

CAPÍTULO I

EL PROYECTO

I EL PROYECTO

I.1 PRESENTACION DEL PROYECTO

I.1.1 TITULO

“Mejoramiento de la gestión administrativa de la empresa "DIGITAL CORPORATION", con plataforma web”.

I.1.2 AREA DEL PROYECTO

Tecnologías de la Información y Comunicación.

I.1.3 RESPONSABLE DEL PROYECTO

Carrera de Ingeniería Informática – Taller III – Grupo 3

I.1.4 ENTIDADES ASOCIADAS

Empresa DIGITAL CORPORATION.

Universidad Autónoma Juan Misael Saracho.

I.1.5 GRUPO RESPONSABLE DEL PROYECTO

Grupo 3 Taller III

Universitario:

Iván Ayala Catarí

I.1.6 DURACION (MESES)

La realización del Proyecto tendrá una duración de 10 meses.

I.1.7 ACTIVIDADES PREVISTAS PARA LOS INTEGRANTES DEL EQUIPO DE INVESTIGACION

Responsable	Actividades
Director Iván Ayala Catari	<u>En cuanto al seguimiento y control del Proyecto</u> - Planificación y control del cronograma del proyecto. - Asignar y gestionar recursos y prioridades a los distintos componentes y actividades del proyecto. - Coordinar las iteraciones entre el equipo de trabajo y los usuarios del proyecto. - Mantener al equipo del proyecto enfocado en los objetivos. - Establecer un conjunto de prácticas que aseguren la calidad e integridad del proyecto. - Supervisar el desarrollo del proyecto. <u>En cuanto al desarrollo de los componentes</u> - Capturar la especificación y validación de requisitos interactuando con los usuarios. - Elaboración del Análisis, Diseño y Desarrollo del Sistema. - Elaboración del Modelo de datos. - Delineación de los Diagramas UML. - Diseño de la Base de Datos del Sistema. - Programación del Sistema Web. - Preparación de pruebas Funcionales del Sistema

	- Planificar el contenido temático de los cursos de capacitación. - Formar al personal en el uso de las TIC para el manejo del producto final. - Definir las estrategias de socialización del proyecto. - Llevar a cabo la socialización del producto final.
--	---

Tabla N° 1: Responsable del Proyecto

I.2 DESCRIPCION DEL PROYECTO

I.2.1 Resumen ejecutivo del Proyecto

- La empresa nació en 2007 ofrece a todas las personas que viven en nuestro departamento una forma sencilla de acceder a un crédito directo en todos los productos con los que cuenta nuestra empresa. DIGITAL CORPORATION Actualmente contamos con 950 clientes activos habiendo hecho felices a varias familias saciando varias necesidades que aquejaban como ser: transporte, comunicación, amenidad, comodidad, etc.
- DIGITAL CORPORATION tienen la característica de pertenecer a un sector cuya escala de operación eficiente es pequeña, pudiendo ésta coexistir con empresas grandes produciendo beneficios parecidos o mejores.
- DIGITAL CORPORATION desde su inicio y hasta el momento ha ido creciendo en gran magnitud, y actualmente debe manejar una gran cantidad de información sobre las ventas que se ejecutan y las diferentes actividades que realizan, por lo cual los medios con los que cuenta para el control de la información han quedado obsoletos.
- El presente proyecto contribuye a su propósito, mejorando la gestión de la información en el área administrativa (Clientes, Productos, Ventas, Control de Plan de Pagos) de DIGITAL CORPORATION. Se pretende desarrollar un

sistema de información que podrá facilitar el acceso a la información de esta manera permitirá medir el éxito que tiene como empresa.

I.2.2 Descripción, fundamentación y Justificación del Proyecto

Este proyecto tiene como uno de sus objetivos el mejoramiento de la gestión de los procesos administrativos y también la difusión de información sobre la Empresa DIGITAL CORPORATION.

Por lo tanto con este proyecto lo que se pretende es brindar información confiable y promover el conocimiento de los diferentes servicios que ofrece la Empresa DIGITAL CORPORATION a la sociedad.

Las ventajas que traerá este proyecto serán muchas por así mencionar el incremento de la clientela, mayor conocimiento de la Empresa por parte de la sociedad.

JUSTIFICACIÓN

JUSTIFICACIÓN SOCIAL Y ECONOMICA

Se pretende que el Trabajo de grado apoye en gran medida a la gestión de los procesos administrativos de la Empresa DIGITAL CORPORATION. También se pretende abastecer información eficiente y confiable a la población.

JUSTIFICACIÓN ACADÉMICA

Para el tratamiento del Sistema se aplicaran conocimientos adquiridos durante la etapa de estudio en la Carrera de Ingeniería Informática, también se aplicara herramientas disponibles en el mundo de la informática.

I.2.3 Objetivos

I.2.3.1 Objetivo General

Mejorar la gestión administrativa de la empresa "DIGITAL CORPORATION" haciendo uso de las nuevas tecnologías.

I.2.3.2 Objetivos Específicos

- Desarrollar un Sistema de Gestión Administrativa para la Empresa DIGITAL CORPORATION.
- Ejecutar la capacitación al personal encargado de la gestión administrativa sobre el manejo y uso del Sistema.

I.2.4 Metodología

Referente al Desarrollo del Proyecto:

Se pretende establecer un ciclo de vida del proyecto compuesto por fases que se van dando de manera secuencial en el transcurso del tiempo.

Se utilizarán las siguientes metodologías para cada componente:

1. Sistema Web de Gestión para la Empresa DIGITAL CORPORATION.

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

2. Talleres de Capacitación en el uso del Sistema Web de Gestión.

Para la ejecución de la capacitación se elaboró un manual de usuario con el objetivo de que dicho manual sea usado en caso de surgir alguna duda sobre el manejo del sistema. Dicho manual cuenta con imágenes impresas de las pantallas del sistema y los posibles mensajes que podrían salir al usuario.

Fase de Iniciación y Planificación:

Se efectuará la presentación formal de la idea a las instituciones involucradas.

Se realizará la planificación temporal de recursos y la definición del fin y objetivos del proyecto.

Se utilizará el Sistema de Marco Lógico para presentar de forma sistemática y lógica los objetivos del proyecto y sus relaciones de causalidad además de mejorar la planificación, el seguimiento y la evaluación del proyecto.

Se realizara un calendario de actividades y un cronograma de control de fechas de inicio y entrega de cada actividad.

Fase de Ejecución:

La recopilación de información se realizará por medio de cuestionarios y entrevistas a las empresas involucradas.

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. El proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. Se usará la notación UML para visualizar, especificar, construir y documentar el sistema de software a desarrollar. Se realizará un modelo de datos basado en estándares que nos permitan mantener la integridad de la información.

Fase de Entrega:

Se comprobara si el producto final funciona correctamente, además se revisará si el proyecto responde a todas las especificaciones aprobadas.

Referente a la Transferencia de Resultados:**Metodología de Enseñanza Socializada**

Se pretende ejecutar una metodología de enseñanza socializada, dirigida a los usuarios finales.

Técnica expositiva

Consiste en la exposición oral, por parte del capacitador; en la que se estimulará la participación de los usuarios, este requiere una buena motivación para atraer la atención de los educandos.

Técnica de la experiencia

La experiencia es un procedimiento eminentemente activo y que procura:

Exponer de forma presencial los pasos a seguir para realizar alguna acción.

I.2.5 Bibliografía Consultada

Páginas Visitadas en Internet

[1]. Enciclopedia Libre: Proceso Unificado de Rational.

http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational [Consulta 05/04/2012].

[2]. Ing. Córdoba Cepal : LA METODOLOGIA DEL MARCO LOGICO COMO BASE PARA CONSTRUIR LA MATRIZ DE INDICADORES. Diciembre 2008.

<http://www.slideshare.net/ahaazd/mml-base-para-matriz-de-indicadores-para-resultados>
[Consulta 05/04/2010].

[3] CASOS DE USO

http://www.cs.ualberta.ca/pfigueroa/soo/uml/casos_uso01.html[Consulta 10/05/2012].

[4] POSTGRESQL

<http://es.tldp.org/postgresql-es/web/navegable/tutorial/tutorial.html>[Consulta 07/07/2012].

[5] UML

<http://es.wikipedia.org/wiki/UML>[Consulta 09/08/2012].

[6] Sistemas de plataforma web Sistemas de plataforma web. Webroom - Webroom Soluções Interativas. 2010

<http://www.webroom.com.br/conteudo.aspx?cont=120&pai=66> [Consulta 09/09/2012].

<http://www.scribd.com/doc/2983518/Marco-Logico-10-casos-practicos> [Consulta 30/04/2011].

[7]. ENCICLOPEDIA-ACE METODOLOGIA DE LA CAPACITACION. 1998-2009 ACE Project.

[8] <http://aceproject.org/ace-es/topics/vo/voe/voe04/voe04b/voe04b-default> [Consulta 03/04/2011].

Curso de: Cátedra abierta SML (Sistema del Marco Lógico) dictada por Silvana Paz – Efraín Torrejón

http://turing.iimas.unam.mx/~coria/dgsca_curso_datware_datmin_2006/presentaciones/session_06/ieee_830_standard_12nov2005.ppt

Libros Consultados

[9] INGENIERÍA DEL SOFTWARE ORIENTADA A OBJETOS

Roger S. Pressman (2002)

[10] Libro de Contabilidad: “Tratamiento Contable de los Bienes de Uso y los Activos Intangibles” autores: Rubén Centellas España y Carlos Gonzáles Alanes

I.2.6 Cronograma de Actividades

N°	Actividad	N° días	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	Presentación del Perfil de Proyecto	21			X	X								
2	Entrega de observaciones	7				X								
3	Entrega de Perfiles modificados	7				X								
4	Primera revisión	23				X	X							
5	Segunda revisión (1ra. Eval.)	31					X	X						
6	Tercera revisión	31						X	X					
7	Cuarta revisión (2da.eval.)	65							X	X	X			
8	Entrega de Borradores	47									X	X	X	
9	Observaciones por parte del tribunal	11											X	X
10	Entrega de borradores finales	21												X
11	Elaboración de rol de defensas	1												X
12	Defensas finales	9												X

Tabla 2: Cronograma de Actividades

I.2.7 Marco Lógico del Proyecto

	Indicadores	Medios de Verificación	Supuestos
Fin Mejorar la atención a clientes en cuanto a la venta de productos y servicios que ofrece DIGITAL CORPORATION	Luego de 2 años finalizado el proyecto y habiendo puesto en ejecución los productos del mismo, al menos un 40% de los clientes de DIGITAL CORPORATION indican que la administración es más eficiente.	Encuesta realizada a clientes de DIGITAL CORPORATION durante un periodo de tiempo referente a la atención de clientes	DIGITAL CORPORATION invierte en la implementación y mantenimiento del Sistema.
Objetivo General (Propósito) Gestión administrativa mejorada de la empresa "DIGITAL CORPORATION" con Plataforma Web.	Al finalizar el proyecto se han automatizado (como criterio de mejora) 75% al menos de los procesos relativos a la gestión administrativa de DIGITAL CORPORATION	Documento emitido por el Gerente de la Empresa DIGITAL CORPORATION que exprese conformidad por los resultados obtenidos en el mejoramiento de la gestión administrativa	Presupuesto suficiente para la implementación del sistema.

Objetivos Específicos (Componentes) 1.-Sistema web de gestión de procesos administrativos desarrollado. 2.-Capacitacion del sistema Implementada.	Al finalizar el proyecto se ha desarrollado el sistema para la gestión de procesos administrativos de la empresa DIGITAL CORPORATION basándose en los requerimientos expresados bajo la norma IEEE-STD-830-1998 Al terminar el proyecto para el uso del sistema web. Se han implementado 3 talleres de capacitación.	Informe otorgado por la Gerencia que garantiza el seguimiento que se hizo al Sistema. Durante su desarrollo. 2.1 Fotos de verificación de la capacitación. 2.2 Certificados de capacitación	1.1 Disponibilidad del equipo y accesorios necesarios para la implementación del sistema. 1.2 El personal involucrado proporciona toda la información necesaria para una correcta recolección de datos. 2.1 Personal de la empresa apoya con todo lo necesario para que se lleven a cabo las jornadas de capacitación. 2.2 Personal involucrado asiste a las jornadas de capacitación.
---	--	--	--

ACTIVIDADES COMPONENTE 1: Sistema web de gestión de procesos administrativos desarrollado. 1. Determinación de requerimientos 2. Análisis y Diseño del Sistema Web. 3. Desarrollo del Sistema Web. 4. Ejecución de Pruebas al Sistema Web. COMPONENTE 2: Capacitación del sistema Implementada 1. Elaboración de los Manuales de Instalación y Manejo del Sistema Web 2. Planificación y Preparación de las Jornadas de Capacitación. 3. Ejecución de las Jornadas de Capacitación.	El componente tendrá un costo de: 29175.175 El componente tendrá un costo de: 983.25 Bs	Instalador del Sistema web listo para ser utilizado. Manual de Usuario e Instalación impresos.	El equipo de trabajo cuenta con todo el apoyo de los involucrados para el correcto desarrollo de todas las actividades del sistema web. • Costos de Bienes y Servicios no afectan al presupuestado. • La Empresa DIGITAL CORPORATION corre con todos los gastos expresados en el presupuesto de capacitación.
--	---	--	---

