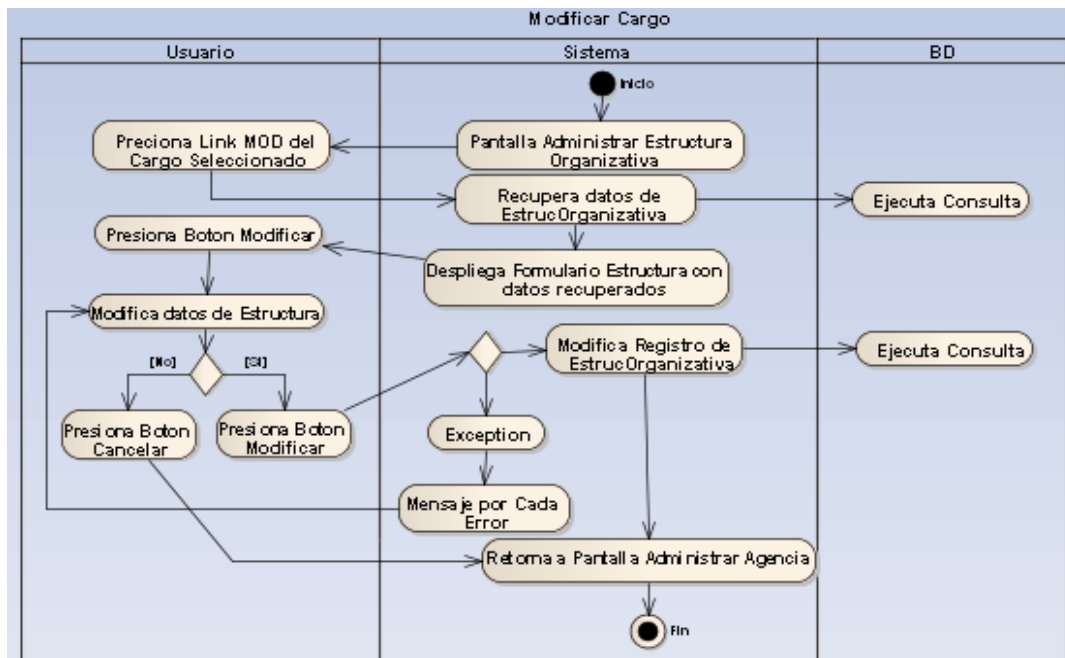


Figura 106. Diagrama de Actividad: Modificar Cargo

1.1.2.2.3.9.1.4.27. Diagrama de Actividad: Caso de Uso Eliminar Cargo

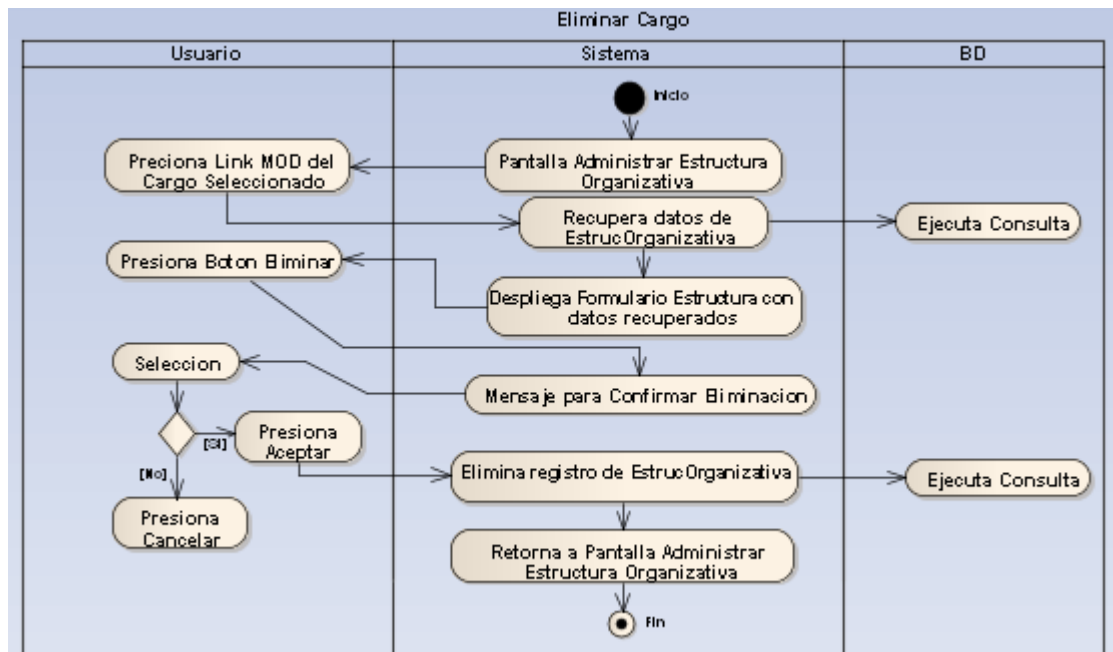


Figura 107. Diagrama de Actividad: Eliminar Cargo

1.1.2.2.3.9.1.4.28. Diagrama de Actividad: Caso de Uso Administrar Proveedor

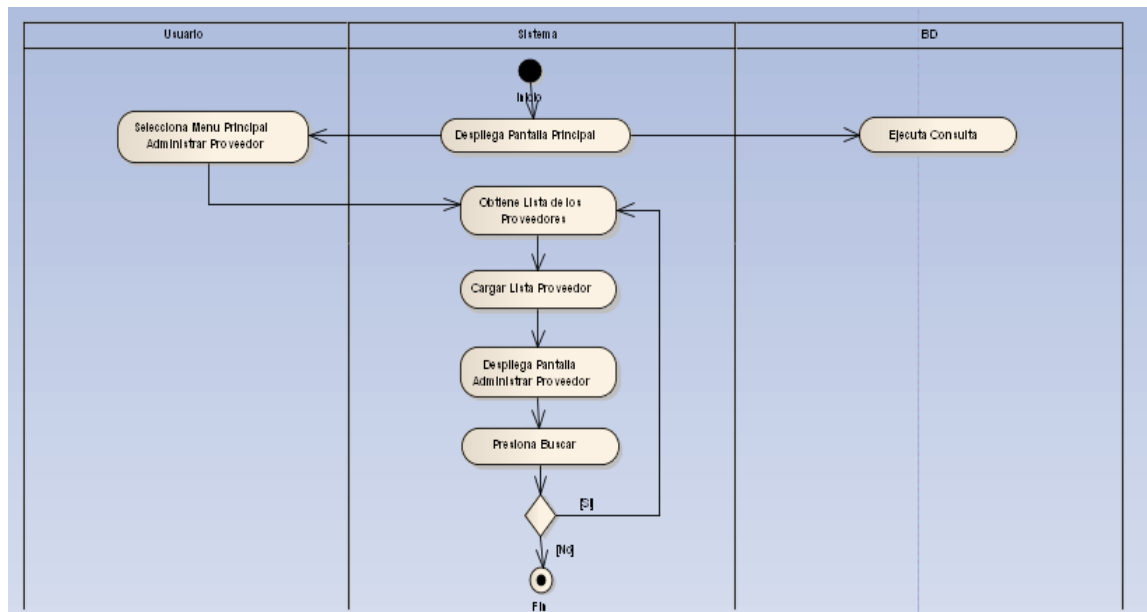


Figura 108. Diagrama de Actividad: Administrar Proveedor

1.1.2.2.3.9.1.4.29. Diagrama de Actividad: Caso de Uso Adicionar Proveedor

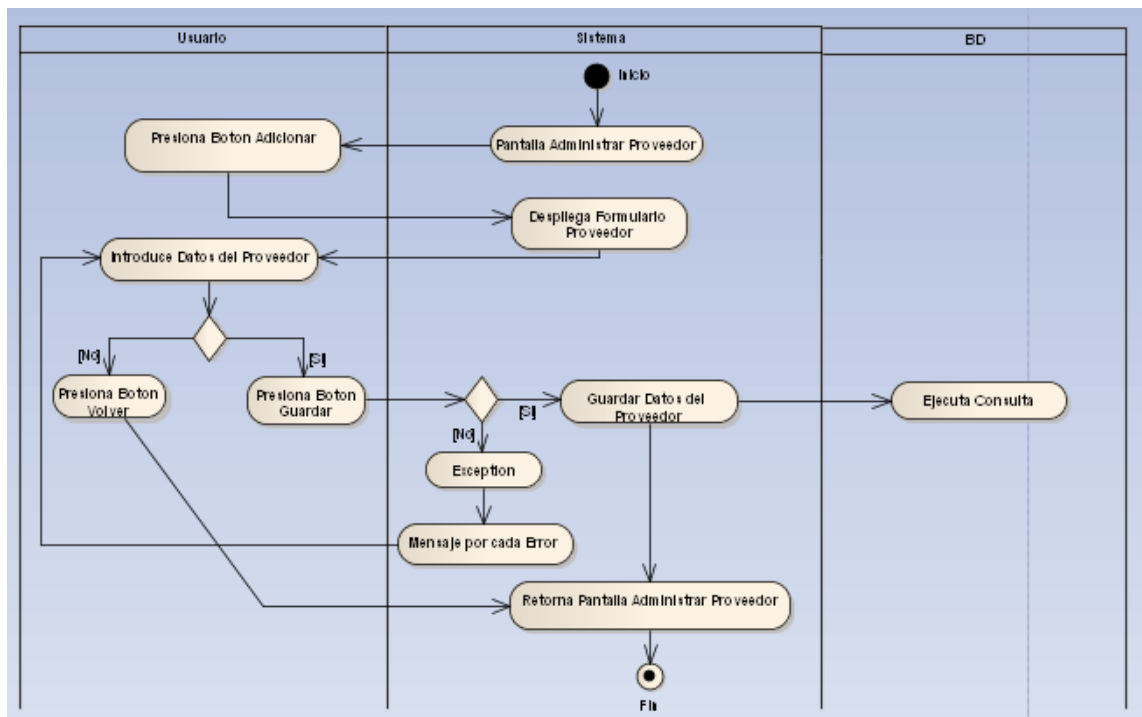


Figura 109. Diagrama de Actividad: Adicionar Proveedor

1.1.2.2.3.9.1.4.30. Diagrama de Actividad: Caso de Uso Modificar Proveedor

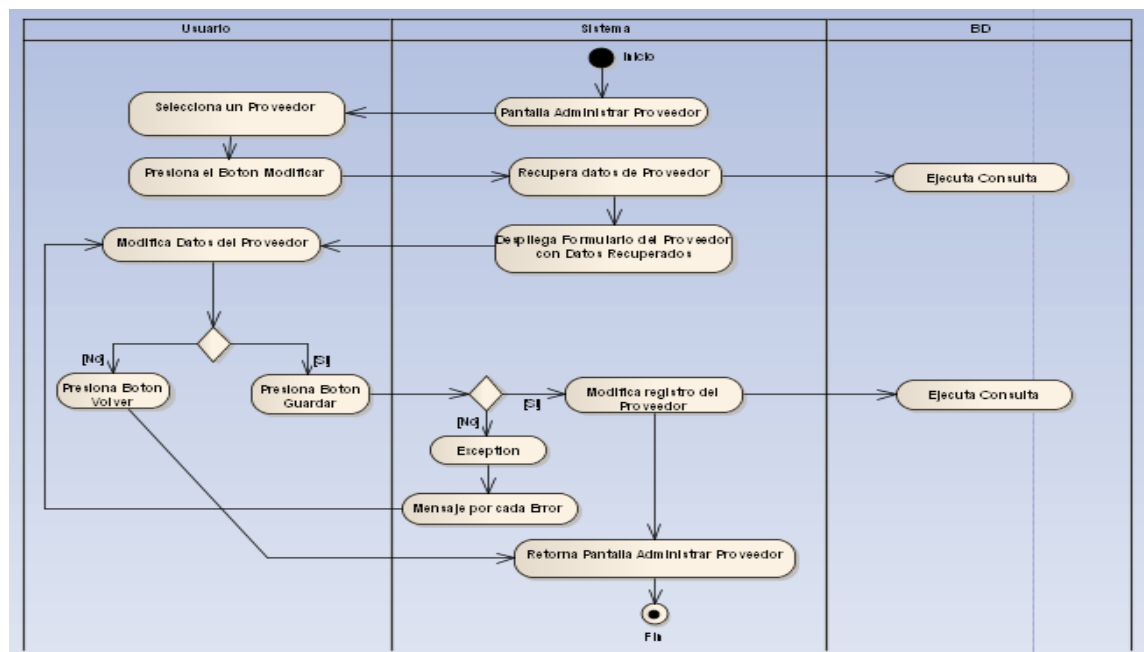


Figura 110. Diagrama de Actividad: Modificar Proveedor

1.1.2.2.3.9.1.4.31. Diagrama de Actividad: Caso de Uso Eliminar Proveedor

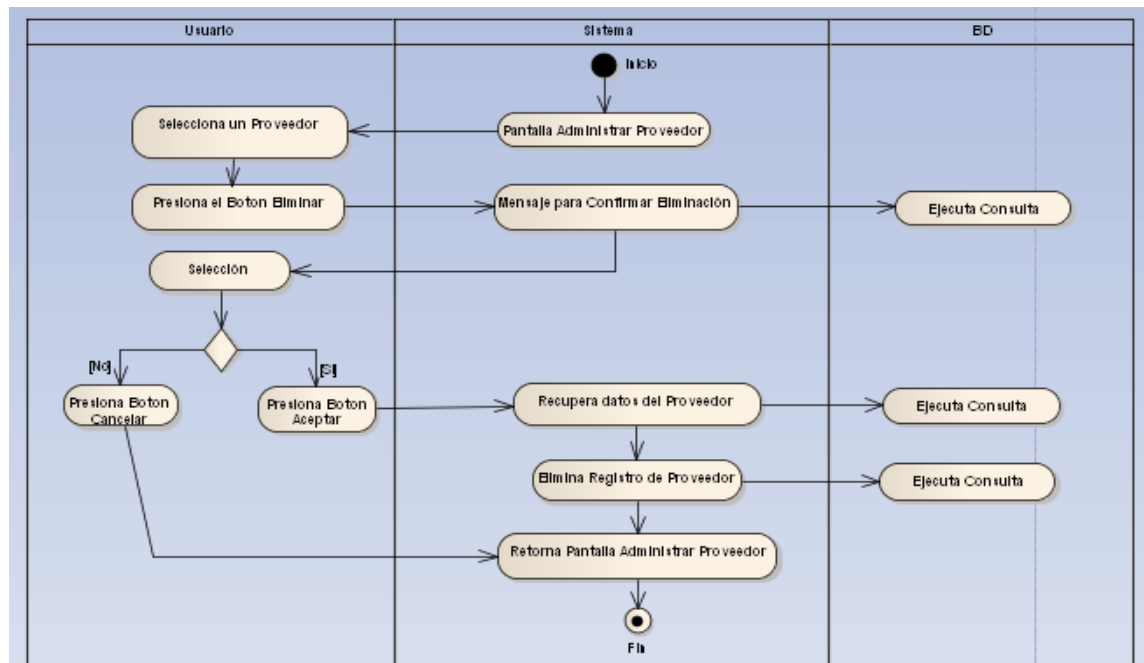


Figura 111. Diagrama de Actividad: Eliminar Proveedor

1.1.2.2.3.9.1.4.32. Diagrama de Actividad: Caso de Uso Administrar Productos

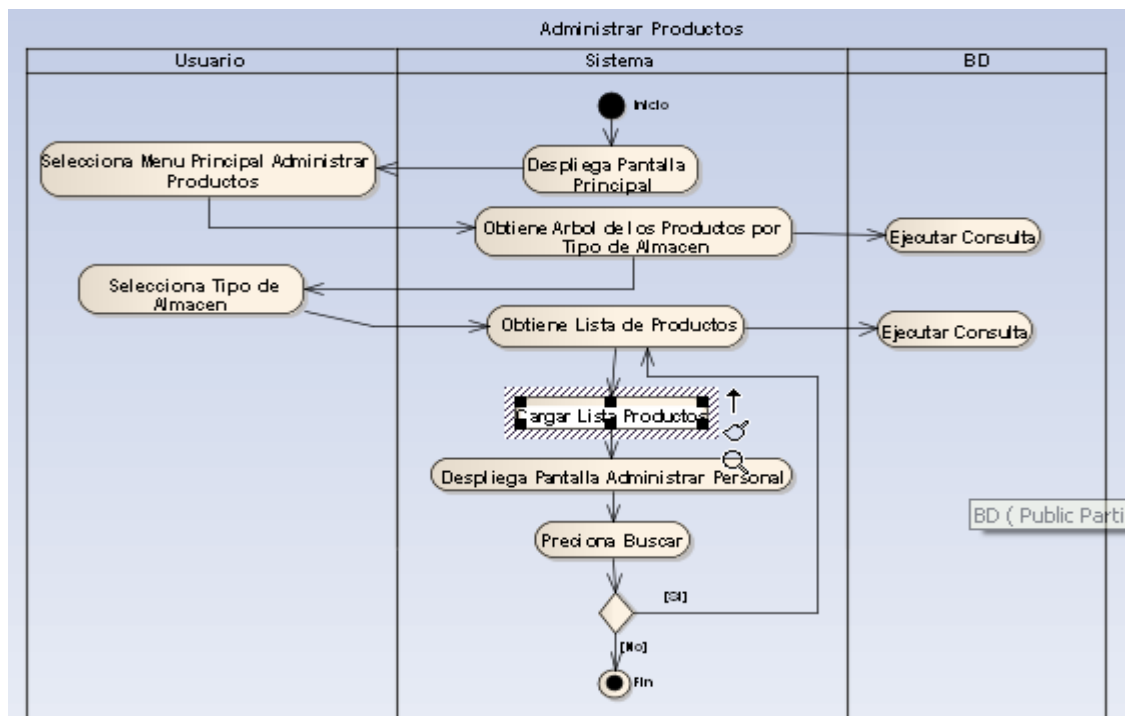


Figura 112. Diagrama de Actividad: Administrar Productos

1.1.2.2.3.9.1.4.33. Diagrama de Actividad: Caso de Uso Adicionar Producto

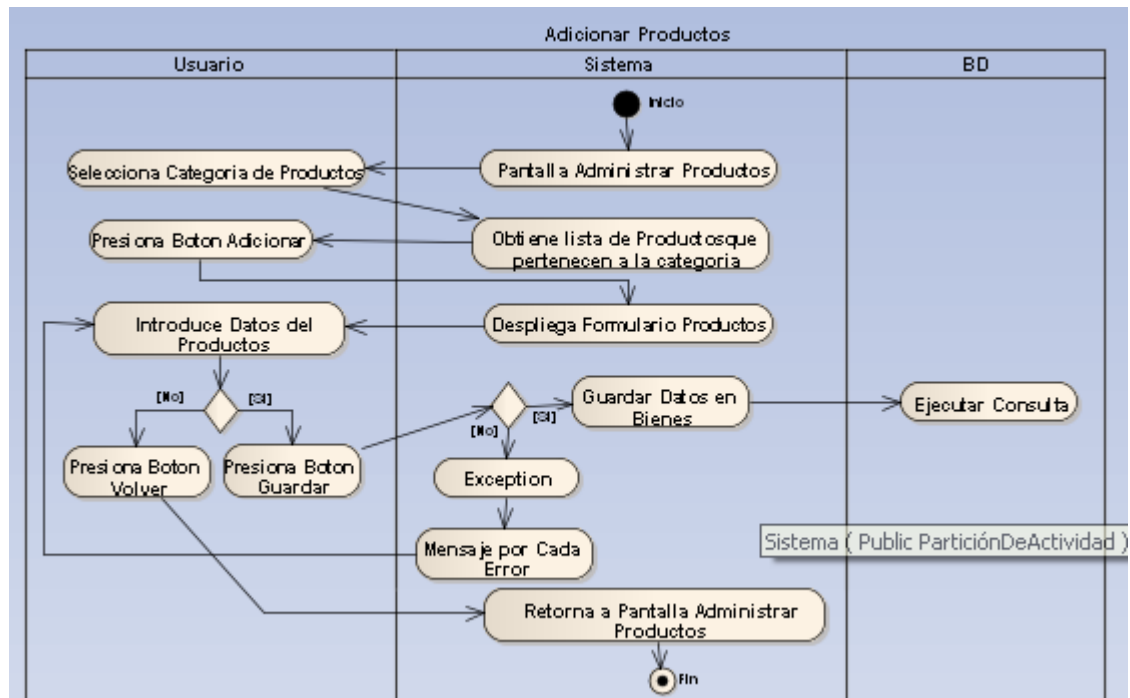


Figura 113. Diagrama de Actividad: Adicionar Producto

1.1.2.2.3.9.1.4.34. Diagrama de Actividad: Caso de Uso Modificar Producto

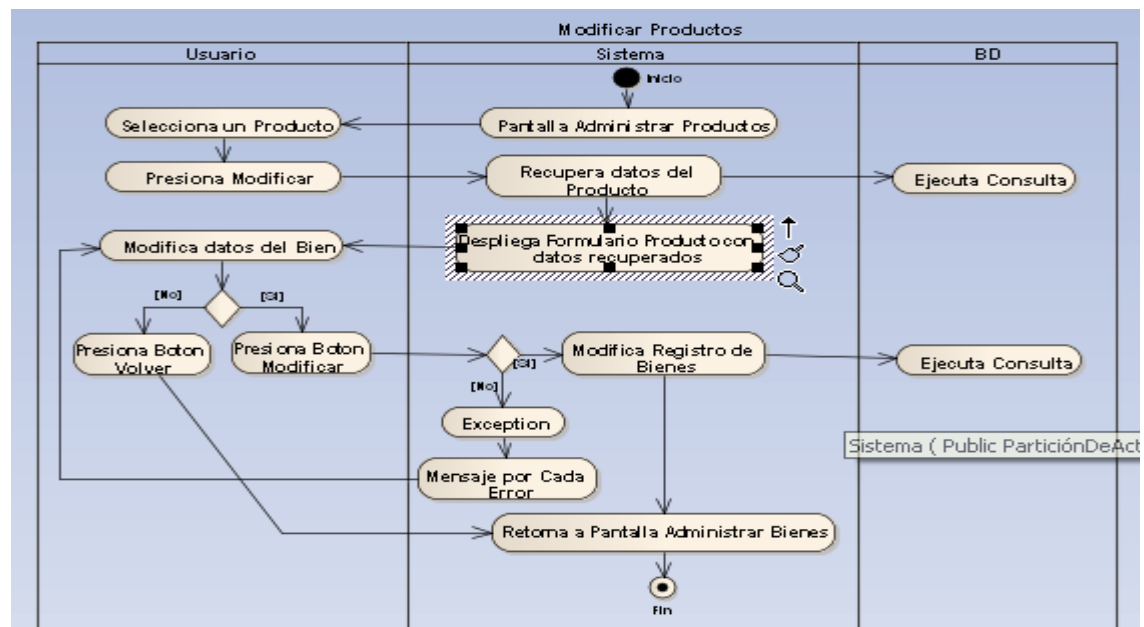


Figura 114. Diagrama de Actividad: Modificar Producto

1.1.2.2.3.9.1.4.35. Diagrama de Actividad: Caso de Uso Eliminar Producto

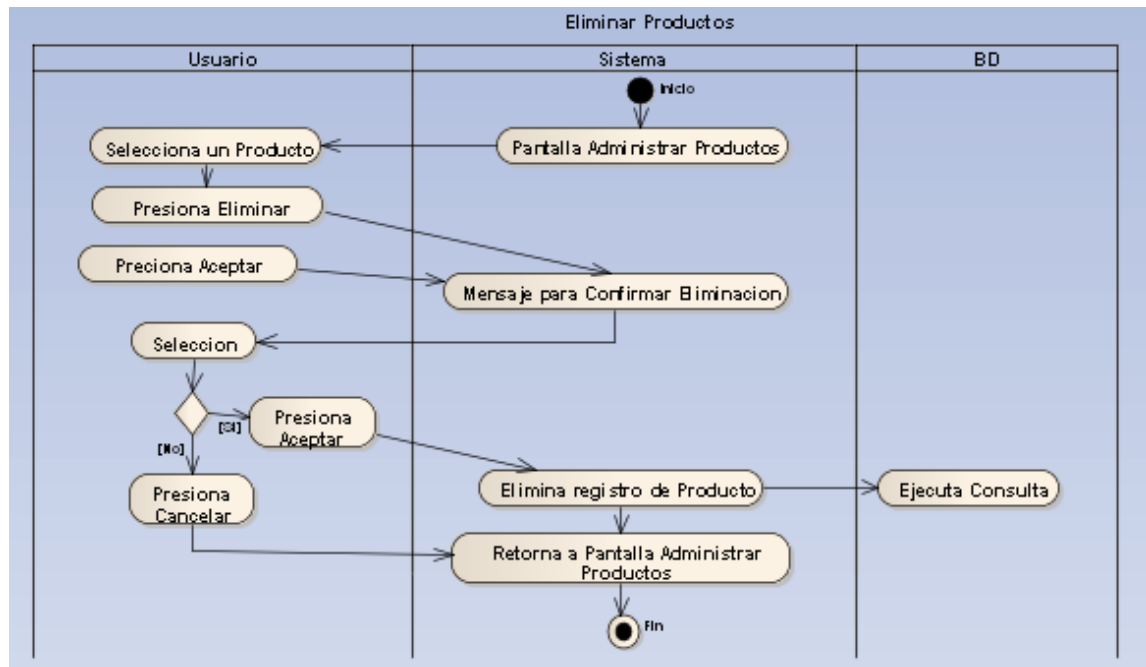


Figura 115. Diagrama de Actividad: Eliminar Producto

1.1.2.2.3.9.2. Modelado de Diagramas de Secuencias

Un Diagrama de Secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. En aplicaciones grandes además de los objetos se muestran también los componentes y Casos de Uso. El mostrar los componentes tiene sentido ya que se trata de objetos reutilizables, en cuanto a los Casos de Uso hay que recordar que se implementan como objetos cuyo rol es encapsular lo definido en el Caso de Uso.

Para mostrar la interacción con el usuario o con otro Sistema se introducen en los Diagramas de Secuencia las *boundary classes*. En las primeras fases de diseño el propósito de introducir estas clases es capturar y documentar los requisitos de interfaz, pero no el mostrar cómo se va a implementar dicha interfaz.

Los Diagramas de Secuencia, formalmente diagramas de traza de eventos o de interacción de objetos, se utilizan con frecuencia para validar los Casos de Uso.

Documentan el diseño desde el punto de vista de los Casos de Uso. Observando qué mensajes se envían a los objetos, componentes o Casos de Uso y viendo a grosso modo cuanto tiempo consume el método invocado, los Diagramas de Secuencia nos ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos. A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del Diagrama de Clases.

1.1.2.2.3.9.2.1. Diagrama de Secuencias

1.1.2.2.3.9.2.1.1. Diagrama de Secuencia: Caso de Uso Ingresar al Sistema

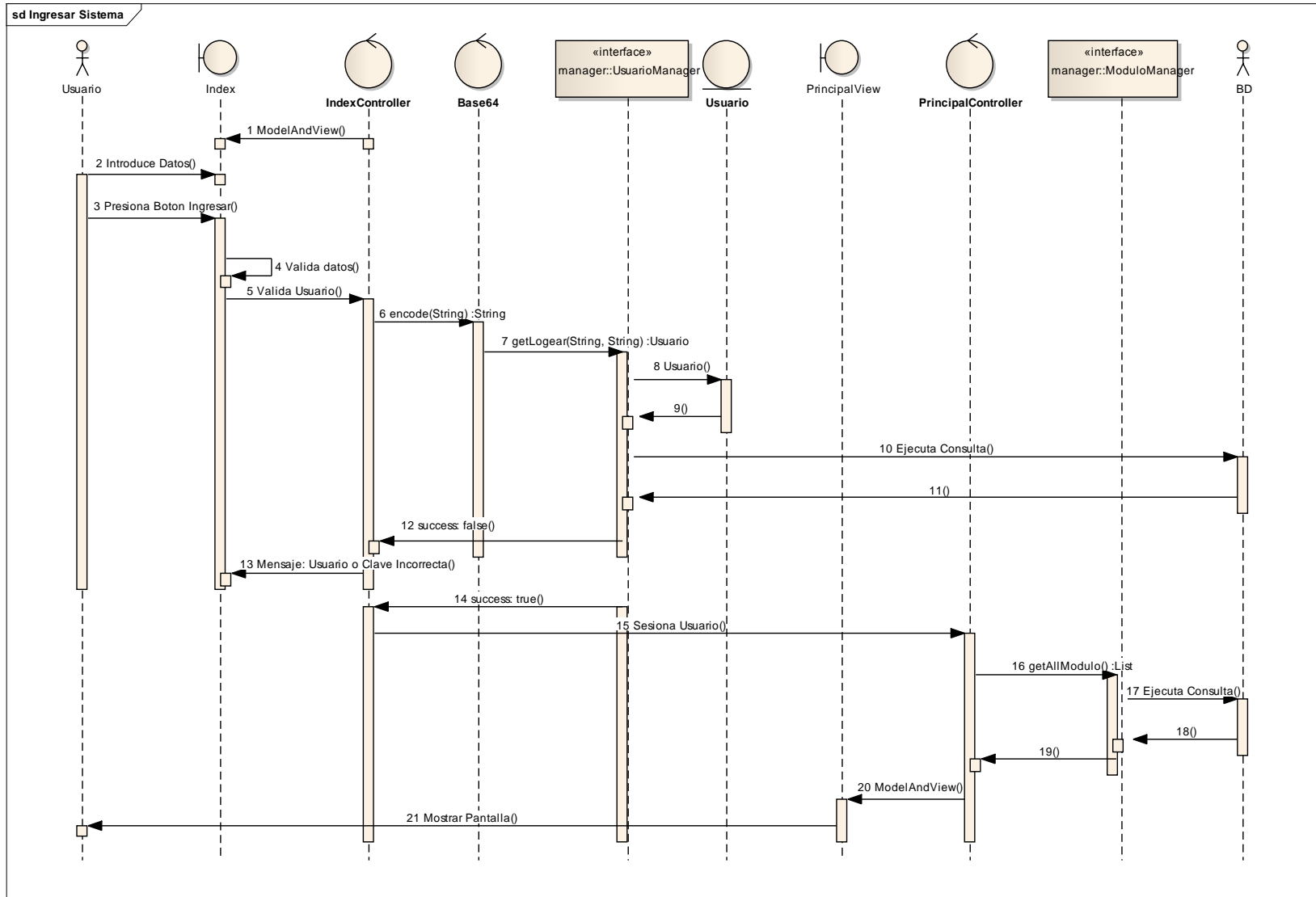


Figura 116. Diagrama de Secuencia: CU Ingresar al Sistema

1.1.2.2.3.9.2.1.2. Diagrama de Secuencia: Caso de Uso Administrar Personal

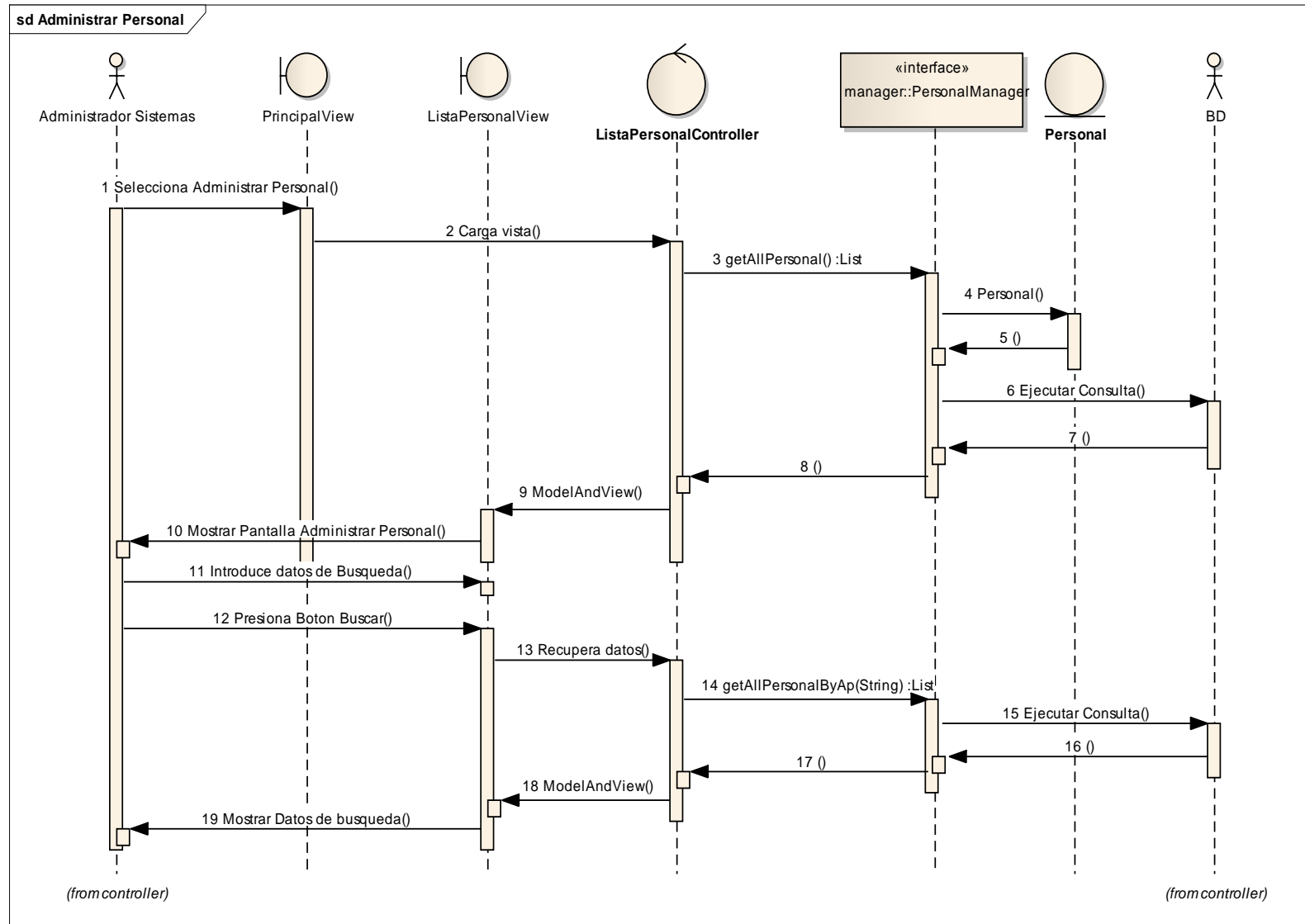


Figura 117. Diagrama de Secuencia: CU Administrar Personal

1.1.2.2.3.9.2.1.3. Diagrama de Secuencia: Caso de Uso Adicionar Personal

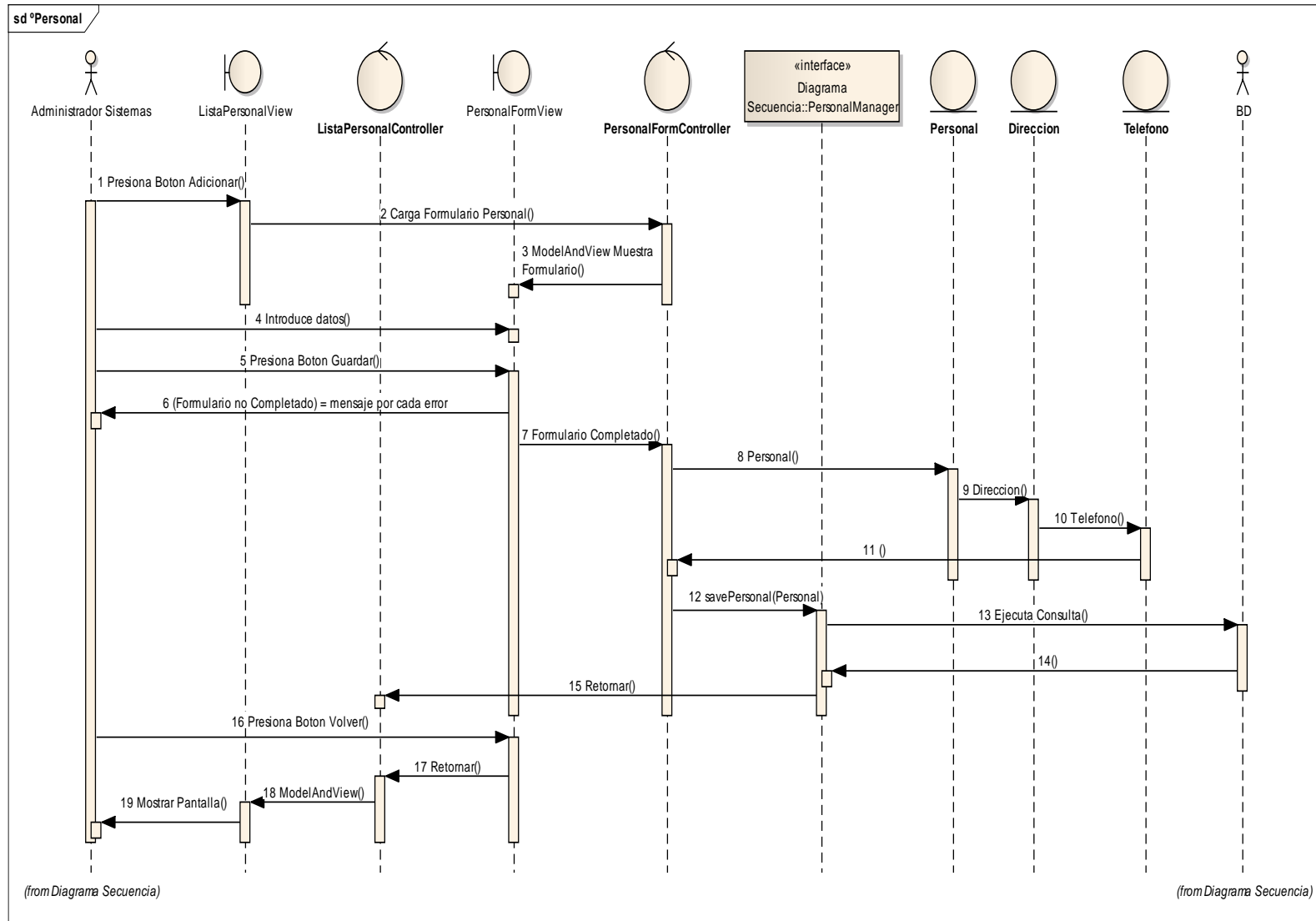


Figura 118. Diagrama de Secuencia: CU Adicionar Personal

1.1.2.2.3.9.2.1.4. Diagrama de Secuencia: Caso de Uso Modificar Personal

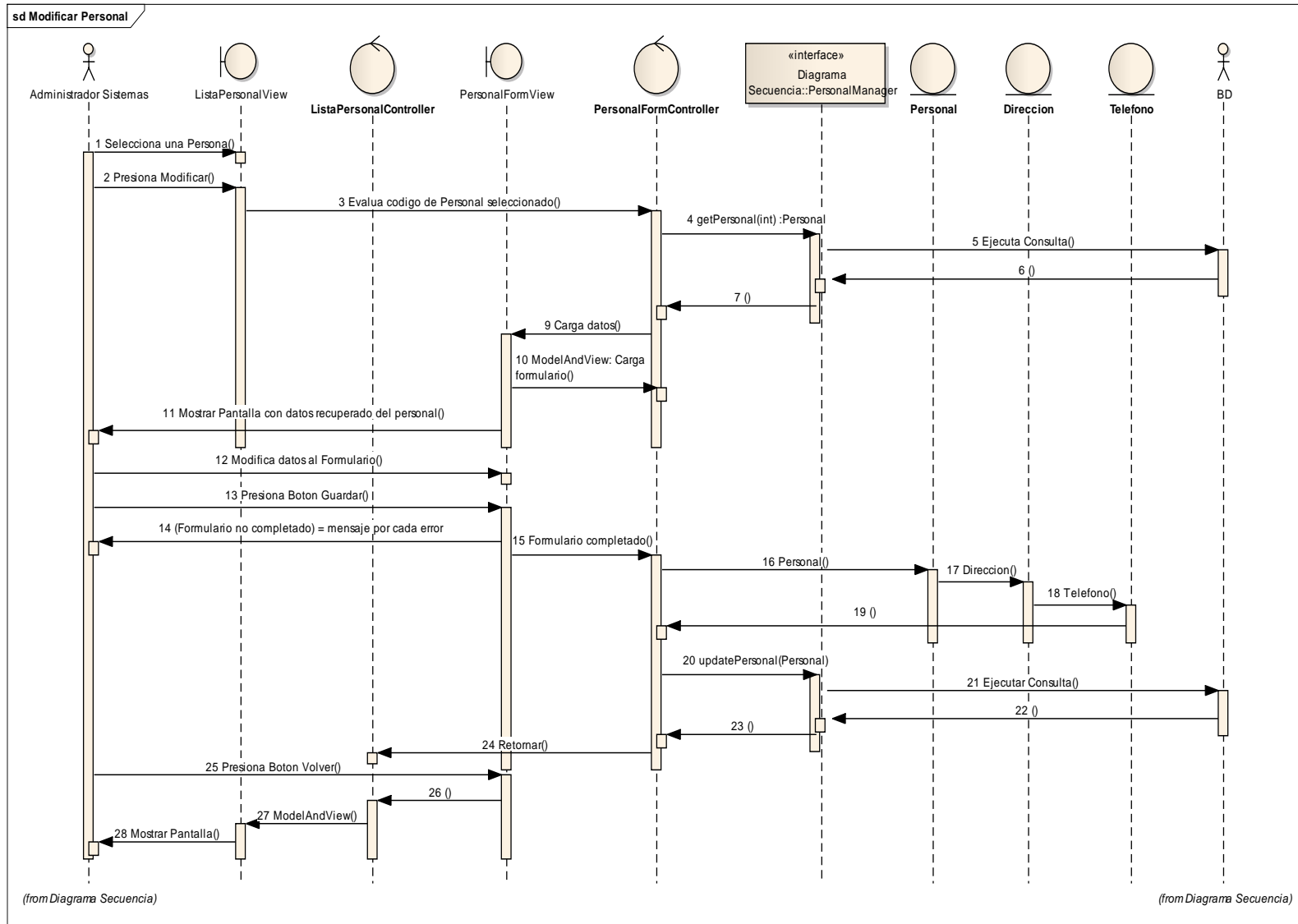


Figura 119. Diagrama de Secuencia: CU Modificar Personal

1.1.2.2.3.9.2.1.5. Diagrama de Secuencia: Caso de Uso Eliminar Personal

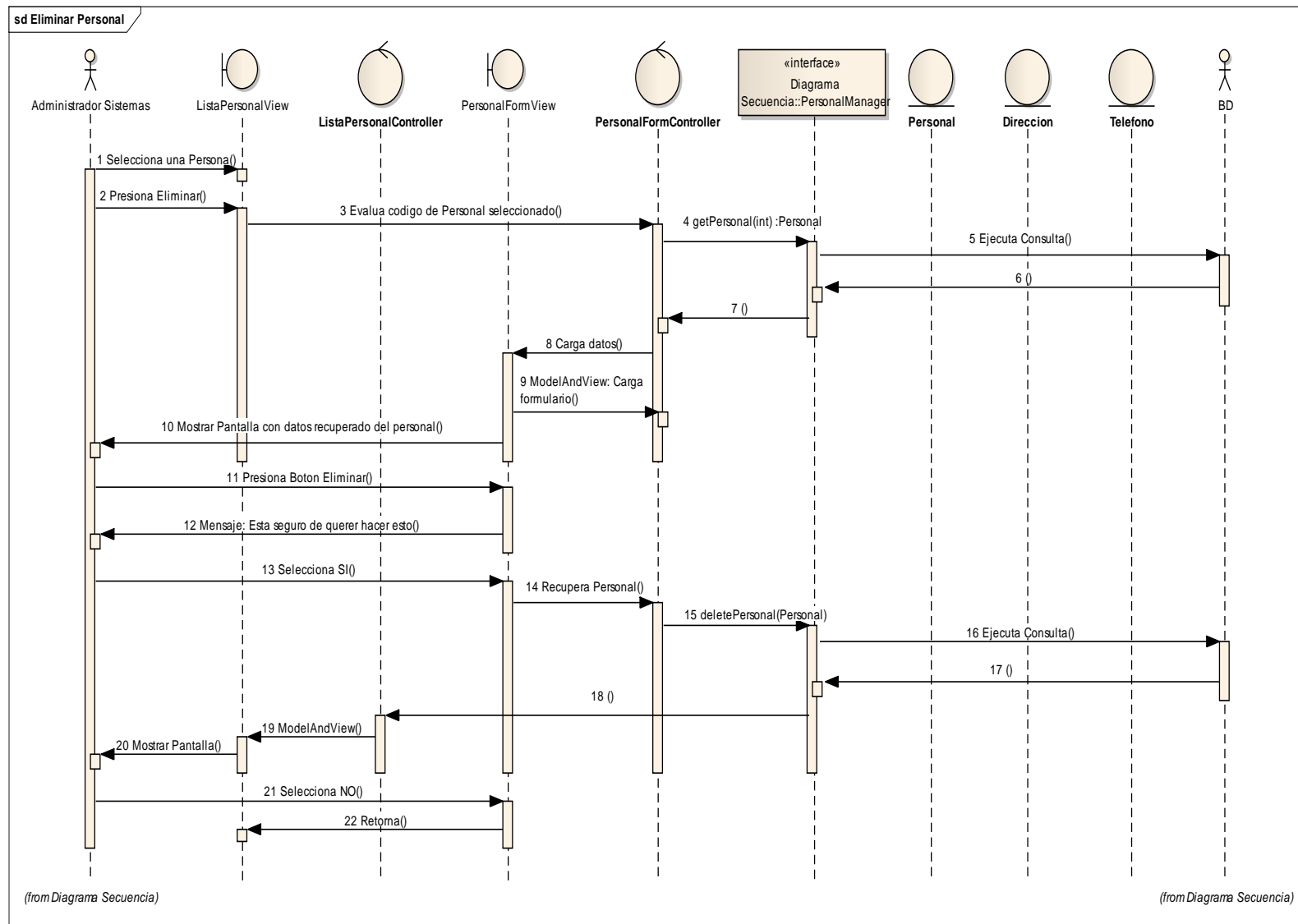


Figura 120. Diagrama de Secuencia: CU Eliminar Personal

1.1.2.2.3.9.2.1.6. Diagrama de Secuencia: Caso de Uso Adicionar Usuario

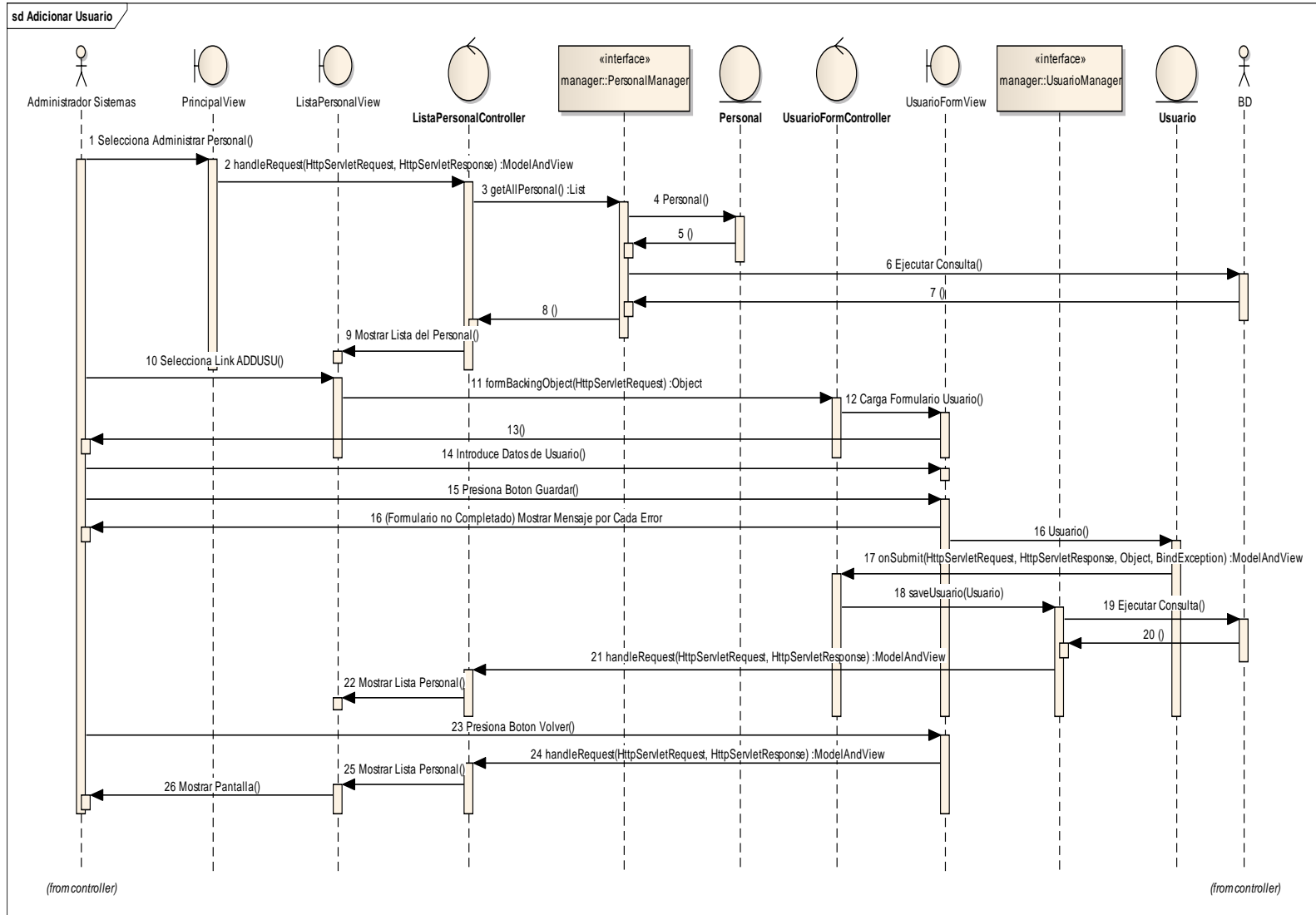


Figura 121. Diagrama de Secuencia: CU Adicionar Usuario

1.1.2.2.3.9.2.1.7. Diagrama de Secuencia: Caso de Uso Modificar Usuario

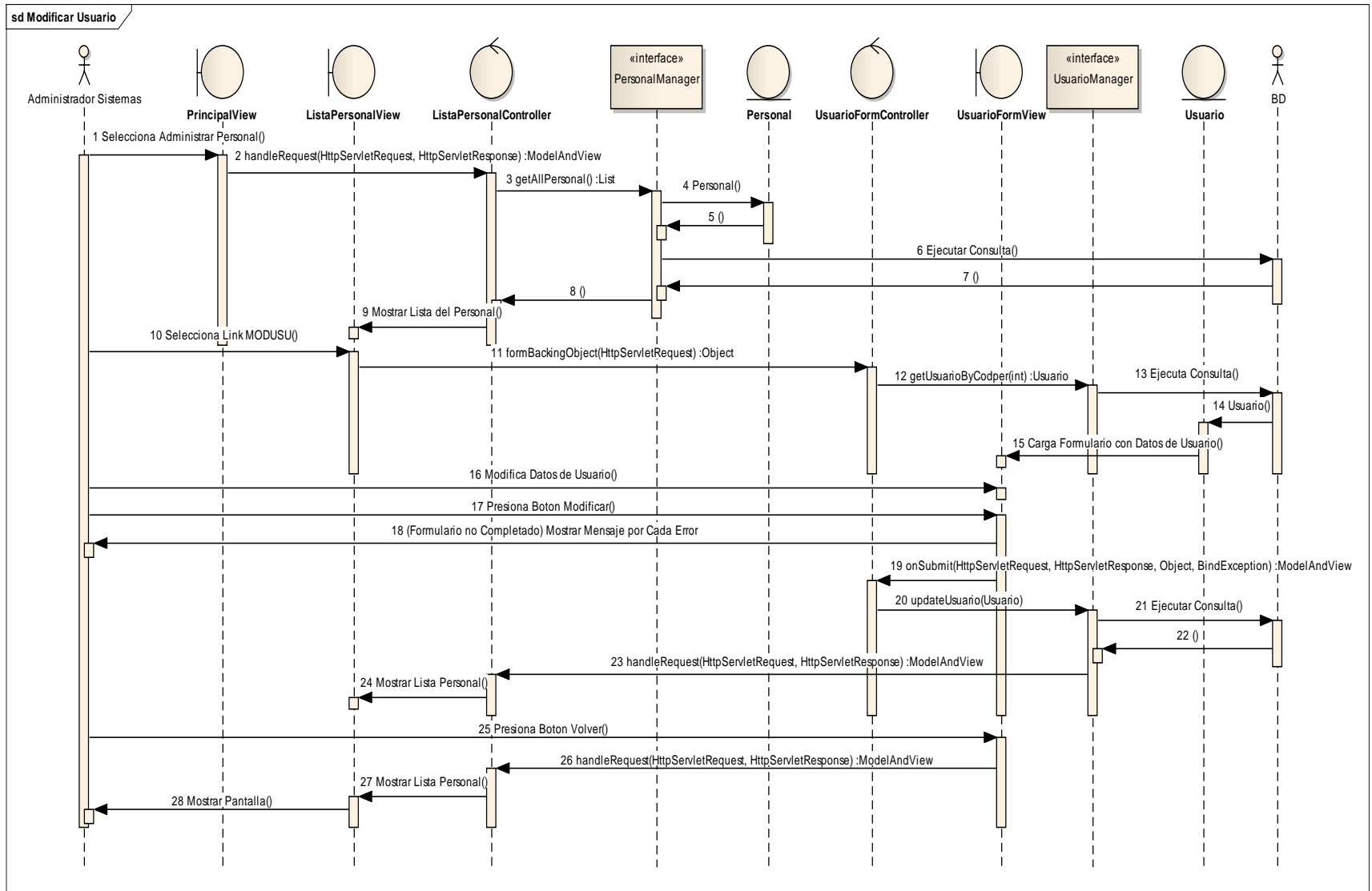


Figura 122. Diagrama de Secuencia: CU Modificar Usuario

1.1.2.2.3.9.2.1.8. Diagrama de Secuencia: Caso de Uso Eliminar Usuario

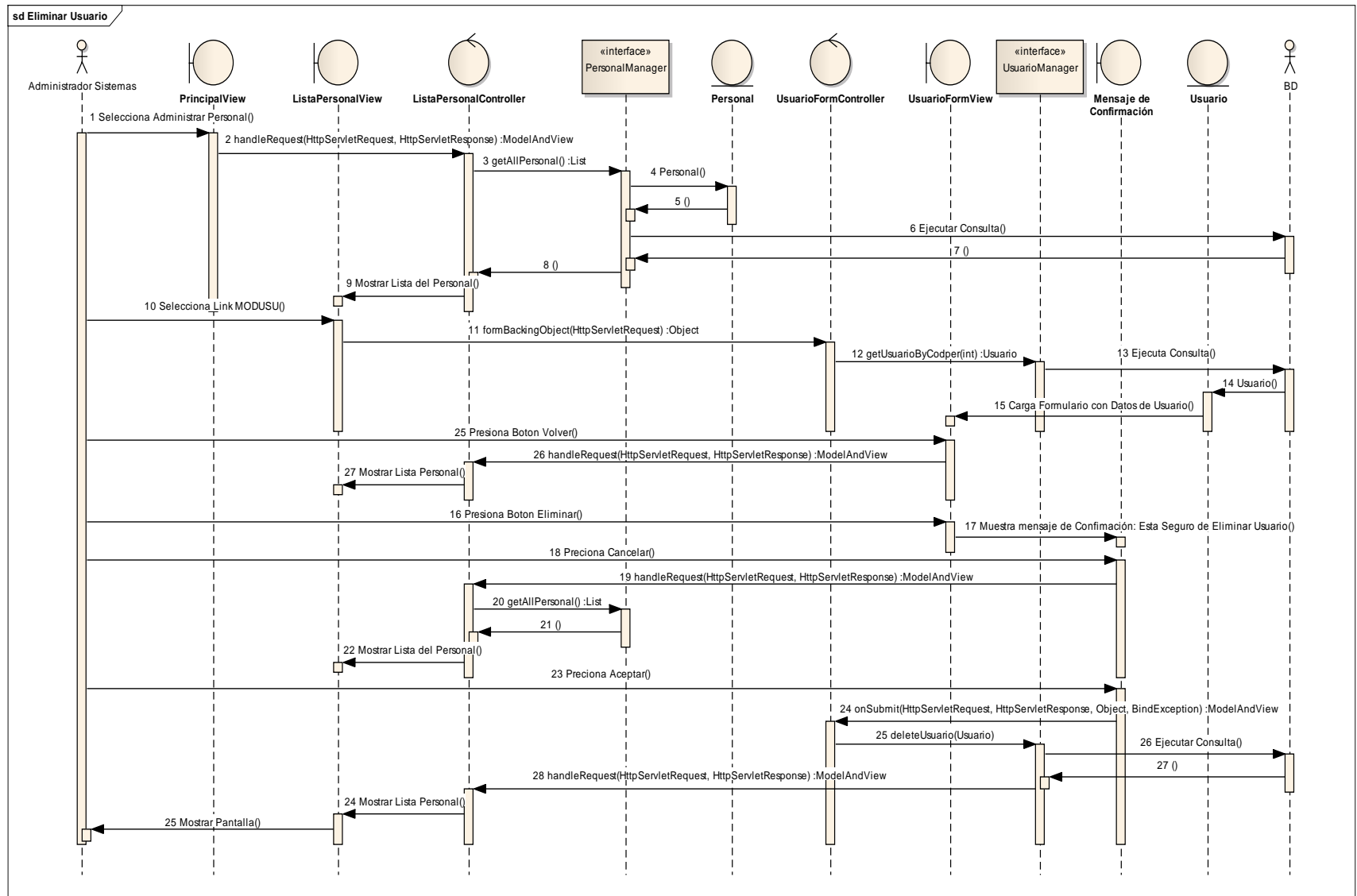


Figura 123. Diagrama de Secuencia: CU Eliminar Usuario

1.1.2.2.3.9.2.1.9. Diagrama de Secuencia: Caso de Uso Mostrar Kardex Personal

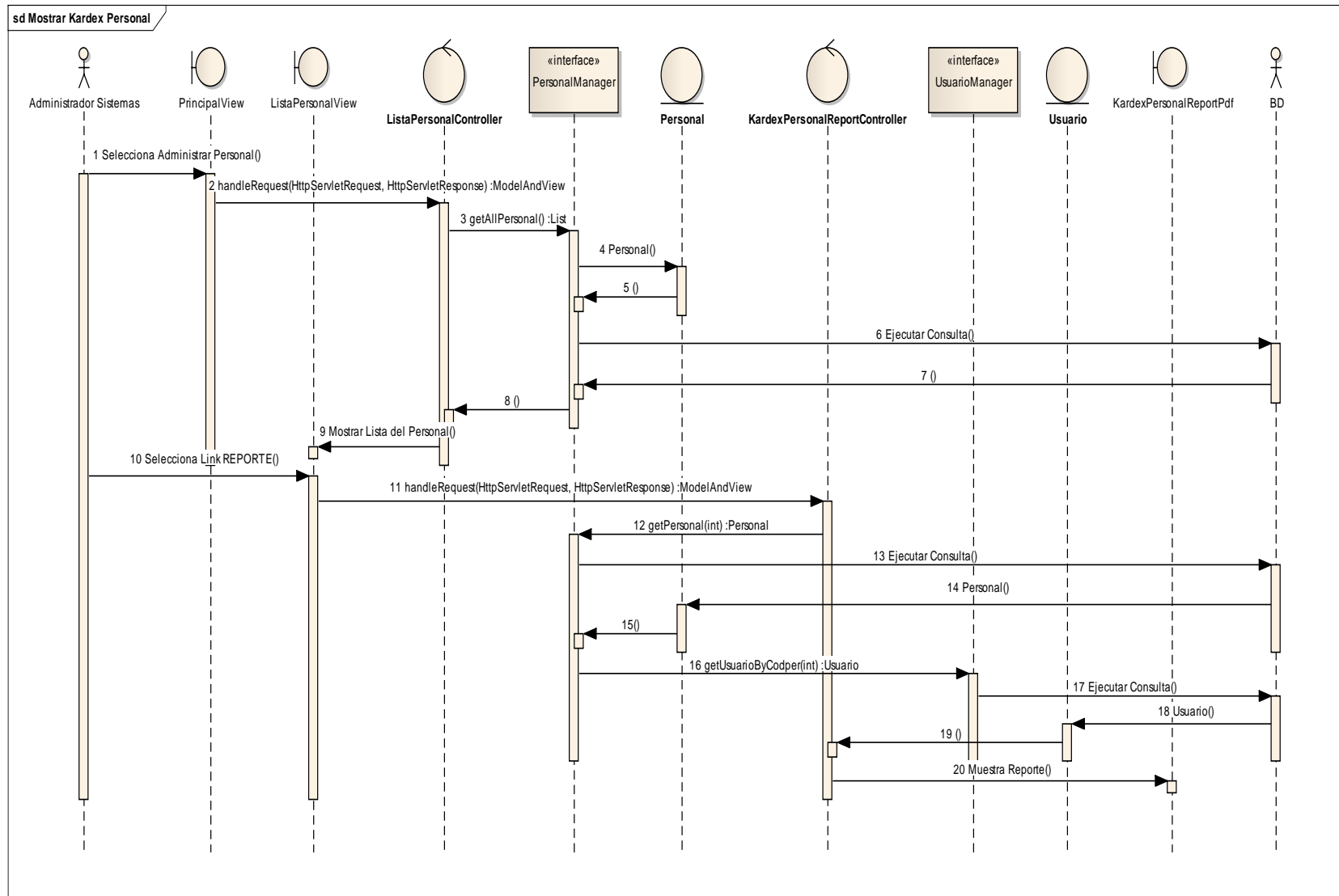


Figura 124. Diagrama de Secuencia: CU Mostrar Kardex Personal

1.1.2.2.3.9.2.1.10. Diagrama de Secuencia: Caso de Uso Administrar Roles

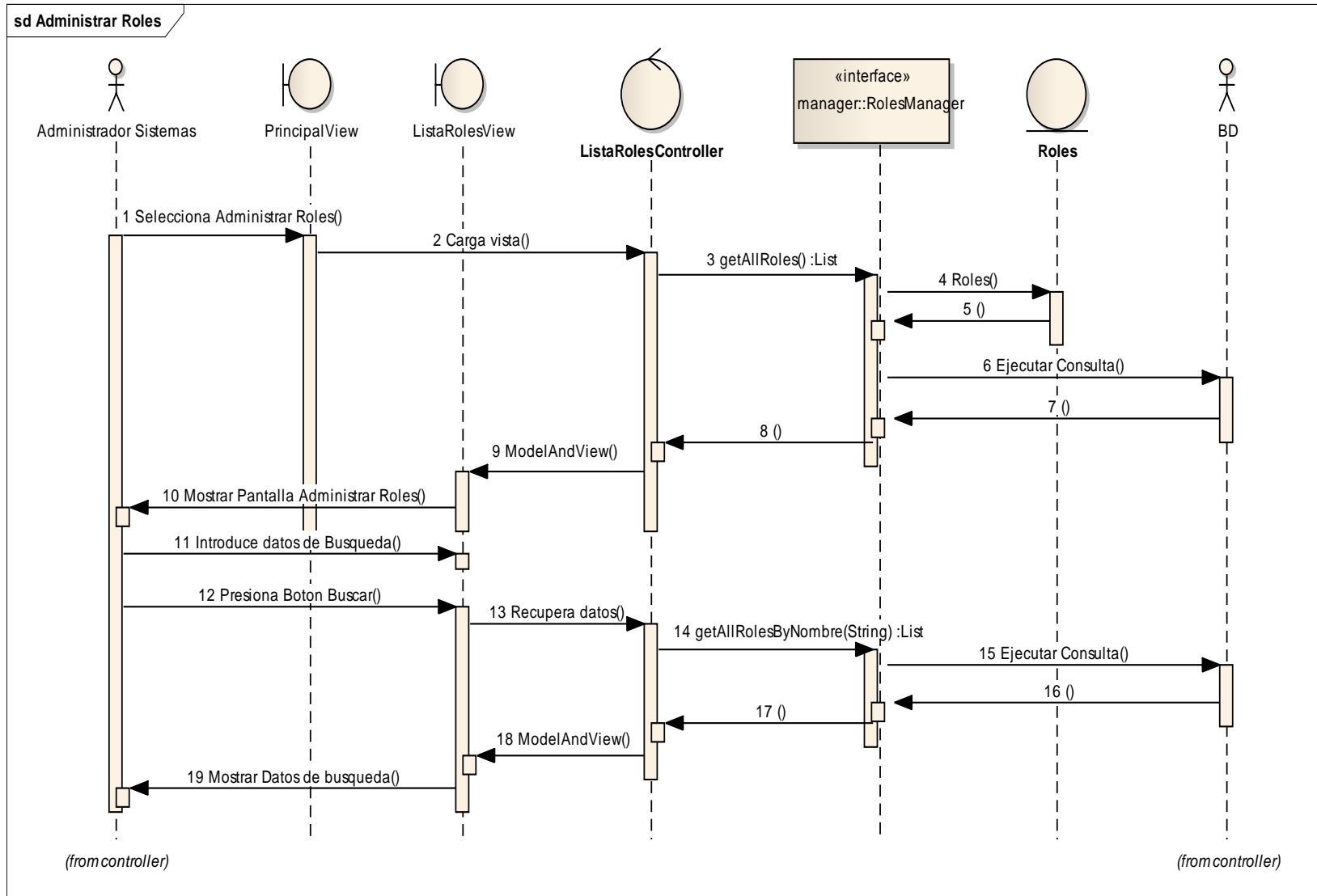


Figura 125. Diagrama de Secuencia: CU Administrar Roles

1.1.2.2.3.9.2.1.11. Diagrama de Secuencia: Caso de Uso Adicionar Roles

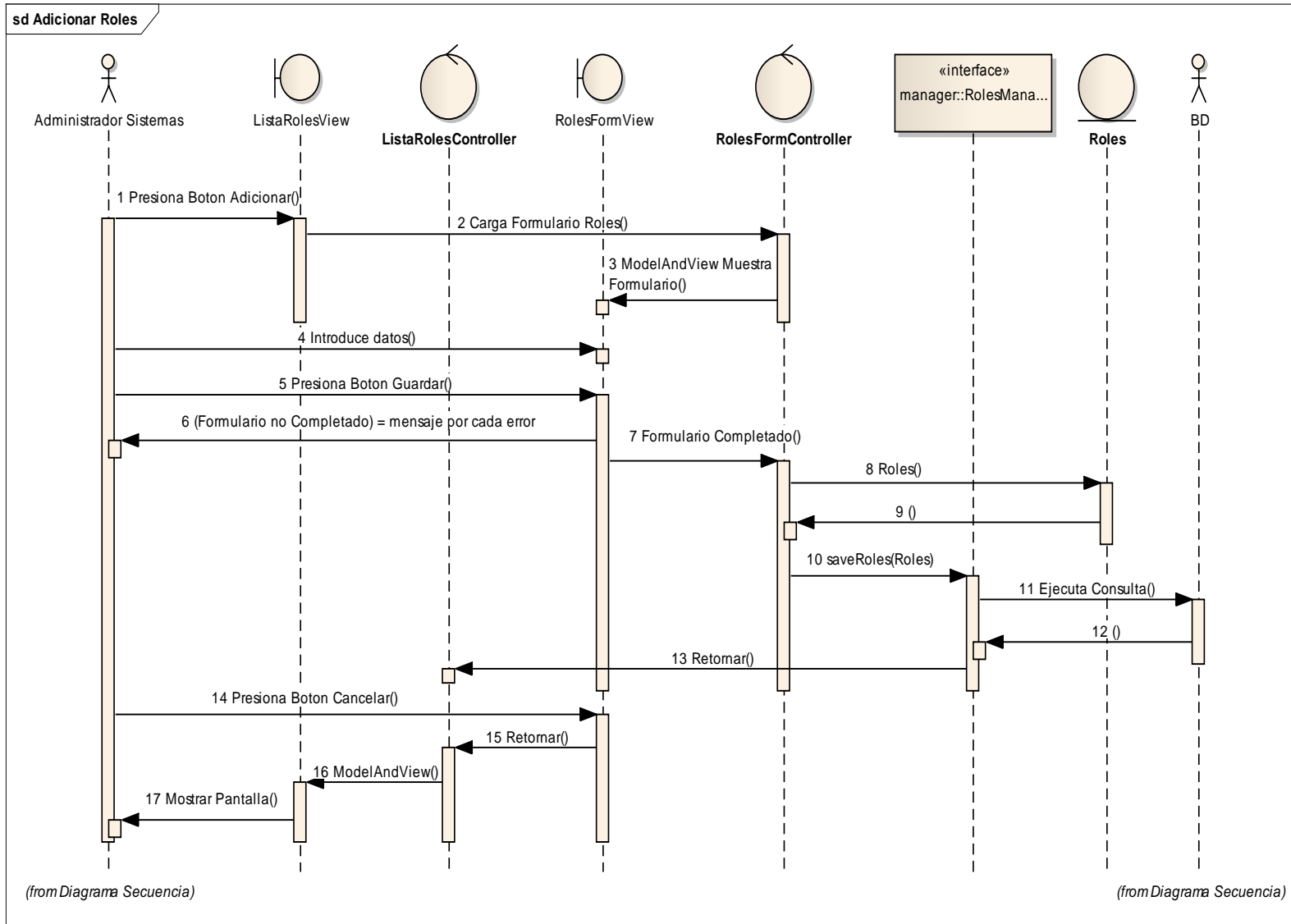


Figura 126. Diagrama de Secuencia: CU Adicionar Roles

1.1.2.2.3.9.2.1.12. Diagrama de Secuencia: Caso de Uso Modificar Roles

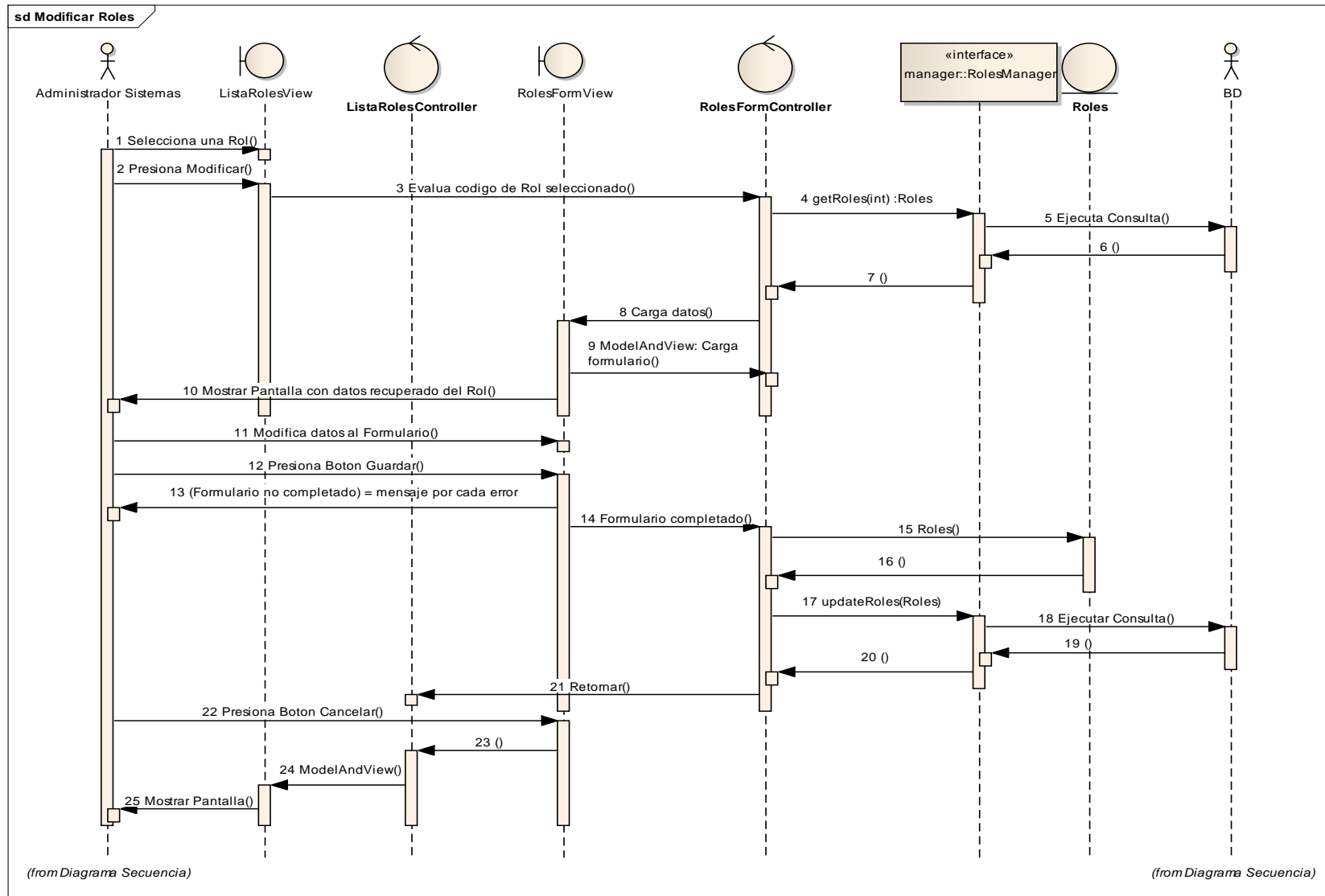


Figura 127. Diagrama de Secuencia: CU Modificar Roles

1.1.2.2.3.9.2.1.13. Diagrama de Secuencia: Caso de Uso Eliminar Roles

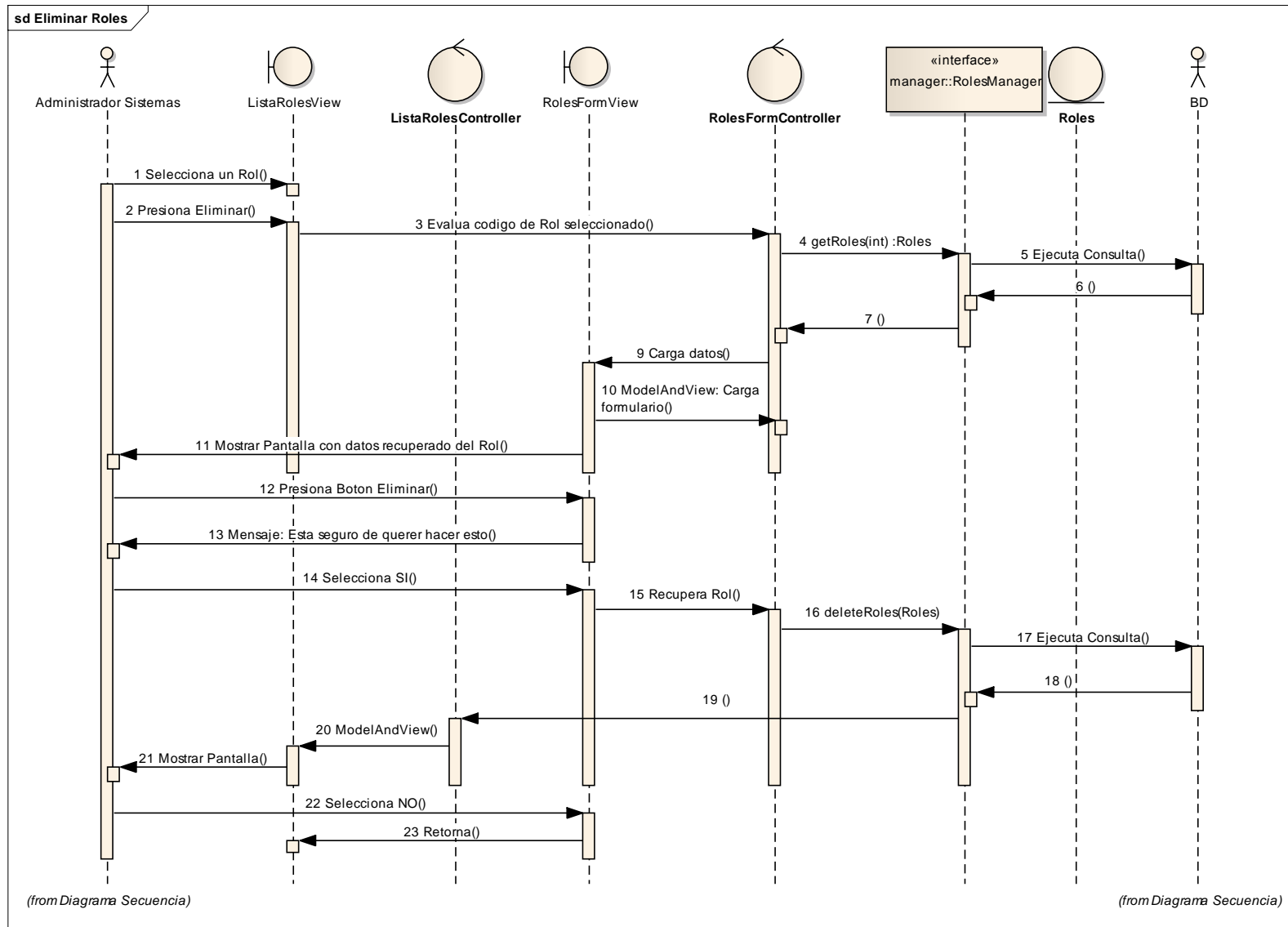


Figura 128. Diagrama de Secuencia: CU Eliminar Roles

1.1.2.2.3.9.2.1.14. Diagrama de Secuencia: Caso de Uso Administrar Asignación y Reasignar de Cargo a Personal

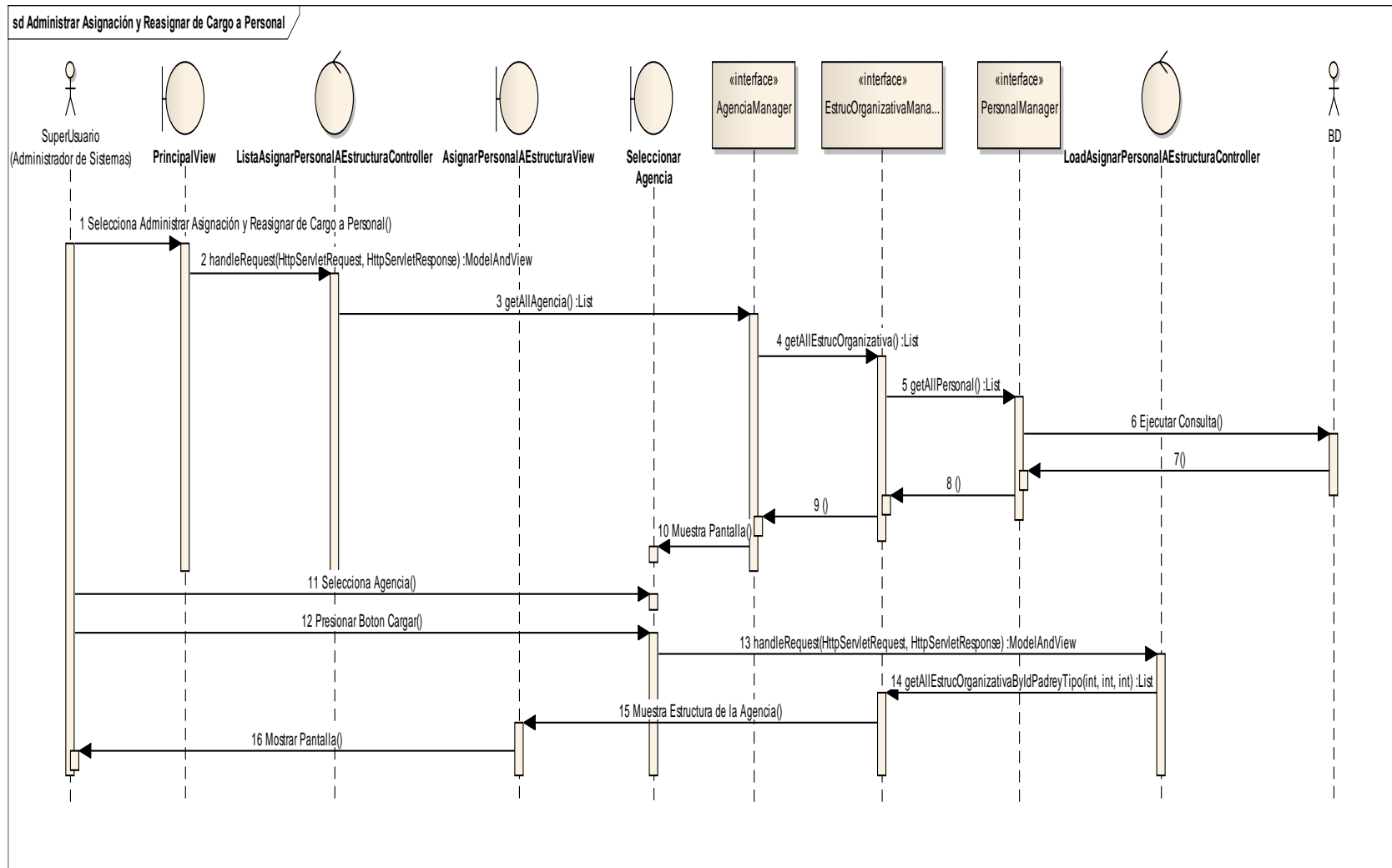


Figura 129. Diagrama de Secuencia: CU Administrar Asignación y Reasignar de Cargo a Personal