Tabla 3: Matriz de Marco Lógico

ÁRBOL DE PROBLEMAS

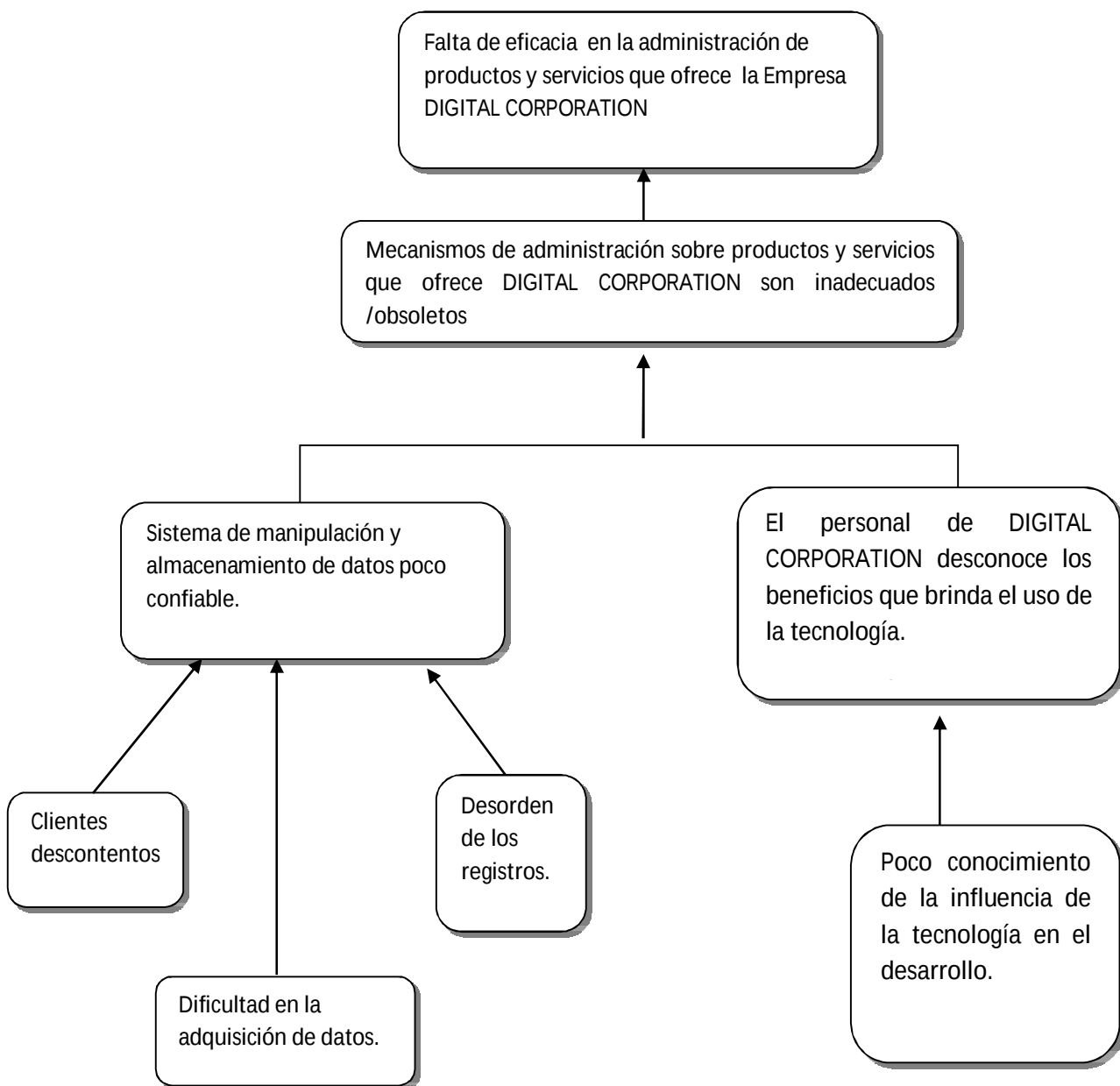


Figura 1: Árbol de Problemas

ÁRBOL DE OBJETIVOS

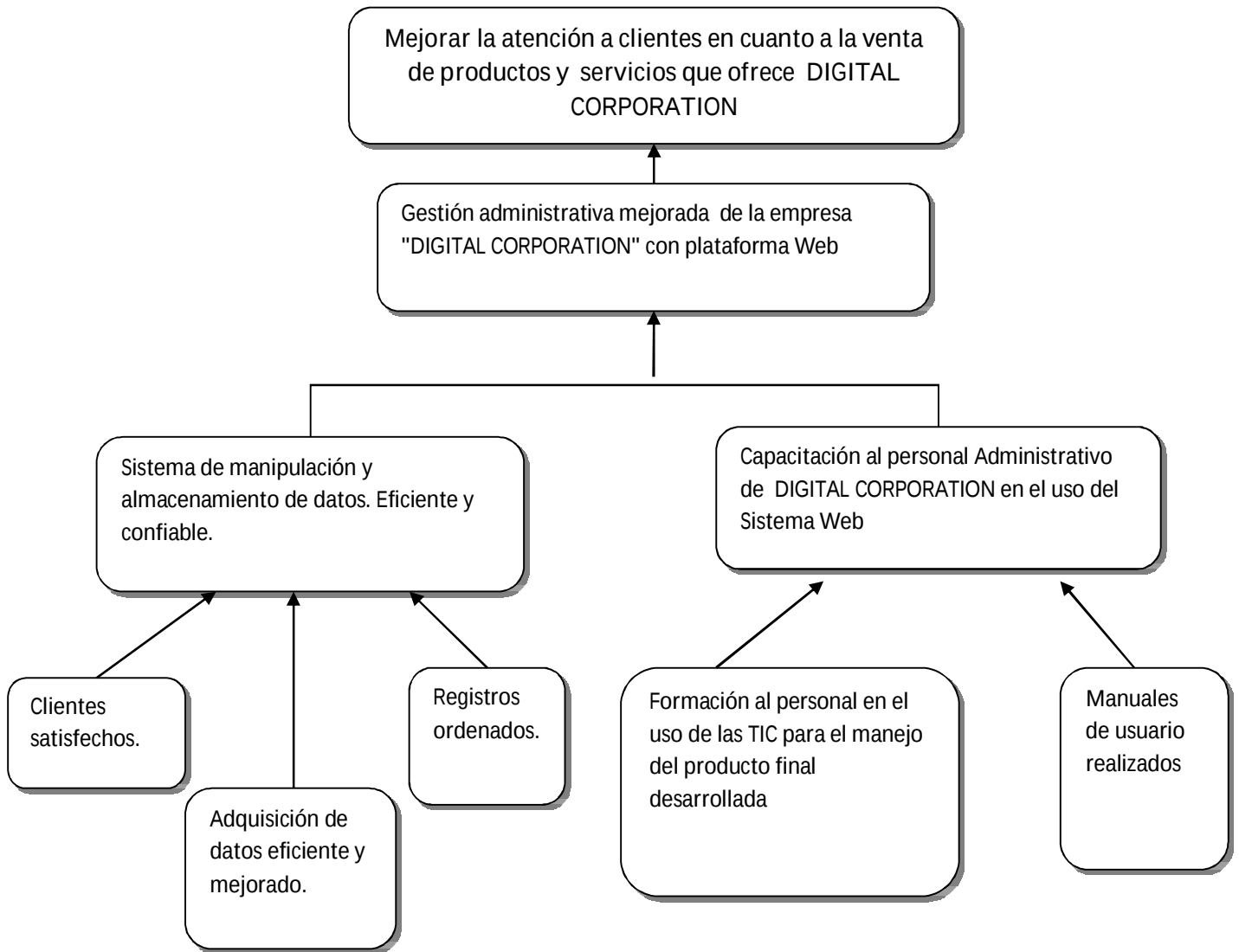


Figura 2: Árbol de Objetivos

CUADRO DE INVOLUCRADOS

GRUPO	INTERESES	PROBLEMAS PERCIBIDOS	RECURSOS Y MANDATOS
Personas de la Sociedad	Mejor Información sobre la Empresa.	- La Empresa Digital Corporación es una de las empresas con menos conocimiento por parte de la población de Tarija. - Escasos medios de difusión de información sobre la Empresa DIGITAL CORPORATI ON.	Disponibilidad de usar el Sistema Web si es confiable.
Clientes	- Atención rápida y eficaz a los requerimientos que se tiene el cliente. - Disponer de la información precios y disponibilidad de productos para realizar cotizaciones. - Recibir un servicio confiable y seguro en la parte de adquisiciones de productos.	- Demora en la atención al cliente. - Falta de conocimiento de la influencia de la tecnología en el desarrollo.	- Alguna disponibilidad de usar el Sistema Web si es confiable. - Mayor desarrollo tecnológico.
Director	Tener un Sistema Web de Gestión y	Demora en la atención a los clientes	Interés en apoyar iniciativas que permitan el

	Informativo. • Ofrecer un buen servicio a sus clientes.	debido a los procesos manuales que se realizan (anotaciones, presentación de catálogos).	desarrollo de la empresa. • Amplio conocimiento del negocio. • Productos de buena calidad y aptos para la competencia.
--	--	---	--

Tabla N° 4: Cuadro de Involucrados

CAPÍTULO II

COMPONENTES

II. 1. COMPONENTE 1

SISTEMA DE GESTION DE LOS PROCESOS ADMINISTRATIVOS VIA WEB DE DIGITAL CORPORATION

II.1.1. COMPONENTE 1.

II.1.1.1. Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como 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 ofertado por el Univ. Iván Ayala Catari; Basado en una metodología de Rational Unified Process en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología, constando únicamente en la tercera fase de dos iteraciones. 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 los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

En la actualidad la Empresa DIGITAL CORPORATION no cuenta con un Portal Web el cual pueda brindar información a los clientes y público en general. Por tal motivo se vio la necesidad de desarrollar un Sistema Web de Gestión y Información para la mejora de procesos administración y que el usuario tenga un fácil acceso a la información y servicios provistos por la Empresa DIGITAL CORPORATION.

Entre las características principales de este nuevo medio tecnológico en línea observamos las siguientes Ventajas:

- Personalizado.
- Disponible.
- Confiable.
- Instantáneo.
- Interactivo.

II.1.1.1.1. Propósito

El propósito del presente documento es proporcionar la información necesaria para controlar el proyecto, en él se describe el enfoque de desarrollo del software.

Con este documento se pretende proporcionar una idea del software a desarrollar exponiendo a la vez su estructura hasta una visión terminada.

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Sistema Web de Gestión Para La Empresa DIGITAL CORPORATION” El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones.

II.1.1.1.2. Alcance

Este instrumento proporcionará una idea del software a desarrollar exponiendo a la vez su estructura hasta una visión terminada.

El mismo describe el plan global usado para el desarrollo del “Sistema Web Para La Empresa DIGITAL CORPORATION”. Para la presente versión de éste escrito, se ha basado en la captura de requisitos por medio del Gerente Propietario de la Empresa mencionada para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Planificación se generará la primera versión del artefacto “Visión”, el cual se empleará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

II.1.1.1.3. 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 y describe cómo se realizará su seguimiento.

II.1.1.2. Vista General del Proyecto

II.1.1.2.1. Propósito, Objetivos y Alcances

Propósito

El problema radica en que en la actualidad la Empresa DIGITAL CORPORATION no cuenta con un medio en el cual se pueda acceder a la información actualizada de manera rápida.

Por tal motivo se proporcionará una herramienta software que permita automatizar el manejo de la información como ser: venta, registro, productos y reportes; para que ayude a alcanzar los objetivos deseados por la institución y por los usuarios de nuestro sistema.

Objetivos

Objetivo General

Diseñar e Implementar un sistema web de gestión el cual contara con Información específica y actualizada de la empresa DIGITAL CORPORATION como ser: gestión de usuarios y clientes; lista de productos; venta y reportes de ventas, sistema que será un medio de ayuda para la Empresa.

Objetivo Específico.

- Analizar y Diseñar la aplicación propuesta.
- Instrumentar el sistema con las siguientes tecnologías:
 - o PostGreSQL.
 - o Java(TM)
- Aplicar el lenguaje de modelado unificado (UML) en toda la fase del desarrollo.
- Aplicar la metodología de desarrollo RUP (Proceso Unificado Racional).
- Diseñar una interfaz gráfica atractiva, amigable y de fácil uso para el manejo del usuario.
- Facilitar el acceso y manejo de la información: El uso de una base de datos que nos permita registrar toda la información producida en el departamento, que permitirá a través de una consulta a la misma acceder a los datos deseados de manera rápida y oportuna en un tiempo eficiente.
- Proveer mayor seguridad en el manejo de la información: El hecho de asignar roles a cada actor (usuarios) que interviene dentro del sistema, nos permitirá tener cierto grado de seguridad en el manejo y acceso a la información dentro del sistema Web.

Alcances

El alcance general del presente trabajo es implementar un Sistema web de Gestión para la Empresa “DIGITAL CORPORATION” que proporcione al usuario una interfaz amigable para que este pueda tener Información clasificada, actualizada y acorde a sus necesidades y requerimiento.

La parte de Gestión de Usuarios: Se toma en cuenta las altas, bajas y modificaciones de los usuarios y sus respectivos roles, donde solo el Administrador tiene permiso de realizar dichas acciones, esto incluye el nombre del usuario, su apellido paterno, apellido materno, Fecha de nacimiento, Dirección, teléfono, celular, login, clave y el nombre de rol.

La parte de Gestión de Clientes: Se toma en cuenta las altas, bajas y modificaciones de los clientes, donde los encargados tienen permiso de realizar dichas acciones, esto

incluye el nombre del cliente, su apellido paterno, apellido materno, Dirección, teléfono.

En la parte de venta nos limitaremos al control y registro del cliente, la persona que realiza la venta, importe, fecha de emisión, y demás datos que se requieren para emitir un comprobante.

Se realizaran reportes de usuarios, productos, clientes y ventas.

El sistema pretende llegar a cubrir las necesidades básicas de la Empresa Comercial “DIGITAL CORPORATION”.

II.1.1.2.2. Suposiciones y Restricciones

Suposiciones

- ✓ Información actualizada.
- ✓ Obtención de información, ágil y rápida.
- ✓ Reducción y detección de errores en cuanto a la información.
- ✓ No existirá pérdida de información de acuerdo al control y seguridad que requiera el software.

Restricciones

- ✓ El sistema no contara con un chat.
- ✓ El sistema requiere capacitación y mantenimiento.
- ✓ El sistema será implementado en las plataformas Windows o Linux.
- ✓ El sistema será diseñado según los requerimientos de la Empresa DIGITAL CORPORATION.

II.1.1.2.3. 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 preciso 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 presenten los objetivos de cada iteración.

1) Plan de Desarrollo del Software

Es el presente documento.

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, 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 usando estereotipos específicos para este modelo.

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.

4) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada. .

5) 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.

6) Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

7) 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 realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, 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.

8) 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.

9) Prototipos de Interfaces de Usuario

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.

10) Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

11) 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.).

12) Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

13) Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

14) 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.

15) Solicitud de Cambio

Los cambios propuestos para los artefactos se formalizan mediante este documento. Mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación e impacto, y se asegura que éstos sean conocidos por el equipo de desarrollo. Los cambios se establecen respecto de la última baseline (el estado del conjunto de los artefactos en un momento determinado del proyecto) establecida. En nuestro caso al final de cada iteración se establecerá una baseline.

16) Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

17) Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento y Sistema de Ayuda en Línea

18) Producto

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

Los artefactos 19, 20 y 21 se generarán a partir de la fase de Construcción, con lo cual se han incluido aquí sólo para dar una visión global de todos los artefactos que se generarán en el proceso de desarrollo.

II.1.1.3. Organización del Proyecto

II.1.1.3.1. Participantes en el Proyecto

Jefe de Proyecto.

Universitario Iván Ayala Catarí, alumno del último curso de la carrera de Ingeniería Informática en la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Con una experiencia modesta en metodologías de desarrollo, herramientas CASE y notaciones, en particular la notación UML y metodología de desarrollo RUP.

Analista de Sistemas.

Universitario de Ingeniería Informática con conocimientos de UML, con experiencia en sistemas afines a la línea del proyecto, labor que llevará a cabo Iván Ayala Catarí.

Analistas - Programadores.

Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final. Este trabajo ha sido encomendado a Iván Ayala Catarí.

Ingeniero de Software.

El perfil establecido estará a cargo del estudiante de Informática que participará realizando labores de Gestión de requisitos, Gestión de configuración, documentación y diseño de datos. Encargado de las pruebas funcionales del sistema, realizará esta labor Iván Ayala Catarí.

II.1.1.3.2. Interfaces Externas

II.1.1.3.3. Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos.

	Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

Tabla N° 5: Roles y Responsabilidades

II.1.1.4. Gestión del Proceso

II.1.1.4.1. Estimaciones del Proyecto

II.1.1.4.1.1. Presupuesto / Justificación

Se desea estimar los costos que tendrá el Sistema Web para la gestión de la información de la Empresa DIGITAL CORPORATION.

Utilizaremos el método COCOMO para estimación de costos y para esto tenemos los siguientes datos: El sistema estima producir 3,5 MIFE (miles de instrucciones fuentes) se pretende que el sistema tenga una complejidad media y para esto se contratará analistas y programadores que tengan conocimientos medios en este tipo de programación del sistema (cobrando alrededor de 400 US\$) pretende brindar una confiabilidad alta utilizar recursos de base de datos muy altos y una complejidad normal.