1.1.2.2.3.9.2.1.15. Diagrama de Secuencia: Caso de Uso Asignar Cargo

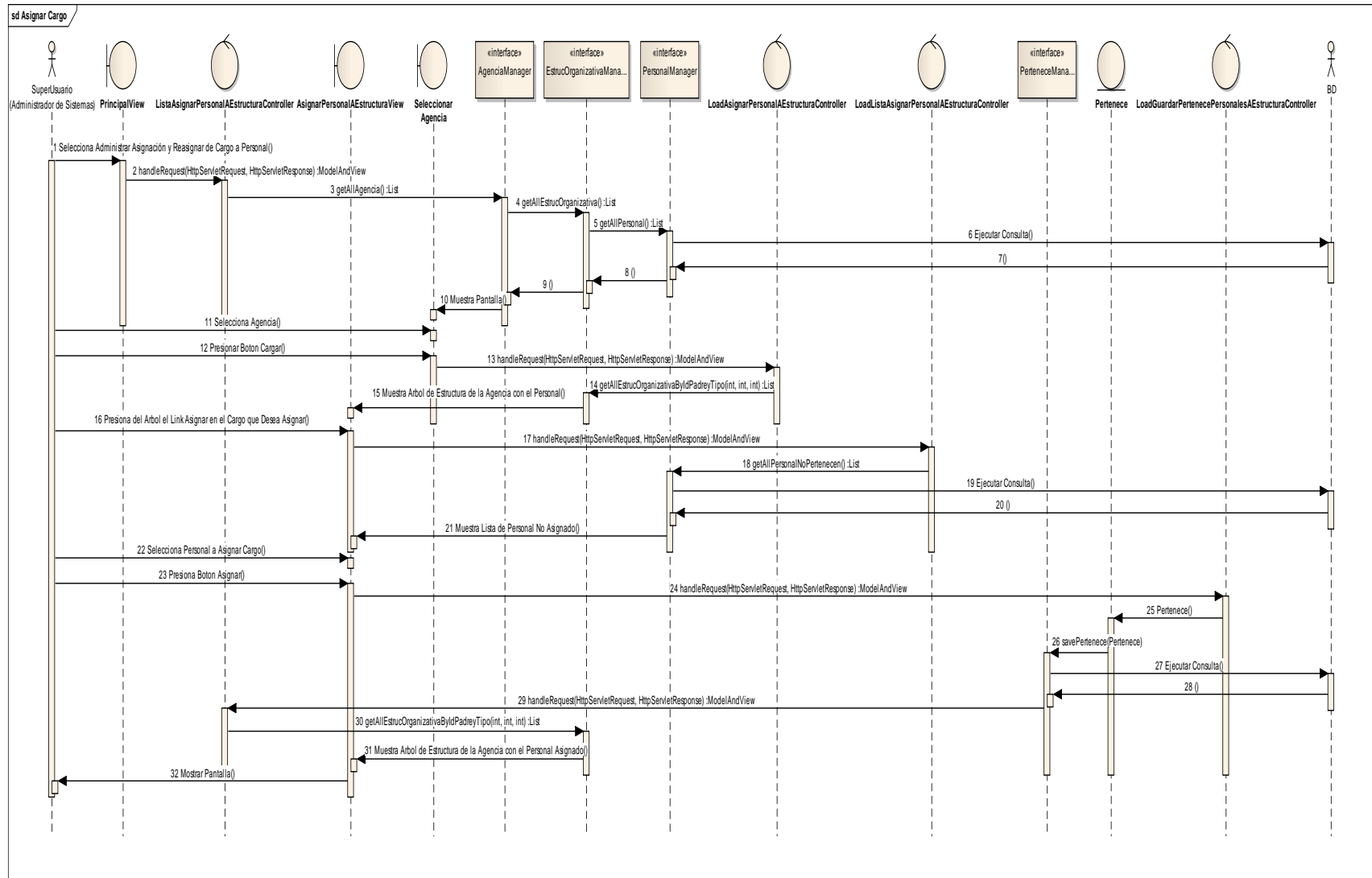


Figura 130. Diagrama de Secuencia: CU Asignar Cargo

1.1.2.2.3.9.2.1.16. Diagrama de Secuencia: Caso de Uso Reasignar Cargo

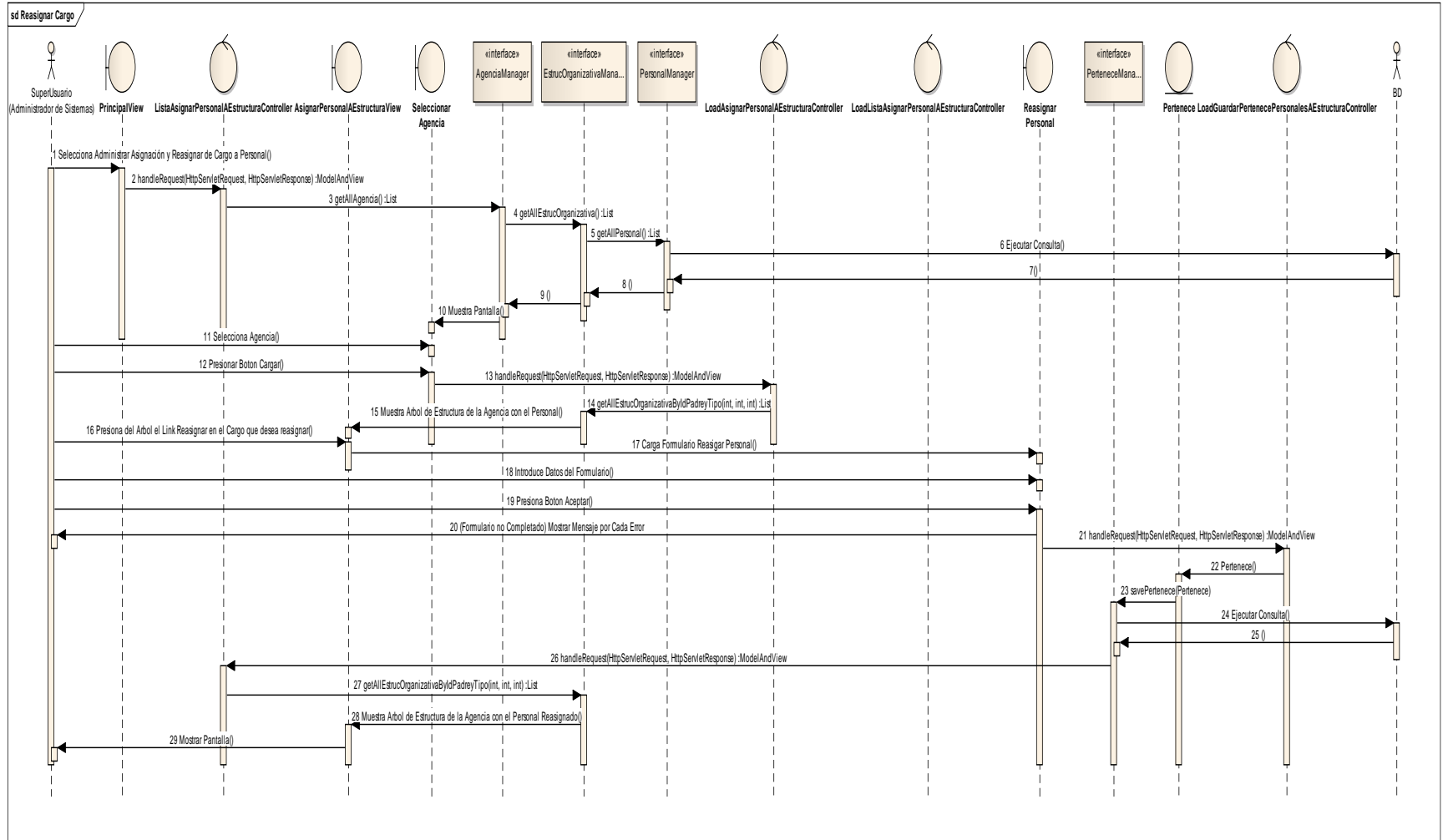


Figura 131. Diagrama de Secuencia: CU Reasignar Cargo

1.1.2.2.3.9.2.1.17. Diagrama de Secuencia: Caso de Uso Administrar Agencia

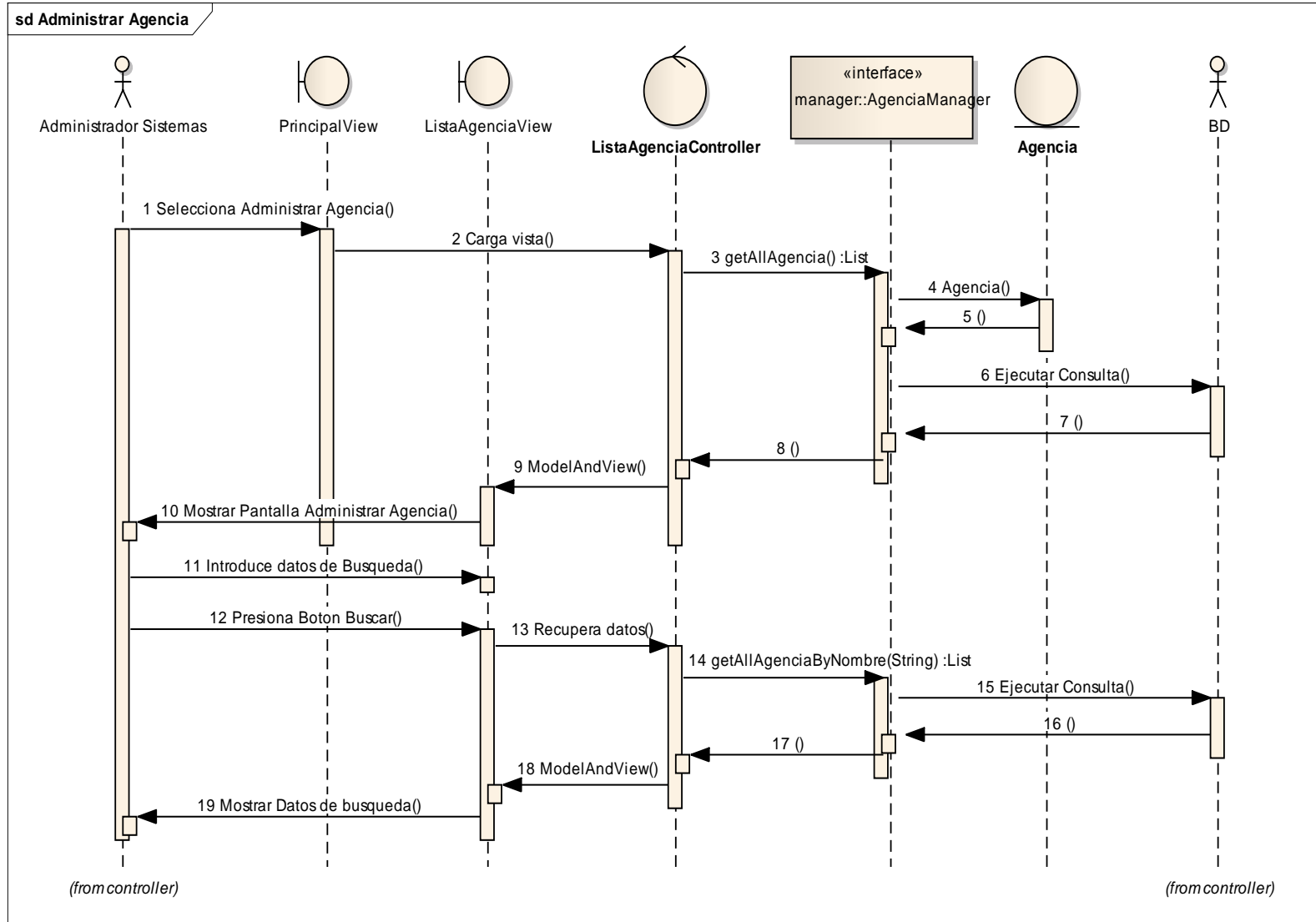


Figura 132. Diagrama de Secuencia: CU Administrar Agencia

1.1.2.2.3.9.2.1.18. Diagrama de Secuencia: Caso de Uso Adicionar Agencia

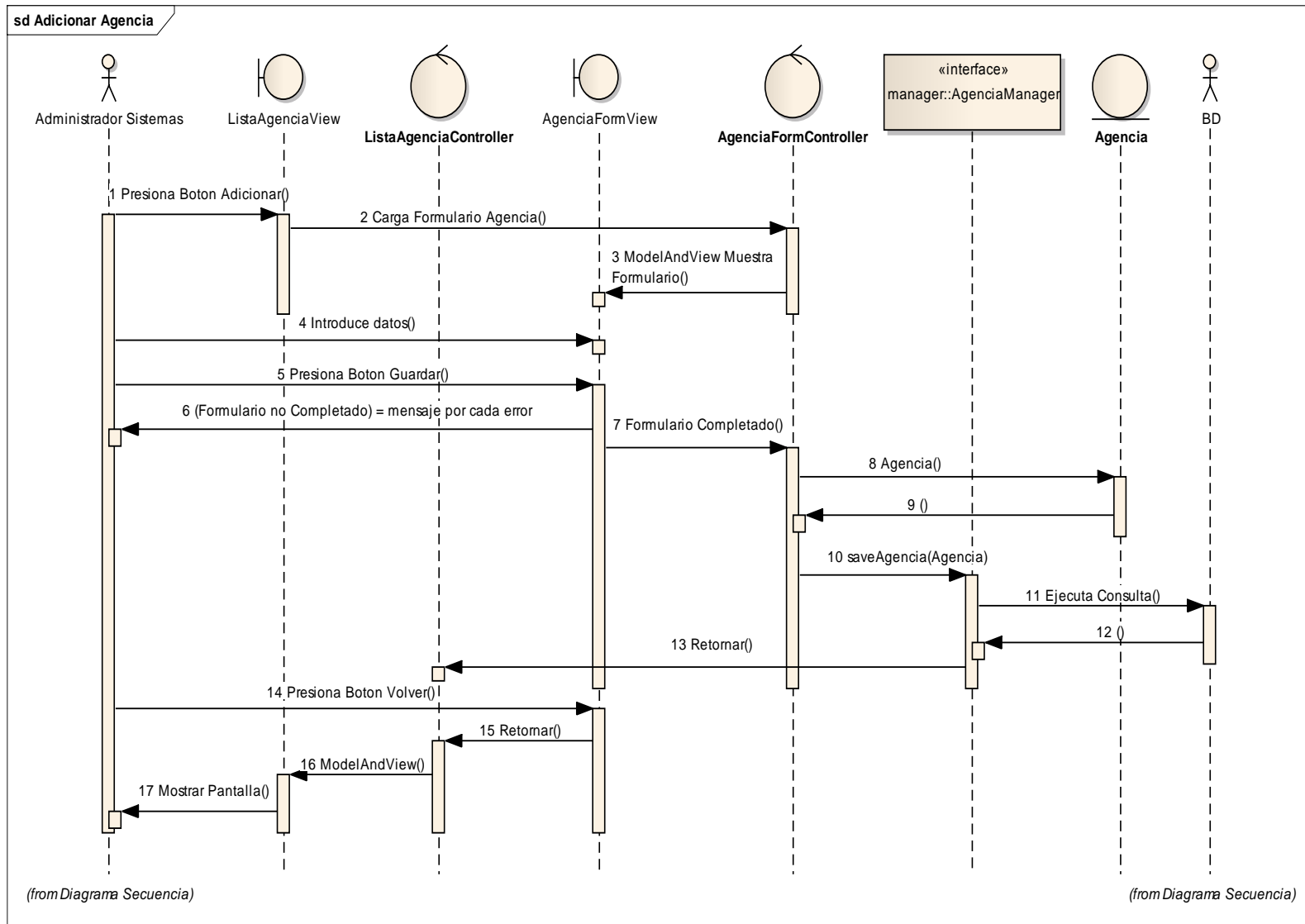


Figura 133. Diagrama de Secuencia: CU Adicionar Agencia

1.1.2.2.3.9.2.1.19. Diagrama de Secuencia: Caso de Uso Modificar Agencia

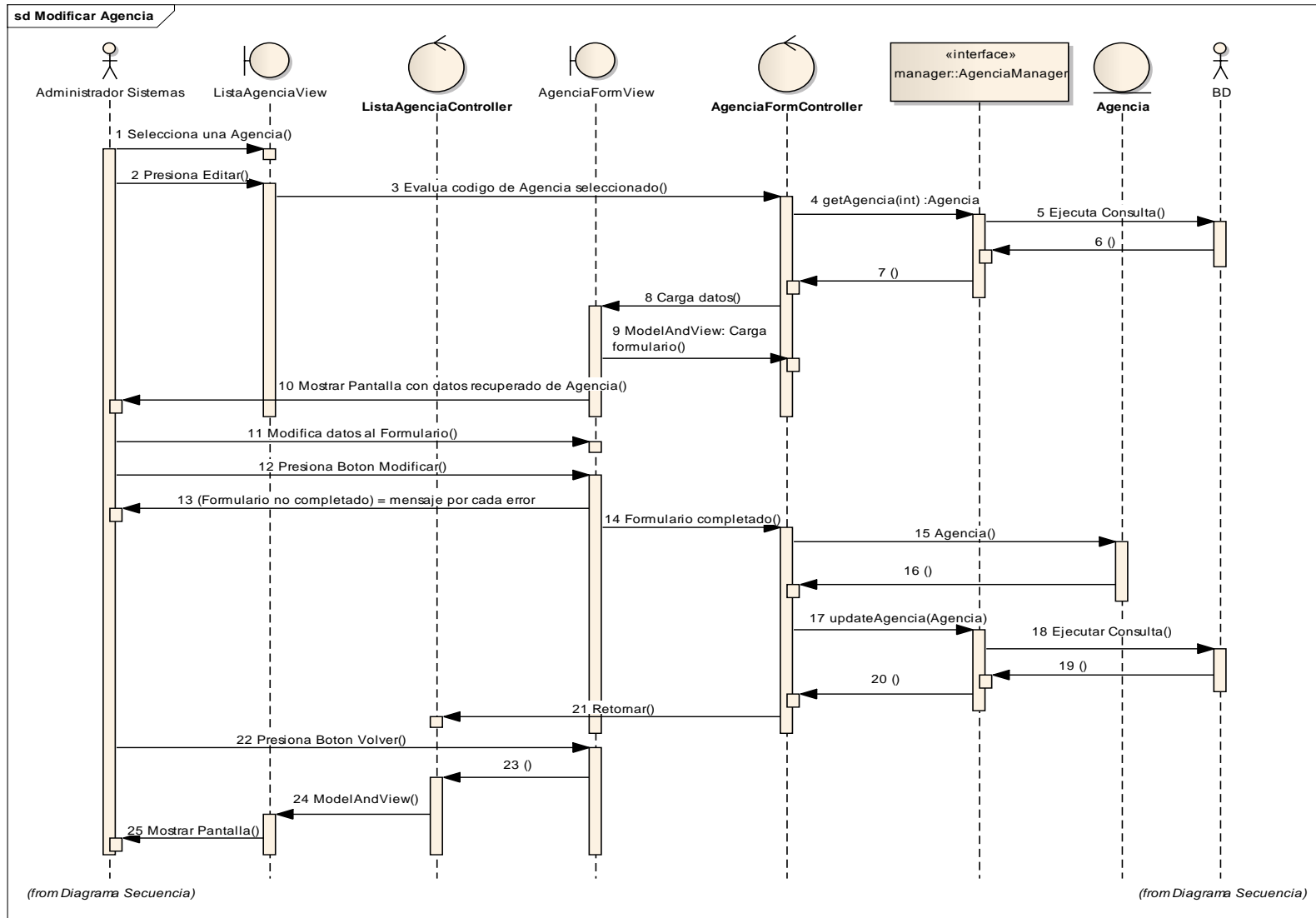


Figura 134. Diagrama de Secuencia: CU Modificar Agencia

1.1.2.2.3.9.2.1.20. Diagrama de Secuencia: Caso de Uso Eliminar Agencia

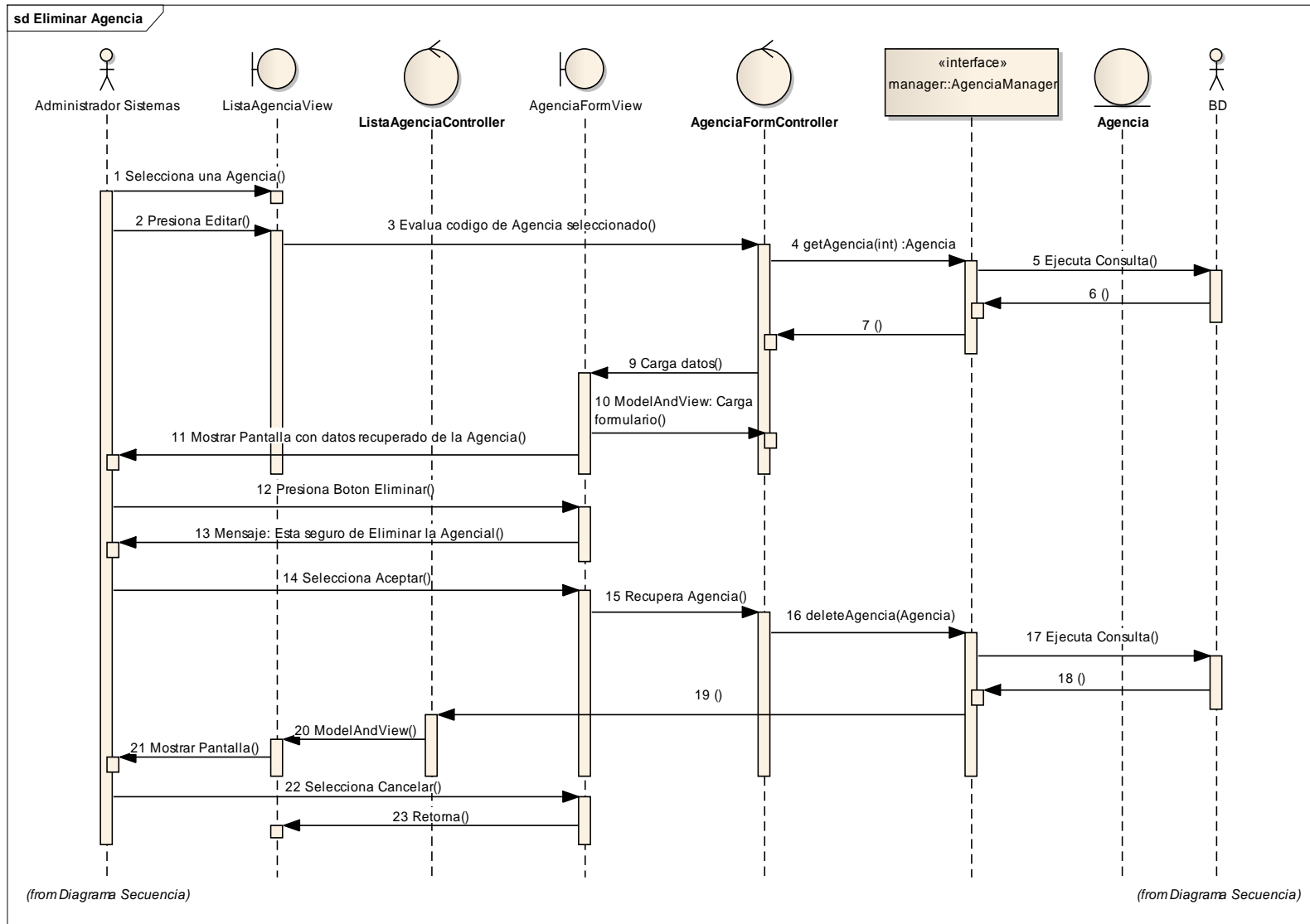


Figura 135. Diagrama de Secuencia: CU Eliminar Agencia

1.1.2.2.3.9.2.1.21. Diagrama de Secuencia: Caso de Uso Administrar Estructura Organizativa

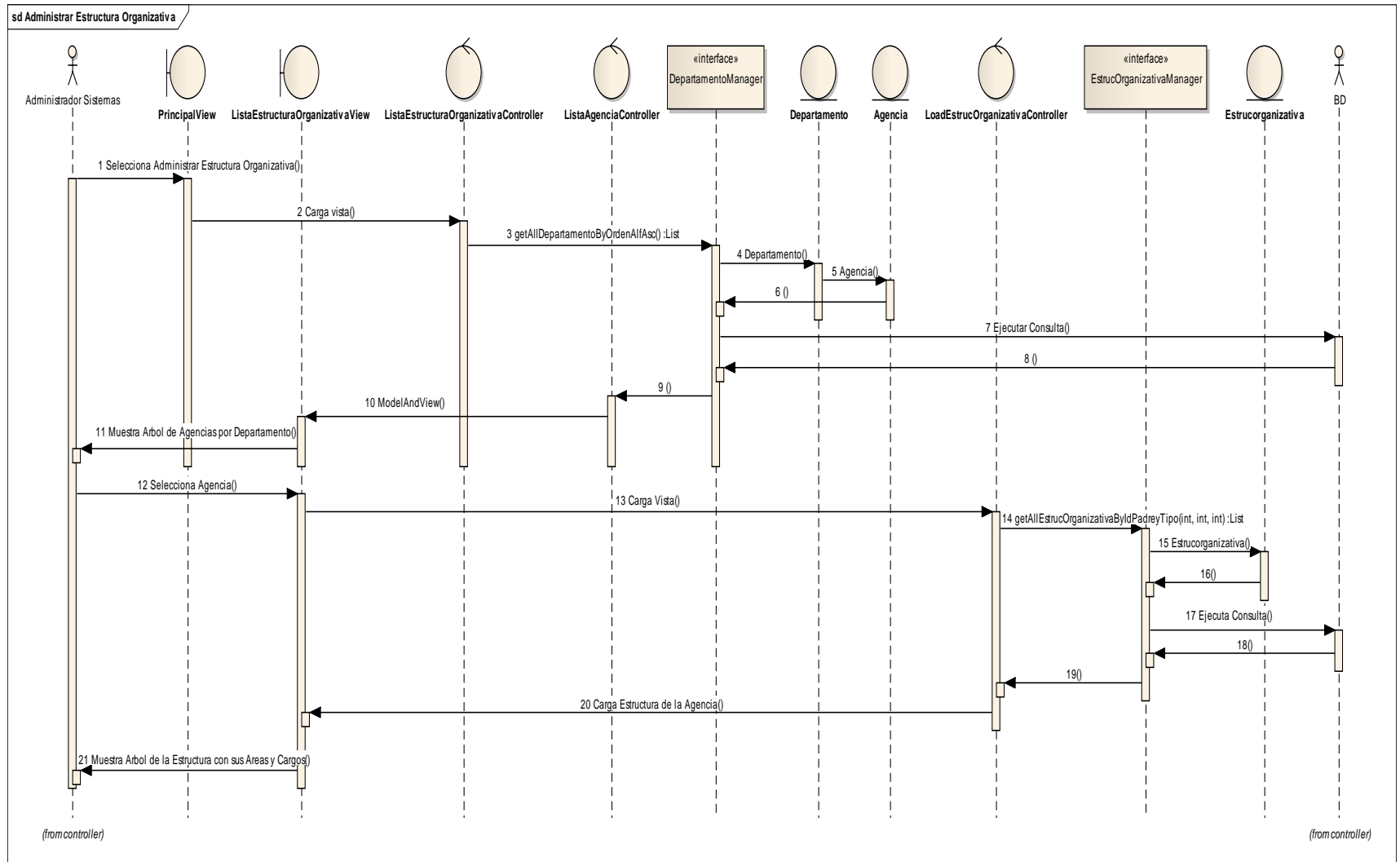


Figura 136. Diagrama de Secuencia: CU Estructura Organizativa

1.1.2.2.3.9.2.1.22. Diagrama de Secuencia: Caso de Uso Adicionar Área

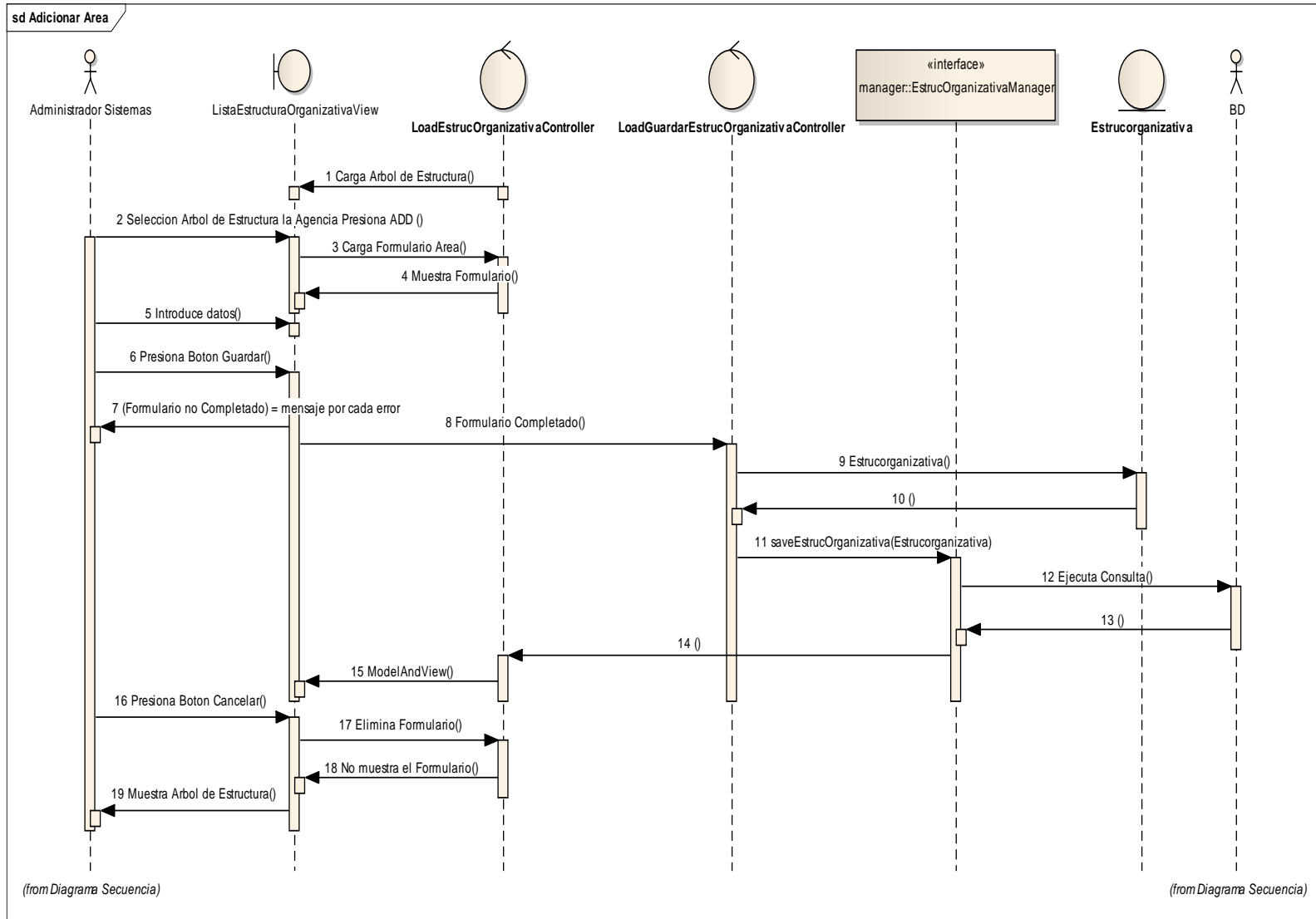


Figura 137. Diagrama de Secuencia: CU Adicionar Área

1.1.2.2.3.9.2.1.23. Diagrama de Secuencia: Caso de Uso Modificar Área

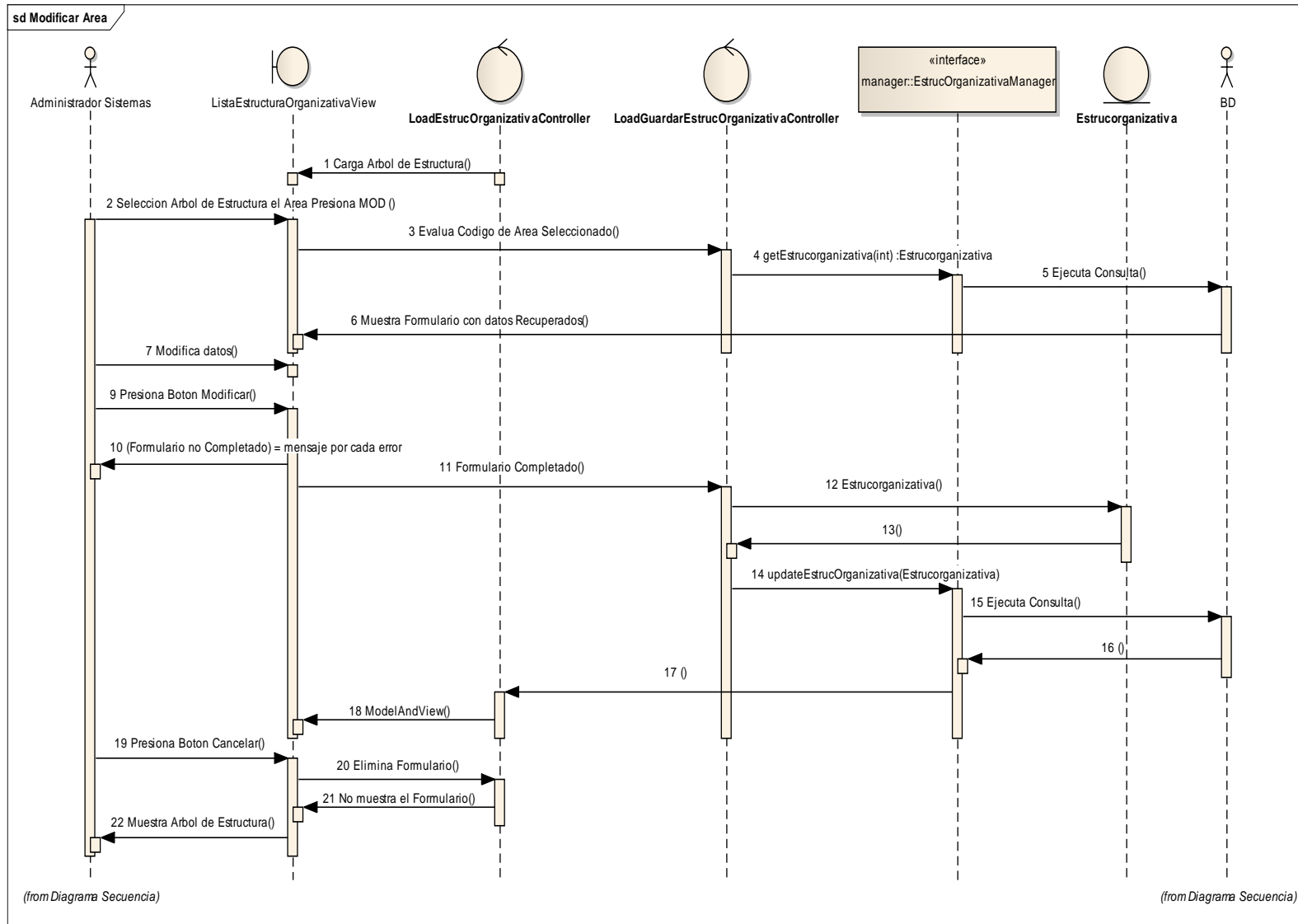


Figura 138. Diagrama de Secuencia: CU Modificar Área

1.1.2.2.3.9.2.1.24. Diagrama de Secuencia: Caso de Uso Eliminar Área

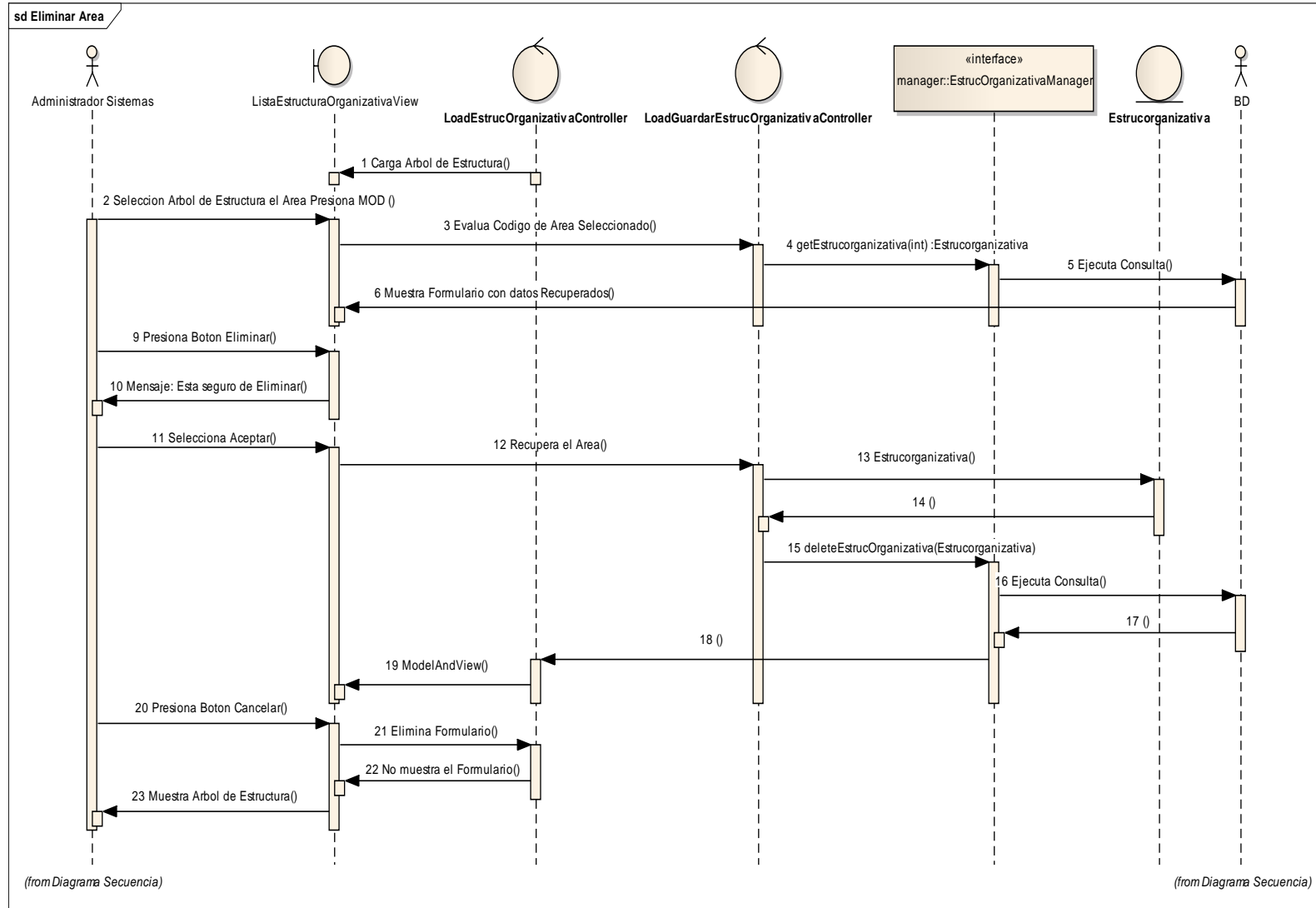


Figura 139. Diagrama de Secuencia: CU Eliminar Área

1.1.2.2.3.9.2.1.25. Diagrama de Secuencia: Caso de Uso Adicionar Cargo

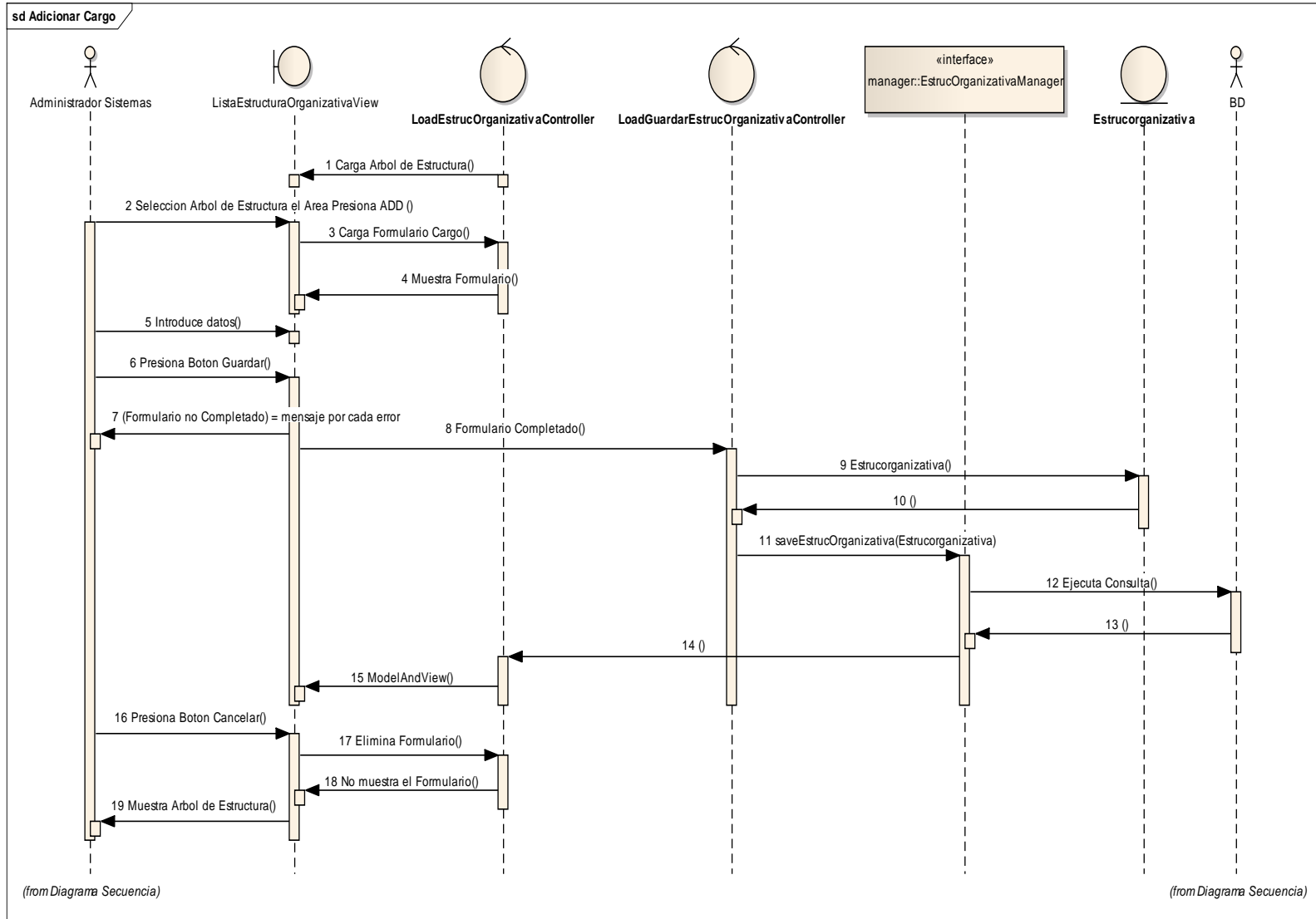


Figura 140. Diagrama de Secuencia: CU Adicionar Cargo

1.1.2.2.3.9.2.1.26. Diagrama de Secuencia: Caso de Uso Modificar Cargo

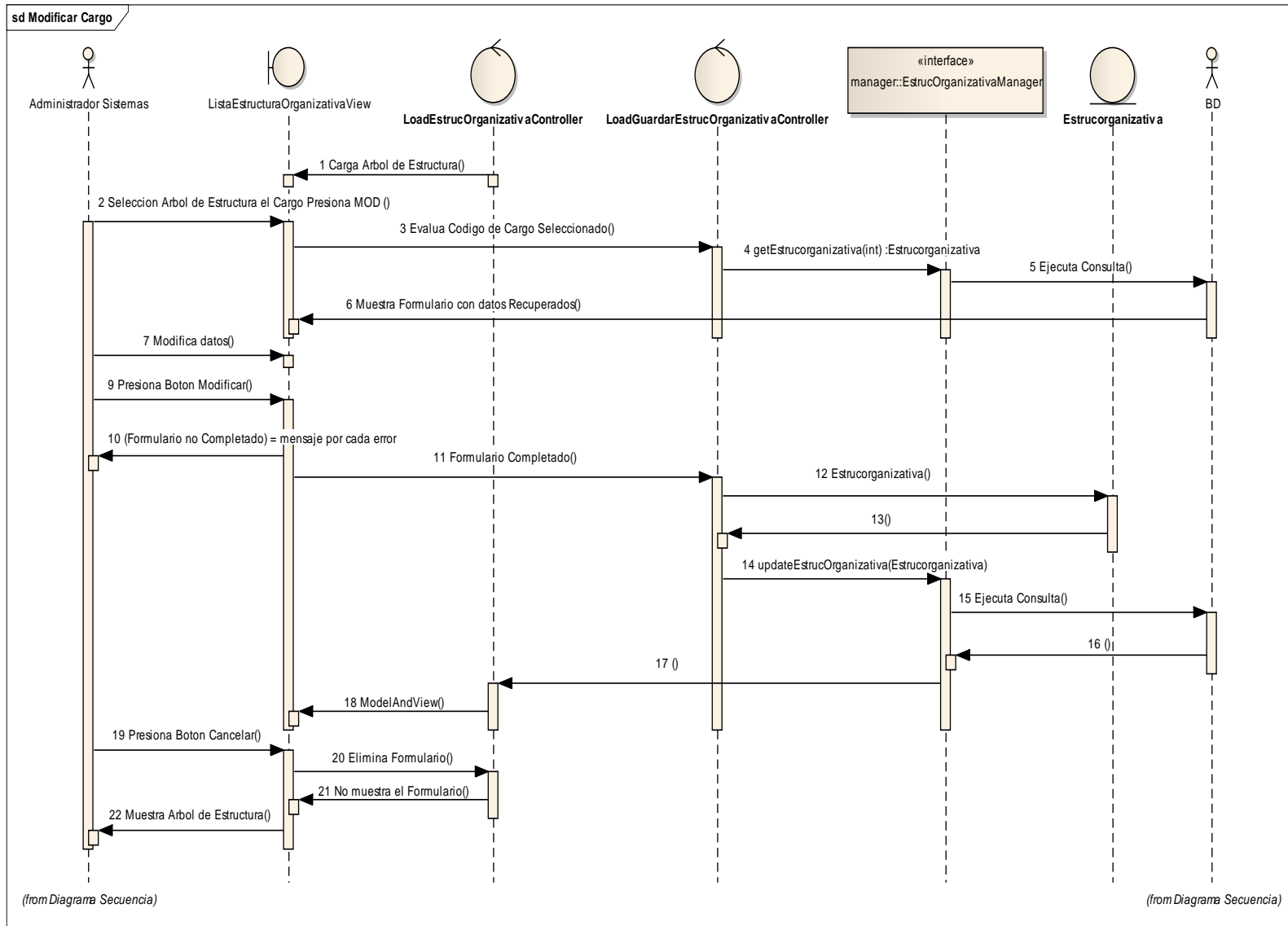


Figura 141. Diagrama de Secuencia: CU Modificar Cargo

1.1.2.2.3.9.2.1.27. Diagrama de Secuencia: Caso de Uso Eliminar Cargo

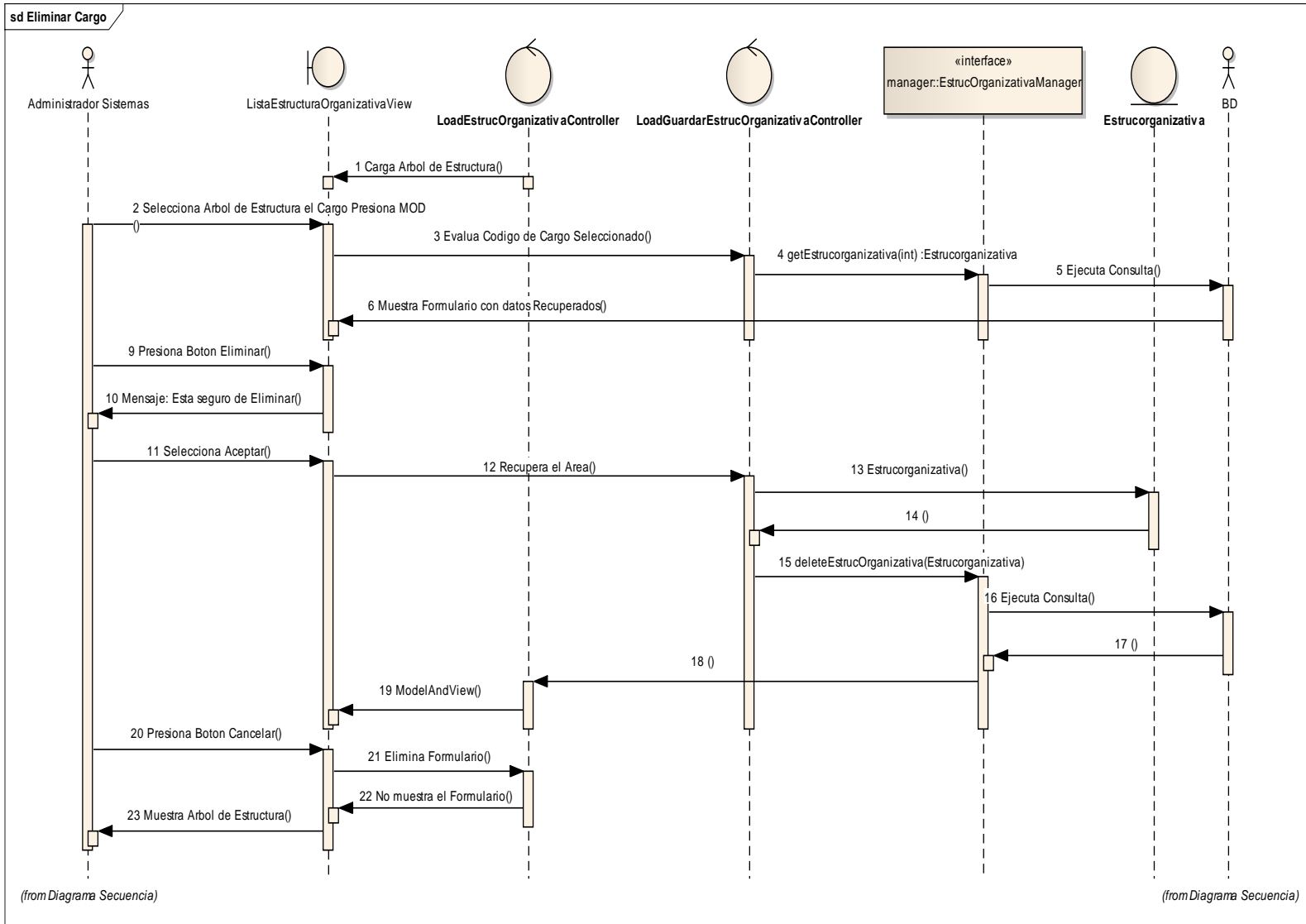


Figura 142. Diagrama de Secuencia: CU Eliminar Cargo

1.1.2.2.3.9.2.1.28. Diagrama de Secuencia: Caso de Uso Administrar Proveedor

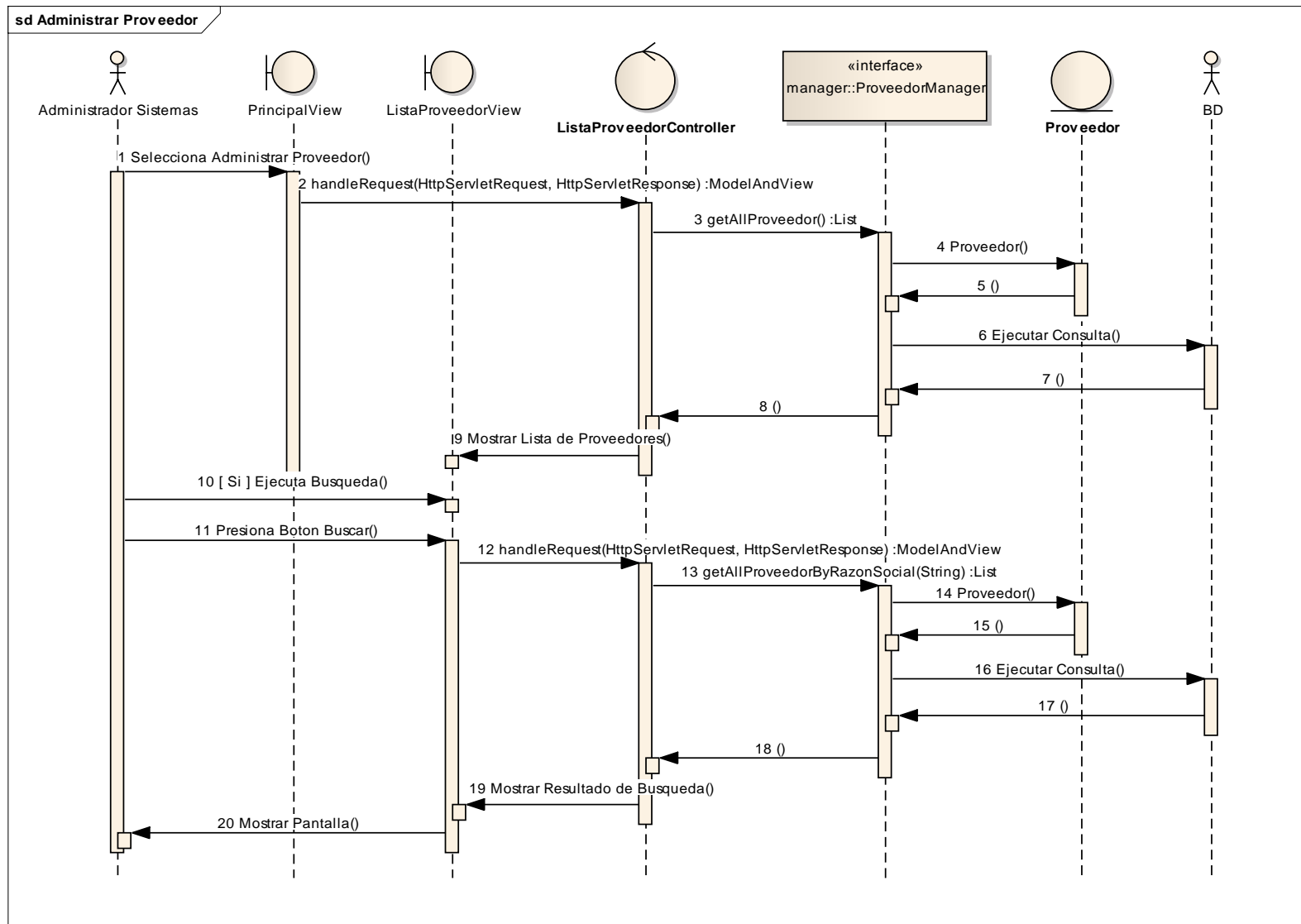


Figura 143. Diagrama de Secuencia: CU Administrar Proveedor

1.1.2.2.3.9.2.1.29. Diagrama de Secuencia: Caso de Uso Adicionar Proveedor

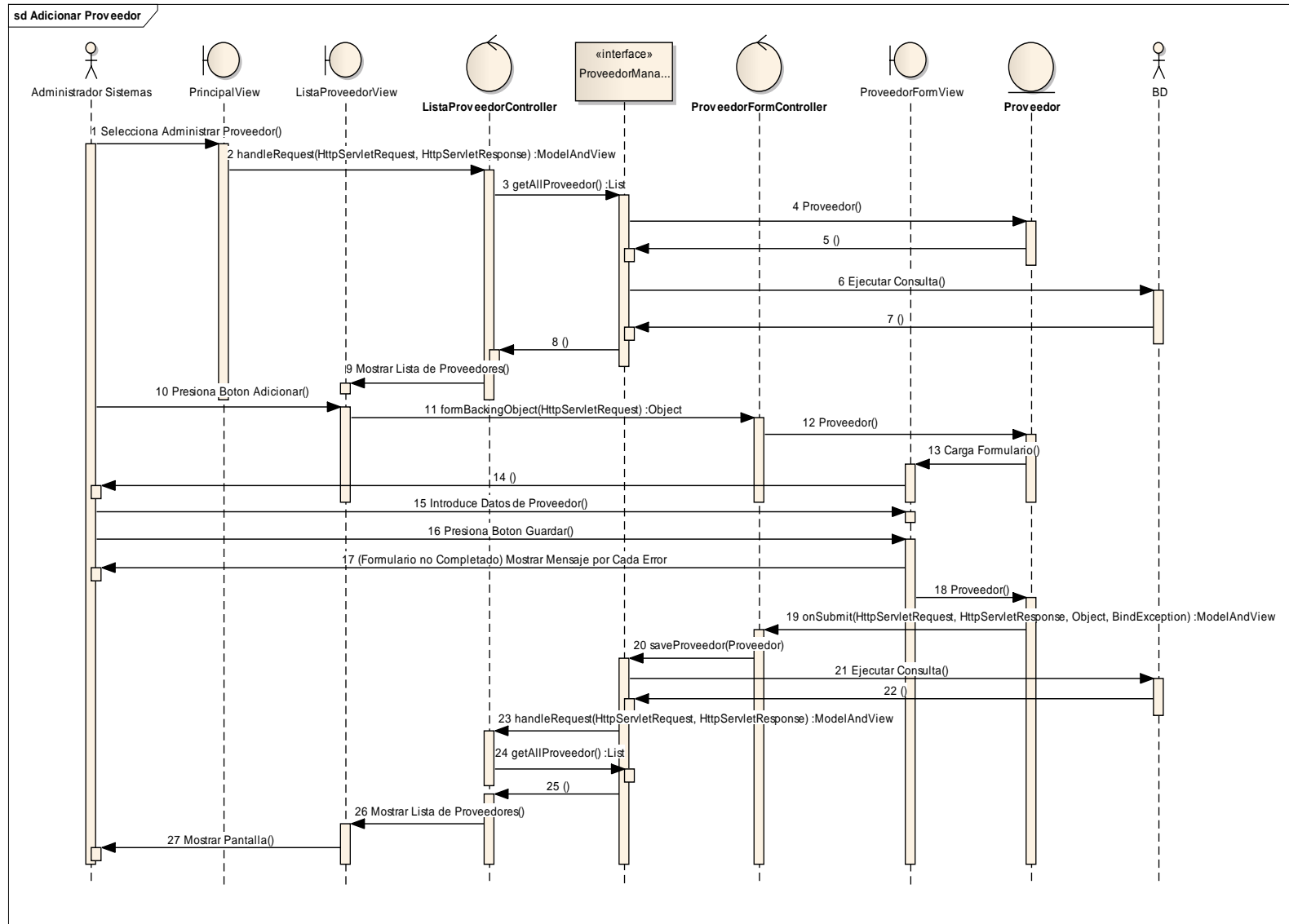


Figura 144. Diagrama de Secuencia: CU Adicionar Proveedor

1.1.2.2.3.9.2.1.30. Diagrama de Secuencia: Caso de Uso Modificar Proveedor

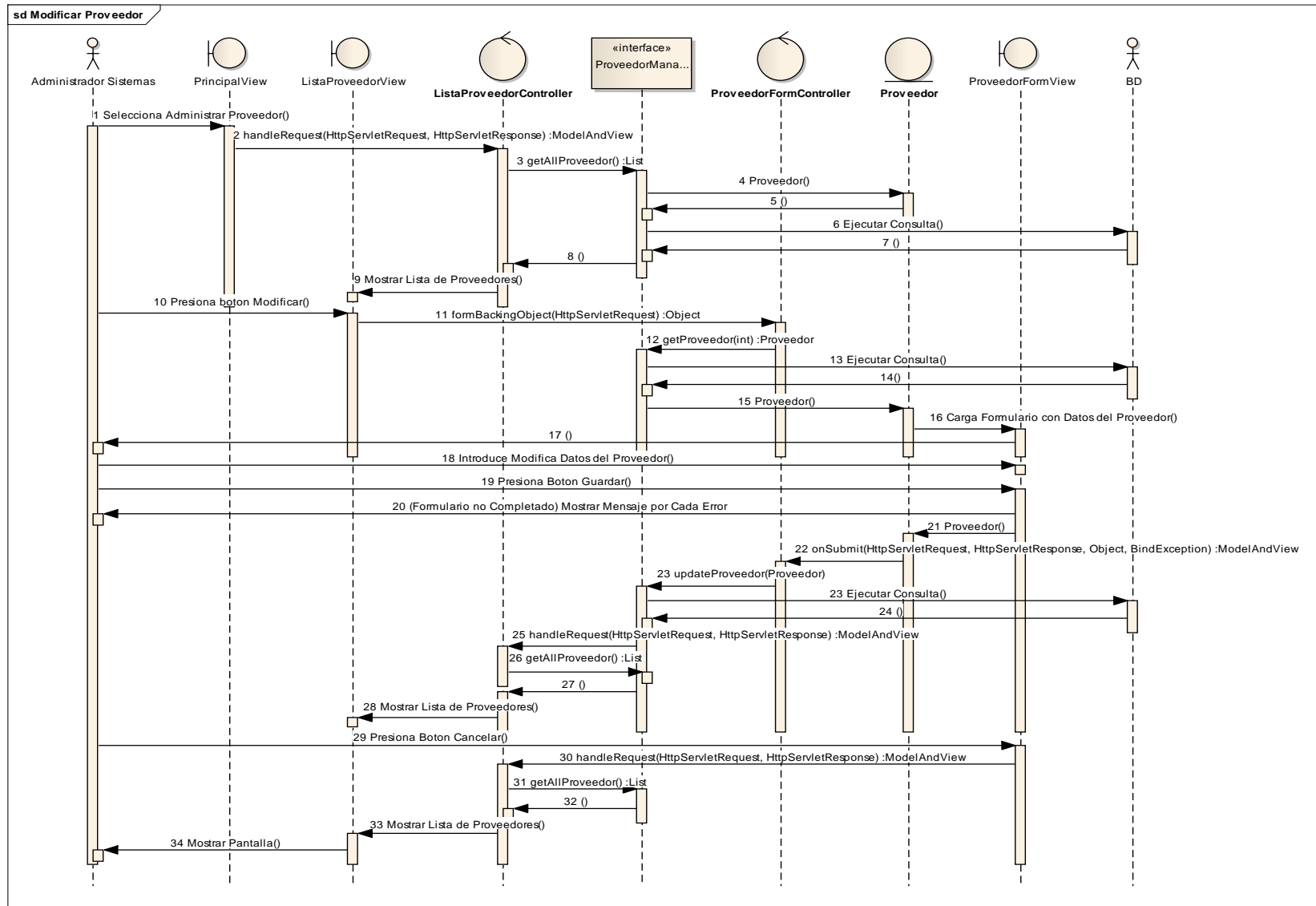


Figura 145. Diagrama de Secuencia: CU Modificar Proveedor

1.1.2.2.3.9.2.1.31. Diagrama de Secuencia: Caso de Uso Eliminar Proveedor

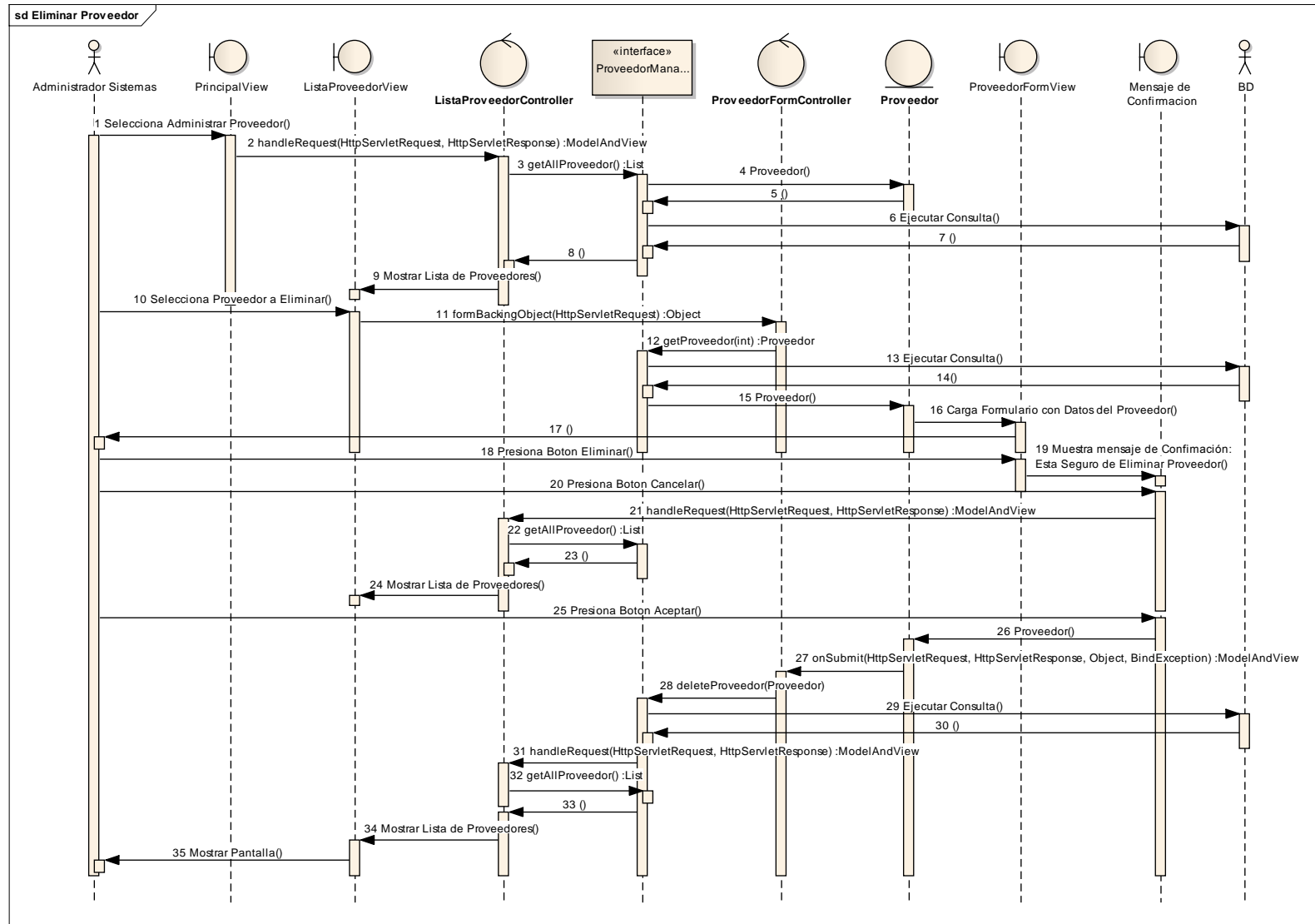


Figura 146. Diagrama de Secuencia: CU Eliminar Proveedor

1.1.2.2.3.9.2.1.32. Diagrama de Secuencia: Caso de Uso Administrar Producto

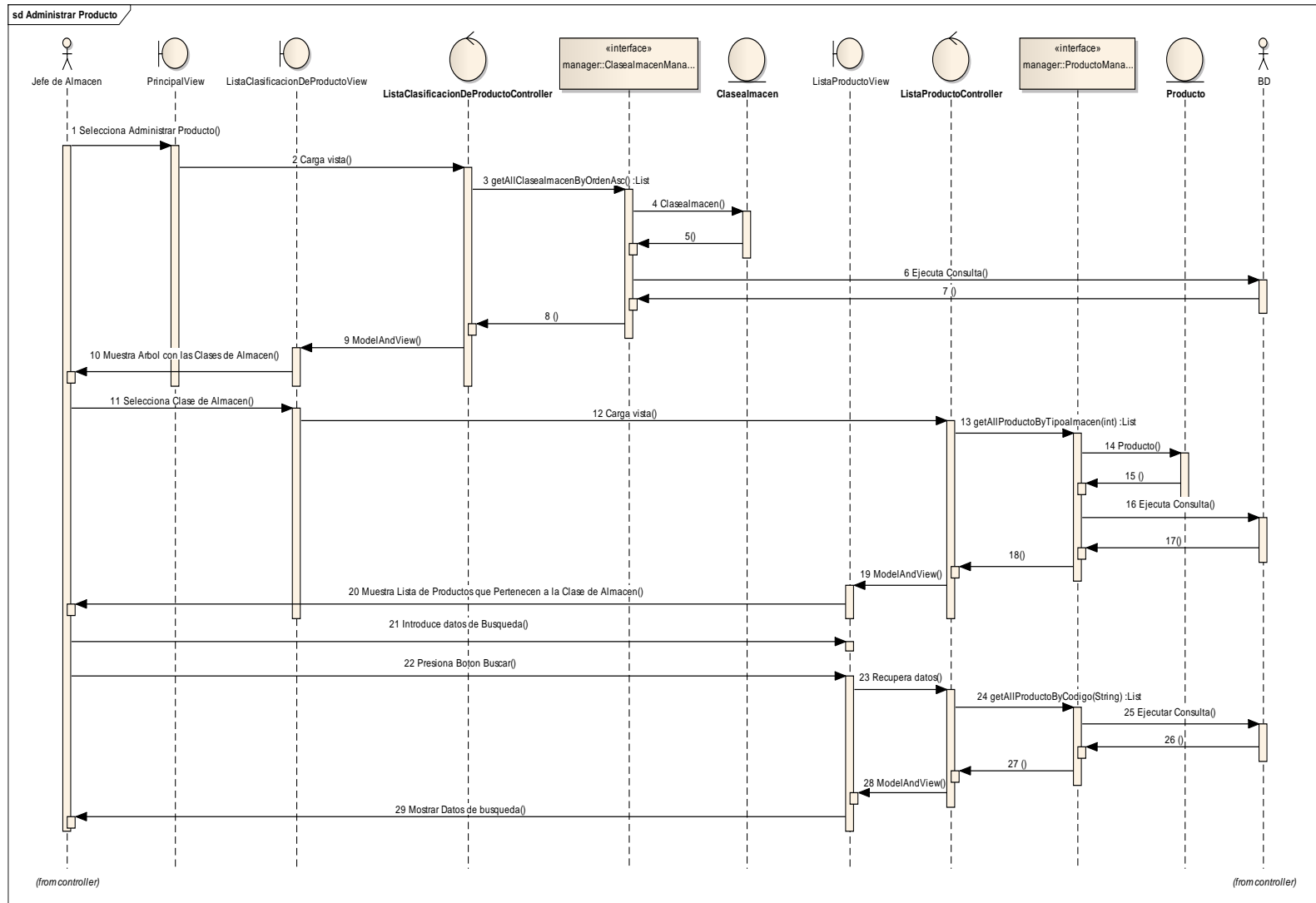


Figura 147. Diagrama de Secuencia: Caso de Uso Administrar Producto

1.1.2.2.3.9.2.1.33. Diagrama de Secuencia: Caso de Uso Adicionar Producto

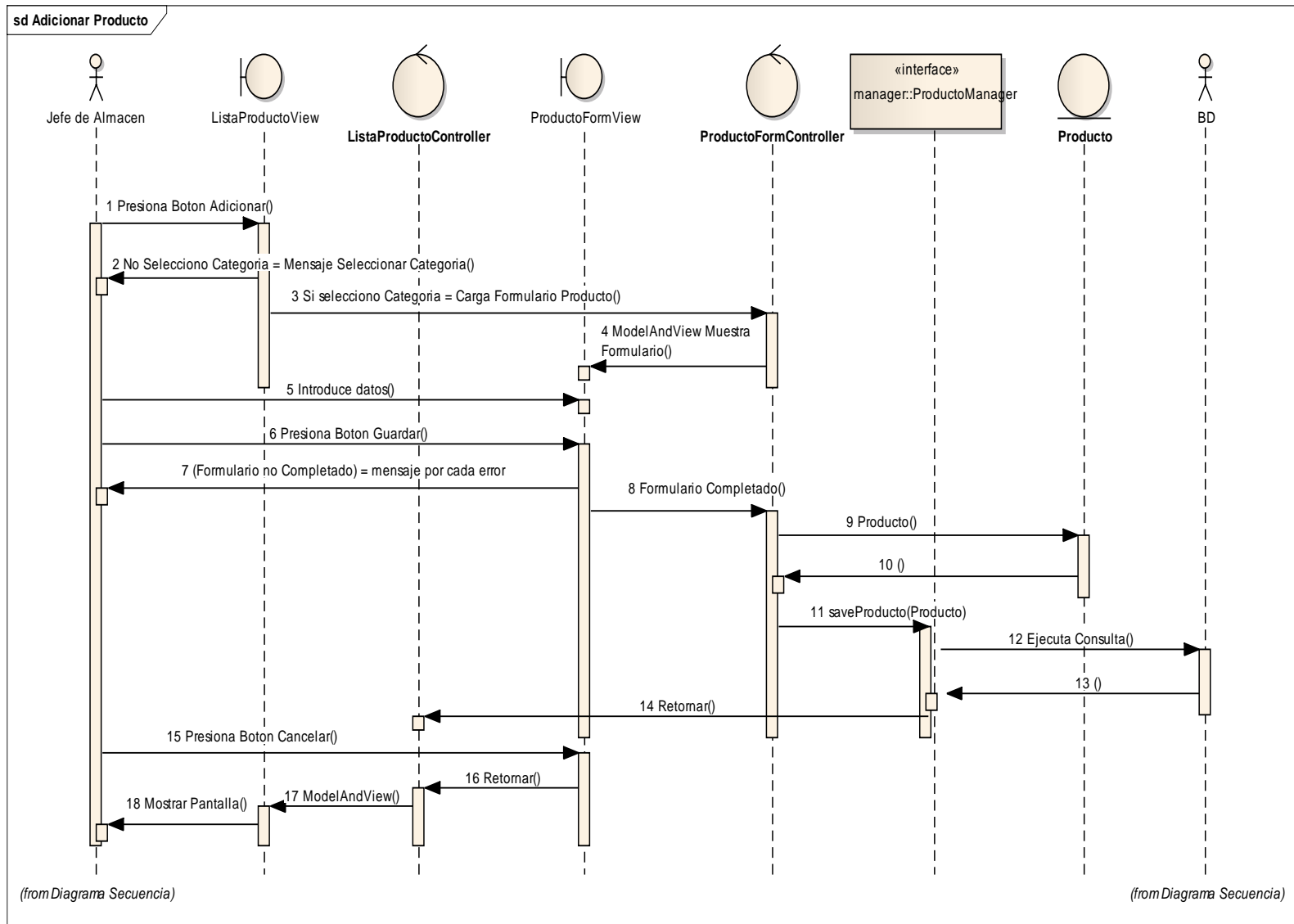


Figura 148. Diagrama de Secuencia: Caso de Uso Adicionar Producto

1.1.2.2.3.9.2.1.34. Diagrama de Secuencia: Caso de Uso Modificar Producto

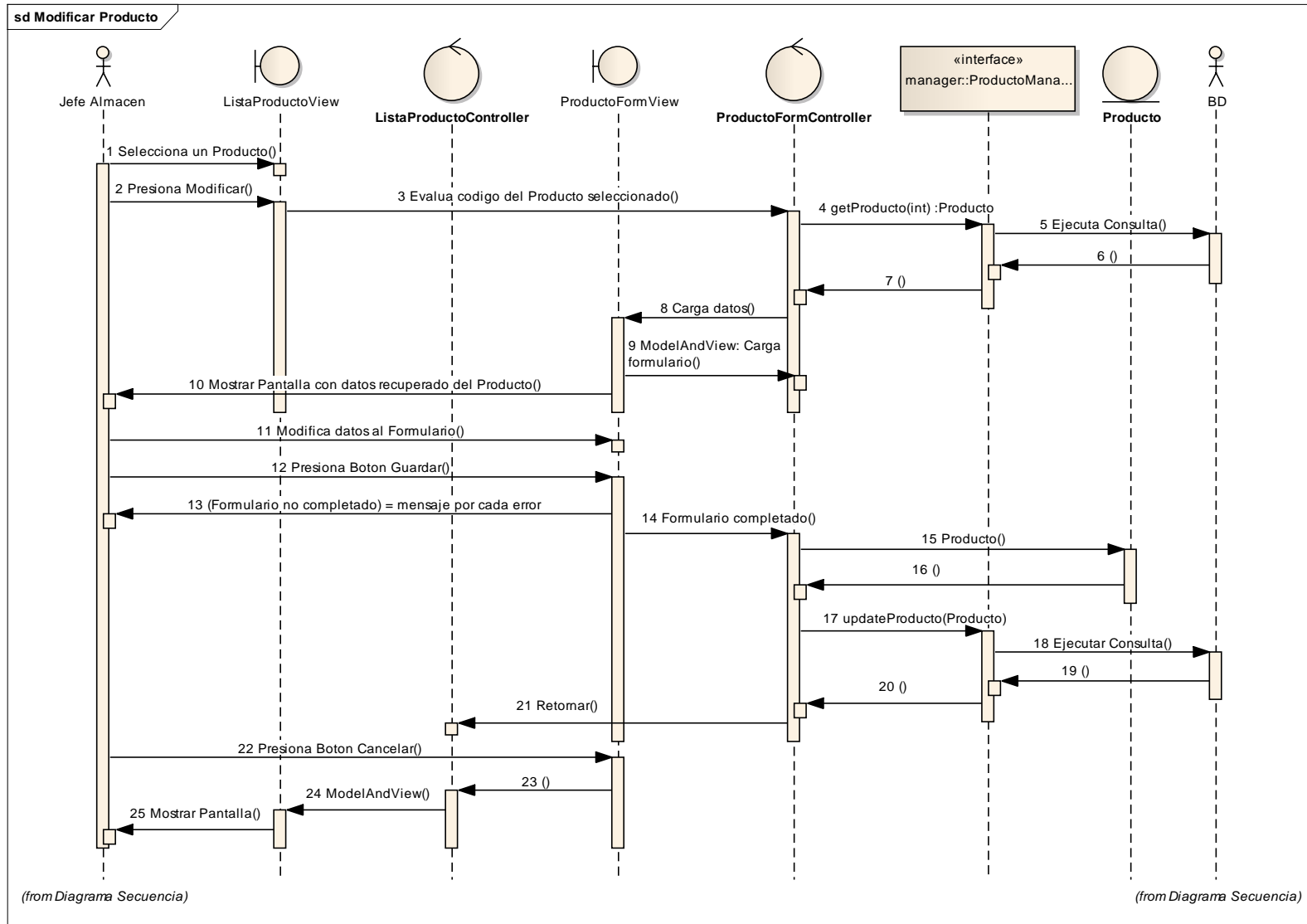


Figura 149. Diagrama de Secuencia: Caso de Uso Modificar Producto

1.1.2.2.3.9.2.1.35. Diagrama de Secuencia: Caso de Uso Eliminar Producto

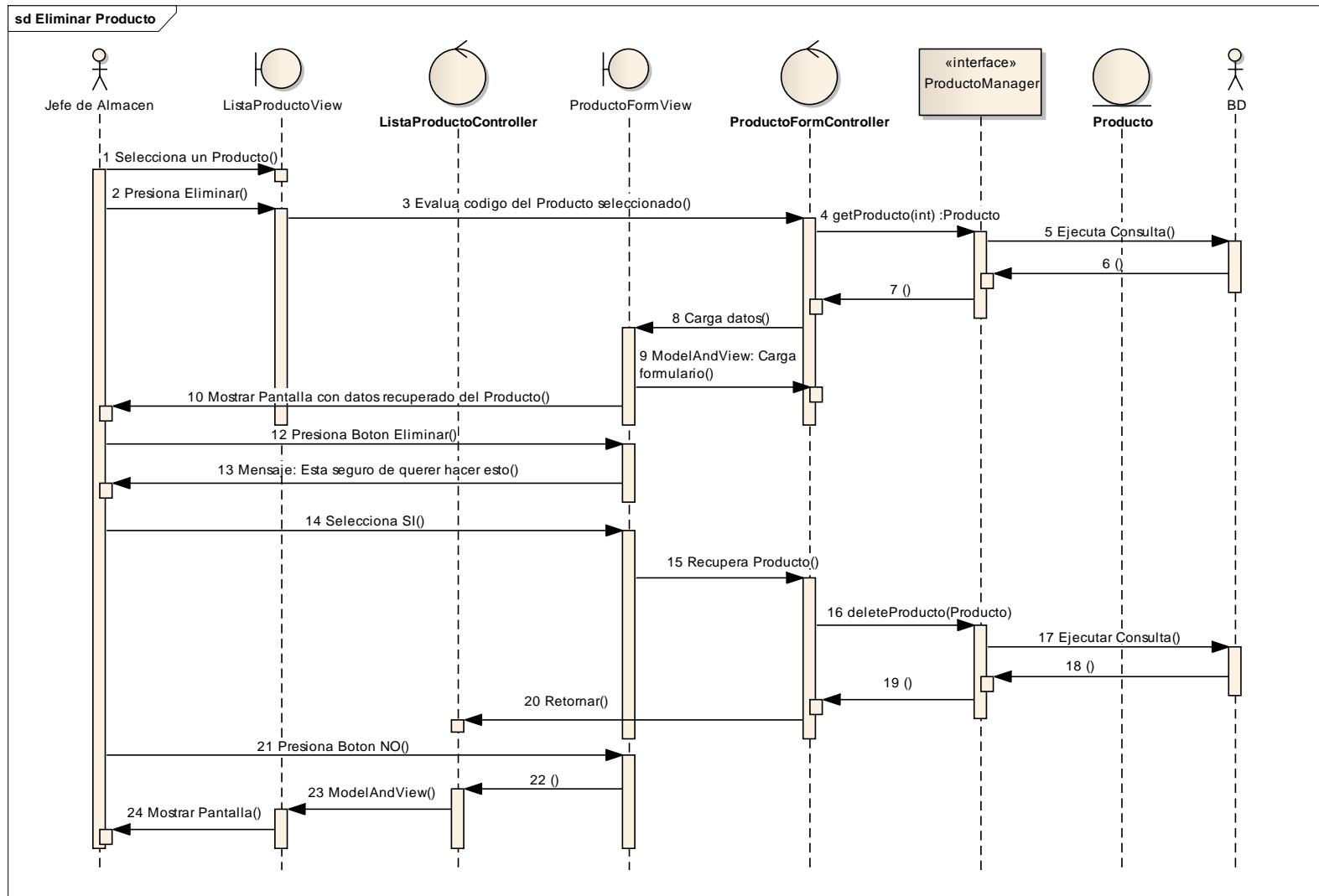


Figura 150. Diagrama de Secuencia: Caso de Uso Eliminar Producto

1.1.2.2.3.9.2.1.36. Diagrama de Secuencia: Caso de Uso Administrar Tipo y Categoría de Almacén

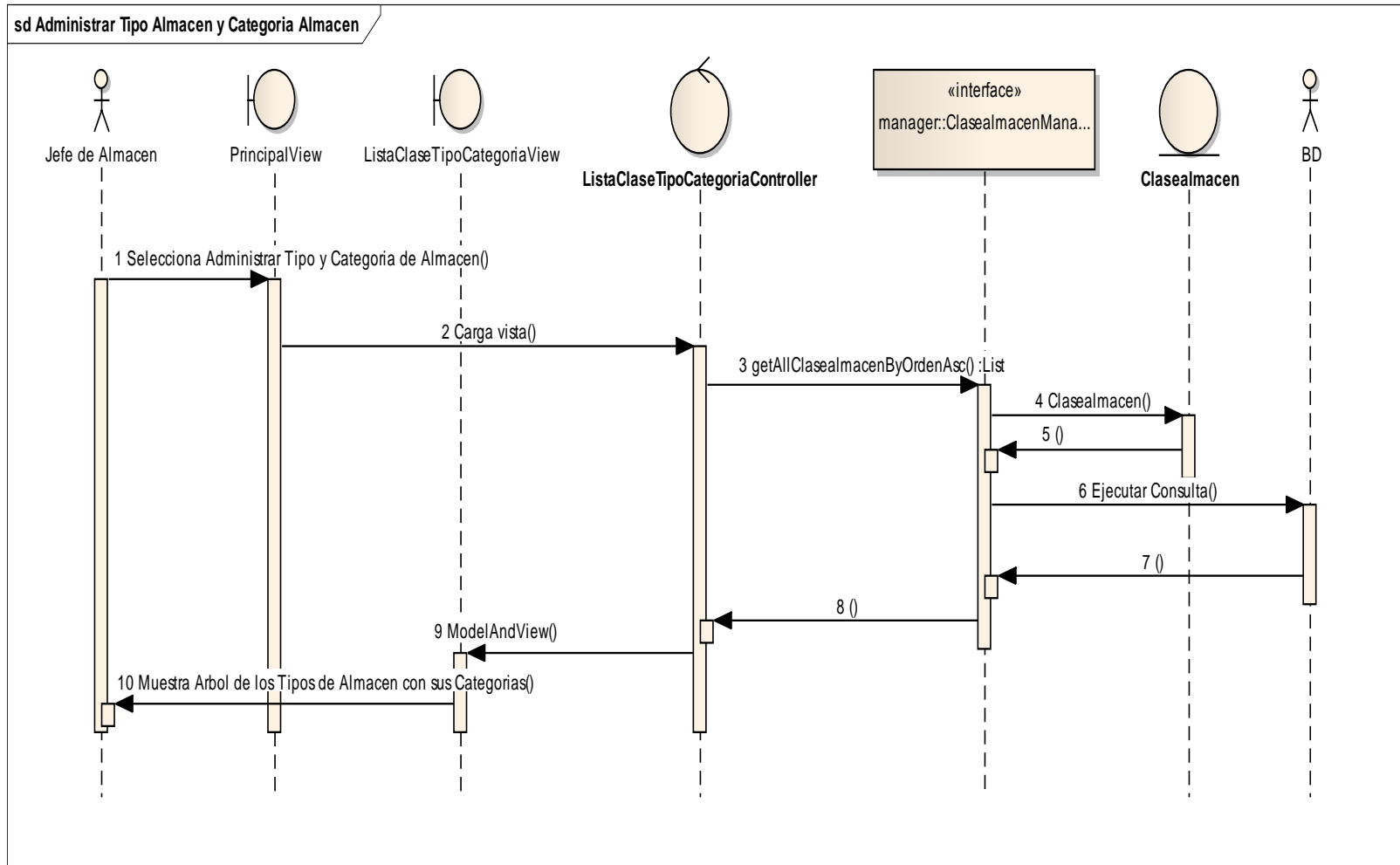


Figura 151. Diagrama de Secuencia: CU Administrar Tipo y Categoría de Almacén

1.1.2.2.3.9.2.1.37. Diagrama de Secuencia: Caso de Uso Adicionar Tipo Almacén

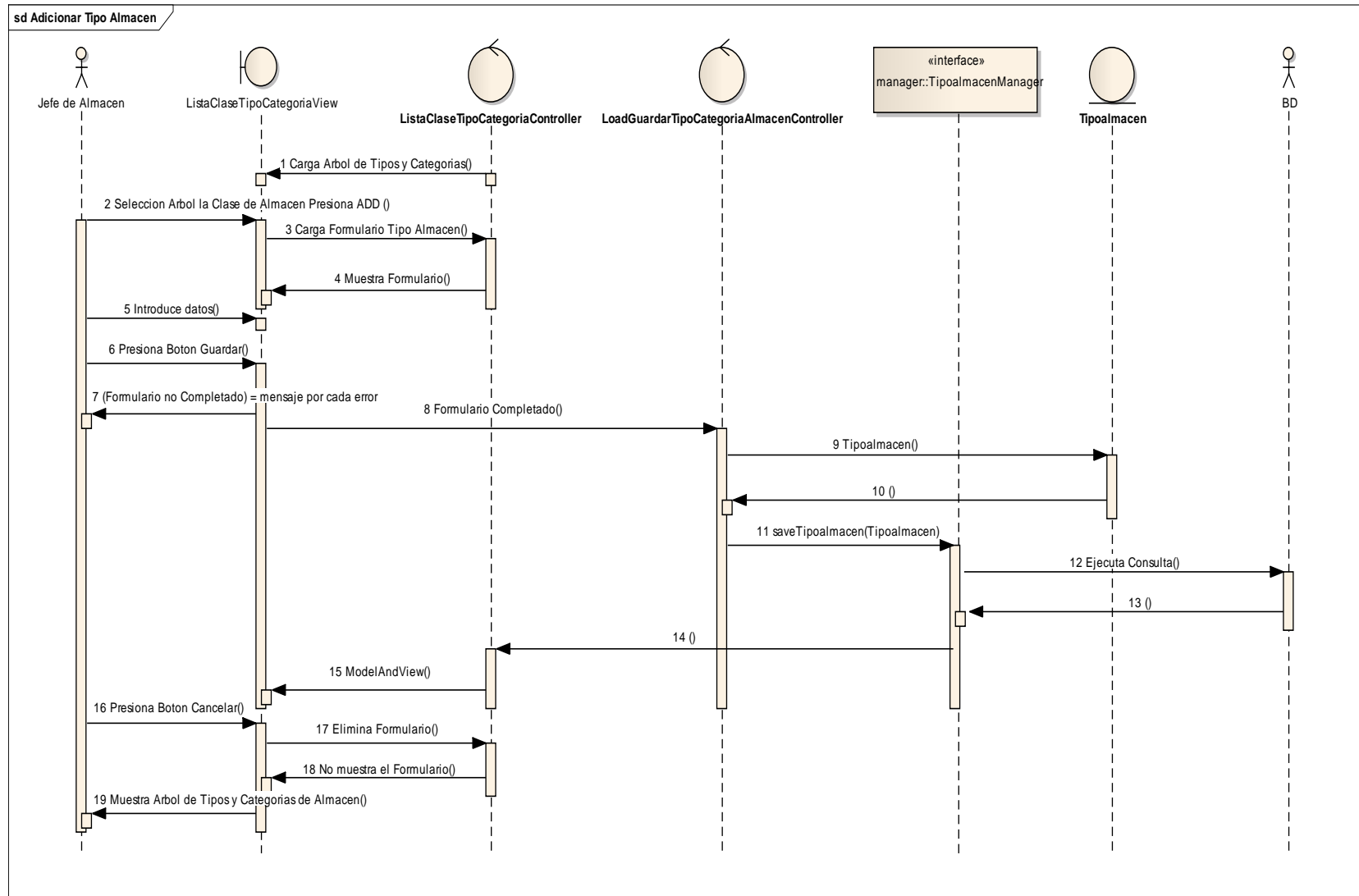


Figura 152. Diagrama de Secuencia: CU Adicionar Tipo Almacén

1.1.2.2.3.9.2.1.38. Diagrama de Secuencia: Caso de Uso Modificar Tipo Almacén

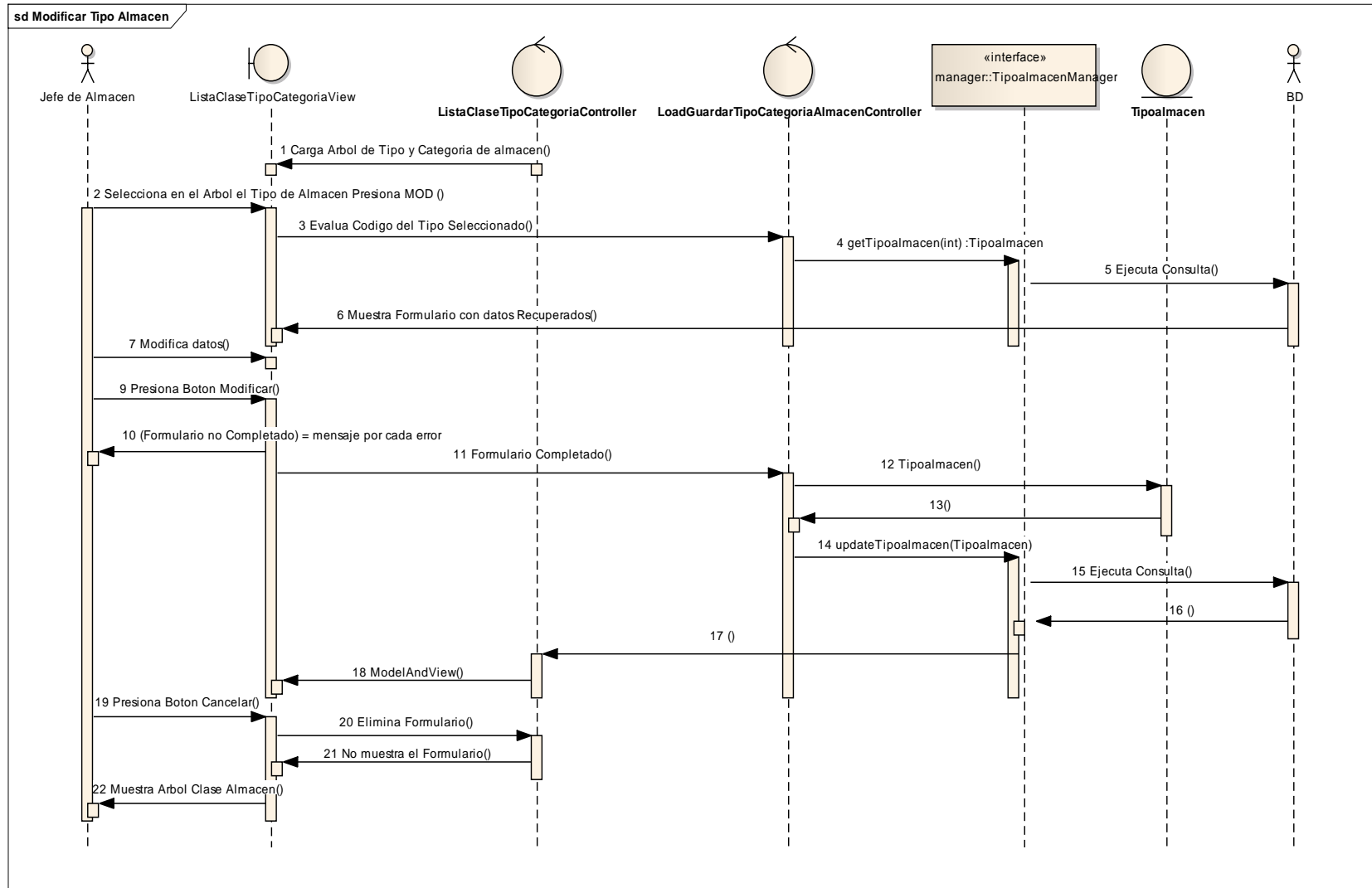


Figura 153. Diagrama de Secuencia: CU Modificar Tipo Almacén

1.1.2.2.3.9.2.1.39. Diagrama de Secuencia: Caso de Uso Eliminar Tipo Almacén

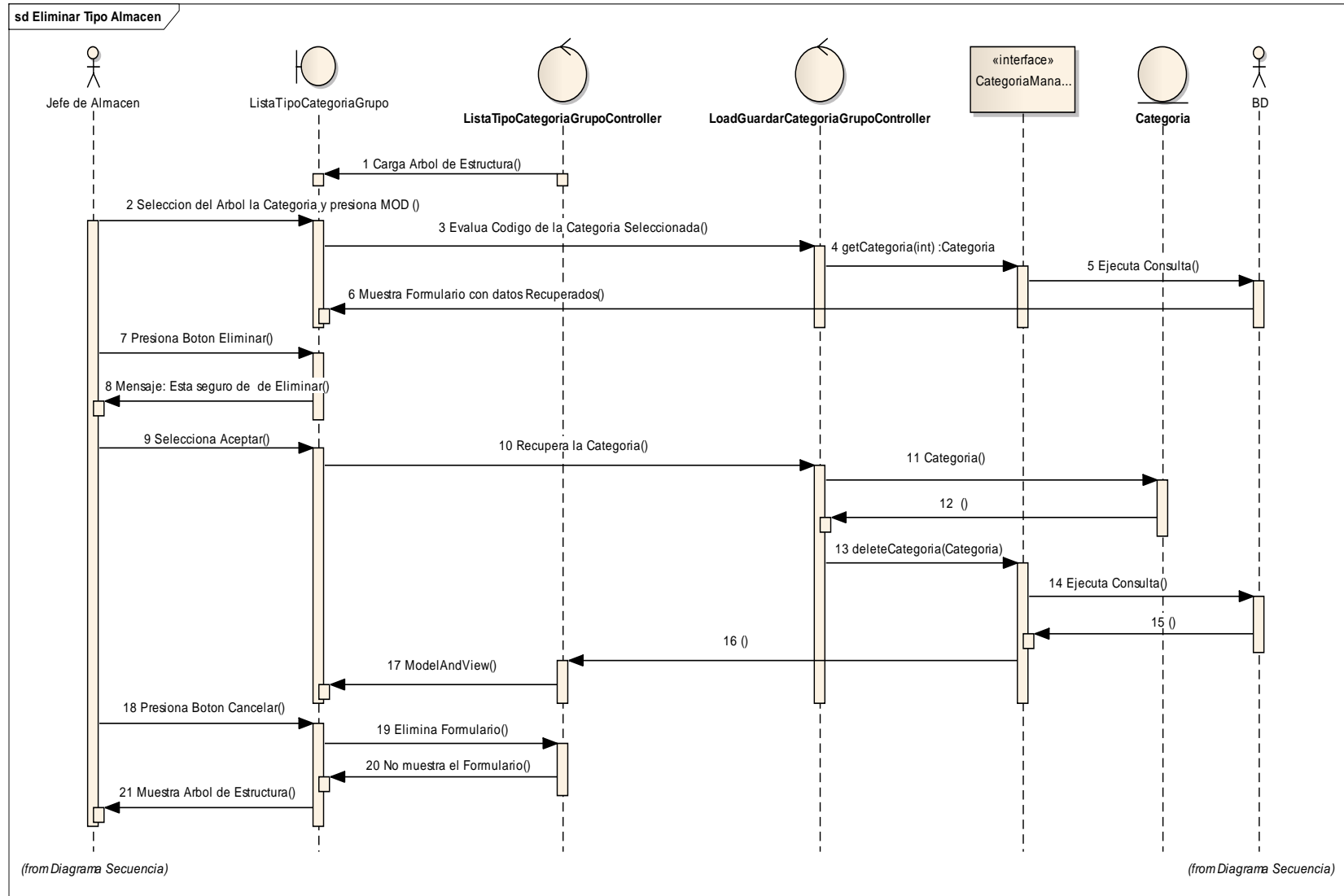


Figura 154. Diagrama de Secuencia: CU Eliminar Tipo Almacén

1.1.2.2.3.9.2.1.40. Diagrama de Secuencia: Caso de Uso Salir Sistema

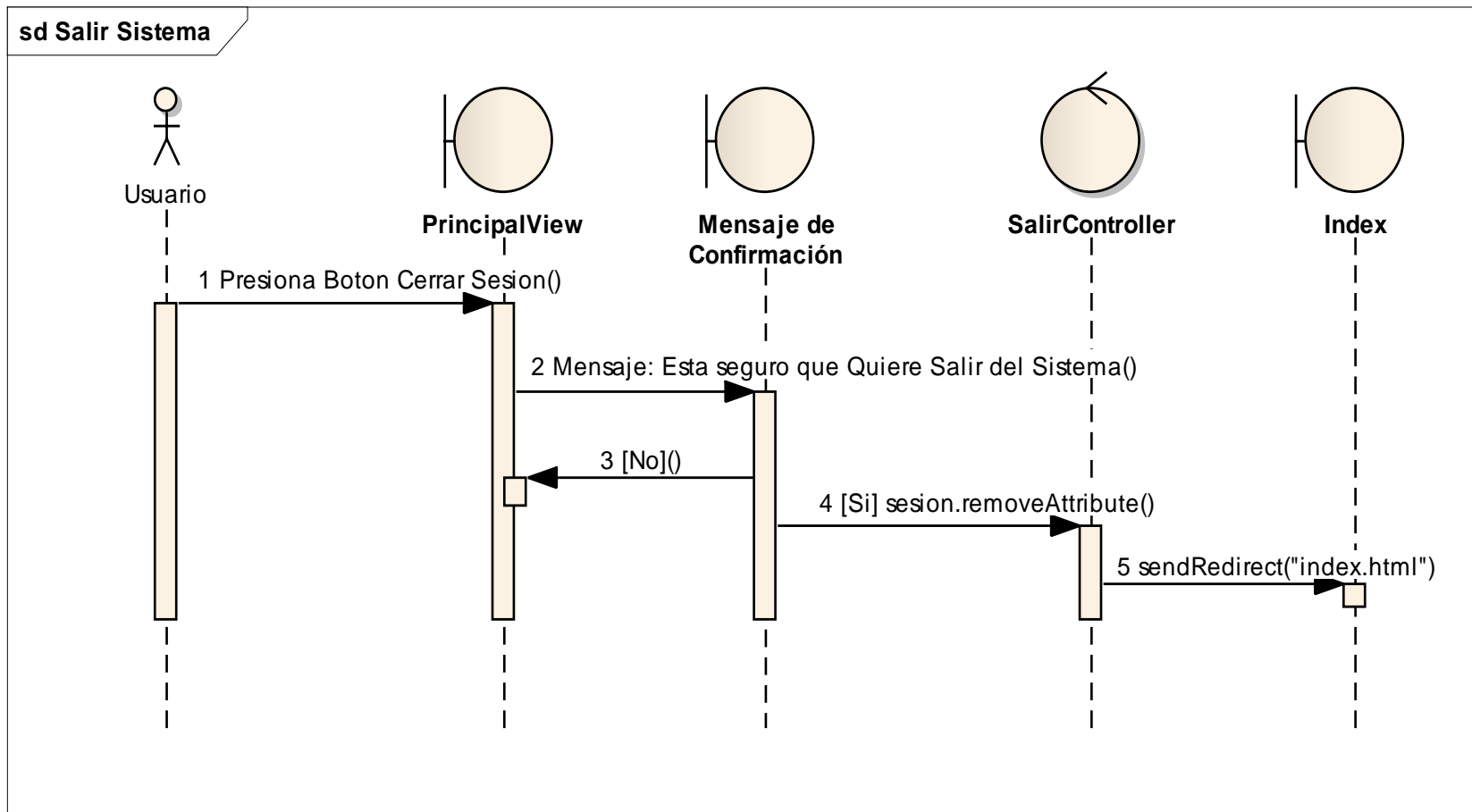


Figura 155. Diagrama de Secuencia: CU Salir Sistema

1.1.2.2.3.9.2.1.41. Diagrama de Secuencia: Caso de Uso Adicionar Ingreso de Producto

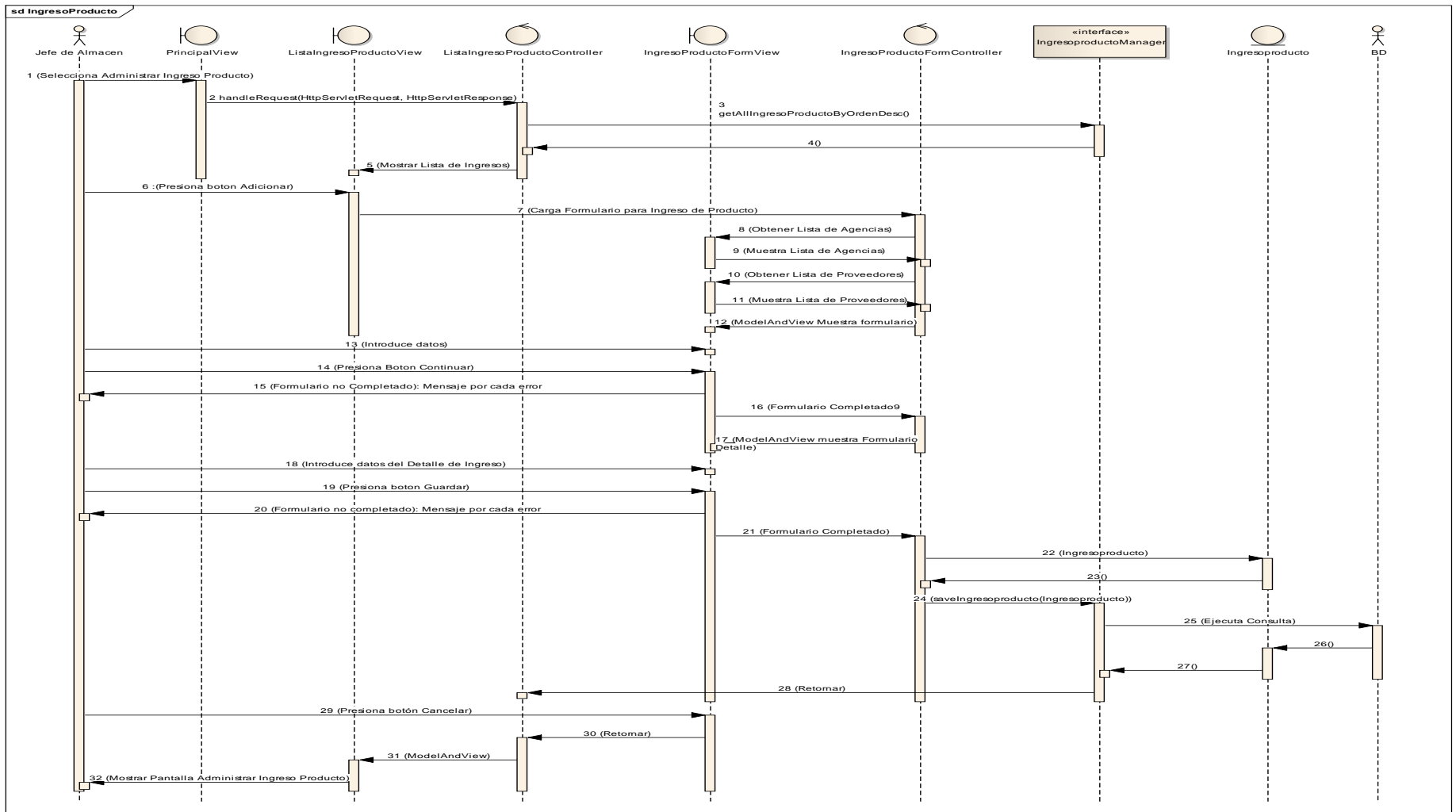


Figura 156. Diagrama de Secuencia: CU Adicionar Ingreso de Producto

1.1.2.2.3.9.2.1.42. Diagrama de Secuencia: Caso de Uso Adicionar Salida de Producto

Figura 157. Diagrama de Secuencia: CU Adicionar Salida de Producto

1.1.2.2.3.10. Modelo de Datos

Previendo que la persistencia de la información del Sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

1.1.2.2.3.10.1. Modelado de Diagrama de Clases

1.1.2.2.3.10.1.1. Introducción

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un Diagrama de Clases representa las clases del Sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de Casos de Uso aporta información para establecer las clases, objetos, atributos y operaciones.

1.1.2.2.3.10.1.2. Mecanismos de Abstracción

- Clasificación / Instanciación
- Composición / Descomposición
- Agrupación / Individualización
- Especialización / Generalización

La clasificación es uno de los mecanismos de abstracción más utilizados. La clase define el ámbito de definición de un conjunto de objetos, y cada objeto pertenece a una clase. Los objetos se crean por instanciación de las clases.

Cada clase se representa en un rectángulo con tres compartimientos:

- Nombre de la clase
- Atributos de la clase
- Operaciones de la clase

Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad para los elementos que son:

- (-) Privado: es el más fuerte. Esta parte es totalmente invisible.
- (#) Los atributos u operaciones protegidas están visibles para las clases de herencia.
- (+) Los atributos u operaciones públicas son visibles desde otras clases y también por clases de herencia.

1.1.2.2.3.10.1.4. Especificación de Métodos

A continuación una especificación de los métodos de los objetos utilizados en el proyecto, se mostrara el pseudo-código preliminar de cada método.

1.1.2.2.3.10.1.4.1. Objeto Agencia

OBJETO:	Agencia
MÉTODO:	getAllAgencia
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>return getHibernateTemplate().loadAll(Agencia.class);</pre>	

OBJETO:	Agencia
MÉTODO:	deleteAgencia
ARGUMENTO:	(Agencia agencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>getHibernateTemplate().delete(agencia);</pre>	

OBJETO:	Agencia
MÉTODO:	saveAgencia
ARGUMENTO:	(Agencia agencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>getHibernateTemplate().save(agencia);</pre>	

OBJETO:	Agencia
MÉTODO:	getAgencia
ARGUMENTO:	(int codagencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>return (Agencia) getHibernateTemplate().get(Agencia.class, codagencia);</pre>	

OBJETO:	Agencia
MÉTODO:	updateAgencia
ARGUMENTO:	(Agencia agencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().update(agencia);	

OBJETO:	Agencia
MÉTODO:	getAllAgenciaByOrdenAsc
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Agencia order by nombre asc");	

OBJETO:	Agencia
MÉTODO:	getAllAgenciaByNombre
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Agencia where (upper(nombre) LIKE upper('"+dato+"%')) order by nombre asc");	

Tabla 61. Objeto Agencia

1.1.2.2.3.10.1.4.2. Objeto Categoría

OBJETO:	Categoría
MÉTODO:	getAllCategoríaByOrdenAsc
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Categoría order by codcat asc");	

OBJETO:	Categoría
MÉTODO:	saveCategoria
ARGUMENTO:	(Categoria categoria);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().save(categoria);	

OBJETO:	Categoría
MÉTODO:	updateCategoria
ARGUMENTO:	(Categoria categoria);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().update(categoria);	

OBJETO:	Categoría
MÉTODO:	deleteCategoria
ARGUMENTO:	(Categoria categoria);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().delete(categoria);	

OBJETO:	Categoría
MÉTODO:	getAllCategoriaByTipoalmacen
ARGUMENTO:	(int codtipoa);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Categoria where tipoalmacen.codtipoa="+codtipoa);	

OBJETO:	Categoría
MÉTODO:	getAllCategoriaByClasealmacen
ARGUMENTO:	(int codclaalm);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Categoria where tipoalmacen.clasealmacen.codclaalm="+codclaalm);	

OBJETO:	Categoría
---------	-----------

MÉTODO:	getCategoria
ARGUMENTO:	(int codcat);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Categoria)getHibernateTemplate().get(Categoria.class, codcat);	

Tabla 62. Objeto Categoría

1.1.2.2.3.10.1.4.3. Clase Almacén

OBJETO:	Clase Almacén
MÉTODO:	getAllClasealmacenByOrdenAsc
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Clasealmacen order by codclaalm asc");	

OBJETO:	Clase Almacén
MÉTODO:	getClasealmacen
ARGUMENTO:	(int codclaalm);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Clasealmacen) getHibernateTemplate().get(Clasealmacen.class, codclaalm);	

Tabla 63. Objeto Clase Almacén

1.1.2.2.3.10.1.4.4. Objeto Departamento

OBJETO:	Departamento
MÉTODO:	getAllDepartamentoByOrdenAlfAsc
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Departamento order by nombre asc");	

Tabla 64. Objeto Departamento

1.1.2.2.3.10.1.4.5. Objeto DetalleIngreso

OBJETO:	DetalleIngreso
MÉTODO:	saveDetalleingreso
ARGUMENTO:	(Detalleingreso detingreso);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().save(detingreso);	

OBJETO:	DetalleIngreso
MÉTODO:	deleteDetalleingreso
ARGUMENTO:	(Detalleingreso detingreso);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().delete(detingreso);	

OBJETO:	DetalleIngreso
MÉTODO:	getDetalleingreso
ARGUMENTO:	(int coddetalle);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Detalleingreso) getHibernateTemplate().get(Detalleingreso.class, coddetalle);	

Tabla 65. Objeto DetalleIngreso

1.1.2.2.3.10.1.4.6. Objeto DetalleSalida

OBJETO:	DetalleSalida
MÉTODO:	getDetallesalida
ARGUMENTO:	(int coddetalle);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Detallesalida) getHibernateTemplate().get(Detallesalida.class, coddetalle);	

OBJETO:	DetalleSalida
MÉTODO:	saveDetallesalida
ARGUMENTO:	(Detallesalida detsalida);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().save(detsalida);	

OBJETO:	DetalleSalida
MÉTODO:	deleteDetallesalida
ARGUMENTO:	(Detallesalida detsalida);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().delete(detsalida);	

Tabla 66. Objeto DetalleSalida

1.1.2.2.3.10.1.4.7. Objeto Direccion

OBJETO:	Direccion
MÉTODO:	saveDireccion
ARGUMENTO:	(Direccion direccion);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().save(direccion);	

OBJETO:	Direccion
MÉTODO:	updateDireccion
ARGUMENTO:	(Direccion direccion);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().update(direccion);	

OBJETO:	Direccion
MÉTODO:	deleteDireccion
ARGUMENTO:	(Direccion direccion);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().delete(direccion);	

OBJETO:	Direccion
MÉTODO:	getDireccionByCodper
ARGUMENTO:	(int codper);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
List lista=getHibernateTemplate().find("from Direccion where (personal.codpersonal="+codper+""); if(lista.size(>0){ return (Direccion)lista.get(0); } return null;	

Tabla 67. Objeto Direccion

1.1.2.2.3.10.1.4.8. Objeto EstrucOrganizativa

OBJETO:	EstrucOrganizativa
MÉTODO:	getAllEstrucOrganizativa
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().loadAll(Estrucorganizativa.class);	

OBJETO:	EstrucOrganizativa
MÉTODO:	getAllEstrucOrganizativaByIdPadreyTipo
ARGUMENTO:	(int idpadre, int tipo, int codagencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Estrucorganizativa where idpadre="+idpadre+" and tipo= "+tipo+" and agencia.codagencia="+codagencia);	

OBJETO:	EstrucOrganizativa
MÉTODO:	getEstrucOrganizativa
ARGUMENTO:	(int idhijo);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (EstrucOrganizativa)getHibernateTemplate().get(EstrucOrganizativa.class, idhijo);	

OBJETO:	EstrucOrganizativa
MÉTODO:	deleteEstrucOrganizativa
ARGUMENTO:	(EstrucOrganizativa estructura);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().delete(estructura);	

OBJETO:	EstrucOrganizativa
MÉTODO:	saveEstrucOrganizativa
ARGUMENTO:	(EstrucOrganizativa estructura);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().save(estructura);	

OBJETO:	EstrucOrganizativa
MÉTODO:	getAllAreaByAgencia
ARGUMENTO:	(int codagencia)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from EstrucOrganizativa where (tipo=1 and agencia.codagencia="+codagencia+"");	

OBJETO:	EstrucOrganizativa
MÉTODO:	updateEstrucOrganizativa
ARGUMENTO:	(EstrucOrganizativa estructura);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().update(estructura);	

OBJETO:	EstrucOrganizativa
MÉTODO:	getAllCargoByAgencia
ARGUMENTO:	(int codagencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Estrucorganizativa where (tipo=2 and agencia.codagen="+codagencia+"");	

Tabla 68. Objeto EstrucOrganizativa

1.1.2.2.3.10.1.4.9. Objeto Ingresoproducto

OBJETO:	Ingresoproducto
MÉTODO:	getAllIngresoproducto
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().loadAll(Ingresoproducto.class);	

OBJETO:	Ingresoproducto
MÉTODO:	getIngresoproducto
ARGUMENTO:	(int codingreso);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Ingresoproducto) getHibernateTemplate().get(Ingresoproducto.class, codingreso);	

OBJETO:	Ingresoproducto
MÉTODO:	saveIngresoproducto
ARGUMENTO:	(Ingresoproducto ingresoproducto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().save(ingresoproducto);	

OBJETO:	Ingresoproducto
MÉTODO:	deleteIngresoproducto
ARGUMENTO:	(Ingresoproducto ingresoproducto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().delete(ingresoproducto);	

OBJETO:	Ingresoproducto
MÉTODO:	updateIngresoproducto
ARGUMENTO:	(Ingresoproducto ingresoproducto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().update(ingresoproducto);	

OBJETO:	Ingresoproducto
MÉTODO:	getAllIngresoproductoByOrdenDes
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Ingresoproducto order by numeropedido asc");	

OBJETO:	Ingresoproducto
MÉTODO:	getAllIngresoproductoByNroPedido
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Ingresoproducto where (upper(numeropedido) LIKE upper('"+dato+"%')) order by numeropedido asc");	

OBJETO:	Ingresoproducto
MÉTODO:	getUltimoIngresoproducto
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>List lista= getHibernateTemplate().find("from Ingresoproducto order by codingreso desc"); if(lista.size(>0){ return (Ingresoproducto)lista.get(0); } return null;</pre>	

Tabla 69. Objeto Ingresoproducto

1.1.2.2.3.10.1.4.10. Objeto Kardex

OBJETO:	Kardex
MÉTODO:	getKardex
ARGUMENTO:	(int codkardex);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>return (Kardex) getHibernateTemplate().get(Kardex.class, codkardex);</pre>	

OBJETO:	Kardex
MÉTODO:	saveKardex
ARGUMENTO:	(Kardex kardex);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>getHibernateTemplate().save(kardex);</pre>	

OBJETO:	Kardex
MÉTODO:	deleteKardex
ARGUMENTO:	(Kardex kardex);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
<pre>getHibernateTemplate().delete(kardex);</pre>	

OBJETO:	Kardex
MÉTODO:	updateKardex
ARGUMENTO:	(Kardex kardex);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
getHibernateTemplate().update(kardex);	

OBJETO:	Kardex
MÉTODO:	getUltimoKardexbyProducto
ARGUMENTO:	(int codproducto, int codagen);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