Método Cocomo Básico

Desarrollo:

Datos: Tamaño=3 MIFE Tarifa mensual= 400 US\$ COSTO=EN*tarifa mensual

Para calcular el Esfuerzo Nominal vamos a utilizar la siguiente fórmula:

$$EN=2.4*(MIFE)^{1.05}$$

Remplazando datos nos dará $EN=2.4*(3,5)^{1.05}$ entonces el esfuerzo nominal será:

$$EN=9 \text{ meses de programador.}$$

Para calcular el Tiempo de Desarrollo vamos a utilizar la siguiente fórmula:

$$TDES=2.5*(EN)^{0.38}$$

Remplazando datos nos dará $TDES=2.5*(9)^{0.38}$ entonces el tiempo de desarrollo será:

$$TDES=6 \text{ meses para su desarrollo.}$$

$$COSTO=5*400=2400 \text{ US\$ costo estimado del Sistema.}$$

Factor Multiplicador	Razonamiento	Valor
Confiabilidad	Nominal	1
Base de Datos	Nominal	1
Tiempo	Nominal	1
Memoria	Alto	1.06
Máquina Virtual	Nominal	1
Tiempo de Retorno	Bajo	0.87
Analistas	Alto	0.86
Programadores	Alto	0.86
Experiencia	Nominal	1
Experiencia	Nominal	1
Experiencia	Nominal	1
Practica	Muy Alto	0.82
Herramientas	Alto	0.91
Calendario	Alto	1.04
Complejidad	Muy Alto	1.30
Factor de Ajuste		0.7

Tabla No. 6: Cuadro de estimaciones

Método Cocomo Intermedio

Desarrollo:

Con el fin de ajustar las estimaciones con respecto a desviaciones del valor inicial obtenido se utiliza el factor de ajuste el cual se calcula con los datos arriba descritos y el cual nos da: $FA=0.7$

Para calcular el esfuerzo nominal ajustado vamos a utilizar la siguiente formula.

$$PM=2.4*(MIFE)1.05 *FA$$

Remplazando datos nos dará $PM=2.4*(3,5)1.05*0.7$ entonces el esfuerzo nominal ajustado será:

$PM=6$ meses de programador.

Para calcular el tiempo de desarrollo ajustado vamos a utilizar la siguiente formula.

$$TDES=2.5*(PM)^{0.38}$$

Remplazando datos nos dará $TDES=2.5*(6)^{0.38}$ entonces el tiempo de desarrollo ajustado será:

$TDES=6$ meses para su desarrollo.

$COSTO=PM*costo\ mensual=6*400=2400\ US\$$ costo estimado del proyecto.

II.1.1.4.1.2. Presupuestos de Gastos

Para que el sistema logre los objetivos previstos anteriormente detallaremos el gasto que se realizara por cada mes del tiempo de desarrollo del sistema, como se realizó una estimación de costo del sistema y del tiempo que durara el sistema nos basaremos en esto para detallar el gasto del sistema por cada mes que dure el mismo.

En el desarrollo de dicho sistema consideraremos los siguientes recursos básicos como ser Costo Analista, Costo Programador, Costo Hardware y Costo Software a continuación describiremos cada uno de estos costos.

Costo Analista.- Este costo se refiere al costo que pagaremos por el servicio del analista a lo largo del desarrollo del sistema.

Costo Programador.- Este costo se refiere al costo que pagaremos por el servicio del programador a lo largo del desarrollo del sistema.

Costo Hardware.- Este costo se refiere al precio que pagaremos por el alquiler del Equipo como computadoras, impresoras y otros, en el tiempo que dure el desarrollo del sistema.

Costo Software.- Este costo se refiere al precio que pagaremos por las diferentes herramientas de software que ocuparemos en el desarrollo del proyecto.

MESES	COSTO HARDWARE	COSTO ANALISTA	COSTO PROGRAMADOR	COSTO SOFTWARE	COSTO TOTALES
1	50	200	-	0	250
2	70	200	-	0	270
3	70	100	100	0	270
4	70	100	400	0	570
5	70	50	400	0	520
6	70	50	400	0	520
TOTAL	400	700	1300	0	2400 U\$\$

Tabla N° 7: Presupuestos de Gastos

II.1.1.4.2. Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

II.1.1.4.2.1. Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar)

Fase	Nro. Iteraciones	Duración
Fase de Inicio	2	3 semanas
Fase de Elaboración	2	4 semanas
Fase de Construcción	2	5 semanas
Fase de Transición	-	

Tabla N° 8: Plan de Fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollarán los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente /usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una

	duración de una semana.
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 2.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla N° 9: Hitos

Ciclo de Vida

➤ Planificación

- o Se escriben los relatos de usuario (“user stories”) o requerimientos funcionales y no funcionales.
- o El plan de entregas crea el cronograma.
- o Entregas pequeñas muy frecuentes.
- o Se mide la velocidad del proyecto.
- o Se divide el proyecto en fases de iteración.
- o La planificación de iteración inicia cada fase.
- o Se obtiene los diagramas de caso de uso, para capturar los requerimientos del usuario.
- o Reuniones de pie, todo los días.

➤ **Diseño y Codificación**

- o Mantener la simplicidad en el diseño.
- o Elegir una metáfora de sistema, un conjunto de nombres ilustrativos de la realidad.
- o Realizar el diseño lógico y físico de datos.
- o Construir la base de datos.
- o Crear soluciones rápidas puntuales (“spike solutions”) para reducir el riesgo.
- o No agregar funcionalidad antes de tiempo.
- o Codificación.
- o El cliente siempre al alcance.
- o El código debe respetar las normas aceptadas.
- o No optimizar hasta el fin del proyecto.
- o
- o No trabajar horas extras.

➤ **Prueba**

- o Todo el código debe tener pruebas de caja blanca.
- o Todo el código debe pasar las pruebas de caja blanca antes de la entrega.
- o Crear pruebas toda vez que aparezcan errores.
- o Correr frecuentes pruebas de aceptación y publicar la puntuación.

II.1.1.4.2.2. Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en

paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

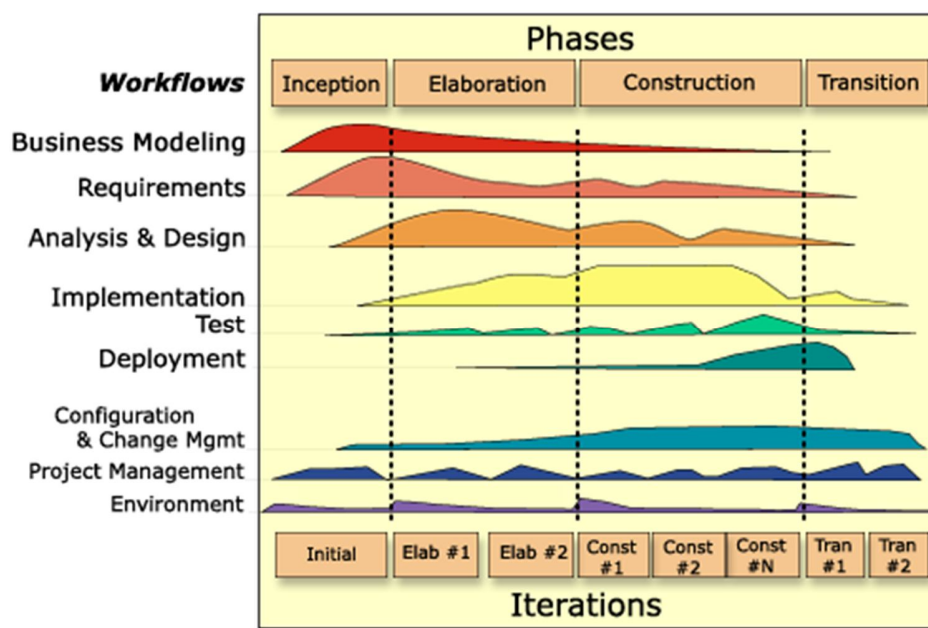


Figura 3: Fases del Proyecto

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

ACTIVIDADES	DURACION	COMIENZO	FIN
Recopilación de la Información	18 días		
Encontrar Actores y Casos de Uso	4 días	17/03/11	20/03/11
Detallar los casos de uso	6 días	21/03/11	28/03/11
Definir prototipo e interfaces	8 días	31/03/11	09/04/11
Análisis y Diseño del Sistema para el Sindicato	74 días		
Analizar los casos de uso	15 días	10/04/11	30/04/11
Analizar cada clase de análisis	15 días	01/05/11	21/05/11
Diseñar la Arquitectura	17 días	22/05/11	13/06/11
Diseñar casos de uso	10 días	16/06/11	27/06/11
Diseñar clases de Diseño	17 días	30/06/11	22/07/11
Programación e Implementación del Sistema	50 días		
Programación del Sistema	25 días	23/07/11	26/08/11
Implementar arquitectura	25 días	27/08/11	30/09/11
Realizar Pruebas al Sistema	14 días		30/04/11
Planificar y diseñar las pruebas de integración de los Sistemas	7 días	01/10/11	09/10/11
Realizar las pruebas de integración y de sistemas	7 días	10/10/11	20/10/11
Capacitación del Personal	21 días		
Diseñar una metodología para los talleres de capacitación en el manejo del sistema	9 días	21/10/11	31/10/11
Realizar Talleres de Capacitación en el Manejo del Sistema	12 días	03/11/11	18/11/11
Diseño e Implementación de la Red	12 días		
Diseño de la Red Local	5 días	19/11/11	15/11/11
Implementación de la Red Local	7 días	26/11/11	04/12/11

Tabla N° 10: Calendario del Proyecto

II.1.1.5. Referencias

- Documentación de Rational Unified Process, manuales de ayuda, tutoriales, etc.
- Diseño de aplicaciones de escritorio usando UML, por Miguel Reynolds.
- UML y Patrones Craig Larman
- Visual Modeling with Rational Rose and UML, Terry Quantani – Addison-Wesley.

II.1.2. REQUERIMIENTOS FUNCIONALES

II.1.2.1. Introducción

En este apartado se describen los requerimientos funcionales del sistema, la funcionalidad o los servicios que se espera que éste provea.

II.1.2.1.1. Descripción Textual de los Requerimientos Funcionales

F1) Gestión de Personal.

Req(04) Listado de Personal. Prioridad: Alta. El sistema deberá permitir el listado de todo el personal debidamente registrados.

Req(05) Añadir Personal. Prioridad: Alta. El sistema deberá registrar o añadir personal.

Req(06) Modificar Personal. Prioridad: Alta. El sistema deberá permitir la modificación de los diferentes personal.

Req(07) Eliminación o baja de Personal. Prioridad: Alta. El sistema deberá permitir la baja de personal.

F2) Gestión de Cliente.

Req(08) Listado de Cliente. Prioridad: Alta. El sistema deberá permitir el listado de todos los clientes debidamente registrados.

Req(09) Añadir Cliente. Prioridad: Alta. El sistema deberá registrar o añadir cliente.

Req(10) Modificar Cliente. Prioridad: Alta. El sistema deberá permitir la modificación de los diferentes clientes.

Req(11) Eliminación o baja de Personal. Prioridad: Alta. El sistema deberá permitir la baja de clientes.

F3) Gestión Productos.

Req(12) Listado de Productos. Prioridad: Alta. El sistema deberá permitir el listado de todos los productos debidamente registrados.

Req(13) Añadir Producto. Prioridad: Alta. El sistema deberá registrar o añadir productos.

Req(14) Modificar Producto. Prioridad: Alta. El sistema deberá permitir la modificación de los diferentes productos.

Req(15) Eliminación o baja de Producto. Prioridad: Alta. El sistema deberá permitir la baja de productos.

F4) Gestión Ventas.

Req(16) Listado de Ventas. Prioridad: Alta. El sistema deberá permitir el listado de todas las ventas debidamente registradas.

Req(17) Agregar Venta. Prioridad: Alta. El sistema deberá registrar o añadir ventas.

Req(18) Modificar Venta. Prioridad: Alta. El sistema deberá permitir la modificación de datos de la venta.

Req(19) Eliminación o baja de Venta. Prioridad: Alta. El sistema deberá permitir la baja de la venta.

F5) Gestión Entrada de Productos.

Req(20) Listado de Entrada de Productos. Prioridad: Alta. El sistema deberá permitir el listado de todos los productos que entran a la empresa.

Req(21) Añadir Entrada de Productos. Prioridad: Alta. El sistema deberá registrar o añadir productos.

Req(22) Modificar Entrada de Productos. Prioridad: Alta. El sistema deberá permitir la modificación de los diferentes productos.

Req(23) Eliminación o baja de Entrada de Productos. Prioridad: Alta. El sistema deberá permitir la baja de productos.

F6) Ver plan de pagos.

Req(24) Listado de Plan de Pagos. Prioridad: Alta. El sistema permitirá el listado de planes de pago del cliente el estado actual de su cuenta.

Req(25) Controlar Plan de Pagos. Prioridad: Alta. El sistema permitirá ver el estado actual de su plan de pago del cliente.

Req(26) Imprimir Plan de Pagos. Prioridad: Alta. El sistema permitirá la opción de impresión del plan de pagos del cliente.

F7) Gestionar Reservas.

Req(27) Reserva. Prioridad: Alta. El sistema permitirá al cliente la selección de productos para su respectiva reserva.

Req(28) Agregar Reserva. Prioridad: Alta. El sistema deberá registrar o añadir reserva realizada por el cliente.

Req(29) Eliminar Reserva. Prioridad: Alta. El sistema permitirá dar de baja a la reserva realizada por el cliente.

F8) Gestionar Cotizaciones.

Req(30) Cotización: Alta. El sistema permitirá llenar un formulario al cliente con todas las características del producto que busca y se le brindara la cotización del producto tanto por unidad o por la cantidad del producto que desea.

Req(31) Imprimir Cotización. Prioridad: Alta. El sistema permitirá la opción de impresión de la cotización realizada.

F9) Administrar Publicaciones.

Req(32) Administrar Publicaciones. Prioridad: Alta. El sistema permitirá al administrador poder publicar promociones de productos, descuentos de productos y avisos que requiera dar a conocer la empresa.

Req(33) Adicionar Publicaciones. Prioridad: Alta. El sistema deberá registrar o añadir publicaciones .

Req(34) Modificar Publicación. Prioridad: Alta. El sistema deberá permitir la modificación de datos de la publicación.

Req(35) Cambiar de Estado Publicación. Prioridad: Alta. El sistema deberá permitir la baja de la publicación.

II.1.3. GLOSARIO

II.1.3.1. Introducción

Este documento recoge términos manejados durante la elaboración del proyecto de desarrollo del sistema. Se trata de un diccionario informal de datos y de definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para el proyecto.

II.1.3.1.1. Propósito

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo.

II.1.3.1.2. Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos. De tal modo que la terminología empleada, se refleja con claridad en este documento.

II.1.3.2. Referencias

El presente glosario hace referencia a los siguientes documentos:

- Documento Plan de Desarrollo Software del Proyecto.
- Documento Visión del Proyecto.

Documentos de Especificación de Casos de Uso del Proyecto.

II.1.3.3. Organización del Glosario

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional de la lengua español.

II.1.3.4. Definiciones

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto de desarrollo.

Administrador

Persona encargada de dirigir el negocio realizando la programación y control de diferentes opciones de ventas.

Apache

Servidor HTTP de dominio público basado en el sistema operativo Linux. Apache fue desarrollado en 1995 y es actualmente uno de los servidores HTTP más utilizados en la red.

Aplicación

Un programa informático que lleva a cabo una función con el objeto de ayudar a un usuario a realizar una determinada actividad. WWW, FTP, correo electrónico y Telnet son ejemplos de aplicaciones en el ámbito de Internet.

Cliente – Servidor

Modelo de comunicación entre ordenadores conectados a una red en el cual hay uno, llamado cliente, que satisface las peticiones realizadas por otro llamado servidor.

Usuario

El usuario es aquella persona y sea perteneciente o no a la Empresa el cual puede realizar varias acciones en el Sistema y poder visualizar varias páginas. El usuario representa uno de los agentes con los que interactúa con el Sistema con ciertos privilegios.