List lista= getHibernateTemplate().find("from Kardex where ((producto.codpro="+codproducto+")and (agencia.codagen="+codagen+")) order by codkardex desc"); if(lista.size(>0){ return (Kardex)lista.get(0); } return null;	

OBJETO:	Kardex
MÉTODO:	getUltimoKardexIorS
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
List lista= getHibernateTemplate().find("from Kardex order by codkardex desc"); if(lista.size(>0){ return (Kardex)lista.get(0); } return null;	

Tabla 70. Objeto Kardex

1.1.2.2.3.10.1.4.11. Objeto Localidad

OBJETO:	Localidad
MÉTODO:	getAllLocalidadByOrdenAlfAsc
ARGUMENTO:	();

CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Localidad order by nombre asc");	

OBJETO:	Localidad
MÉTODO:	getAllLocalidadByProvincia
ARGUMENTO:	(int codprovincia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Localidad where provincia.codprov='"+codprovincia);	

OBJETO:	Localidad
MÉTODO:	getLocalidad
ARGUMENTO:	(int codlocalidad);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Localidad)getHibernateTemplate().get(Localidad.class, codlocalidad);	

Tabla 71. Objeto Localidad

1.1.2.2.3.10.1.4.12. Objeto Marca

OBJETO:	Marca
MÉTODO:	getAllMarca
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Marca order by codmarca asc");	

OBJETO:	Marca
MÉTODO:	getMarca
ARGUMENTO:	(int codmarca);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Marca) getHibernateTemplate().get(Marca.class, codmarca);	

Tabla 72. Objeto Marca

1.1.2.2.3.10.1.4.13. Objeto Módulo

OBJETO:	Módulo
MÉTODO:	getAllModulo
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return getHibernateTemplate().find("from Modulo order by codmod asc");	

OBJETO:	Módulo
MÉTODO:	getModulo
ARGUMENTO:	(int codmod);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN:	
return (Modulo) getHibernateTemplate().get(Modulo.class, codmod);	

Tabla 73. Objeto Módulo

1.1.2.2.3.10.1.4.14. Objeto Personal

OBJETO:	Personal
MÉTODO:	getAllPersonal
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().loadAll(Personal.class);	

OBJETO:	Personal
MÉTODO:	deletePersonal
OBJETO:	Personal
MÉTODO:	(Personal personal); savePersonal
ARGUMENTO:	Public (Personal personal);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(personal);	

OBJETO:	Personal
MÉTODO:	updatePersonal
ARGUMENTO:	(Personal personal);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(personal);	

OBJETO:	Personal
MÉTODO:	getPersonal
ARGUMENTO:	(int codpersonal);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Personal) getHibernateTemplate().get(Personal.class, codpersonal);	

OBJETO:	Personal
MÉTODO:	getAllPersonalByAp
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Personal where (estado=1)and ((upper(ap) LIKE upper('"+dato+"%')) or (upper(am) LIKE upper('"+dato+"%')) or (upper(nombre) LIKE upper('"+dato+"%')))) order by ap asc");	

OBJETO:	Personal
MÉTODO:	getAllPersonalByEstado
ARGUMENTO:	(int estado);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Personal where (estado="+estado+") order by ap asc");	

OBJETO:	Personal
MÉTODO:	getAllPersonalNoPertenece
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Personal where codpersonal NOT IN (select p.personal.codpersonal from Pertenece p where (p.estado=1))");	

OBJETO:	Personal
MÉTODO:	getAllPersonalNoPertenece
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Personal where (codpersonal NOT IN (select p.personal.codpersonal from Pertenece p where (p.estado=1))) and ((upper(nombre) LIKE upper('"+dato+"%')) or (upper(ap) LIKE upper('"+dato+"%'))or (upper(am) LIKE upper('"+dato+"%'))) order by nombre asc");	

Tabla 74. Objeto Personal

1.1.2.2.3.10.1.4.15. Objeto Pertenece

OBJETO:	Pertenece
---------	-----------

MÉTODO:	getAllPertenece
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Pertenece where (estado=1)");	

OBJETO:	Pertenece
MÉTODO:	getPertenece
ARGUMENTO:	(int codpertenece);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Pertenece) getHibernateTemplate().get(Pertenece.class, codpertenece);	

OBJETO:	Pertenece
MÉTODO:	savePertenece
ARGUMENTO:	(Pertenece pertenece);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(pertenece);	

OBJETO:	Pertenece
MÉTODO:	deletePertenece
ARGUMENTO:	(Pertenece pertenece);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(pertenece);	

OBJETO:	Pertenece
MÉTODO:	getAllPertenecebyEstadoandAgencia
ARGUMENTO:	(int estado, int codagencia);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Pertenece where (estado='"+estado+"') and (estrucorganizativa.agencia.codagen='"+codagencia+"')");	

Tabla 75. Objeto Pertenece

OBJETO:	Pertenece
MÉTODO:	updatePertenece
ARGUMENTO:	(Pertenece pertenece);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(pertenece);	

1.1.2.2.3.10.1.4.16. Objeto Producto

OBJETO:	Producto
MÉTODO:	getAllProducto
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Producto where (estado!=0)");	

OBJETO:	Producto
MÉTODO:	getAllProductoByCategoria
ARGUMENTO:	(int codcat);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Producto where (estado!=0) and (categoria.codcat="+codcat+"");	

OBJETO:	Producto
MÉTODO:	getAllProductoByTipoalmacen
ARGUMENTO:	(int codtipoa);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Producto where (estado!=0) and (categoria.tipoalmacen.codtipoa="+codtipoa+"");	

OBJETO:	Producto
MÉTODO:	getAllProductoByClasealmacen
ARGUMENTO:	(int codclaalm);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Producto where (estado!=0) and (categoria.tipoalmacen.clasealmacen.codclaalm="+codclaalm+"");	

OBJETO:	Producto
MÉTODO:	saveProducto
ARGUMENTO:	(Producto producto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(producto);	

OBJETO:	Producto
MÉTODO:	updateProducto
ARGUMENTO:	(Producto producto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(producto);	

OBJETO:	Producto
MÉTODO:	deleteProducto
ARGUMENTO:	(Producto producto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
producto.setEstado(0);	

OBJETO:	Producto
MÉTODO:	getProducto
ARGUMENTO:	(int codpro);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Producto)getHibernateTemplate().get(Producto.class, codpro);	

OBJETO:	Producto
MÉTODO:	getAllProductoByEstado
ARGUMENTO:	(int estado);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Producto where (estado="+estado+"");	

OBJETO:	Producto
MÉTODO:	getAllProductoByCodigo
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public

ESPECIFICACIÓN	
return getHibernateTemplate().find("from Producto where (estado!=0) and (upper(codigo) LIKE upper(''+dato+'%')) order by codigo asc");	

Tabla 76. Objeto Producto

1.1.2.2.3.10.1.4.17. Objeto Proveedor

OBJETO:	Proveedor
MÉTODO:	getAllProveedor
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Proveedor where (estado=1)");	

OBJETO:	Proveedor
MÉTODO:	getAllProveedorByRazonSocial
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Proveedor where (upper(empresa) LIKE upper(''+dato+'%')) order by empresa asc");	

OBJETO:	Proveedor
MÉTODO:	getProveedor
ARGUMENTO:	(int codproveedor);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Proveedor) getHibernateTemplate().get(Proveedor.class, codproveedor);	

OBJETO:	Proveedor
MÉTODO:	saveProveedor
ARGUMENTO:	(Proveedor proveedor);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(proveedor);	

OBJETO:	Proveedor
MÉTODO:	updateProveedor
ARGUMENTO:	(Proveedor proveedor);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(proveedor);	

OBJETO:	Proveedor
MÉTODO:	deleteProveedor
ARGUMENTO:	(Proveedor proveedor);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(proveedor);	

Tabla 77. Objeto Proveedor

1.1.2.2.3.10.1.4.18. Objeto Provincia

OBJETO:	Provincia
MÉTODO:	getAllProvinciaByOrdenAlfAsc
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Provincia order by nombre asc");	

OBJETO:	Provincia
MÉTODO:	getAllProvinciaByDepartamento
ARGUMENTO:	(int coddepartamento);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Provincia where departamento.coddep="+coddepartamento);	

Tabla 78. Objeto Provincia

1.1.2.2.3.10.1.4.19. Objeto Roles

OBJETO:	Roles
MÉTODO:	getAllRoles
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().loadAll(Roles.class);	

OBJETO:	Roles
MÉTODO:	deleteRoles
ARGUMENTO:	(Roles roles);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(roles);	

OBJETO:	Roles
MÉTODO:	saveRoles
ARGUMENTO:	(Roles roles);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(roles);	

OBJETO:	Roles
MÉTODO:	updateRoles
ARGUMENTO:	(Roles roles);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(roles);	

OBJETO:	Roles
MÉTODO:	getRoles
ARGUMENTO:	(int codrol);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Roles) getHibernateTemplate().get(Roles.class, codrol);	

OBJETO:	Roles
MÉTODO:	getAllRolesByOrdenDes
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Roles order by nombre asc");	

OBJETO:	Roles
MÉTODO:	getAllRolesByNombre
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Roles where (upper(nombre) LIKE upper('"+dato+"%')) order by nombre asc");	

Tabla 79. Objeto Roles

1.1.2.2.3.10.1.4.20. Objeto Salidaproducto

OBJETO:	Salidaproducto
MÉTODO:	getAllSalidaproducto
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().loadAll(Salidaproducto.class);	

OBJETO:	Salidaproducto
MÉTODO:	getSalidaproducto
ARGUMENTO:	(int codsalida);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Salidaproducto) getHibernateTemplate().get(Salidaproducto.class, codsalida);	

OBJETO:	Salidaproducto
MÉTODO:	saveSalidaproducto
ARGUMENTO:	(Salidaproducto salidaproducto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(salidaproducto);	

OBJETO:	Salidaproducto
MÉTODO:	deleteSalidaproducto
ARGUMENTO:	(Salidaproducto salidaproducto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(salidaproducto);	

OBJETO:	Salidaproducto
MÉTODO:	updateSalidaproducto
ARGUMENTO:	(Salidaproducto salidaproducto);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(salidaproducto);	

OBJETO:	Salidaproducto
MÉTODO:	getAllSalidaproductoByNroForm
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Salidaproducto where (upper(numeroform) LIKE upper("'" + dato + "%')) order by numeroform asc");	

OBJETO:	Salidaproducto
MÉTODO:	getUltimoSalidaproducto
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
List lista= getHibernateTemplate().find("from Salidaproducto order by codsalida desc");	
<pre> if(lista.size(>0){ return (Salidaproducto)lista.get(0); } return null; </pre>	

Tabla 80. Objeto Salidaproducto

1.1.2.2.3.10.1.4.21. Objeto Telefono

OBJETO:	Telefono
MÉTODO:	saveTelefono
ARGUMENTO:	(Telefono telefono)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(telefono);	

OBJETO:	Telefono
MÉTODO:	updateTelefono
ARGUMENTO:	(Telefono telefono);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(telefono);	

OBJETO:	Telefono
MÉTODO:	deleteTelefono
ARGUMENTO:	(Telefono telefono);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(telefono);	

OBJETO:	Telefono
MÉTODO:	getTelefonoByCodper
ARGUMENTO:	(int codper);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
List lista=getHibernateTemplate().find("from Telefono where (personal.codpersonal='"+codper+""); if(lista.size(>0){ return (Telefono)lista.get(0); } return null;	

Tabla 81. Objeto Telefono

1.1.2.2.3.10.1.4.22. Objeto Tipoalmacen

OBJETO:	Tipoalmacen
MÉTODO:	getAllTipoalmacenByOrdenAsc
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Tipoalmacen order by codtipoa asc");	

OBJETO:	Tipoalmacen
MÉTODO:	getAllTipoalmacenByNombre
ARGUMENTO:	(String dato);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Tipoalmacen where (upper(nombre) LIKE upper('"+dato+"%')) order by nombre asc");	

OBJETO:	Tipoalmacen
MÉTODO:	saveTipoalmacen
ARGUMENTO:	(Tipoalmacen tipoalmacen);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(tipoalmacen);	

OBJETO:	Tipoalmacen
MÉTODO:	updateTipoalmacen
ARGUMENTO:	(Tipoalmacen tipoalmacen);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(tipoalmacen);	

OBJETO:	Tipoalmacen
MÉTODO:	deleteTipoalmacen
ARGUMENTO:	(Tipoalmacen tipoalmacen);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(tipoalmacen);	

OBJETO:	Tipoalmacen
MÉTODO:	getAllTipoalmacenByClasealmacen
ARGUMENTO:	(int codclaalm);
CONTROL DE EXPORTACIÓN:	Public

ESPECIFICACIÓN	
return getHibernateTemplate().find("from Tipoalmacen where (clasealmacen.codclaalm="+codclaalm+") order by nombre asc");	
OBJETO:	Tipoalmacen
MÉTODO:	getTipoalmacen
ARGUMENTO:	(int codtipoa);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Tipoalmacen)getHibernateTemplate().get(Tipoalmacen.class, codtipoa);	

Tabla 82. Objeto Tipoalmacen

1.1.2.2.3.10.1.4.23. Objeto Unidad

OBJETO:	Unidad
MÉTODO:	getAllUnidad
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Unidad order by codunidad asc");	

OBJETO:	Unidad
MÉTODO:	getUnidad
ARGUMENTO:	(int codunidad);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Unidad) getHibernateTemplate().get(Unidad.class, codunidad);	

Tabla 83. Objeto Unidad

1.1.2.2.3.10.1.4.24. Objeto Usuario

OBJETO:	Usuario
MÉTODO:	getAllUsuario
ARGUMENTO:	();
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().loadAll(Usuario.class);	

OBJETO:	Usuario
MÉTODO:	deleteUsuario
ARGUMENTO:	(Usuario usuario);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(usuario);	

OBJETO:	Usuario
MÉTODO:	saveUsuario
ARGUMENTO:	(Usuario usuario);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(usuario);	

OBJETO:	Usuario
MÉTODO:	updateUsuario
ARGUMENTO:	(Usuario usuario);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(usuario);	

OBJETO:	Usuario
MÉTODO:	getUsuario
ARGUMENTO:	(int codusuario);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return (Usuario) getHibernateTemplate().get(Usuario.class, codusuario);	

OBJETO:	Usuario
MÉTODO:	getAllUsuarioByEstado
ARGUMENTO:	(int estado);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return getHibernateTemplate().find("from Usuario where (estado='"+estado+"') order by ap asc");	

OBJETO:	Usuario
MÉTODO:	getUsuarioByCodper
ARGUMENTO:	(int codper);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
List lista=getHibernateTemplate().find("from Usuario where (personal.codpersonal='"+codper+"')"); if(lista.size(>0){ return (Usuario)lista.get(0); } return null;	

OBJETO:	Usuario
MÉTODO:	getLogear
ARGUMENTO:	(String usuario, String clave);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
List lista = getHibernateTemplate().find("from Usuario where (login='"+usuario+"') and (password='"+clave+"') "); if(lista.size(>0){ return (Usuario) lista.get(0); } return null;	

Tabla 84. Objeto Usuario

1.1.2.2.3.10.1.4.25. Objeto Usuro1

OBJETO:	Usuro1
MÉTODO:	deleteUsuro1
ARGUMENTO:	(Usuro1 usuro1);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().delete(usuro1);	

OBJETO:	Usuro1
MÉTODO:	saveUsuro1
ARGUMENTO:	(Usuro1 usuro1);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().save(usuro1);	

OBJETO:	Usuro1
MÉTODO:	updateUsuro1
ARGUMENTO:	(Usuro1 usuro1);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
getHibernateTemplate().update(usuro1);	

OBJETO:	Usuro1
MÉTODO:	getUsuro1ByCodusuYcodrol
ARGUMENTO:	(int codusuario, int codrol);
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> List lista=getHibernateTemplate().find("from Usuro1 where (usuario.codusuario="+codusuario+") and (roles.codrol="+codrol+") and (estado="+1+")"); if(lista.size(>0){ return (Usuro1)lista.get(0); } </pre>	

Tabla 85. Objeto Usuro1

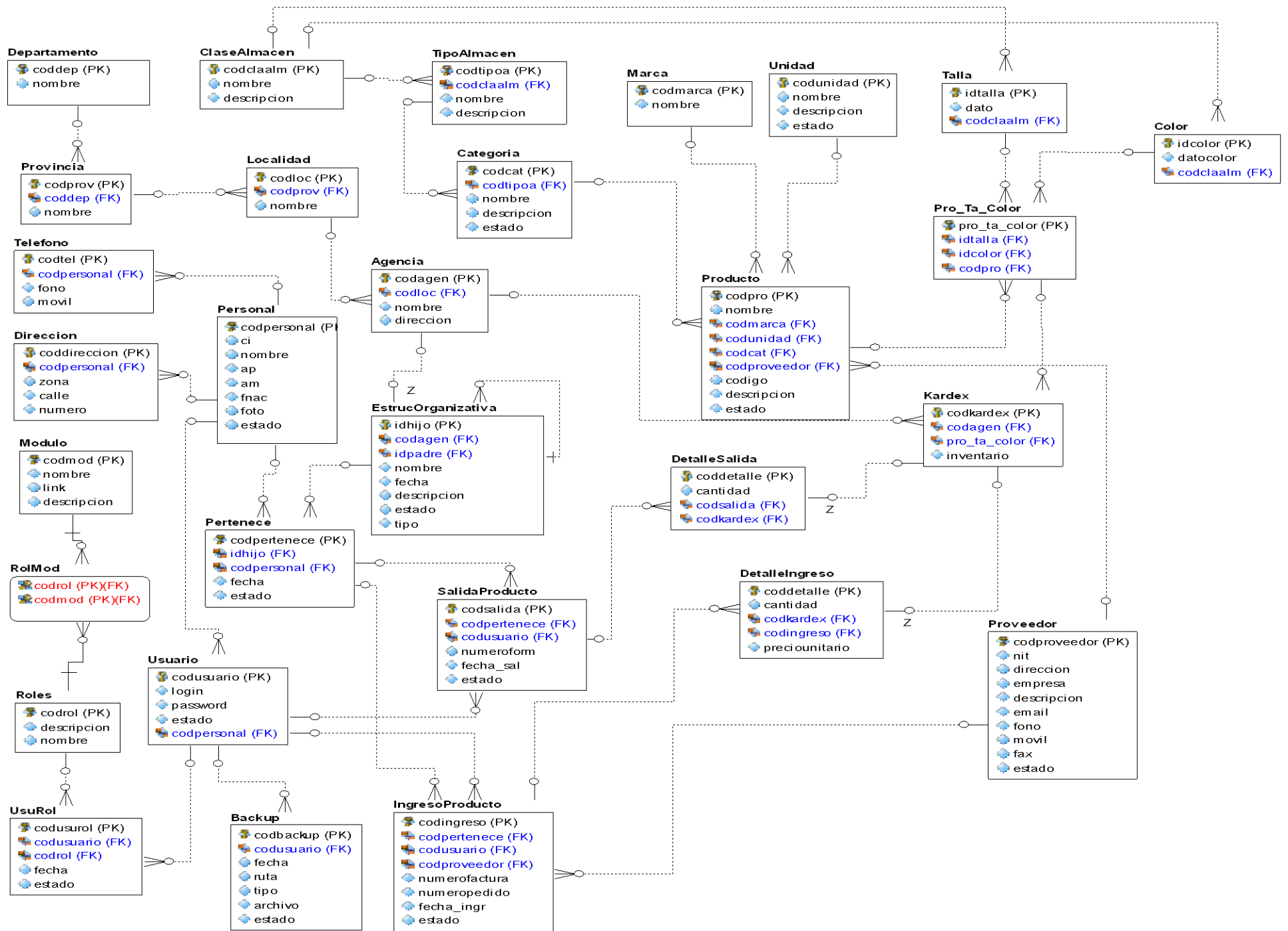
1.1.2.2.3.10.2. Modelado Entidad Relación

1.1.2.2.3.10.2.1. Introducción

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

Se suele cometer el error de ir creando nuevas tablas a medida que se van necesitando, haciendo así el modelo de datos y la construcción física de las tablas simultáneamente. El resultado de esto acaba siendo un Sistema de información parcheado, con datos dispersos que terminan por no cumplir adecuadamente los requisitos necesarios.