Código Fuente

Conjunto de instrucciones que componen un programa informático. Estos programas se escriben en determinados lenguajes; el lenguaje que se utiliza para elaborar una página web, que puede considerarse en cierto sentido un programa, es el HTML.

Computadora

Máquina electrónica capaz de procesar información siguiendo instrucciones almacenadas en programas. Antes que electrónicas estas máquinas fueron mecánicas o electromecánicas.

Correo

Envío de mensajes de correo electrónico en cuya cabecera no aparece la identificación del remitente del mensaje. Suele realizarse a través de servidores especializados, que eliminan dicha información. Dichos servidores han tenido, y aún tienen, problemas legales en algunos países, pero, dejando a un lado las cuestiones legales, esta modalidad de correo electrónico es también discutible desde el punto de vista ético, como lo es en general la correspondencia anónima, dentro y fuera de la red.

Hardware

Conjunto de los componentes que integran la parte material para el funcionamiento de una computadora.

HTML (Lenguaje de Marcado de Hipertexto)

Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.

HTTP (Protocolo de Transferencia de Hipertexto)

Protocolo usado para la transferencia de documentos WWW.

Internet

Red de telecomunicaciones nacida en 1969 en los EE.UU. a la cual están conectadas centenares de millones de personas, organismos y empresas en todo el mundo, mayoritariamente en los países más desarrollados, y cuyo rápido desarrollo está teniendo importantes efectos sociales, económicos y culturales, convirtiéndose de esta manera en uno de los medios más influyentes de la llamada Sociedad de la Información y en la Autopista de la Información por excelencia. Fue conocida como ARPANET hasta 1974.

Requerimientos funcionales

Se refiere a la funcionalidad o los servicios que se espera que el sistema provea.

Requerimientos no funcionales

Los requerimientos no funcionales tienen que ver con las características que de una u otra forma puedan limitar el sistema como son: el rendimiento (en tiempo y espacio), confiabilidad, interfaces, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

Rol

Asignación de un cargo a los usuarios del negocio.

RUP

Rational Unified Process. Primero se ve en que principios se basa, luego se trata su estructura desde dos puntos de vista: las cuatro fases y los nueve flujos de trabajo. El Proceso Unificado de Rational es un proceso de ingeniería del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles.

En definitiva el RUP es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos software. Además Rational proporciona herramientas para todos los pasos del desarrollo así como documentación en línea para sus clientes.

Servidor Web

Un servidor Web es el programa, que se administra a través de la computadora, que maneja los dominios y páginas Web, interpretando lenguajes como html, servlet y spring, entre otros. Ejemplos: Apache y Microsoft IIS.

Software

Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

TIC

Las **tecnologías de la información y la comunicación** (TIC o bien NTIC para *Nuevas Tecnologías de la Información y de la Comunicación* o IT para *Information Technology*) agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, internet y telecomunicaciones.

Técnicas usadas para conseguir, recibir, adquirir, procesar, guardar y diseminar información numérica, textual, pictórica, audible, visible (multimedia) a través de accesorios o dispositivos basados en combinación de la microelectrónica, la computación y las telecomunicaciones.

Web Site

Conjunto de páginas Web que comparten un mismo tema e intención y que generalmente se encuentra en un sólo servidor. Punto de la red con una dirección única y al que pueden acceder los usuarios para obtener información.

II.1.3.5. Estereotipos UML

Actor del Negocio	
Caso de Uso del Negocio	
Comunicación	
Relación	

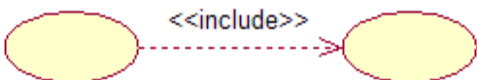
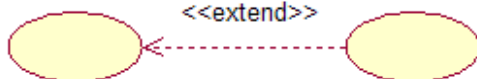
Trabajador del Negocio	
Entidad del Negocio	
Actor	
Caso de Uso	
Relación de Inclusión	
Relación de Extensión	

Tabla N° 11: Estereotipos de UML

II.1.4. VISIÓN

II.1.4.1. Introducción

II.1.4.1.1. Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema web informativo para el apoyo a la difusión de Información sobre la empresa DIGITAL CORPORATION. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

II.1.4.1.2. Alcance

El documento visión se ocupa del sistema web de gestión e informativo para el apoyo de los procesos administrativos y difusión de Información sobre la Empresa DIGITAL CORPORATION. Dicho sistema será desarrollado por el Universitario: Iván Ayala Catarí. Alumno de quinto año de la Carrera de Ingeniería Informática, Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho (U. A. J. M. S.) Tarija.

II.1.4.2. Definiciones Acrónicas

- RUP: Son las siglas de Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.
- UML: Son las siglas de Lenguaje Unificado de Modelación. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

II.1.4.3. Referencias

- Glosario.
- Plan de desarrollo del software.
- RUP (Racional Unified Process).
- Diagrama de casos de uso.
- Modelo de negocio.

II.1.5. MODELO DE CASOS DE USO DEL NEGOCIO

II.1.5.1. Introducción

El modelado de Casos de Uso del Negocio se basa en dos diagramas principales, el modelo de casos de uso del negocio y los modelos de objetos del negocio.

II.1.5.2. Propósito

- Representar la funcionalidad provista de la organización como un todo.
- Comprender mejor el funcionamiento de la organización

II.1.5.3. Alcance

- Identificar los objetos de Negocio
- Describe los procesos del Negocio

II.1.5.4. Diagrama de casos de uso del negocio

Modelo de Casos de Uso del Negocio del Encargado de ventas

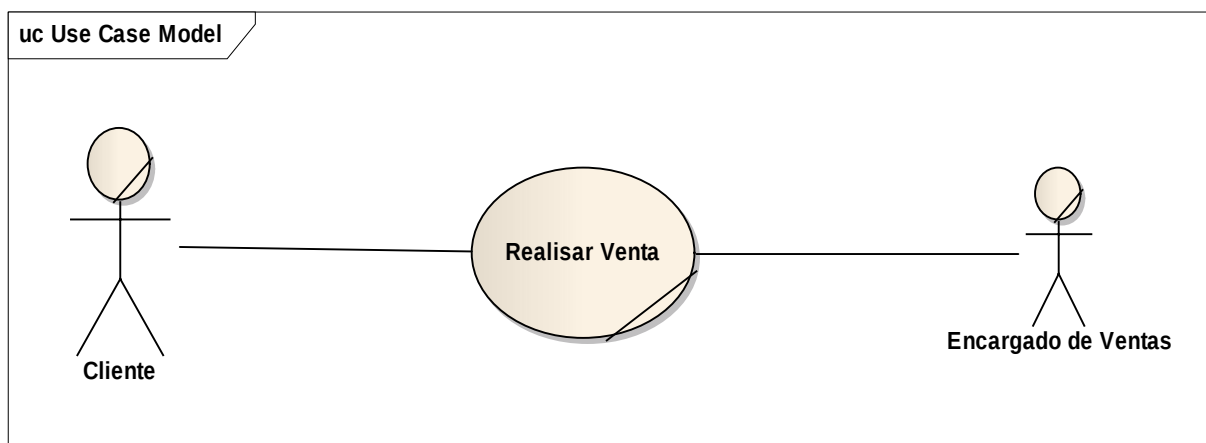


Figura N° 4: Modelo de Casos de Uso del Encargado de ventas

Modelo de Casos de Uso del Negocio del Administrador

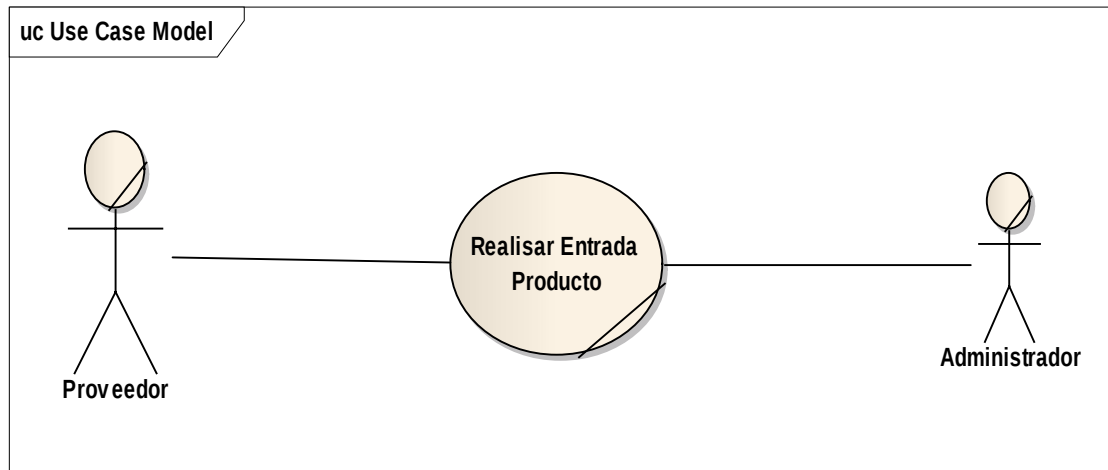


Figura 5: Modelos de Casos de Uso del Negocio del Administrador

II.1.5.5. Descripción de los actores del negocio

- **Encargado de Ventas:**

Es el que el que registra o realiza las ventas al cliente, las salidas de los productos.

- **Administrador:**

Es el encargado de administrar a la persona que trabaja en la tienda, recibe el reporte mensual sobre las ventas.

II.1.5.6. Descripción de los casos de uso del negocio

II.1.5.6.1. Realizar Venta

Esto sucede cuando la el encargado de ventas realiza el registra de un venta.

II.1.5.6.2. Registro Entrada Producto

Esto sucede cuando el administrador pide a sus proveedores el producto requerido.

II.1.6. MODELO DE OBJETOS DEL NEGOCIO

II.1.6.1. Introducción

El modelado del negocio se basa en los diagramas de modelos de objetos del negocio.

II.1.6.2. Propósito

Comprender la Estructura dinámica de los Casos de Uso de Negocio.

II.1.6.3. Alcance

- Describe los procesos del negocio
- Identificar y definir los objetos de negocio.

II.1.6.4. Modelo de Objeto del Negocio

II.1.6.4.1. Realizar Venta

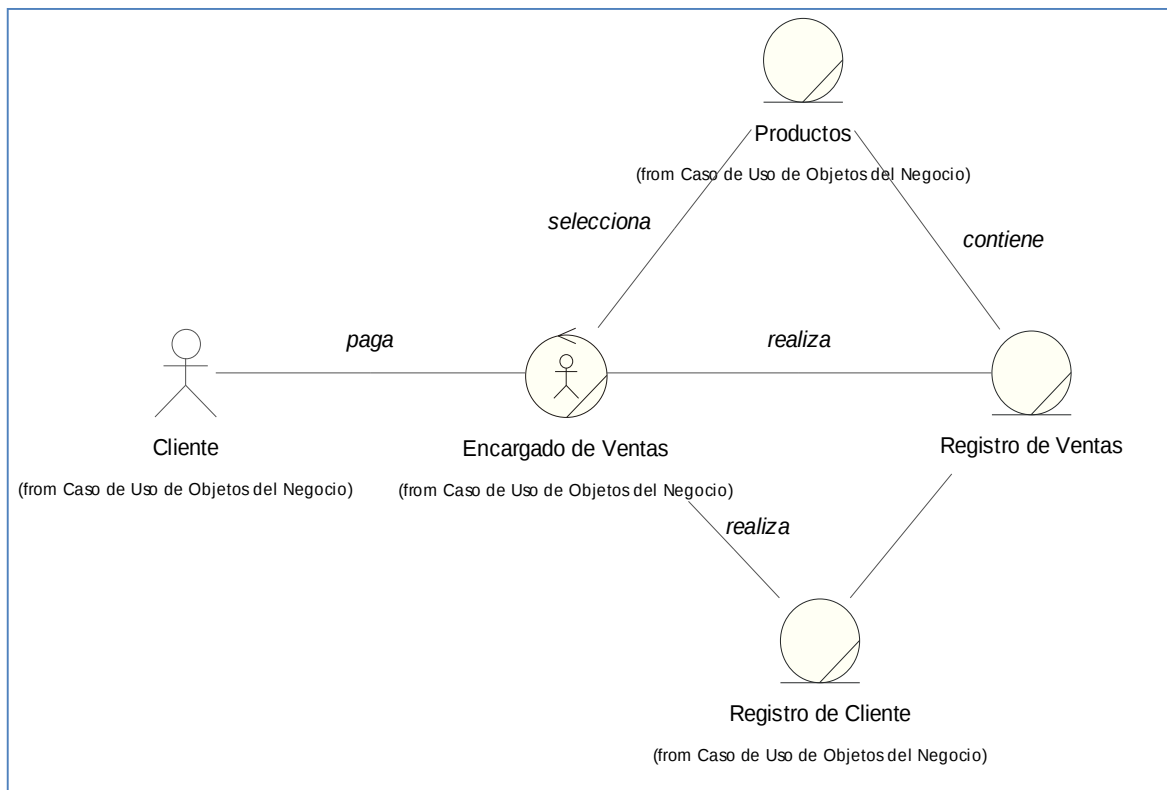


Figura N° 6: Realizar Venta

II.1.6.4.2. Realizar Entrada Producto

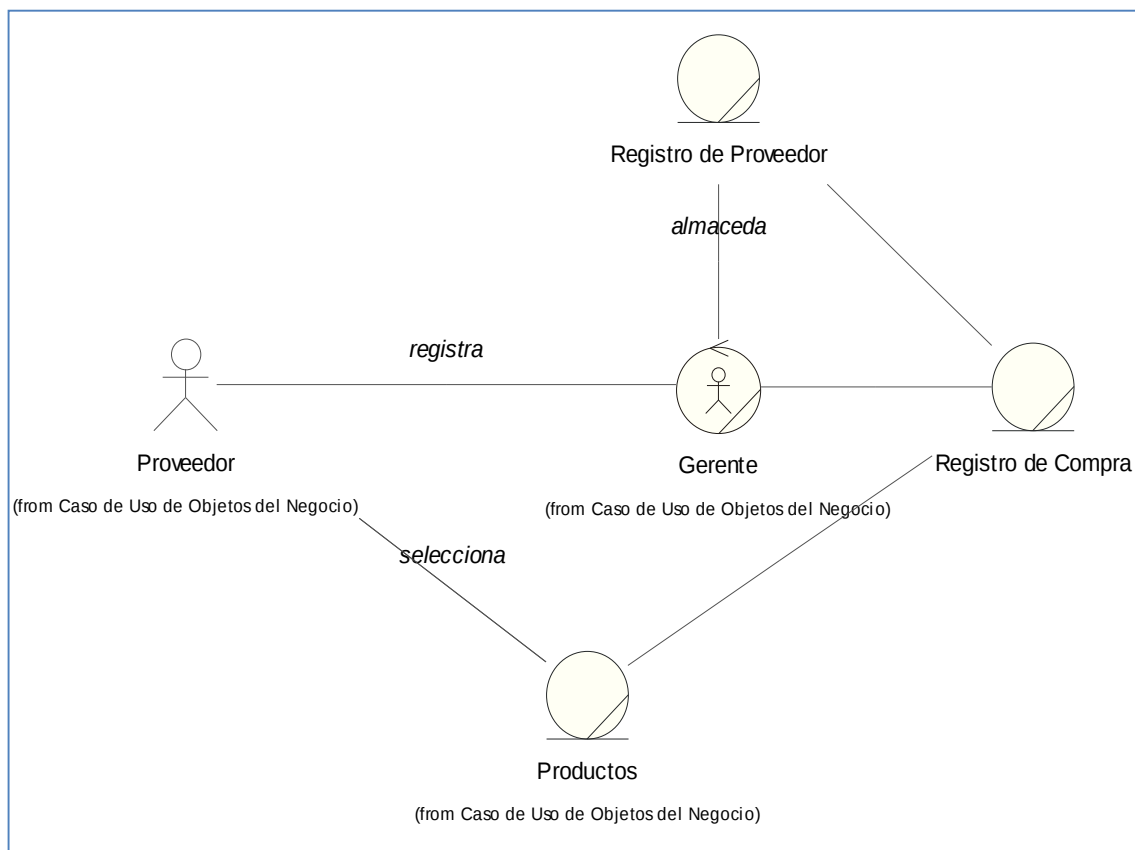


Figura N° 7: Realizar Entrada Producto

II.1.6.5. MODELO DE CASOS DE USO

II.1.6.5.1. Introducción

Un diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuario u otras aplicaciones). Es una herramienta esencial para la captura de requerimientos y para la planificación y control de un proyecto interactivo.