1.1.2.2.3.10.2.2. Modelo Entidad / Relación



1.1.2.2.3.10.2.3. Creación de la Base de Datos (SQL)

```
--  
-- ER/Studio 7.5 SQL Code Generation  
-- Company : NOEMI  
-- Project : BDAlmacenN.dm1  
-- Author : NOEMI  
--  
-- Date Created : Saturday, August 17, 2013 23:19:25  
-- Target DBMS : PostgreSQL 8.0  
  
--  
-- TABLE: "Agencia"  
--  
  
CREATE TABLE "Agencia"(  
  
    codagen int4 NOT NULL,  
    codloc int4,  
    nombre varchar(100),  
    direccion varchar(150),  
    CONSTRAINT "PK4" PRIMARY KEY (codagen)  
  
)  
;  
  
--  
-- TABLE: "Backup"  
--
```

```
CREATE TABLE "Backup"(  
    codbackup int4 NOT NULL,  
    codusuario int4,  
    fecha date,  
    ruta varchar(50),  
    tipo varchar(50),  
    archivo varchar(50),  
    CONSTRAINT "PK38" PRIMARY KEY (codbackup)  
  
)  
;  
  
--  
-- TABLE: "Categoria"  
--  
  
CREATE TABLE "Categoria"(  
  
    codcat int4 NOT NULL,  
    codtipoa int4,  
    nombre varchar(60),  
    descripcion varchar(150) NOT NULL,  
    estado int4,  
    CONSTRAINT "PK9" PRIMARY KEY (codcat)  
  
)  
;  
  
--  
-- TABLE: "ClaseAlmacen"
```

```
--  
CREATE TABLE "ClaseAlmacen"(  
    codclaalm    int4        NOT NULL,  
    nombre      varchar(150),  
    descripcion  varchar(200),  
    CONSTRAINT "PK5" PRIMARY KEY (codclaalm)  
  
)  
;
```

```
--  
-- TABLE: "Departamento"  
--
```

```
CREATE TABLE "Departamento"(  
  
    coddep    int4        NOT NULL,  
    nombre    varchar(20),  
    CONSTRAINT "PK1" PRIMARY KEY (coddep)  
  
)  
;
```

```
--  
-- TABLE: "DetalleIngreso"  
--
```

```
CREATE TABLE "DetalleIngreso"(  

```

```
    coddetalle    int4    NOT NULL,
    codkardex     int4,
    codingreso    int4,
    cantidad      int4,
    preciounitario float4,
    CONSTRAINT "PK15" PRIMARY KEY (coddetalle)

)
;

--
-- TABLE: "DetalleSalida"
--

CREATE TABLE "DetalleSalida"(

    coddetalle    int4    NOT NULL,
    codsalida     int4,
    codkardex     int4,
    cantidad      int4,
    CONSTRAINT "PK13" PRIMARY KEY (coddetalle)

)
;

--
-- TABLE: "Direccion"
--

CREATE TABLE "Direccion"(
```

```
coddireccion int4 NOT NULL,
codpersonal int4,
zona varchar(100),
calle varchar(100),
numero varchar(15),
CONSTRAINT "PK8_1" PRIMARY KEY (coddireccion)

)
;

--
-- TABLE: "EstrucOrganizativa"
--

CREATE TABLE "EstrucOrganizativa"(
    idhijo int4 NOT NULL,
    codagen int4,
    idpadre int4 NOT NULL,
    nombre varchar(150) NOT NULL,
    fecha date,
    descripcion varchar(200),
    estado int4,
    tipo int4 NOT NULL,
    CONSTRAINT "PK33" PRIMARY KEY (idhijo)

)
;

--
-- TABLE: "IngresoProducto"
```

```
--  
  
CREATE TABLE "IngresoProducto"(  
  
    codingreso    int4    NOT NULL,  
    codpertenece  int4,  
    codusuario    int4,  
    codproveedor  int4,  
    numerofactura varchar(20),  
    numeropedido  varchar(20),  
    fecha_ingr    date,  
    CONSTRAINT "PK10" PRIMARY KEY (codingreso)  
  
)  
;
```

```
--  
-- TABLE: "Kardex"  
--
```

```
CREATE TABLE "Kardex"(  
  
    codkardex    int4    NOT NULL,  
    codagen      int4,  
    codpro       int4,  
    inventario   int4,  
    tipo_kar     int4    NOT NULL,  
    fecha        date,  
    CONSTRAINT "PK14" PRIMARY KEY (codkardex)
```

```
)  
;  
  
--  
-- TABLE: "Localidad"  
--  
  
CREATE TABLE "Localidad"(  
  
    codloc int4 NOT NULL,  
    codprov int4,  
    nombre varchar(50),  
    CONSTRAINT "PK3" PRIMARY KEY (codloc)  
  
)  
;  
  
--  
-- TABLE: "Marca"  
--  
  
CREATE TABLE "Marca"(  
  
    codmarca int4 NOT NULL,  
    nombre varchar(60),  
    CONSTRAINT "PK48" PRIMARY KEY (codmarca)  
  
)  
;
```

```
--  
-- TABLE: "Modulo"  
--  
  
CREATE TABLE "Modulo"(  
    codmod    int4    NOT NULL,  
    nombre    varchar(60),  
    link      varchar(60),  
    descripcion varchar(60),  
    CONSTRAINT "PK39" PRIMARY KEY (codmod)  
  
)  
;  
  
--  
-- TABLE: "Personal"  
--  
  
CREATE TABLE "Personal"(  
  
    codpersonal int4    NOT NULL,  
    ci          varchar(15) NOT NULL,  
    nombre      varchar(60) NOT NULL,  
    ap         varchar(60),  
    am         varchar(60) NOT NULL,  
    fnac       date,  
    foto       varchar(100),  
    estado     int4    NOT NULL,  
    CONSTRAINT "PK1_1" PRIMARY KEY (codpersonal)
```

```
)  
;  
  
--  
-- TABLE: "Pertenece"  
--  
CREATE TABLE "Pertenece"(  
  
    codpertenece int4 NOT NULL,  
    idhijo       int4,  
    codpersonal  int4,  
    fecha        date,  
    estado       int4,  
    CONSTRAINT "PK34" PRIMARY KEY (codpertenece)  
  
)  
;  
  
--  
-- TABLE: "Producto"  
--  
CREATE TABLE "Producto"(  
  
    codpro       int4 NOT NULL,  
    codmarca     int4,  
    codunidad   int4,  
    codcat       int4,  
    codigo       varchar(15),  
    descripcion  varchar(150),
```

```
estado    int4,  
CONSTRAINT "PK7" PRIMARY KEY (codpro)  
  
)  
;  
  
--  
-- TABLE: "Proveedor"  
--  
  
CREATE TABLE "Proveedor"(  
  
    codproveedor int4    NOT NULL,  
    nit          varchar(20),  
    direccion   varchar(150),  
    empresa     varchar(100),  
    descripcion  varchar(150),  
    email       varchar(50),  
    fono        varchar(15),  
    movil       varchar(15),  
    fax         varchar(15),  
    estado      int4    NOT NULL,  
    CONSTRAINT "PK11" PRIMARY KEY (codproveedor)  
  
)  
;  
  
--  
-- TABLE: "Provincia"  
--
```

```
CREATE TABLE "Provincia"(  
    codprov int4 NOT NULL,  
    coddep int4,  
    nombre varchar(50),  
    CONSTRAINT "PK2" PRIMARY KEY (codprov)  
  
)  
;
```

```
--  
-- TABLE: "Roles"  
--
```

```
CREATE TABLE "Roles"(  
  
    codrol integer NOT NULL,  
    descripcion varchar(100),  
    nombre varchar(60) NOT NULL,  
    CONSTRAINT "PK4_1" PRIMARY KEY (codrol)  
  
)  
;
```

```
--  
-- TABLE: "RolMod"  
--
```

```
CREATE TABLE "RolMod"(  
  
    codrol int4 NOT NULL,
```

```
codmod int4 NOT NULL,  
CONSTRAINT "PK40" PRIMARY KEY (codrol, codmod)  
  
)  
;  
  
--  
-- TABLE: "SalidaProducto"  
--
```

```
CREATE TABLE "SalidaProducto"(  
  
    codsalida int4 NOT NULL,  
    codpertenece int4,  
    codusuario int4,  
    numeroform varchar(20),  
    fecha_sal date,  
    CONSTRAINT "PK12" PRIMARY KEY (codsalida)  
  
)  
;  
  
--  
-- TABLE: "Telefono"  
--
```

```
CREATE TABLE "Telefono"(  
    codtel int4 NOT NULL,  
    codpersonal int4,
```

```
fono      varchar(15),
movil     varchar(15),
CONSTRAINT "PK7_1" PRIMARY KEY (codtel)

)
;

--
-- TABLE: "TipoAlmacen"
--

CREATE TABLE "TipoAlmacen"(

codtipoa  int4      NOT NULL,
codclaalm int4,
nombre    varchar(50),
descripcion varchar(10),
CONSTRAINT "PK35" PRIMARY KEY (codtipoa)

)
;

--
-- TABLE: "Unidad"
--

CREATE TABLE "Unidad"(

codunidad int4      NOT NULL,
nombre    varchar(50),
```

```
descripcion varchar(150),
estado int4,
CONSTRAINT "PK19" PRIMARY KEY (codunidad)

)
;

--
-- TABLE: "Usuario"
--
CREATE TABLE "Usuario"(

codusuario int4 NOT NULL,
login varchar(50) NOT NULL,
password varchar(50) NOT NULL,
estado int4 NOT NULL,
codpersonal int4,
CONSTRAINT "PK6" PRIMARY KEY (codusuario)

)
;

--
-- TABLE: "UsuRol"
--

CREATE TABLE "UsuRol"(
codusurol int4 NOT NULL,
codusuario int4,
codrol int4,
```

```
fecha    date,
estado   int4  DEFAULT 0 NOT NULL,
CONSTRAINT "PK5_1" PRIMARY KEY (codusurol)

)
;

--
-- TABLE: "Agencia"
--

ALTER TABLE "Agencia" ADD CONSTRAINT "RefLocalidad31"

    FOREIGN KEY (codloc)
    REFERENCES "Localidad"(codloc)
;

--
-- TABLE: "Backup"
--

ALTER TABLE "Backup" ADD CONSTRAINT "RefUsuario411"

    FOREIGN KEY (codusuario)
    REFERENCES "Usuario"(codusuario)
;

--
-- TABLE: "Categoria"
--
```

```
ALTER TABLE "Categoria" ADD CONSTRAINT "RefTipoAlmacen371"
```

```
    FOREIGN KEY (codtipoa)
    REFERENCES "TipoAlmacen"(codtipoa)
;
```

```
--
-- TABLE: "DetalleIngreso"
--
```

```
ALTER TABLE "DetalleIngreso" ADD CONSTRAINT "RefIngresoProducto101"
```

```
    FOREIGN KEY (codingreso)
    REFERENCES "IngresoProducto"(codingreso)
;
```

```
ALTER TABLE "DetalleIngreso" ADD CONSTRAINT "RefKardex121"
```

```
    FOREIGN KEY (codkardex)
    REFERENCES "Kardex"(codkardex)
;
```

```
--
-- TABLE: "DetalleSalida"
--
```

```
ALTER TABLE "DetalleSalida" ADD CONSTRAINT "RefKardex131"
```

```
    FOREIGN KEY (codkardex)
    REFERENCES "Kardex"(codkardex)
```

;

```
ALTER TABLE "DetalleSalida" ADD CONSTRAINT "RefSalidaProducto141"
```

```
    FOREIGN KEY (codsalida)
```

```
    REFERENCES "SalidaProducto"(codsalida)
```

;

--

-- TABLE: "Direccion"

--

```
ALTER TABLE "Direccion" ADD CONSTRAINT "RefPersonal281"
```

```
    FOREIGN KEY (codpersonal)
```

```
    REFERENCES "Personal"(codpersonal)
```

;

--

-- TABLE: "EstrucOrganizativa"

--

```
ALTER      TABLE      "EstrucOrganizativa"      ADD      CONSTRAINT  
"RefEstrucOrganizativa231"
```

```
    FOREIGN KEY (idpadre)
```

```
    REFERENCES "EstrucOrganizativa"(idhijo)
```

;

```
ALTER TABLE "EstrucOrganizativa" ADD CONSTRAINT "RefAgencia321"
```

```
    FOREIGN KEY (codagen)  
    REFERENCES "Agencia"(codagen)
```

```
;
```

```
--
```

```
-- TABLE: "IngresoProducto"
```

```
--
```

```
ALTER TABLE "IngresoProducto" ADD CONSTRAINT "RefProveedor91"
```

```
    FOREIGN KEY (codproveedor)  
    REFERENCES "Proveedor"(codproveedor)
```

```
;
```

```
ALTER TABLE "IngresoProducto" ADD CONSTRAINT "RefUsuario451"
```

```
    FOREIGN KEY (codusuario)  
    REFERENCES "Usuario"(codusuario)
```

```
;
```

```
ALTER TABLE "IngresoProducto" ADD CONSTRAINT "RefPertenece651"
```

```
    FOREIGN KEY (codpertenece)  
    REFERENCES "Pertenece"(codpertenece)
```

```
;
```

```
--
```

```
-- TABLE: "Kardex"
```

```
--  
  
ALTER TABLE "Kardex" ADD CONSTRAINT "RefProducto571"  
  
    FOREIGN KEY (codpro)  
    REFERENCES "Producto"(codpro)  
    ;  
  
ALTER TABLE "Kardex" ADD CONSTRAINT "RefAgencia631"  
  
    FOREIGN KEY (codagen)  
    REFERENCES "Agencia"(codagen)  
    ;  
  
--  
-- TABLE: "Localidad"  
--  
  
ALTER TABLE "Localidad" ADD CONSTRAINT "RefProvincia21"  
  
    FOREIGN KEY (codprov)  
    REFERENCES "Provincia"(codprov)  
    ;  
  
--  
-- TABLE: "Pertenece"  
--  
  
ALTER TABLE "Pertenece" ADD CONSTRAINT "RefPersonal261"
```

```
FOREIGN KEY (codpersonal)
REFERENCES "Personal"(codpersonal)
;

ALTER TABLE "Pertenece" ADD CONSTRAINT "RefEstrucOrganizativa271"

FOREIGN KEY (idhijo)
REFERENCES "EstrucOrganizativa"(idhijo)
;

--
-- TABLE: "Producto"
--

ALTER TABLE "Producto" ADD CONSTRAINT "RefCategoria51"

FOREIGN KEY (codcat)
REFERENCES "Categoria"(codcat)
;

ALTER TABLE "Producto" ADD CONSTRAINT "RefUnidad151"

FOREIGN KEY (codunidad)
REFERENCES "Unidad"(codunidad)
;

ALTER TABLE "Producto" ADD CONSTRAINT "RefMarca551"

FOREIGN KEY (codmarca)
```

```
REFERENCES "Marca"(codmarca)
;

--
-- TABLE: "Provincia"
--

ALTER TABLE "Provincia" ADD CONSTRAINT "RefDepartamento11"

FOREIGN KEY (coddep)
REFERENCES "Departamento"(coddep)
;

--
-- TABLE: "RolMod"
--

ALTER TABLE "RolMod" ADD CONSTRAINT "RefRoles421"

FOREIGN KEY (codrol)
REFERENCES "Roles"(codrol)
;

ALTER TABLE "RolMod" ADD CONSTRAINT "RefModulo431"
FOREIGN KEY (codmod)
REFERENCES "Modulo"(codmod)
;

--
```

```
-- TABLE: "SalidaProducto"
```

```
--
```

```
ALTER TABLE "SalidaProducto" ADD CONSTRAINT "RefUsuario441"
```

```
    FOREIGN KEY (codusuario)
```

```
    REFERENCES "Usuario"(codusuario)
```

```
;
```

```
ALTER TABLE "SalidaProducto" ADD CONSTRAINT "RefPertenece591"
```

```
    FOREIGN KEY (codpertenece)
```

```
    REFERENCES "Pertenece"(codpertenece)
```

```
;
```

```
--
```

```
-- TABLE: "Telefono"
```

```
--
```

```
ALTER TABLE "Telefono" ADD CONSTRAINT "RefPersonal311"
```

```
    FOREIGN KEY (codpersonal)
```

```
    REFERENCES "Personal"(codpersonal)
```

```
;
```

```
--
```

```
-- TABLE: "TipoAlmacen"
```

```
--
```

```
ALTER TABLE "TipoAlmacen" ADD CONSTRAINT "RefClaseAlmacen361"
```

```
FOREIGN KEY (codclaalm)
REFERENCES "ClaseAlmacen"(codclaalm)
;

--
-- TABLE: "Usuario"
--

ALTER TABLE "Usuario" ADD CONSTRAINT "RefPersonal251"

FOREIGN KEY (codpersonal)
REFERENCES "Personal"(codpersonal)
;

--
-- TABLE: "UsuRol"
--

ALTER TABLE "UsuRol" ADD CONSTRAINT "RefRoles221"

FOREIGN KEY (codrol)
REFERENCES "Roles"(codrol)
;

ALTER TABLE "UsuRol" ADD CONSTRAINT "RefUsuario241"

FOREIGN KEY (codusuario)
REFERENCES "Usuario"(codusuario)
;
```

1.1.2.2.3.10.2.4. Diccionario de Datos

Agencia

Nombre Agencia

Entidad

Tipo Entidad Dependiente

Primary Keys codagencia

Definición Agencias que tiene la institución

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codagencia	integer	✓	✓	Representa la Clave Primaria
	✓	codlocalidad	integer			Representa la clave primaria de la tabla Localidad que será generada por la base de datos
		nombre	varchar(100)			Representa el nombre de la Agencia
		direccion	varchar(150)			Representa la Dirección de la Agencia

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
--------	-------	----------	----------

<u>agencia_codlocalidad_fk</u> <u>ey</u>	codlocalidad	<u>localidad</u>	codlocalidad
---------------------------------------------	--------------	------------------	--------------

Dirección

Nombre Dirección

Entidad

Tipo Entidad Dependiente

Primary Keys Coddireccion

Definición Dirección que tiene el Personal

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		coddireccion	integer	✓	✓	Representa la Clave Primaria
		zona	varchar(100)			Representa la zona donde vive el Personal
		calle	varchar(100)			Representa la calle donde vive el Personal
		numero	varchar(15)			Representa el numero de Vivienda del Personal
	✓	codpersonal	integer			Representa la clave primaria de la tabla personal que será generada por la base de

						datos
--	--	--	--	--	--	-------

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
<u>direccion_codpersonal_fkey</u>	codpersonal	<u>public.personal</u>	codpersonal

Estrucorganizativa

Nombre Estrucorganizativa

Entidad

Tipo Entidad Dependiente

Primary Keys idhijo

Definición Estructura organizativa que tiene una Agencia

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		idhijo	integer	✓	✓	Representa la Clave Primaria
	✓	idpadre	integer	✓		Representa la clave primaria de la tabla estrucorganizativa que será generada por la base de datos
		nombre	varchar(150)	✓		Representa el Nombre de la Estructura

		fecha	date			Representa la Fecha de creación de la Estructura
		descripcion	varchar(200)			Representa la descripción para la estructura
		estado	integer	✓		Representa el estado de la Estructura 0 Inactivo 1 Activo
		tipo	integer	✓		Representa el tipo de la Estructura 1 Área, 2 Cargo
	✓	codagencia	integer			Representa la clave primaria de la tabla agencia que será generada por la base de datos

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
<u>estrucorganizativa_codagencia_fkey</u>	codagencia	<u>public.agencia</u>	codagencia
<u>estrucorganizativa_idpadre_fkey</u>	idpadre	<u>public.estrucorganizativa</u>	idhijo

Módulo

Nombre Módulo
Entidad

Tipo Entidad Independiente

Primary Keys Codmod

Definición Módulos del Sistema

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codmod	integer	✓	✓	Representa la Clave Primaria
		nombre	varchar(100)			Representa el Nombre de los Módulos del Sistema
		link	varchar(150)			Representa el Link del Módulo
		descripcion	varchar(200)			Representa la Descripción del Módulo

Localidad

Nombre Localidad

Entidad

Tipo Entidad Dependiente

Primary Keys Codlocalidad

Definición Localidad que tienen las provincias

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codlocalidad	integer	✓	✓	Representa la Clave Primaria
	✓	codprov	integer			Representa la clave primaria de la tabla provincia que será generada por la base de datos
		nombre	varchar(100)			Representa el Nombre de la Localidad

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
<u>localidad_codprov_fkey</u>	codprov	<u>public.provincia</u>	codprov

Personal

Nombre Personal

Entidad

Tipo Entidad Independiente

Primary Keys Codpersonal

Definición Personal que tiene lainstitución

Notas

PK	FK	Nombre	Tipo De	Not	Unique	Descripción
----	----	--------	---------	-----	--------	-------------

			Datos	null		
✓		codpersonal	integer	✓	✓	Representa la clave primaria
		ci	varchar(15)	✓		Representa el CI del Personal
		nombre	varchar(60)	✓		Representa el nombre de la Personal
		ap	varchar(60)			Representa el Apellido Paterno de la Personal
		am	varchar(60)	✓		Representa el Apellido Materno de la Personal
		fnac	date			Representa Fecha de Nacimiento del Personal
		foto	varchar(100)			Representa el atributo para cargar la Foto de la Personal
		estado	integer	✓		Representa el Estado de la Personal 0 Inactivo 1 Activo

Proveedor

Nombre Proveedor

Entidad

Tipo Entidad Independiente

Primary Keys Codproveedor

Definición Proveedores de los Productos

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codproveedor	integer	✓	✓	Representa la Clave Primaria
		nit	varchar(20)	✓		Representa el Nit del Proveedor
		razonsocial	varchar(150)	✓		Representa la Razón Social del Proveedor
		email	varchar(100)			Representa el Email del Proveedor
		fax	varchar(20)			Representa el Nro del Fax del Proveedor
		fono	varchar(15)			Representa el Telefono Fijo del Proveedor
		movil	varchar(15)			Representa el Telefono Movil del Proveedor
		estado	integer	✓		Representa el Estado del Proveedor 0 Inactivo 1 Activo
		direccion	varchar(150)			Representa la Dirección

						del Proveedor
		descripcion	varchar(200)			Representa la Descripción del Proveedor

Provincia

Nombre Provincia

Entidad

Tipo Entidad Dependiente

Primary Keys Codprov

Definición Provincias que tienen los Departamentos

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codprov	integer	✓	✓	Representa la Clave Primaria
	✓	coddep	integer			Representa la clave primaria de la tabla departamento que será generada por la base de datos
		nombre	varchar(100)			Representa el Nombre de la Provincia

Foreign Keys

Name	Fields	FK Table	FK Fields
<u>provincia_coddep_fkey</u>	coddep	<u>public.departamento</u>	coddep

Roles

Nombre Roles

Entidad

Tipo Entidad Independiente

Primary Keys Codrol

Definición Roles establecidos para los Usuarios del Sistema

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codrol	integer	✓	✓	Representa la Clave Primaria
		descripcion	varchar(200)			Representa la Descripción del Rol
		nombre	varchar(60)	✓		Representa el Nombre del Rol

RolMod

Nombre RolMod

Entidad

Tipo Entidad Dependiente

Definición Módulos asignadas para los Roles

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓	✓	codmod	integer	✓		Representa la clave primaria de la tabla Módulo que será generada por la base de datos
✓	✓	codrol	integer	✓		Representa la clave primaria de la tabla roles que será generada por la base de datos

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
<u>rolfun_codfun_fkey</u>	codfun	<u>public.funcionalidad</u>	codfun
<u>rolfun_codrol_fkey</u>	codrol	<u>public.roles</u>	codrol

Telefono

Nombre Telefono

Entidad

Tipo Entidad Dependiente

Primary Keys Codtel

Definición Teléfonos del Personal

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codtel	integer	✓	✓	Representa la Clave Primaria
		fono	varchar(15)			Representa el Telefono Fijo del Personal
		movil	varchar(15)			Representa el Telefono Movil del Personal
	✓	codpersonal	integer			Representa la clave primaria de la tabla personal que será generada por la base de datos

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
<u>telefono_codpersonal_fkey</u>	codpersonal	<u>public.personal</u>	codpersonal

Usuario

Nombre Usuario

Entidad

Tipo Entidad Dependiente

Primary Keys Codusuario

Definición Usuario para acceder al Sistema

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codusuario	integer	✓	✓	Representa la clave primaria
		login	varchar(50)	✓		Representa el login para el ingreso al sistema del Usuario
		password	varchar(50)	✓		Representa el password para el ingreso al sistema del Usuario
		estado	integer	✓		Representa el Estado del Usuario 1 Activo 0 Inactivo
	✓	codpersonal	integer			Representa la clave primaria de la tabla personal que será generada por la base de datos

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
--------	-------	----------	----------

<u>usuario_codpersonal_fkey</u>	codpersonal	<u>public.personal</u>	codpersonal
---------------------------------	-------------	------------------------	-------------

Usurol

Nombre Usurol

Entidad

Tipo Entidad Dependiente

Primary Keys Codusurol

Definición Rol asignado al usuario

Notas

PK	FK	Nombre	Tipo De Datos	Not null	Unique	Descripción
✓		codusurol	integer	✓	✓	Representa la clave primaria
	✓	codusuario	integer			Representa la clave primaria de la tabla usuario que será generada por la base de datos
	✓	codrol	integer			Representa la clave primaria de la tabla rol que será generada por la base de datos
		fecha	date			Representa la Fecha del de Asignación al

						Usuario el Rol
		estado	integer	*		Representa el Estado 0 Inactivo 1 Activo

Foreign Keys

Nombre	Campo	FK Tabla	FK Campo
<u>usurol_codrol_fkey</u>	codrol	<u>public.roles</u>	codrol
<u>usurol_codusuario_fkey</u>	codusuario	<u>public.usuario</u>	codusuario

1.1.2.2.3.11. Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

1.1.2.2.3.11.1. Plan de Pruebas

1.1.2.2.3.11.1.1. Descripción de Aspectos Generales

Esta sección establece el alcance y el objetivo del Plan de Pruebas. Es aquí donde se describen los aspectos fundamentales del esfuerzo que se hará para probar cada una de las gestiones que conforman el Sistema descrito en este Plan de Desarrollo de Software, independiente las características y tamaño que ésta pueda tener.

1.1.2.2.3.11.1.1.1. Objetivo

Este Plan de Pruebas fue desarrollado con el fin de guiar el proceso de pruebas al proyecto CONAL Sistema Automatizado de Control de Almacenes desarrollado por nuestro grupo de trabajo, esto con el fin de asegurar una excelente calidad del software

desarrollado, encontrando para ello errores que puedan perjudicar en el funcionamiento de este, los cuáles una vez corregidos, se podrán elaborar un documento que presente las evaluaciones correspondientes en el plazo más corto posible.

1.1.2.2.3.11.1.1.2. Entorno o Marco

La necesidad de aplicar este plan de pruebas radica en la necesidad de obtener un software de calidad, que cumpla con los requerimientos del Comercial el Carmen, además que sea aprobado por la comunidad de usuarios finales del Sistema.

1.1.2.2.3.11.1.1.3. Arquitectura Técnica

Para la realización de las pruebas se tomará en cuenta los módulos desarrollados en la fase de construcción y los diagramas realizados en la fase de elaboración.

1.1.2.2.3.11.1.1.4. Especificaciones del Software y Hardware

Corresponde a una lista individualizada de todo el hardware y el software que utiliza la aplicación, incluyendo proveedores y versiones.

HARDWARE
<p>Computadora con la suficiente capacidad de soportar el software</p> <p>Los requisitos de hardware del Servidor Web son los siguientes:</p> <ul style="list-style-type: none"> • Procesador: Core i3 de 2º generación de 2.93 Ghz. • Memoria RAM: DDR3 4GB. • Lector de CD: Lecto / Grabador de DVD. • Monitor: Color de 15". • Espacio libre en la unidad de disco rígido: 200 Mb aproximadamente. • Conexión a Internet. O en su Defecto Conexión Intranet.

Tabla 86. Hardware

SOFTWARE

Aplicación	Versión	Descripción
Sistema de Control de Almacenes	0.3	Sistema desarrollado por el grupo de trabajo.
Rational Rose Enterprise Edition	7.6	Herramienta de Diseño y Desarrollo de Diagramas UML para RUP
Enterprise Architect Version 8.0	8	Herramienta de Diseño y Desarrollo de Diagramas UML para RUP
PostgreSQL	9.0	Motor de Base de Datos Relacional.

Tabla 87. Software

OTROS		
Descripción	Versión	Descripción
Documentación	0.3	Documentación de Desarrollo de Software

Tabla 88. Otros

1.1.2.2.3.11.1.1.5. Alcance

Este plan describe las pruebas del Sistema, que serán aplicadas a los componentes del Sistema de Control y Administración de Almacenes.

Se asume que el código final que ha sido revisado como exitoso será probado

Las siguientes medidas de desempeño serán probadas:

Validación correcta de las Entradas de Dato.

1.1.2.2.3.11.1.1.6. Referencias

Glosario del Proyecto.

Plan de Desarrollo de Software.

1.1.2.2.3.11.2. Descripción de las Pruebas Planeadas

1.1.2.2.3.11.2.1. Descripción de las Pruebas Incluidas

Las siguientes pruebas serán realizadas a todo el Sistema Informático:

Pruebas Funcionales.

Pruebas de Validación de Datos.

1.1.2.2.3.11.3. Planeación de la Ejecución de las Pruebas

La siguiente es una descripción de los tipos y técnicas de pruebas que se harán sobre el Sistema Informático, junto con la manera en la que éstas se ejecutarán.

1.1.2.2.3.11.4. Tipos y Técnicas de Pruebas

1.1.2.2.3.11.4.1. Pruebas de Validación de Datos

Las pruebas de validación de datos verifican la interacción del usuario con el software y la información que introduce. El objetivo de las pruebas de la interfaz de usuario es asegurar que dicha interfaz proporciona al usuario el acceso y validación a todos los datos introducidos por el Usuario con motivo de dar Integridad la información.

Objetivo de la Técnica:	Verificar y probar la Interfaz de entrada Usuario- Sistema Verificar todas las posibles combinaciones de caracteres para entradas del Sistema.
Técnica:	Crear o modificar pruebas para cada ventana para verificar todas las entradas de datos.
Hitos:	El testear verificara todas las entradas del Sistema.

Herramientas Requeridas:	-
Criterio de Éxito:	Todas las entradas han sido verificadas mediante combinaciones de caracteres para dar con los posibles errores del Sistema.