II.1.6.5.2. Propósito

- Comprender la estructura y la dinámica del sistema deseado para la organización.
- Identificar posibles mejoras.

II.1.6.5.3. Alcance

- Describe los procesos de sistema y los usuarios.
- Identificar y definir los **procesos del sistema** según los **objetivos** de la organización.
- Definir un **caso de uso** para cada proceso del sistema (el *diagrama de casos de uso* puede mostrar el contexto y los límites de la organización).

II.1.6.6. Diagrama de Casos de Uso

Caso de Uso: Administrar el Sistema

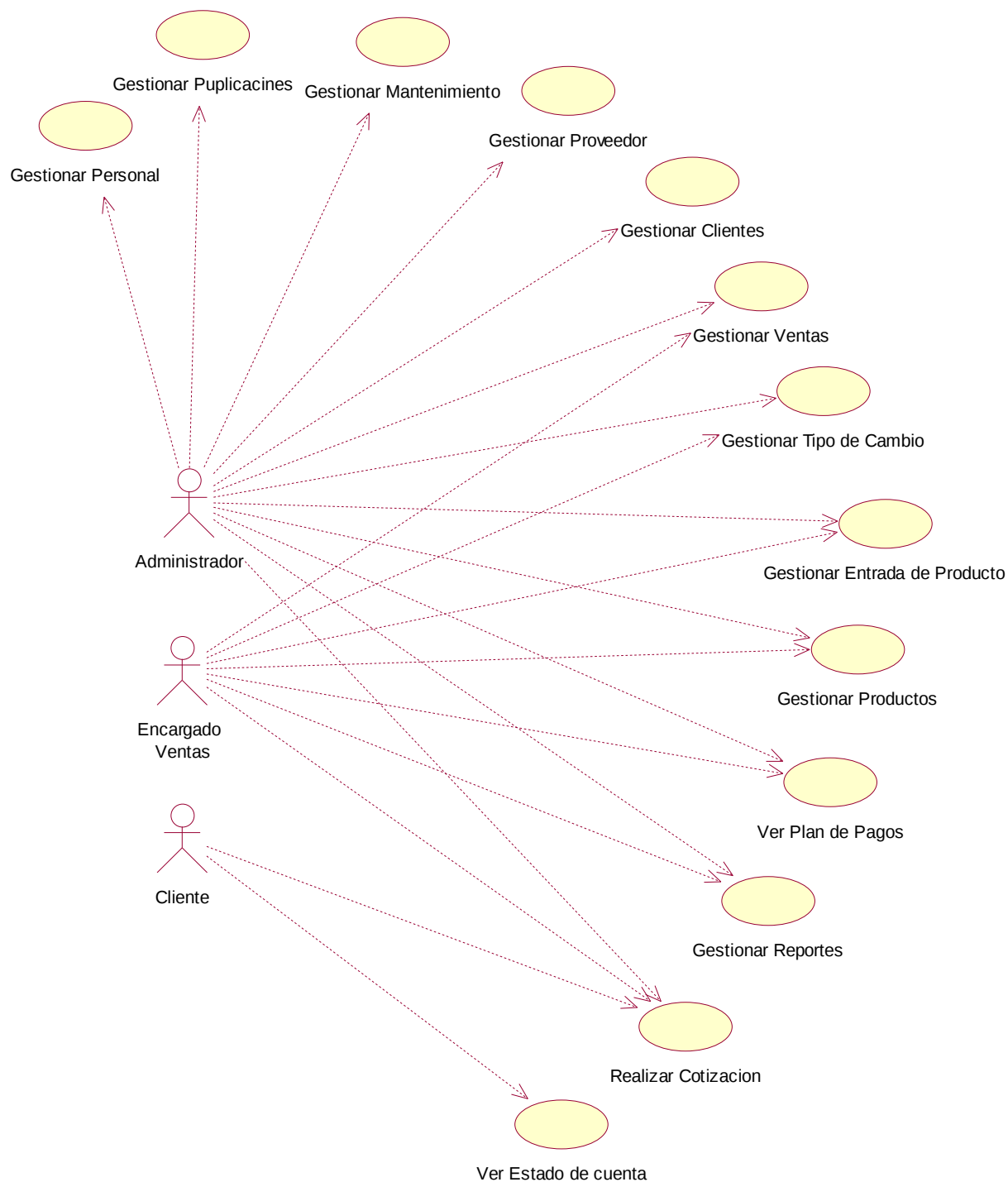
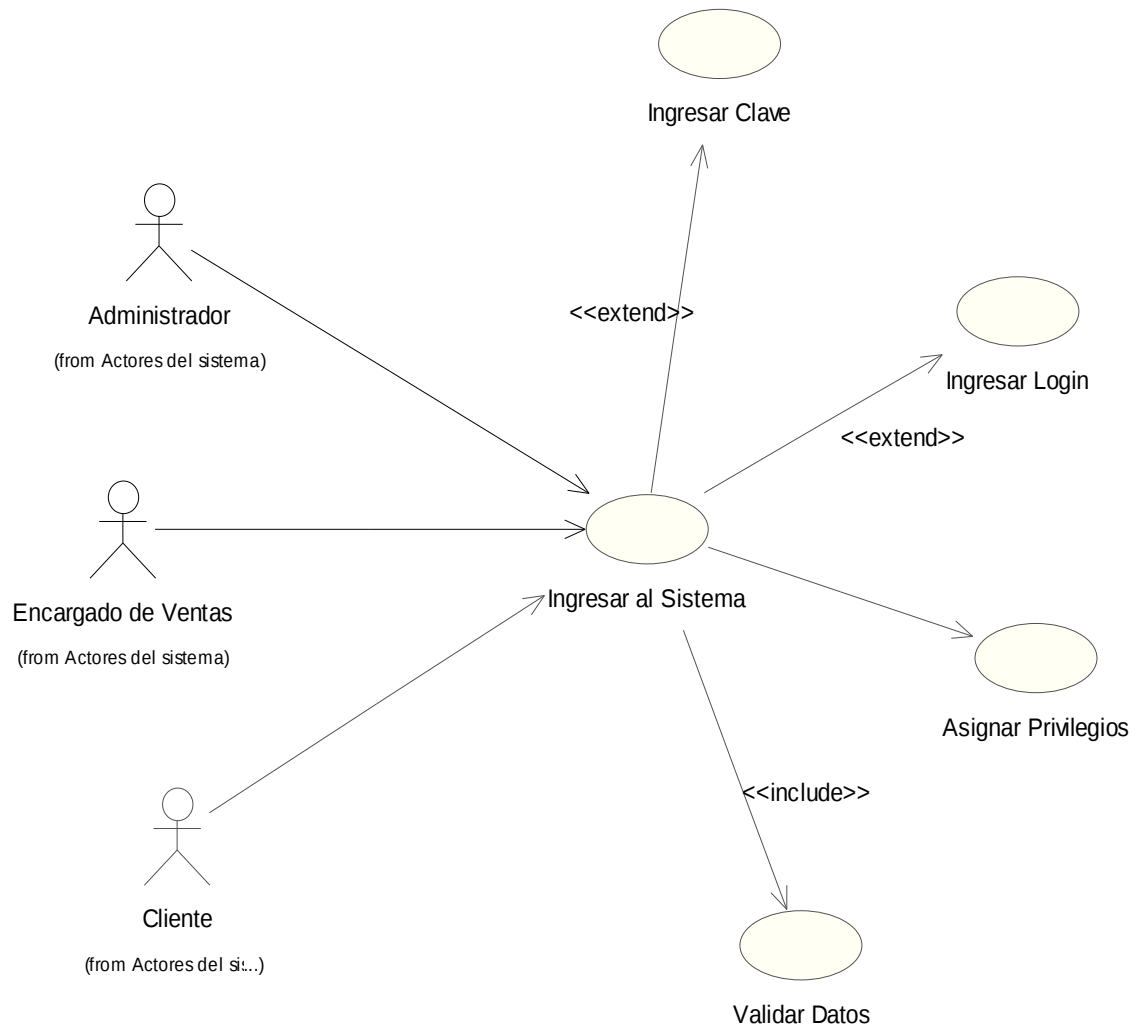
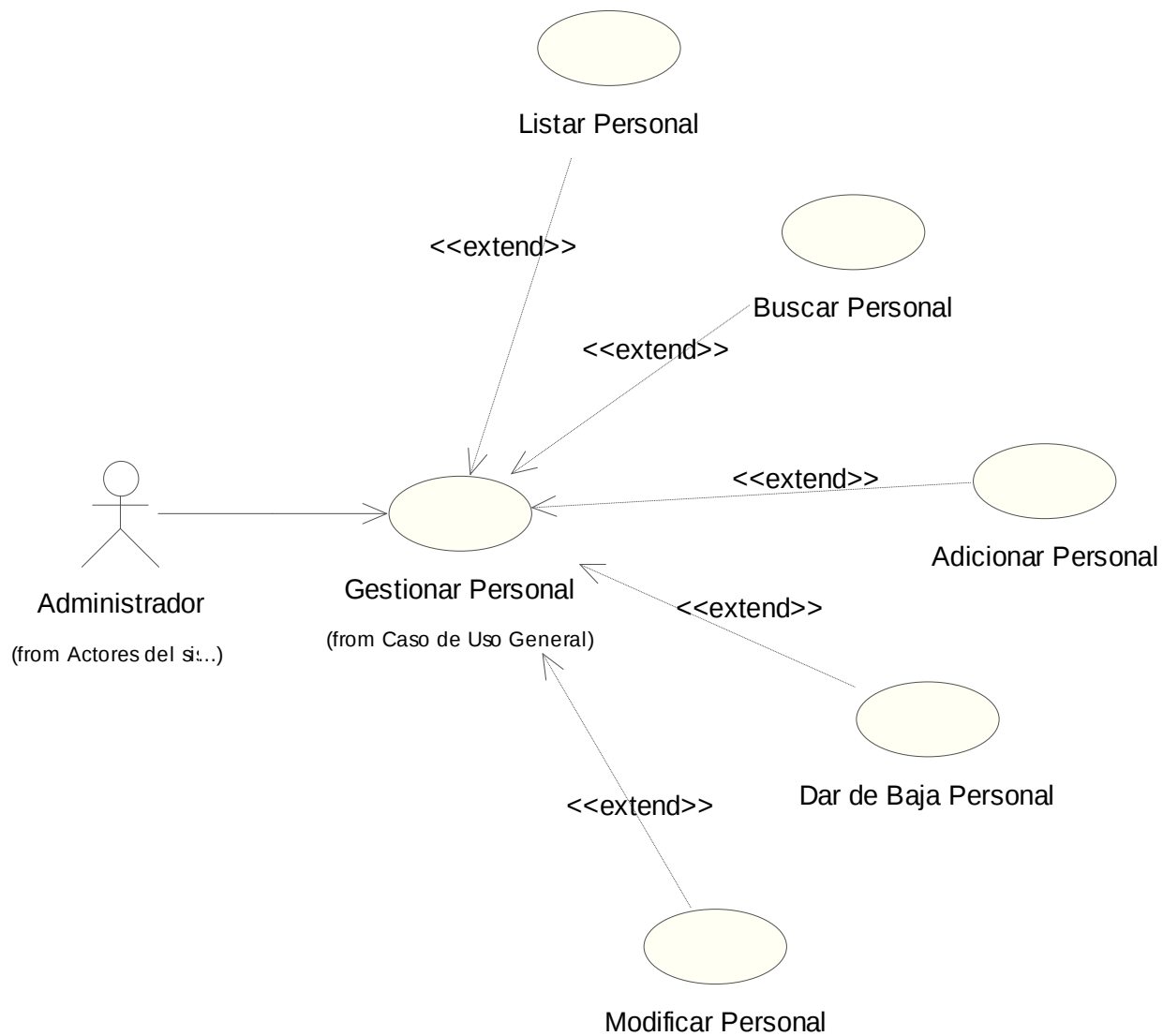
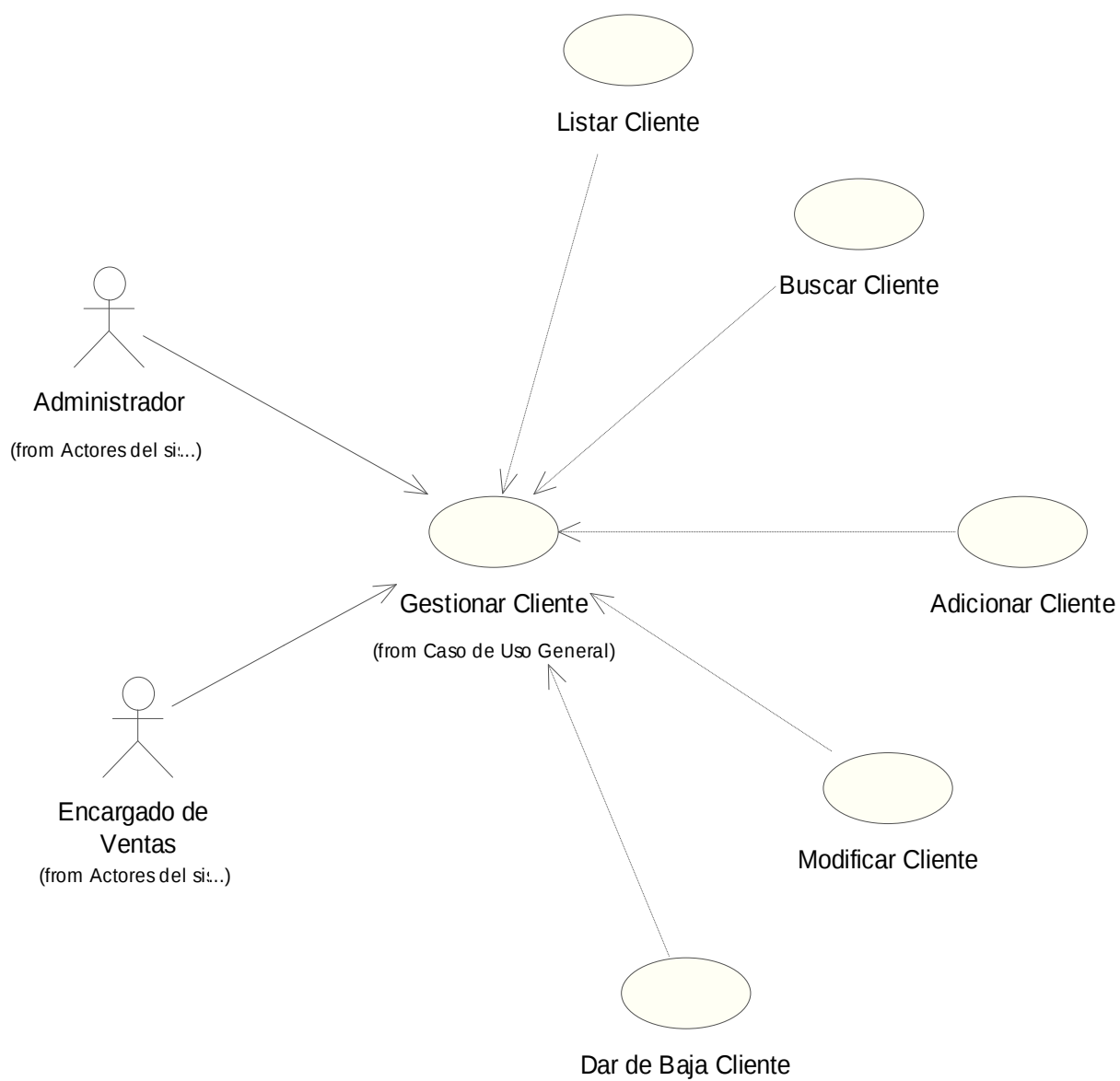


Figura N° 8: Administrar el Sistema

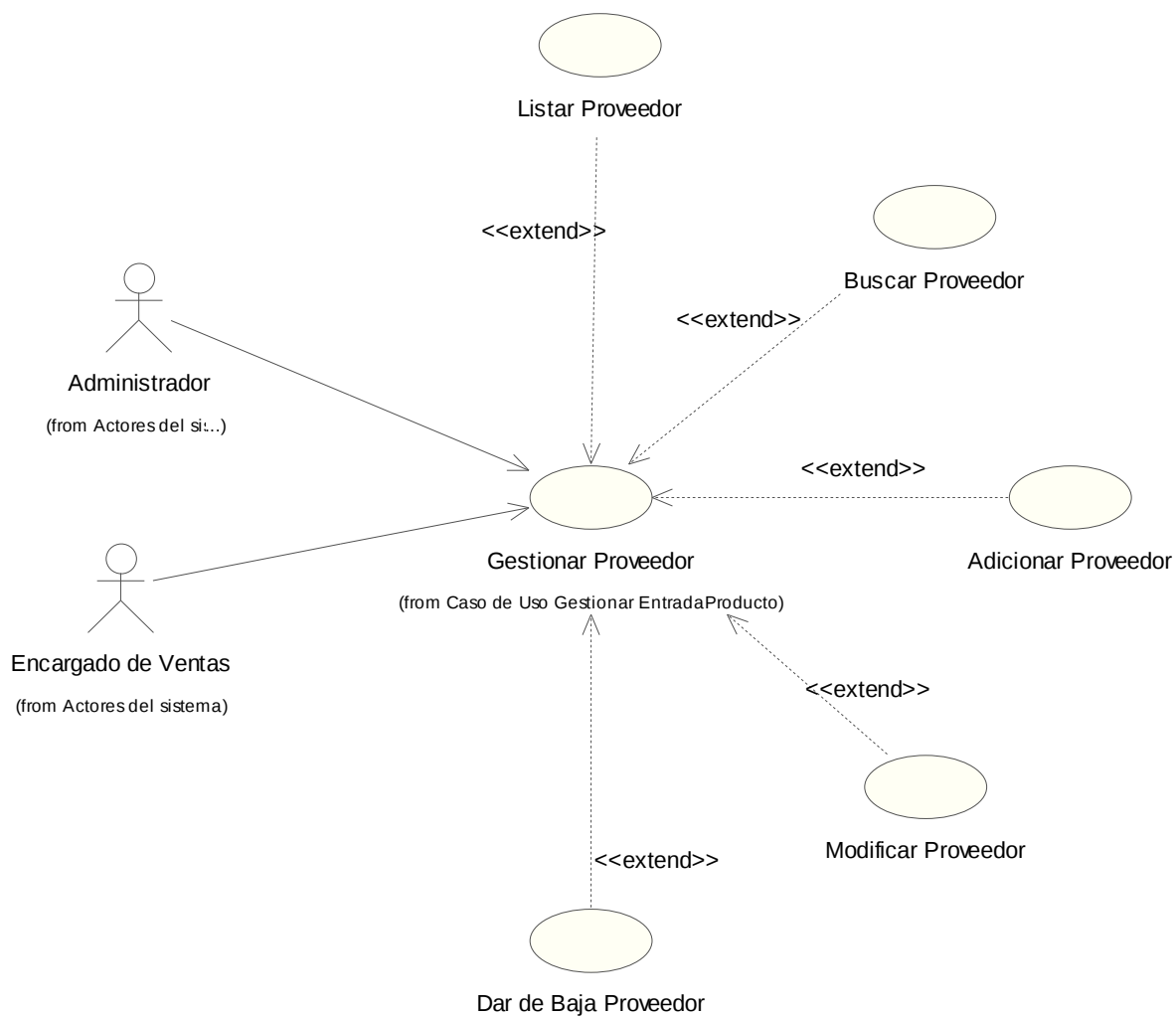
CASO DE USO: Ingresar al Sistema**Figura N° 9: Ingresar al Sistema**

CASOS DE USO ESPECÍFICOS**1 Caso De Uso: Gestionar Personal****Figura N° 10:** Caso De Uso: Gestionar Personal