Tabla 89. Pruebas de Validación de Datos

1.1.2.2.3.11.5. Ejecución de las Pruebas

1.1.2.2.3.11.5.1. Demostración y Resultados de la Prueba de Validación de Datos

1.1.2.2.3.11.5.1.1. Partición Equivalente: Casos de Prueba de Caja Negra

Interfaz: Ingresar al Sistema

- Usuario: Alfanumérico 50 caracteres
- Clave: Alfanumérico 50 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas

Tipo: Usuario	1_ Alfanumérico	3_ Vacío
Tamaño: Usuario	2_ 9<=Caracteres<=50	4_ Caracteres >50
Tipo: Clave	4_ Alfanumérico	6_ Vacío
Tamaño: Clave	5_ 9<=Caracteres<=50	7_ Caracteres >50

Tabla 90. Partición Equivalente. Interfaz: Ingresar al Sistema

Casos de Prueba válidas

CP1	
Usuario	Clave
ADMINISTRADOR	ADMINISTRADOR

Cubre las clases de equivalencias válidas: 1-2-4-5

Casos de Pruebas Inválidas

CP2	
Usuario	Clave
	ADMINISTRADOR

Cubre las clases de equivalencias inválidas: 3

CP3	
Usuario	Clave
ADMINISTRADOR	

Cubre las clases de equivalencias inválidas: 6

Interfaz: Formulario (ABM) Personal

- Cédula Identidad: Alfanumérico 15 caracteres
- Nombre: Alfabético 60 caracteres
- Apellido Paterno: Alfabético 60 caracteres
- Apellido Materno: Alfabético 60 caracteres
- Fecha Nacimiento: Date
- Zona: Alfanumérico 100 caracteres
- Calle: Alfanumérico 100 caracteres
- Nro Casa: Alfanumérico 15 caracteres
- Celular: Alfanumérico 15 caracteres
- Fijo: Alfanumérico 15 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Cédula Identidad	1_ Alfanumérico	3_ Vacío
Tamaño: Cédula Identidad	2_ $1 \leq \text{Caracteres} \leq 15$	4_ Caracteres > 15

Tipo: Nombre	5_ Alfabético	7_ Vacío
Tamaño: Nombre	6_ 1<=Caracteres<=60	8_ Caracteres >60
Tipo: Apellido Paterno	9_ Alfabético	11_ Caracteres >60
Tamaño: Apellido Paterno	10_ 0<=Caracteres<=60	
Tipo: Apellido Materno	12_ Alfabético	14_ Vacío
Tamaño: Apellido Materno	13_ 1<=Caracteres<=60	15_ Caracteres >60
Tipo: Fecha Nacimiento	16_Date	17_Vacío
Tamaño: Fecha Nacimiento		
Tipo: Zona	18_ Alfanumérico	20_Caracteres >100
Tamaño: Zona	19_ 0<=Caracteres<=100	
Tipo: Calle	21_ Alfanumérico	23_Caracteres >100
Tamaño: Calle	22_ 0<=Caracteres<=100	
Tipo: Nro Casa	24_ Alfanumérico	26_ Caracteres >15
Tamaño: Nro Casa	25_ 1<=Caracteres<=15	
Tipo: Celular	27_ Alfanumérico	29_ Caracteres >15
Tamaño: Celular	28_ 1<=Caracteres<=15	
Tipo: Fijo	30_ Alfanumérico	32_ Caracteres >15
Tamaño: Fijo	31_ 1<=Caracteres<=15	

Tabla 91. Partición Equivalente. Interfaz: Formulario (ABM) Personal

Cédula Identid	Nomb re	Apelli do	Apelli do	Fecha Nacimie	Zon a	Calle	Nro Cas	Celula r	Fijo
-------------------	------------	--------------	--------------	------------------	----------	-------	------------	-------------	------

ad		Patern o	Mater no	nto			a		
788	Estela	Rueda	García	06/09/84	Tej ar	Celedon io	646 6	71658 45	66548 77

Cubre las clases de equivalencias válidas: 1-2-5-6-9-10-12-13-16-18-19-21-22-24-2527-28-30-31

Casos de Pruebas Inválidas

Cédula Identid ad	Nomb re	Apelli do Patern o	Apelli do Mater no	Fecha Nacimie nto	Zon a	Calle	Nro Cas a	Celula r	Fijo
	Estela	Rueda	García	06/09/84	Tej ar	Celedon io	646 6	71658 45	66548 77

Cubre las clases de equivalencias inválidas: 3

Cédula Identid ad	Nomb re	Apelli do Patern o	Apelli do Mater no	Fecha Nacimie nto	Zon a	Calle	Nro Cas a	Celula r	Fijo
789987		Rueda	García	06/09/84	Tej ar	Celedon io	646 6	71658 45	66548 77

Cubre las clases de equivalencias inválidas: 7

Cédula Identidad	Nombre	Apellido Paterno	Apellido Materno	Fecha Nacimiento	Zona	Calle	Nro Casaca	Celular	Fijo
789987	Estela	Rueda		06/09/84	Tejara	Celedonio	6466	7165845	6654877

Cubre las clases de equivalencias inválidas: 14

Interfaz: Formulario (ABM) Usuario

- Usuario: Alfanumérico 50 caracteres
- Clave: Alfanumérico 50 caracteres
- Clave Verificar: Alfanumérico 50 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Usuario	1_ Alfanumérico	3_ Vacío
Tamaño: Usuario	2_ $9 \leq \text{Caracteres} \leq 50$	4_ Caracteres > 50

Tipo: Clave	4_ Alfanumérico	6_ Vacío
Tamaño: Clave	5_ 9<=Caracteres<=50	7_ Caracteres >50
Tipo: Clave Verificar	8_ Alfanumérico	10_ Vacío
Tamaño: Clave Verificar	9_ 9<=Caracteres<=50	11_ Caracteres >50

Tabla 92. Partición Equivalente. Interfaz: Formulario (ABM) Usuario

Casos de Prueba válidas

CP1		
Usuario	Clave	Clave Verificar
ADMINTRADOR	RUEDA1234	RUEDA1234

Cubre las clases de equivalencias válidas: 1-2-4-5

Casos de Pruebas Inválidas

CP2		
Usuario	Clave	Clave Verificar
	RUEDA1234	RUEDA1234

Cubre las clases de equivalencias inválidas: 3

CP3		
Usuario	Clave	Clave Verificar
ADMINTRADOR		RUEDA1234

Cubre las clases de equivalencias inválidas: 6

CP4		
Usuario	Clave	Clave Verificar
ADMINTRADOR	RUEDA1234	

Cubre las clases de equivalencias inválidas: 10

Interfaz: Formulario (ABM) Roles

ADICIONAR ROL

DATOS DEL ROL

Nombre: Descripción:

DATOS DE SELECCION ACCESO A LOS MODULOS DEL ROL

ID	NOMBRE DEL MODULO	DESCRIPCIÓN
1 24	\$modulo.nombre	\$modulo.descripcion
2 23	fdsf	\$modulo.descripcion
3 22	sdfds	\$modulo.descripcion
4 21	sdf	\$modulo.descripcion
5 20	fsdfds	\$modulo.descripcion
6 19	dsf	\$modulo.descripcion
7 18	dsf	\$modulo.descripcion
8 17	dsfds	\$modulo.descripcion

Página 1 de 1

Mostrando 1 - 24 de 24

Guardar Cancelar

- Nombre: Alfanumérico 60 caracteres
- Descripción: Alfanumérico 100 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Nombre	1_ Alfanumérico	3_ Vacío

Tamaño: Nombre	2_ 1<=Caracteres<=60	4_ Caracteres >60
Tipo: Descripción	5_ Alfanumérico	7_ Caracteres >100
Tamaño: Descripción	6_ 0<=Caracteres<=100	

Tabla 93. Partición Equivalente. Interfaz: Formulario (ABM) Roles

Casos de Prueba válidas

CP1	
Nombre	Descripción
Usuario Principal	Este es el Usuario Principal del Sistema

Cubre las clases de equivalencias válidas: 1-2-5-6

Casos de Pruebas Inválidas

CP2	
Nombre	Descripción
	Este es el Usuario Principal del Sistema

Cubre las clases de equivalencias inválidas: 3

Interfaz: Formulario Asignar Cargo

George	Nombre	Apellido	Apellido	Seleccionar
1	Alberto	Miranda	Torrez	<input type="radio"/>
6	noemi	portal	rivera	<input checked="" type="radio"/>
7	ertyul	fgjkl	bnm	<input type="radio"/>

Fecha:

- Fecha: Date

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Fecha Tamaño: Fecha	1_ Date	2_ Cualquier otra cosa

Tabla 94. Partición Equivalente. Interfaz: Formulario Asignar Cargo

Casos de Prueba válidas

CP1
Fecha
26/11/2011

Cubre las clases de equivalencias válidas: 1

Casos de Pruebas Inválidas

CP2

Fecha
Hi!!!!!!!!!!!!!!!!!!!!!!

Cubre las clases de equivalencias inválidas: 2

Interfaz: Formulario Reasignar Cargo

- Fecha: Date

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Fecha	1_ Date	2_ Cualquier otra cosa
Tamaño: Fecha		

Tabla 95. Partición Equivalente. Interfaz: Formulario Reasignar Cargo

Casos de Prueba válidas

CP1
Fecha
26/11/2011

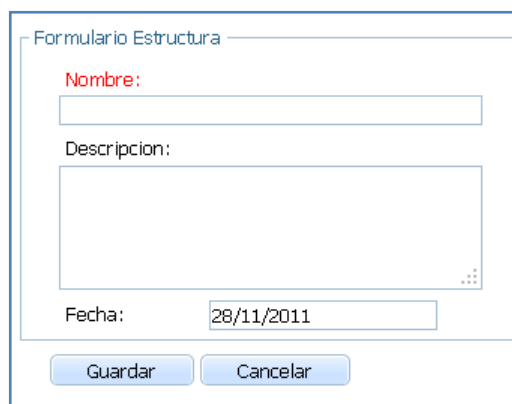
Cubre las clases de equivalencias válidas: 1

Casos de Pruebas Inválidas

CP2
Fecha
26 de agosto

Cubre las clases de equivalencias inválidas: 2

Interfaz: Formulario (ABM) Área



The image shows a software window titled "Formulario Estructura". It contains three input fields: "Nombre:" (a single-line text box), "Descripción:" (a multi-line text area), and "Fecha:" (a date picker box containing "28/11/2011"). At the bottom of the window are two buttons: "Guardar" and "Cancelar".

- Nombre: Alfanumérico 150 caracteres
- Descripción: Alfanumérico 200 caracteres
- Fecha: Date

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Nombre Tamaño: Nombre	1_ Alfanumérico 2_ 1<=Caracteres<=150	3_ Vacío 4_ Caracteres >150
Tipo: Descripción Tamaño: Descripción	5_ Alfanumérico 6_ 0<=Caracteres<=200	7_ Caracteres >200
Tipo: Fecha Tamaño: Fecha	8_Date	9_Cualquier otra cosa

Tabla 96. Partición Equivalente. Interfaz: Formulario (ABM) Área

Casos de Prueba válidas

CP1		
Nombre	Descripción	Fecha
Auditoria	Esta es el área de Auditoria	25/11/2011

Cubre las clases de equivalencias válidas: 1-2-5-6-8

Casos de Pruebas Inválidas

CP2		
Nombre	Descripción	Fecha
	Esta es el área de Auditoria	Hola

Cubre las clases de equivalencias inválidas: 3-9

Interfaz: Formulario (ABM) Cargo

- Nombre: Alfanumérico 150 caracteres
- Descripción: Alfanumérico 200 caracteres
- Fecha: Date

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Nombre	1_ Alfanumérico	3_ Vacío
Tamaño: Nombre	2_ 1<=Caracteres<=150	4_ Caracteres >150
Tipo: Descripción	5_ Alfanumérico	7_ Caracteres >200

Tamaño: Descripción	6_ 0<=Caracteres<=200	
Tipo: Fecha	8_Date	9_Cualquier otra cosa
Tamaño: Fecha		

Tabla 97. Partición Equivalente. Interfaz: Formulario (ABM) Cargo

Casos de Prueba válidas

CP1		
Nombre	Descripción	Fecha
Contador	Cargo del Contador General	30/11/2011

Cubre las clases de equivalencias válidas: 1-2-5-6-8

Casos de Pruebas Inválidas

CP2		
Nombre	Descripción	Fecha
	Cargo del Contador General	Taller

Cubre las clases de equivalencias inválidas: 3-9

Interfaz: Formulario (ABM) Proveedor

CODIGO	NIT	RAZON SOCIAL	DIRECCION	EMAIL	TELEFONOS
2	78874555	Empresa Nort Stard	C/ Moto mendez	nortestlar@hotmail.com	Telf. - Cel: 784545

- Nit: Alfanumérico 20 caracteres
- Razón Social: Alfanumérico 100 caracteres
- Email: Alfanumérico 50 caracteres
- Telf. Fijo: Alfanumérico 15 caracteres
- Telf. Celular: Alfanumérico 15 caracteres
- Fax: Alfanumérico 15 caracteres
- Dirección: Alfanumérico 150 caracteres
- Descripción: Alfanumérico 150 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Nit	1_ Alfanumérico	3_ Vacío
Tamaño: Nit	2_ $1 \leq \text{Caracteres} \leq 20$	4_ Caracteres > 20

Tipo: Razón Social	5_ Alfanumérico	7_ Vacío
Tamaño: Razón Social	6_ 1<=Caracteres<=100	8_ Caracteres >100
Tipo: Email	9_ Alfanumérico	11_ Caracteres >50
Tamaño: Email	10_ 0<=Caracteres<=50	
Tipo: Telf. Fijo	12_ Alfanumérico	14_ Caracteres >15
Tamaño: Telf. Fijo	13_ 0<=Caracteres<=15	
Tipo: Telf. Celular	15_ Alfanumérico	17_ Caracteres >15
Tamaño: Telf. Celular	16_ 0<=Caracteres<=15	
Tipo: Fax	18_ Alfanumérico	20_ Caracteres >100
Tamaño: Fax	19_ 0<=Caracteres<=100	
Tipo: Dirección	21_ Alfanumérico	23_ Caracteres >150
Tamaño: Dirección	22_ 0<=Caracteres<=150	
Tipo: Descripción	24_ Alfanumérico	26_ Caracteres >150
Tamaño: Descripción	25_ 0<=Caracteres<=150	

Tabla 98. Partición Equivalente. Interfaz: Formulario (ABM) Proveedor

CP1							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción
12654782	Matic Soluciones	matic@gmail.com	66-487421	75484588	1277741	Entre Daniel 154	Accesorios para PC

Cubre las clases de equivalencias válidas: 1-2-5-6-9-10-12-13-15-16-18-19-21-22-24-25

Casos de Pruebas Inválidas

CP2							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción
	Matic Soluciones	matic@gmail.com	66-487421	75484588	1277741	Entre Daniel 154	Accesorios para PC

Cubre las clases de equivalencias inválidas: 3

C3							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción
12654782		matic@gmail.com	66-487421	75484588	1277741	Entre Daniel 154	Accesorios para PC

Cubre las clases de equivalencias inválidas: 7

C4							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción

1265478 2	Matic Solucion es	matic@g mail.com	66- 487421	7548458 8	1277741	Entre Daniel 154	Accesori os para PC
--------------	-------------------------	------------------------------	---------------	--------------	---------	------------------------	---------------------------

Cubre las clases de equivalencias inválidas: 11

C5							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción
1265478 2	Matic Solucion es	matic@g mail.com	66- 487421	7548458 8	1277741	Entre Daniel 154	Accesori os para PC

Cubre las clases de equivalencias inválidas: 14

C6							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción
1265478 2	Matic Solucion es	matic@g mail.com	66- 487421	7548458 8	1277741	Entre Daniel 154	Accesori os para PC

Cubre las clases de equivalencias inválidas: 17

C7							
Nit	Razón Social	Email	Telf. Fijo	Telf. Celular	Fax	Dirección	Descripción
12654782	Matic Soluciones	matic@gmail.com	66-487421	75484588	1277741	Entre Daniel 154	Accesorios para PC

Cubre las clases de equivalencias inválidas: 11

Interfaz: Formulario (ABM) Producto

PRODUCTOS

DATOS DEL PRODUCTO

Cod. Producto:

Nombre:

Unidad:

Marca:

Descripcion:

- Código: Alfanumérico 8 caracteres
- Unidad: Unidad
- Marca: Unidad
- Descripción: Alfanumérico 150 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Código	1_ Alfanumérico	3_ Vacío
Tamaño: Código	2_ 1<=Caracteres<=8	4_ Caracteres >8
Tipo: Unidad	5_ Unidad	6_ Vacío
Tamaño: Unidad		
Tipo: Marca	7_ Marca	8_ Vacío
Tamaño: Marca		
Tipo: Descripción	9_ Alfanumérico	11_Caracteres >150
Tamaño: Descripción	10_ 0<=Caracteres<=150	

Tabla 99. Partición Equivalente. Interfaz: Formulario (ABM) Producto

Casos de Prueba válidas

CP1			
Código	Unidad	Marca	Descripción
12345678	Par	Adidas	Es un producto

Cubre las clases de equivalencias válidas: 1-2-5-7-9-10

Casos de Pruebas Inválidas

CP1			
Código	Unidad	Marca	Descripción
			Es un producto

Cubre las clases de equivalencias inválidas: 3-6-8

1.2.Componente 2: Capacitación del Personal encargado de realizar el registro de los Productos sobre el manejo y uso del Sistema, socialización a la Empresa de la importancia del uso de TIC en las entidades financieras ejecutado.

1.2.1. Introducción

De acuerdo al modo de vida actual del mundo, el termino Capacitación y Sistemas de Información están modificando las formas de trabajo de las empresas, los sistemas de información ayudan a acelerar los procesos, por lo tanto las organizaciones que los implantan logran ventajas competitivas al adoptarlos en sus funciones. Pero si el personal no está debidamente capacitado la producción de las empresas será deficiente y en muchos casos incompleto.

En tanto la socialización es un proceso entre una persona y sus semejantes que resulta de aceptar las pautas de comportamiento social y de adaptarse a ellas. La socialización es vista como el proceso mediante el cual se inculca el conocimiento a los miembros de la sociedad, donde los individuos adquieren conocimientos específicos y desarrollan sus potencialidades, habilidades necesarias para la participación adecuada en la vida social. Por lo que la socialización a la Empresa tratara sobre la importancia del uso de TIC ya que las mismas ayudan a las personas a ejecutar su trabajo de una manera más eficiente, ya que es un proceso que lleve a la mejora continua de la producción.

1.2.2. Propósito

El propósito de la capacitación en el uso del nuevo Sistema a implementar es de mejorar el manejo de la información de los Almacenes a través del buen conocimiento del Sistema y de la buena utilización. Así también concientizar el Uso de TIC en la Institución.

1.2.3. Técnicas y Métodos de Enseñanza

1.2.3.1.Principios Didácticos

Los métodos y técnicas de la enseñanza, independiente de las teorías que los originen deben sujetarse a algunos principios comunes, teniendo en cuenta el desarrollo y madurez pedagógica alcanzada hasta el presente. Algunos de ellos son:

- **Principio de proximidad:** Integrar la enseñanza lo más cerca posible en la vida cotidiana del educando.
- **Principio de dirección:** Tornar claros y precisos los objetivos a alcanzar.
- **Principio de marcha propia y continúa:** Procura respetar las diferencias individuales, no exigiendo la misma realización de todos los educandos.
- **Principio de ordenamiento:** Con el establecimiento de un orden se busca facilitar la tarea de aprendizaje.
- **Principio de adecuación:** Es necesario que las tareas y objetivos de la enseñanza sean acordes con la necesidades del educando.
- **Principio de eficiencia:** El ideal: mínimo esfuerzo máxima eficiencia en el aprendizaje.
- **Principio de realidad psicológica:** Previene que no se debe perder de vista la edad evolutiva de los alumnos, así como tampoco sus diferencias individuales.
- **Principio de dificultad o esfuerzo:** Es preciso tener el cuidado de no colocar al educando ante situaciones de las que tenga Posibilidades de salir bien. Pues el fracaso continuado es peor veneno para la criatura humana.
- **Principio de participación:** El educando es parte activa y dinámica del proceso.
- **Principio de espontaneidad:** Cualquier proceso emprendido debe favorecer las manifestaciones naturales del educando.
- **Principio de transparencia:** El conocimiento aprendido debe replicarse en otras situaciones de la vida diaria.
- **Principio de evaluación:** Con un proceso continuo de evaluación, el docente podrá identificar a tiempo dificultades en el proceso de aprendizaje.
- **Principio reflexión:** Inducir al pensamiento reflexivo en el alumno como parte integral de actuar del ser humano.
- **Principio de responsabilidad:** Encaminar todo el proceso de enseñanza de modo que el educando madure en cuanto a comportamiento responsable.

1.2.3.2.Métodos de Enseñanza

1.2.3.2.1. Métodos de Enseñanza individualizada

Su objetivo máximo es ofrecer oportunidades de desenvolvimiento individual más eficiente, y llevar al educando a un completo desarrollo de sus posibilidades personales.

Método de Proyectos

Lleva al alumno a la realización efectiva de algo, es activo y lo lleva para que realice, actúe es en suma determinar una tarea y que el alumno la efectúe. Ofrece pasos para solucionar problemas con la solución para la realización que da experiencia al alumno.

- ✓ Solución de problemas por ejecución.
- ✓ Carácter general o global abarca conjunto de disciplinas.
- ✓ Restringido comprende una o dos disciplinas.
- ✓ Desenvuelve espíritu de iniciativa, responsabilidad, solidaridad y libertad.
- ✓ Cadena organizada de actividades para realizar algo.

Plan Dalton

Este plan se basa en la actividad individual y la libertad. El objetivo consiste en desenvolver la actividad individual e incentiva la iniciativa al dejar al alumno escoger los trabajos y el momento de elaborarlos.

El alumno hace contratos semanales en lo que está interesado, trabajando inmediatamente a su ritmo y posibilidades.

Este método tiene como particularidades que se dan conferencias del profesor con alumnos para organizar trabajos, también un boletín mural donde el profesor consigna instrucciones y la hoja de tareas como documento esencial del plan que contienen explicaciones sobre trabajos, ejercicios e indicaciones de fuentes de consulta.

Algunos inconvenientes que se presentan con este método es que se acentúa exageradamente la individualidad. Tiene un carácter esencialmente individual.

Técnica Winnetka

Busca conjugar las ventajas del trabajo individual con las del trabajo colectivo, sin perder las diferencias individuales.

Enseñanza por unidades o Plan Morrison

Las fases del plan empiezan por la exploración, donde se sondean los conocimientos, luego una presentación donde se expone el tema, sigue la fase de asimilación, en la cual el alumno amplía su conocimiento y continúa con una fase de organización, que es el trabajo de integración para culminar con una recitación, que es la presentación oral de la unidad.

Enseñanza programada

Es el método más reciente para individualizar y permitir que cada alumno trabaje según su propio ritmo y posibilidades.

Enseñanza personalizada

Defiende la tesis que cada educando debe desarrollarse y estudiar a su propio ritmo de aprendizaje.

1.2.3.2.2. Métodos de Enseñanza Socializada

Tiene por objeto la integración social, sin descuidar la individualización.

El estudio en grupo

Se basa en los siguientes principios:

- Crear un ambiente de estudio
- Creación de liderazgo distribuido
- Formulación de objetivos
- Permitir el consenso
- Facilitar la comprensión del proceso

- Asegurar una evaluación permanente

Método de la discusión

Sirve de orientación a la clase para realizar de forma Cooperativa el estudio de una unidad o tema. Se designan un coordinador y un secretario y el resto de grupo de clase.

Método de la asamblea

Toma la misma forma de una discusión ampliada pero con la diferencia como si fuera un cuerpo colegiado gubernamental: Por ejemplo asamblea de estudiantes por la paz.

Método de panel

Es una reunión de especialistas para la discusión general de un tema determinado, el cuál es el área de dominio de los participantes. Hay tres formas básicas, panel simple, simple con alternativa y el panel de interrogadores.

1.2.3.3. Técnicas de Enseñanza

Hay muchas técnicas para hacer llegar nuestro conocimiento y lograr un aprendizaje apropiado:

Técnica expositiva

Consiste en la exposición oral, por parte del profesor; esta debe estimular la participación del alumno en los trabajos de la clase, requiere una buena motivación para atraer la atención de los educandos. Esta técnica favorece el desenvolvimiento del autodomínio, y el lenguaje.

Técnica del dictado

Consiste en que el profesor hable pausadamente en tanto los alumnos van tomando nota de lo que él dice.

Este constituye una marcada pérdida de tiempo, ya que mientras el alumno escribe no puede reflexionar sobre lo que registra en sus notas

Técnica biográfica

Consiste en exponer los hechos o problemas a través del relato de las vidas que participan en ellos o que contribuyen para su estudio. Es más común en la historia, filosofía y la literatura.

Técnica exegética

Consiste en la lectura comentada de textos relacionados con el asunto en estudio, requiere la consulta de obras de autores.

Su finalidad consiste en acostumbrar a leer las obras representativas de un autor, de un tema o una disciplina.

Técnica cronológica

Esta técnica consiste en presentar o desenvolver los hechos en el orden y la secuencia de su aparición en el tiempo.

Esta técnica puede ser progresiva o regresiva-progresiva cuando los hechos son abordados partiendo desde el pasado hasta llegar al presente.

Regresiva cuando esos mismos hechos parten desde el presente en sentido inverso hacia el pasado.

Técnica del interrogatorio

Uno de los mejores instrumentos del campo didáctico como auxiliar en la acción de educar, este permite conocer al alumno y resaltar sus aspectos positivos. Puede ser empleado para:

- Motivación de la clase.
- Estimulo para la reflexión.
- Recapitulación y síntesis de lo aprendido.

Técnica del diálogo

El gran objetivo del diálogo es el de orientar al alumno para que reflexione, piense y se convenza que puede investigar valiéndose del razonamiento.

Técnica de la discusión

Exige el máximo de participación de los alumnos en la disposición de conceptos y en la elaboración misma de la clase.

Consiste en la discusión de un tema, por parte de los alumnos, bajo la dirección del profesor y requiere preparación anticipada.

Técnica del debate

Puede versar sobre:

- Temas que hayan provocado divergencias durante el desarrollo de una clase
- Tópicos del programa
- Dudas surgidas y no aclaradas
- Temas de actualidad social.

Técnica del seminario

El seminario es una técnica más amplia que la discusión o el debate, pudiéndose incluir ambas en su desarrollo.

- El profesor plantea lo fundamental del tema.
- Los estudiantes exponen los resultados de sus estudios, donde los llevan al debate.
- Cuando no se queda aclarado el profesor presta ayuda en el tema.
- Al final son coordinadas las conclusiones, con el auxilio del profesor
- Para un seminario eficiente todos los estudiantes deben prepararse para dicho tema.

Técnica del estudio de casos

Consiste en la presentación de un caso o problema para que la clase sugiera o presente soluciones.

- El profesor es orientador

- La presentación de un caso es presentado por el profesor, un alumno, o una autoridad.
- La participación puede llevarse: las opiniones pueden ser dadas individualmente, por los alumnos el tema es subdividido en subtemas que serán dados a grupos para estudiarlos.

Técnica de problemas

Estudia los problemas de las diversas disciplinas en el orden en que van surgiendo y evolucionando, tiene por objeto desarrollar el razonamiento del alumno, a fin de prepararlo para enfrentar situaciones problemáticas que la vida puede presentarle a cada instante.

Técnica de la demostración

Es el procedimiento más deductivo y puede asociarse a cualquier otra técnica de enseñanza cuando sea necesario comprobar afirmaciones no muy evidentes o ver cómo funciona, en la práctica, lo que fue estudiado teóricamente.

Esta técnica tiene por objetivos:

- ✓ Confirmar explicaciones orales o escritas
- ✓ Ilustrar lo que fue expuesto teóricamente
- ✓ Iniciar teóricamente una técnica para evitar errores
- ✓ Propiciar un esquema de acción correcto para la ejecución de una tarea.
- ✓ Demostrar racionalmente en cuanto al a veracidad de proposiciones abstractas.

Técnica de la experiencia

La experiencia es un procedimiento eminentemente activo y que procura:

- Repetir un fenómeno ya conocido
- Explicar un fenómeno que no es conocido
- Comprobar, con razones lo que va a suceder, partiendo de experiencias

- Conferir confianza para actuar en el terreno de la realidad de manera lógica
- Convencer a cerca de la veracidad de la ley de causa y efecto
- Fortalecer la confianza en sí mismo
- Formar la mentalidad científica
- Orientar para solucionar problemas
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

1.2.4. Propuesta del Proyecto referente a la Capacitación

1.2.4.1.Descripción del Curso

Este curso de 5 días está diseñado para preparar a los o las participantes en el conocimiento y utilización de las TIC, esto con el fin de que puedan manejar y administrar el Sistema Web desarrollado en el proyecto de manera individual y sin depender de algún otro experto.

Este curso está dirigido con preferencia al Área de Gerencia de la Empresa, pero puede adaptarse de acuerdo a las circunstancias (falta de participación, petición de interesados en participar) a la inclusión de otros participantes de las otras Áreas con las que cuenta la institución, además queda abierta la posibilidad de extender el curso de acuerdo a la respuesta.

1.2.4.2.Metas y objetivos generales del Curso

Influenciar positivamente las actitudes de los o las participantes respecto a los beneficios de emplear el Sistema Web desarrollado en el proyecto, el uso y manejo de las TIC como medio alternativo para incrementar su competitividad.

Proporcionarle al o la participante los conocimientos y las habilidades necesarios para manejar y administrar correctamente el software de gestión

1.2.4.3.Objetivos del Aprendizaje del participante

Después de asistir a los cursos de capacitación sobre las TIC y manejo del Sistema Web desarrollado en el proyecto, el o la participante estará en capacidad de administrar el Sistema de Control de Almacenes de la Empresa.

1.2.4.4.Descripción de los métodos de Capacitación y Socialización

El proyecto plantea llevar a cabo para la capacitación y socialización una:

✓ Metodología de Enseñanza Socializada

El cuálpretende ejecutar una metodología de enseñanza socializada, dirigida a los gerentes con lo que se busca una integración social sin descuidar la individualización, aplicando **métodos de asamblea y Panel**.

Para llevarlas a cabo se desarrollará el curso usando las siguientes técnicas:

✓ Técnica expositiva

Consiste en la exposición oral, por parte del capacitador, en la que se estimulará la participación del personal en los trabajos que se realicen, este requiere una buena motivación para atraer la atención de los participantes.

✓ Técnica de la experiencia

La experiencia es un procedimiento eminentemente activo y que procura:

- Exponer de forma presencial los pasos a seguir para efectuar alguna acción. Explicar el funcionamiento de algo mientras es observado.
- Comprobar, con razones lo que va a suceder, partiendo de experiencias.
- Conferir confianza para actuar en el terreno de la realidad de manera lógica.
- Convencer a cerca de la veracidad de la ley de causa y efecto.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

✓ **Metodología de Enseñanza Centrada en el Estudiante**

Esta metodología permite que los estudiantes construyan e integren los nuevos conocimientos adquiridos.

Los profesores que practican la forma de enseñanza centrada en el estudiante tienen un rol primordial para lograr que los estudiantes construyan e integren los nuevos conocimientos.

Demostración: Pruebas, test y ensayos del Sistema Web.

Para la Capacitación se empleará Manuales de usuarios que contienen toda la información detallada sobre el manejo de Sistema de Web y llevaremos un registro del personal que asiste a la capacitación.

1.2.4.5.Descripción de los materiales de Capacitación

Materiales impresos: Guías del curso, manuales de usuario.

Materiales audiovisuales: Equipo de computación, proyector, micrófono y parlantes

Otros materiales: Marcadores, pizarrón

1.2.4.6.Criterios para la selección de los participantes

Los o las participantes de este curso deben ser personal de la Cooperativa, no se exige a los participantes tener conocimientos acerca de computación.

1.2.4.7.Métodos para la evaluación del curso

Cuestionario pre curso: para evaluar los conocimientos pre curso acerca de computación.

Guías de aprendizaje: Los/las participantes evalúan sus propias habilidades

Verificación de lo aprendido: El/la capacitador/a evalúa las habilidades de los/las participantes en el manejo del Sistema de gestión.

Evaluación del curso de capacitación: Los/las participantes evalúan el curso.

1.2.4.8.Duración del curso

El curso tendrá una duración de 5 días, cada clase tendrá una duración aproximada de 90min por día

1.2.4.9. Número sugerido de participantes

Preferiblemente se recomienda no sobrepasar la cantidad de 30 participantes en el curso, esto debido a la falta de instalaciones para acoger a más personas, al presupuesto del proyecto y para favorecer al propio ambiente de aprendizaje.

1.2.4.10. Fecha del curso

De acuerdo al cronograma del proyecto, el curso se realizará a partir del 28 noviembre de 2012 hasta el 2 de diciembre de 2012, salvo algún contratiempo no planificado.

1.2.4.11. Organizador del curso de capacitación

El curso será organizado por el director del proyecto, bajo asesoramiento de los docentes de la materia de Taller III y del Gerente General de la Empresa.

1.2.4.12. Contenido del Curso sobre “Las TIC”

- **La Internet**

Algunos definen Internet como "La Red de Redes", y otros como "La Autopista de la Información".

Efectivamente, Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores, o sea de unos pocos ordenadores en un mismo edificio o empresa.

Por la Red Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información. Hay 50 millones de "Internautas", es decir, de personas que "navegan" por Internet en todo el mundo. Se dice "navegar" porque es normal el ver información que proviene de muchas partes distintas del mundo en una sola sesión.

- **Sistemas de Información**

Pudiésemos determinar entonces como Sistema de información al conjunto de elementos relacionados y ordenados, según ciertas reglas que aporta al Sistema objeto, es decir, a la organización a la que sirve y que marca sus directrices de funcionamiento, la información necesaria para el cumplimiento de sus fines; para ello, debe recoger, procesar y

almacenar datos, procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación.

- **Introducción a Sistemas Web**

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

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

Actualmente, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con los sistemas de información para satisfacer sus necesidades de información

- **Que son TIC**

Las Tecnologías de Información y Comunicación (Tics), se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de Sistema informático.

- Hardware
- Software
- Servicios de TI
- Telecomunicaciones

- **Globalización**

La globalización es el proceso por el que la creciente comunicación e interdependencia entre los distintos países del mundo unifica mercados, sociedades y culturas, a través de una serie de transformaciones sociales, económicas y políticas que les dan un carácter global. Así, los modos de producción y de movimientos de capital se configuran a escala planetaria, mientras los gobiernos van perdiendo atribuciones ante lo que se ha denominado la sociedad en red.

En este contexto el desarrollo tecnológico, y refiriéndome en sí a la incorporación de las TIC en cada nación, hará que el desarrollo en esta llegue sin tardar demasiado, la competitividad de un país en un Sistema globalizado depende en gran medida de la tecnología que adquiera, por eso es tan estrecha la relación entre TIC y globalización. Esperemos que nuestros países comiencen a incorporar las TIC en sus procesos productivos y así reducir la dependencia de potencias mundiales.

- **Las TIC y la competitividad**

La utilización de las TIC y el Comercio Electrónico por parte del sector empresarial, abre nuevas oportunidades de negocios que redundan en una serie de beneficios económicos y sociales, los cuáles van desde el mejoramiento y la facilitación de la comunicación entre las empresas, hasta el manejo más eficiente de los recursos. En tal sentido, las tecnologías deben ser empleadas por el sector en las diferentes etapas de la cadena de valor.

- Calidad
- Productividad
- Servicio
- Imagen

Los procesos centrados en la producción: En esta etapa, las TIC's pueden ser utilizadas para diseñar y probar nuevos productos, e-procurement, procesos de pagos, sistemas de gestión automática de stocks, diferentes tipos de links electrónicos con proveedores, sistemas de control y procesos más relacionados con la producción, entre otros.

Los procesos internos: En lo referente a administración de personal, entrenamiento, reclutamiento interno, compartir y diseminar información de la compañía vía electrónica, entre otros.

El Software para la empresa es vital debido a:

- Necesidad de optimizar

1.2.4.13. Capacitación de Uso de las TIC's

Fecha a Realizar	Sesión de Aprendizaje del Curso	Horas C/Módulo	Nº de Sesiones	Dirigido A
28/11/2012	Sesión 1. La Internet	25 min.	1	Personal del Comercial el Carmen
	Sesión 2. Sistemas de Información	25 min.	1	
	Sesión 3. Introducción a Sistemas Web	20 min.	1	
	Sesión 4. Que son TIC	20 min	1	
	Sesión 5. Las TIC y la Competitividad	30 min	1	
	Sesión 6. Presentación y Manejo del	30 min.	1	Personal de la Empresa en Especial al

	Sistema Informático CONAL.			Personal del Área de Almacenes y Ventas.
TOTALES		3 Horas	6 Sesiones	

Tabla 100. Capacitación en Uso de las TIC's

1.2.4.14. Capacitación en el Uso del Sistema Informático Desarrollado

Fecha a Realizar	Capacitación de Aprendizaje del Curso	Horas C/Módulo	Nº de Sesiones	Dirigido A
30/11/2012	Capacitación en el Manejo del Sistema Informático CONAL.	90 min.	1	Encargado de Sistemas y el Área de Almacenes y Gerencia General.
01/12/2012	Capacitación en el Manejo del Sistema Informático CONAL	90 min.	1	Encargado de Sistemas y el Área de Almacenes y Gerencia General.
02/12/2012	Capacitación en el Manejo del	90 min.	1	Encargado de Sistemas y el

	Sistema Informático CONAL			Área de Almacenes y Gerencia General.
TOTALES		4 Horas, 30 min	1 Sesiones	

Tabla 101. Capacitación en el uso del Sistema Informático Desarrollado

III. Capítulo III: Conclusiones y Recomendaciones

3.1.1. Conclusiones y Recomendaciones

- **Conclusiones**

Para el desarrollo del Sistema de Control y Administración de Almacenes, se realizó un Análisis de Requerimientos utilizando la metodología RUP, logrando conocer así los problemas y necesidades de la Empresa, para después analizar la situación actual y proponer una solución.

El sistema que se desarrollo con el proyecto cumple con todo los requisitos fijados del usuario, por lo tanto se ha llegado a cumplir con los objetivos propuestos del componente 1.

La información es un factor necesario para el proceso de toma de decisiones en las organizaciones, instituciones y sobre todo en las empresas que buscan satisfacer las necesidades de servicios de calidad.

La metodología RUP nos permite realizar mejoras al software, puesto que el proceso de esta permite iterar y realizar las modificaciones necesarias

El número de organizaciones, instituciones que hacen uso de las TIC's (Tecnologías de Información y Comunicación) para la gestión y automatización de sus procesos administrativos, están en incremento.

El uso correcto de las TIC en las empresas, conlleva mejores controles administrativos y operativos, por lo tanto logramos mejores niveles de productividad, satisfacción del cliente y oportunidades de negocio.

- **Recomendaciones**

Para el desarrollo de futuros proyectos con características similares a éste se recomienda hacer uso de las metodologías utilizadas como (UML, XP), etc. Ya que son estándares para el desarrollo de sistemas.

La inserción del Sistema de Control de Almacenes (CONAL) contribuirá en la Gestión de Almacenes en sus diferentes etapas, como: Ingresos de Productos, Salida de Productos, Administración de Proveedores, y otros inherentes al proceso administrativo de los Almacenes de la Empresa, actividades que en su conjunta permitirán optimar los recursos, reduciendo la administración manual de datos.

Hacer conocer a otras empresas e instituciones las ventajas de contar con un sistema automatizado.

Tratar de aplicar los conocimientos infundidos sobre las TIC en futuros proyectos de mejora para institución

Se recomienda que RR.HH en el Área informático debe cumplir con el perfil adecuado en conocimiento de las nuevas metodologías de desarrollo y en el manejo de las últimas tecnologías.

3.1.2. Medios de Verificación

3.1.2.1.Registro de los participantes de los cursos de capacitación realizados por los desarrolladores del proyecto

El registro de asistencia se lo presenta en ANEXOS