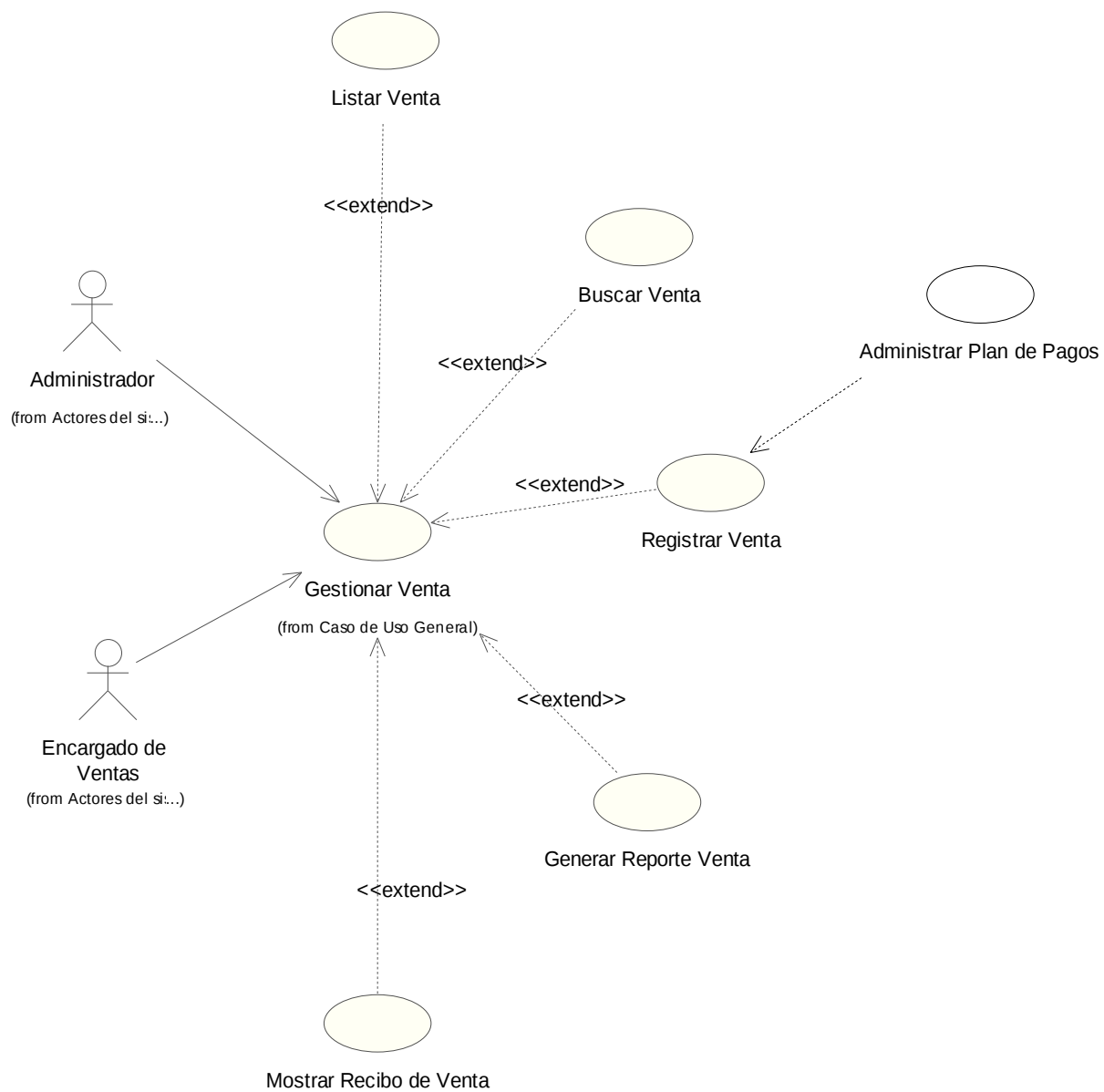
2 Caso De Uso: Gestionar Cliente

**Figura N° 11:** Caso De Uso: Gestionar Cliente

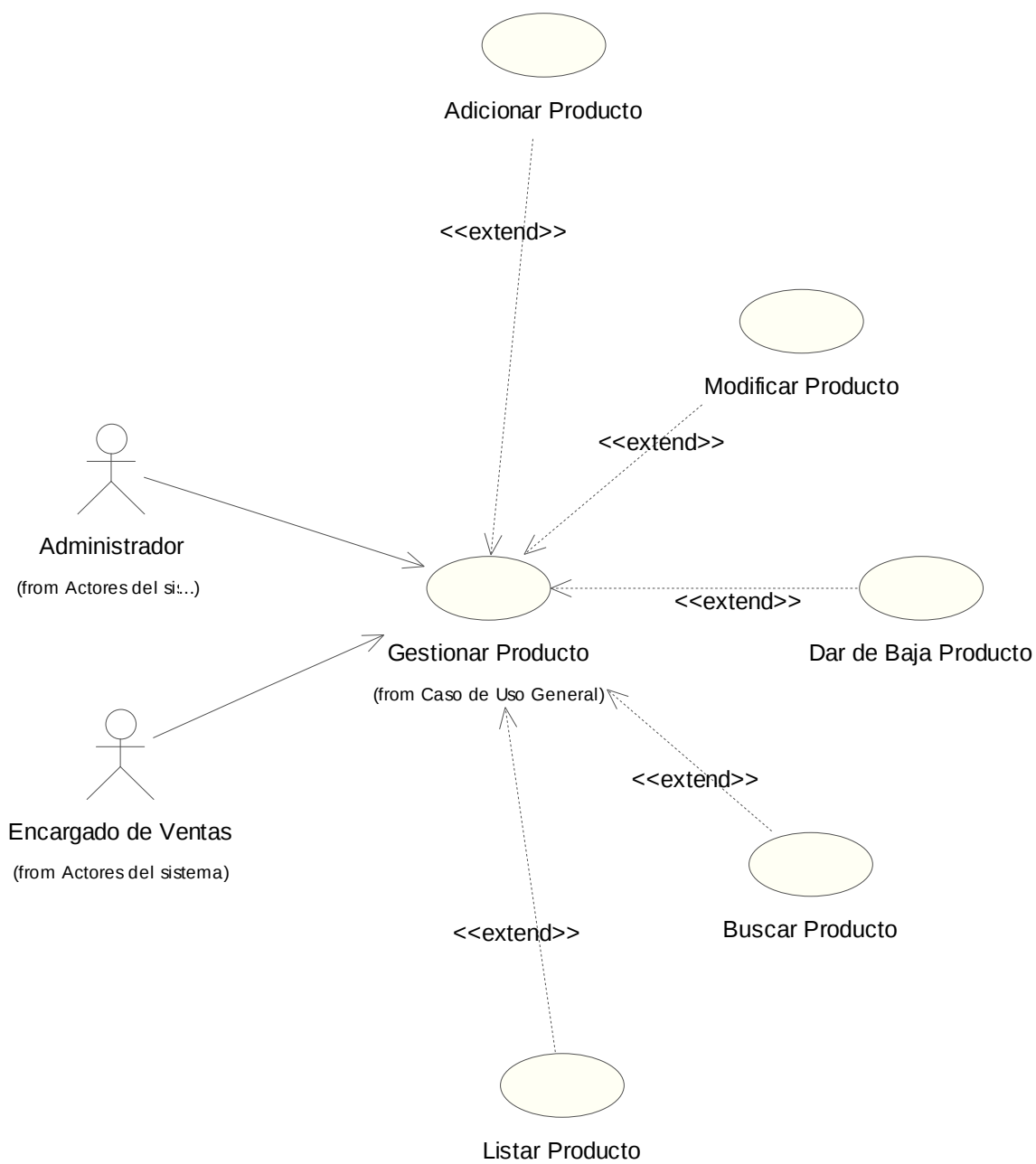
3 Caso De Uso: Gestionar Proveedor

**Figura N° 12:** Caso de Uso: gestionar Proveedor

4 Caso De Uso: Gestionar Ventas

**Figura N° 13:** caso de Uso: Gestionar Ventas

5 Caso De Uso: Gestionar Productos

**Figura N° 14:** Gestionar Productos

6 Caso De Uso: Ver Plan de Pagos

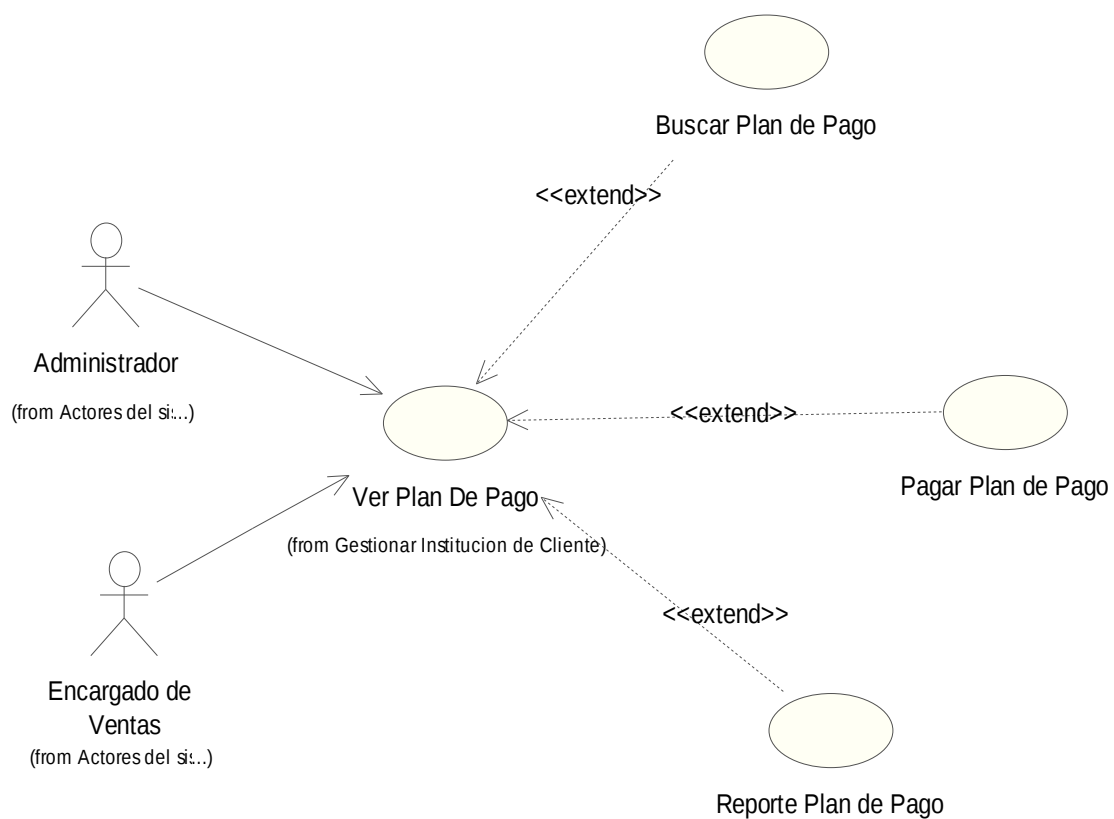


Figura N° 15: Ver Plan de Pagos

7. Caso De Uso: Ver Estado de Cuenta

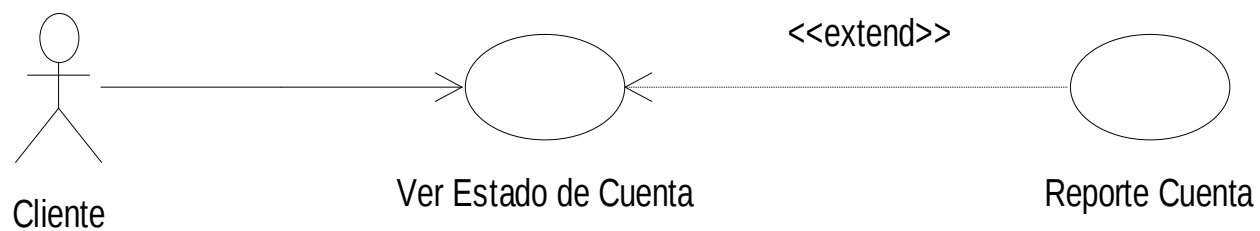
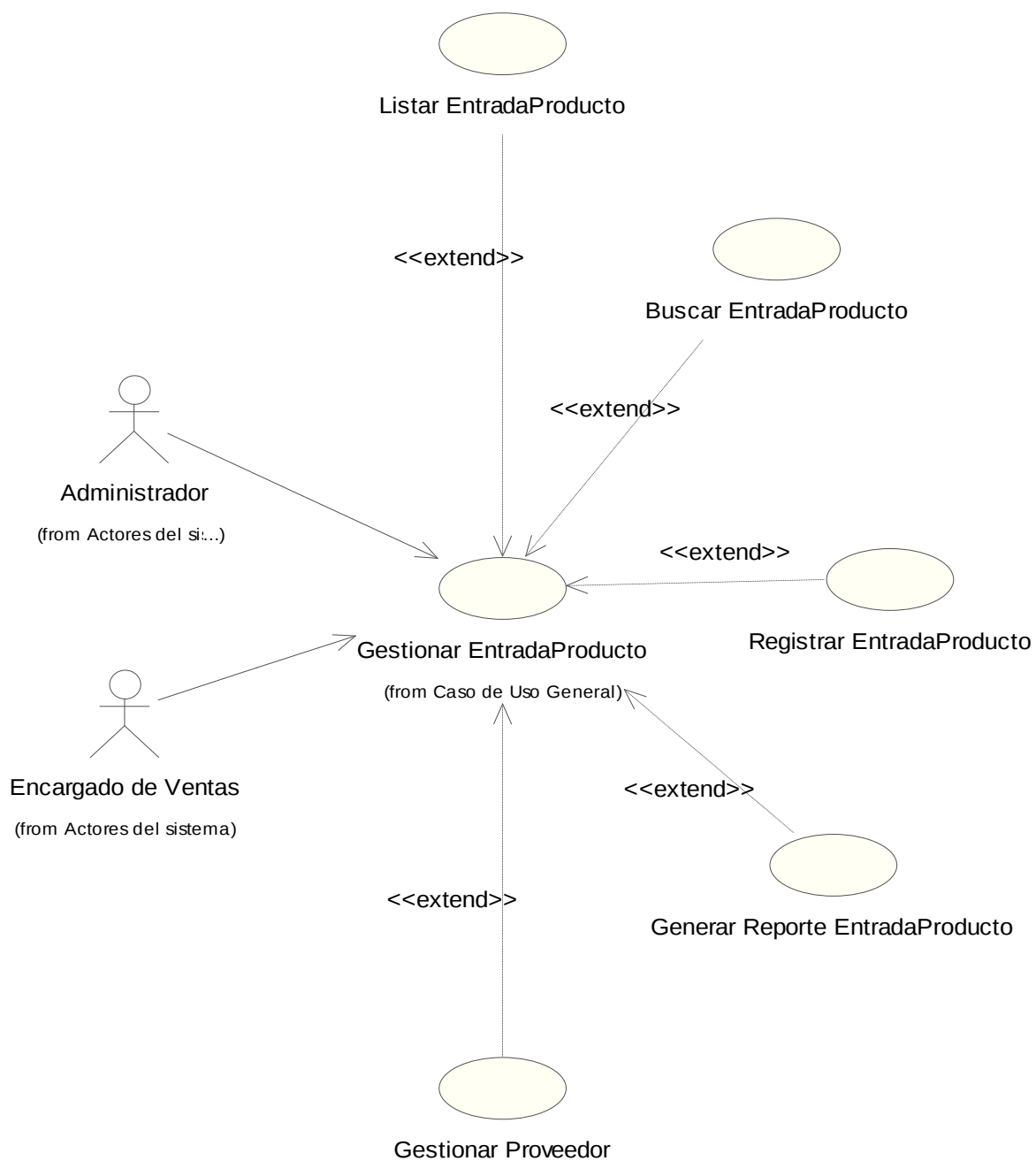
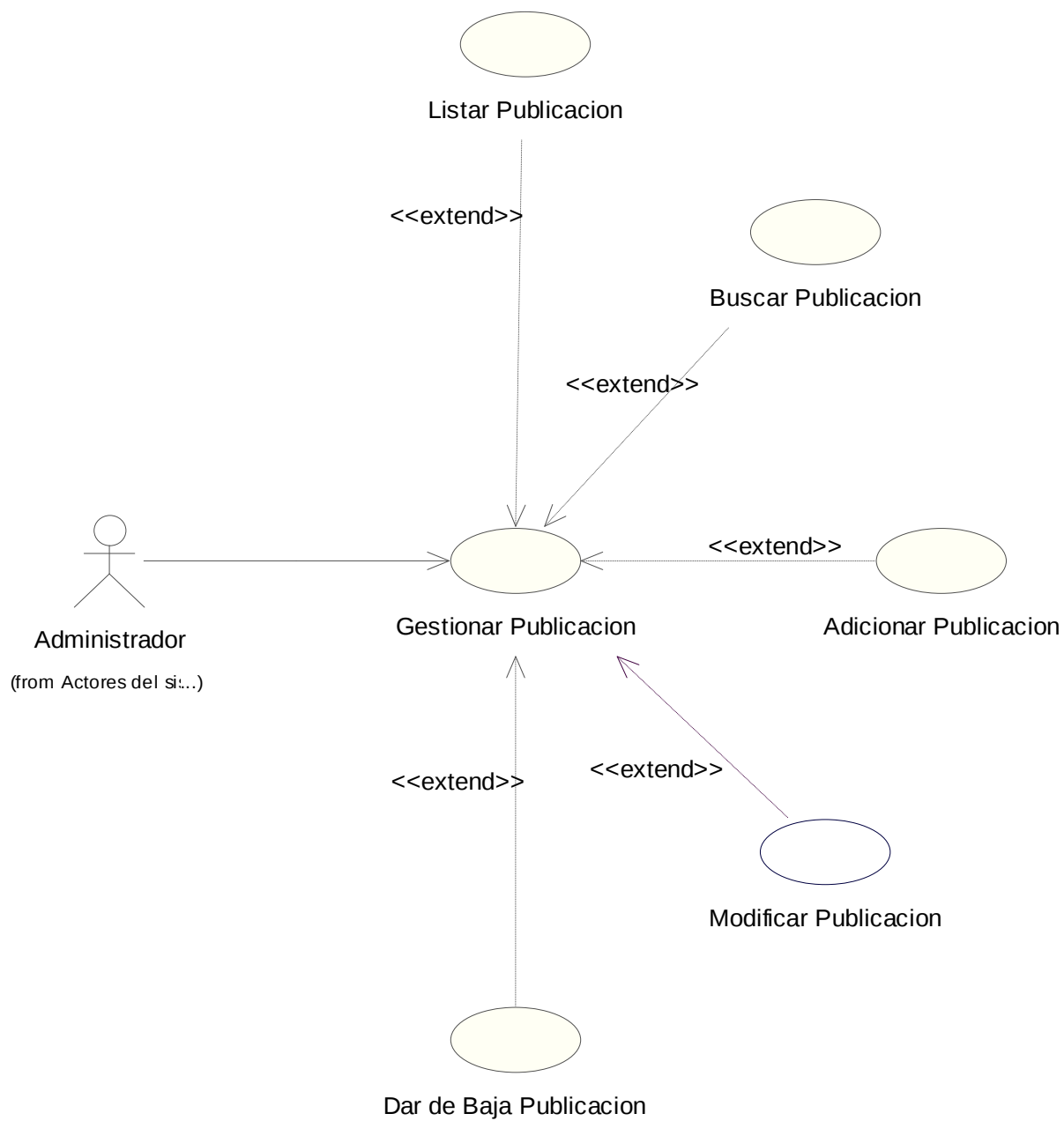


Figura N° 16: Ver Estado de Cuenta

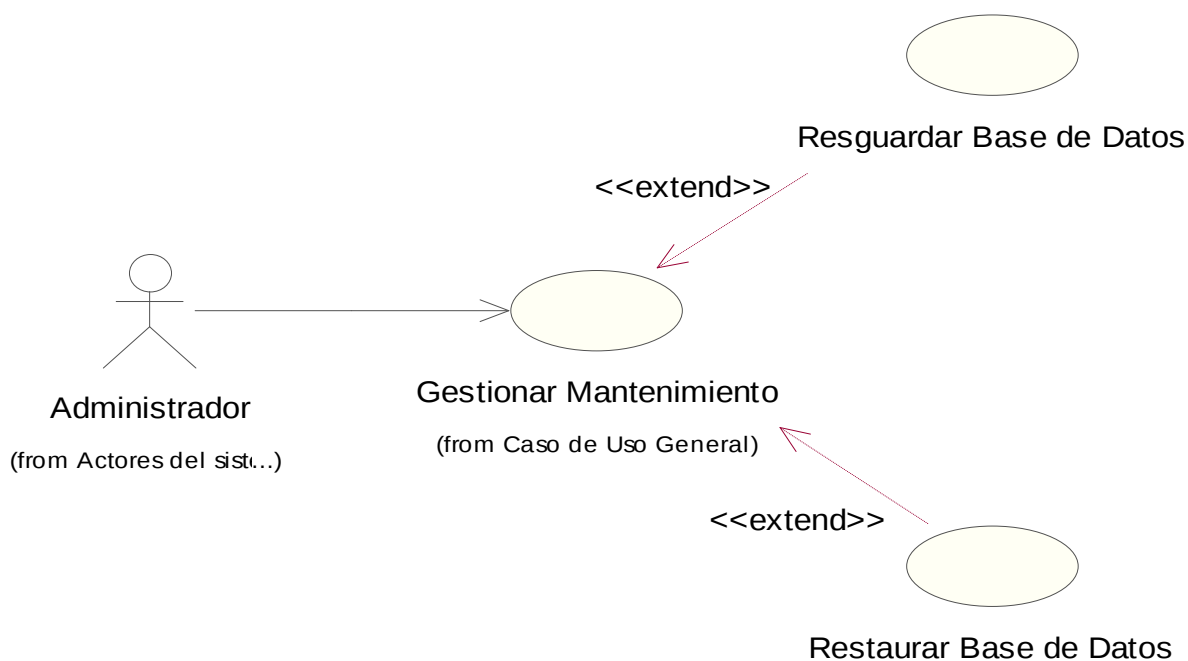
8. Caso De Uso: Gestionar Entrada de Producto

**Figura N° 18:** Gestionar Entrada de Producto

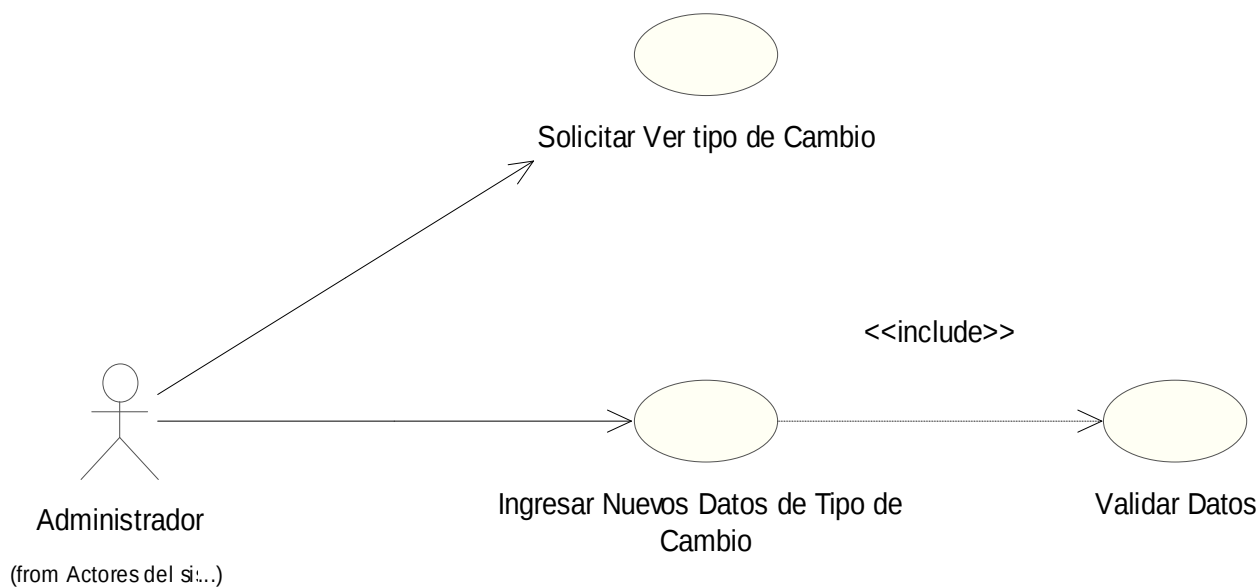
9. Caso de Uso: Administrar Publicaciones

**Figura N° 18:** Administrar Publicaciones

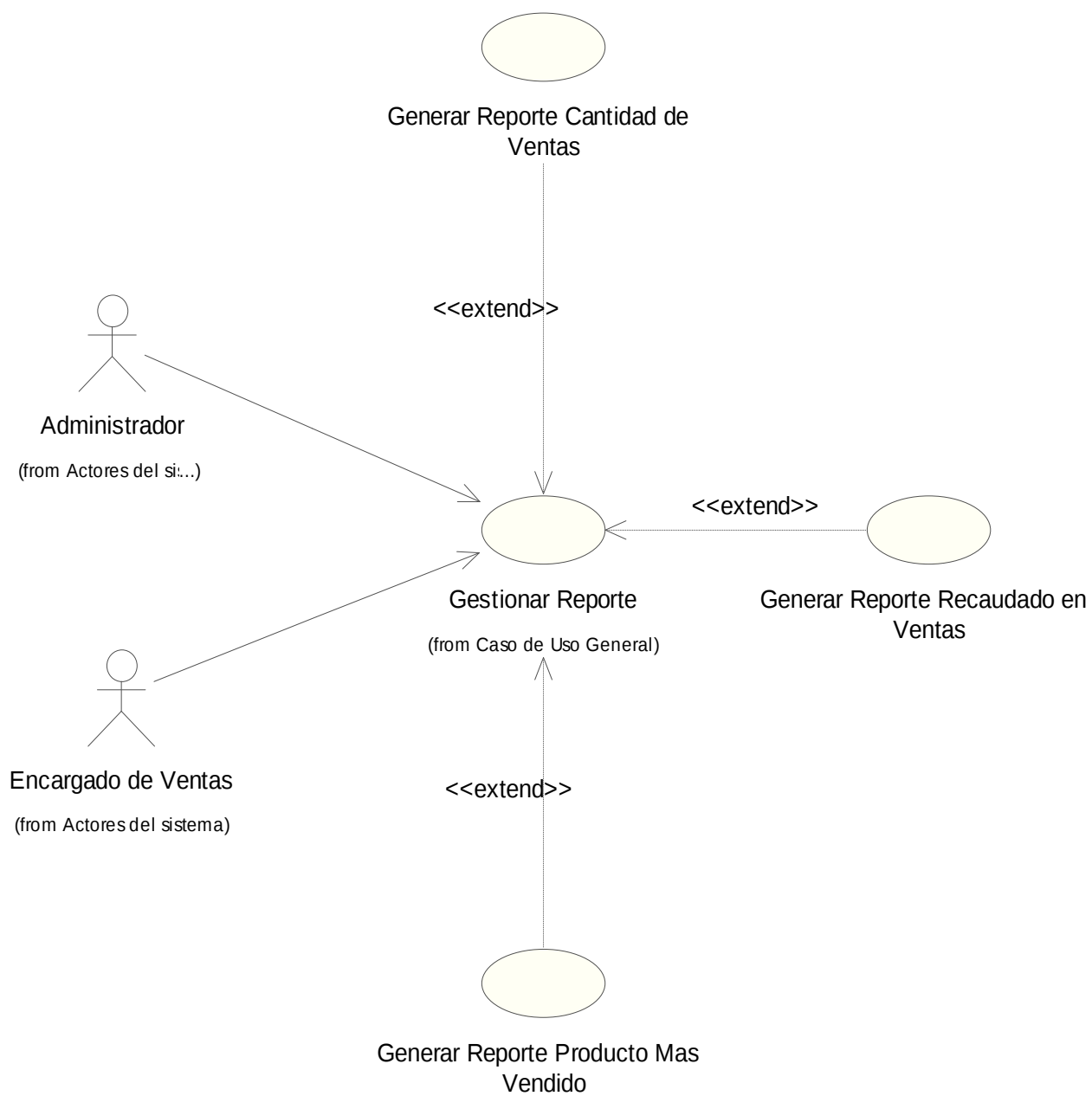
10. Caso de Uso: Gestionar Mantenimiento

**Figura N 19:** Administrar Mantenimiento

11. Caso de Uso: Gestionar Tipo de Cambio

**Figura N° 20:** Gestionar Tipo de Cambio

12. Caso de Uso: Gestionar Reportes

**Figura N° 21:** Gestionar Reportes

II.1.7. REQUERIMIENTOS NO FUNCIONALES

II.1.7.1. Introducción

A continuación se describen los requerimientos no funcionales del sistema bajo los siguientes criterios: rendimiento, confiabilidad, interfaces, fiabilidad, mantenimiento, seguridad, portabilidad y estándares.

Requisitos de Rendimiento

Req(36) Prioridad: Alta. El tiempo de respuesta en las operaciones debe ser el esperable en cualquier aplicación web.

Restricciones de Diseño

Req(37) Prioridad: Alta. El ciclo de vida elegido para desarrollar el producto será el de R.U.P., de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

Req(38) Prioridad: Alta. En cuanto a la infraestructura informática necesaria parece claro que se requiere un servidor web con un Sistema de Gestión de Base de Datos. La opción es PostgreSQL, pero se podrían evaluar otras alternativas como DB4o.

Atributos del Sistema

Disponibilidad

Req(19) Prioridad: Alta. El sistema estará accesible las 24 horas los 7 días de la semana.

Integridad de la información

Req(20) Prioridad: Alta. Es importante disponer de un sistema de backups automatizado, que permita volver a un estado anterior tras un desastre informático.

Mantenibilidad

Req(21) Prioridad: Alta. El sistema requerirá el mínimo posible de mantenimiento y, siempre que sea posible, las tareas de mantenimiento deberían realizarse vía web.

Seguridad

Req(22) *Prioridad: Alta.* El acceso se hará mediante una sencilla validación de usuario y contraseña. El sistema de gestión de base de datos proporcionará funciones de auditoría, registrando la fecha, hora y usuario (login) de toda transacción.

Req(23) *Prioridad: Alta.* El acceso al sistema se realizará exclusivamente mediante conexión cifrada, de modo que se preserve la privacidad de los datos de los usuarios y los identificadores de acceso cuando acceda al sistema desde redes públicas (ejemplo Internet).

II.1.8. ESPECIFICACIÓN DE LOS CASOS DE USO

II.1.8.1. Introducción

Cuando hablamos de las especificaciones de los casos de uso nos referimos a la descripción de los casos de uso. En realidad un caso de uso no es un caso de uso si no tiene una descripción detallada de la misma. Un caso de Uso es Texto no simplemente simbología.

II.1.8.2. Propósito

Entender con claridad el propósito de cada caso de uso.

Entender la relación que tendrá el sistema con el caso de uso.

II.1.8.3. Alcance

Describir el proceso que cumple el caso de uso

Detallar los flujos de cada caso de uso según lo establecido por la institución involucrado.

II.1.8.4. Especificación de los casos de Uso Generales

Los casos de uso más importantes involucrados en el sistema son:

CASO DE USO: Ingresar al Sistema.

Caso de uso:	Ingresar al Sistema
Actor:	Administrador, Encargado de Ventas y Cliente.
Descripción: Este caso de uso nos muestra el ingreso de usuarios al sistema mediante un usuario y clave, cuando el usuario entra al sistema puede gestionar distintos módulos de acuerdo al rol que cumple en la organización.	
Curso Normal	Alternativas
1. El usuario ingresa su nombre de usuario.	
2. El usuario ingresa su clave de acceso.	
3. El sistema valida los datos introducidos.	3.1. Si los datos no son válidos el sistema muestra un mensaje de "usuarios o contraseña no validos" y se retorna a la pantalla de ingreso.

Tabla N° 12: Caso de Uso: Ingresar al Sistema

MODULO GESTIONAR PRODUCTO**CASO DE USO:** Listar Producto.

Caso de uso:	Listar Producto.
Actor:	Administrador, Encargado de Ventas .
Explicación: Este caso de uso nos permite listar a todos los productos que se encuentran en lista.	
Curso Normal	Alternativas

1. El sistema muestra la pantalla Gestionar Producto.	
2. El Usuario lista los productos por tipo y estado y presiona el botón listar.	
3. El sistema actualiza la lista de productos de acuerdo a la elección del usuario.	

Tabla N° 13: Caso de Uso: Listar Producto.

CASO DE USO: Buscar Producto.

Caso de uso:	Buscar Producto.	
Actor:	Administrador, Encargado de Ventas .	
Explicación: Este caso de uso nos permite buscar un producto de cualquier tipo o estado.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Producto.		
2. El Usuario busca el producto por el nombre o código de producto.		
3. El sistema muestra el resultado de la búsqueda del producto.		3.1. Si no se encuentra ningún resultado el sistema muestra un mensaje de información que indica que no se encontró ningún elemento.

Tabla N° 14: Caso de Uso: Buscar Producto.

CASO DE USO: Adicionar Producto.

Caso de uso:	Adicionar Producto.	
Actor:	Administrador, Encargado de Ventas .	
Explicación: Este caso de uso nos permite registrar a los productos que van a ser vendidos en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Producto y el usuario selecciona la opción Adicionar Producto.		
2. El sistema muestra el formulario Adicionar Producto.		
3.- El usuario llena el formulario de adición de producto.		
4. Si todos los campos son válidos el sistema almacena los datos del formulario en la tabla Producto.		4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados del formulario.

Tabla N° 15: Caso de Uso: Adicionar Producto.

CASO DE USO: Modificar Producto.

Caso de uso:	Modificar Producto.	
Actor:	Administrador, Encargado de Ventas.	
Explicación: Este caso de uso nos permite modificar los productos que están registrados en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Producto y el usuario selecciona un producto y presiona el enlace modificar Producto.		
2. El sistema muestra el formulario Modificar Producto.		
3.- El usuario llena el formulario de modificación de producto.		
4. Si todos los campos son válidos el sistema actualiza los datos del formulario en la tabla Producto.		4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados del formulario.

Tabla N° 16: Caso de Uso: Modificar Producto.

CASO DE USO: Dar de Baja Producto.

Caso de uso:	Dar de Baja Producto.	
Actor:	Administrador, Encargado de Ventas .	
Explicación: Este caso de uso nos permite borrar un producto registrado en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Producto y el usuario selecciona un producto y presiona el enlace borrar Producto.		
2. El sistema muestra un mensaje de confirmación de borrado, donde el usuario presiona el botón aceptar.		
3.- Si el producto no pertenece a una venta es borrado de la tabla Producto.		3.1. Si el producto pertenece a una venta no puede ser borrado.

Tabla N° 17: Caso de Uso: Dar de Baja Producto.

GESTIONAR TIPO DE PRODUCTO**CASO DE USO:** Listar Tipo Producto.

Caso de uso:	Listar Tipo Producto.
Actor:	Administrador
Descripción: Este caso de uso nos permite listar todas los tipos de los productos registrados en el sistema.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Tipos.	
2. El Usuario lista los tipos por estado.	
3. El sistema actualiza la lista de las Tipos en Pantalla.	

Tabla N° 18: Caso de Uso: Listar Tipo Producto.**CASO DE USO:** Buscar Tipo Producto.

Caso de uso:	Buscar Tipo Producto.	
Actor:	Administrador	
Descripción: Este caso de uso nos permite buscar un tipo de producto registrada en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Tipo.		
2. El Usuario busca a un tipo por el nombre o por el código de tipo.		
3. El sistema muestra el resultado de la búsqueda del tipo en pantalla.		3.1. Si no se encuentra ningún resultado el sistema muestra un mensaje de información que indica que no se encontró ningún elemento.

Tabla N° 19: Caso de Uso: Buscar Tipo Producto.

CASO DE USO: Adicionar Tipo Producto.

Caso de uso:	Adicionar Tipo Producto.	
Actor:	Administrador	
Descripción: Este caso de uso nos permite registrar o adicionar un tipo de producto.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Tipo y el usuario selecciona la opción Adicionar Tipo.		
2. El sistema muestra el formulario Adicionar Tipo.		
3. El usuario llena el formulario adicionar Tipo.		
4. Si todos los campos son válidos el sistema almacena los datos del formulario en la tabla Tipo.		4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados del formulario.

Tabla N° 20: Caso de Uso: Adicionar Tipo Producto.**CASO DE USO:** Modificar Tipo Producto.

Caso de uso:	Modificar Tipo Producto.	
Actor:	Administrador	
Descripción: Este caso de uso nos permite modificar los datos de una categoría.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Tipo y el usuario ingresa al enlace Modificar Tipo.		
2. El sistema muestra el formulario Modificar Tipo.		
3.- El usuario Modifica el formulario de Tipo.		

4. Si todos los campos son válidos el sistema actualiza los datos del formulario en la tabla Tipo.	4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados del formulario.
--	--

Tabla N° 21: Caso de Uso: Modificar Tipo Producto.

CASO DE USO: Dar de Baja Tipo **Producto**.

Caso de uso:	Dar de Baja Tipo Producto.	
Actor:	Administrador	
Descripción: Este caso de uso nos permite borrar una Tipo registrada en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Categoría y el usuario selecciona un tipo y presiona él en lace Borrar tipo.		
2. El sistema muestra un mensaje de confirmación de borrado, donde el usuario presiona el botón aceptar.		
3.- Si la categoría no tiene relación con un producto es borrado de la tabla Tipo.		3.1. Si el tipo tiene relación con un producto no puede ser borrado.

Tabla N° 22: Caso de Uso: Dar de Baja Tipo Producto.

MODULO GESTIONAR CLIENTE
CASO DE USO: Listar Cliente.

Caso de uso:	Listar Cliente.
Actor:	Administrador, Administrador Auxiliar, Encargado de Ventas.
Descripción: Este caso de uso nos permite listar a todos los clientes que se encuentran registrados en el sistema.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Cliente.	
2. El Usuario lista los clientes por tipo y estado.	
3. El sistema actualiza la lista de los clientes registrados.	

Tabla N° 23: Caso de Uso: Listar Cliente.

CASO DE USO: Buscar Cliente.

Caso de uso:	Buscar Cliente.
Actor:	Administrador y Encargado de Ventas.
Descripción: Este caso de uso nos permite buscar un cliente que se encuentra registrado en el sistema.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Cliente.	
2. El Usuario busca los clientes por el nombre, ci o código del cliente, el cual es asignado por el sistema.	
3. El sistema muestra el resultado de la búsqueda del	3.1. Si no se encuentra ningún

cliente.	resultado el sistema muestra un mensaje de información que indica que no se encontró ningún elemento.
----------	---

Tabla N° 24: Caso de Uso: Buscar Cliente.

CASO DE USO: Adicionar Cliente.

Caso de uso:	Adicionar Cliente.	
Actor:	Administrador y Encargado de Ventas.	
Descripción: Este caso de uso nos permite registrar los datos de un cliente.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Cliente y el usuario selecciona la opción Adicionar Cliente.		
2. El sistema muestra el formulario Adicionar Cliente.		
3. El usuario llena el formulario adicionar cliente.		
4. Si todos los campos son válidos el sistema almacena los datos del formulario en la tabla Persona, Cliente, Dirección, Teléfono.		4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados.

Tabla N° 25: Caso de Uso: Adicionar Cliente.

CASO DE USO: Modificar Cliente.

Caso de uso:	Modificar Cliente.	
Actor:	Administrador y Encargado de Ventas.	
Descripción: Este caso de uso nos permite modificar los datos del cliente ya registrado.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Cliente y el usuario ingresa al enlace Modificar Cliente.		
2. El sistema muestra el formulario Modificar Cliente.		
3.- El usuario Modifica el formulario de Cliente.		
4. Si todos los campos son válidos el sistema actualiza los datos del formulario en la tabla Persona, Cliente, Dirección, Teléfono.		4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados del formulario.

Tabla N° 26: Caso de Uso: Modificar Cliente.**CASO DE USO:** Dar de Baja Cliente.

Caso de uso:	Dar de Baja Cliente.	
Actor:	Administrador y Encargado de Ventas.	
Descripción: Este caso de uso nos permite borrar un cliente registrado en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Cliente y el usuario selecciona un cliente de la lista presentada.		
2. El sistema muestra un mensaje de confirmación de borrado, donde el usuario presiona el botón aceptar.		
3.- Si el cliente no tiene relación con una venta es borrado		3.1. Si el cliente tiene relación

de la tabla Cliente, Dato, Institución.	con una venta no puede ser borrado.
---	-------------------------------------

Tabla N° 27: Caso de Uso: Dar de Baja Cliente.

MODULO GESTIONAR PERSONAL

CASO DE USO: Listar Personal.

Caso de uso:	Listar Personal.
Actor:	Administrador.
Descripción: Este caso de uso nos permite listar a todo el personal de la tienda.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Personal.	
2. El Usuario lista al personal por cargo, estado y presiona el botón Listar.	
3. El sistema actualiza la lista del personal registrados en el sistema.	

Tabla N° 28: Caso de Uso: Listar Personal.

CASO DE USO: Buscar Personal.

Caso de uso:	Buscar Personal.
Actor:	Administrador.
Descripción: Este caso de uso nos permite buscar al personal registrado en el sistema.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Personal.	
2. El Usuario busca a un miembro del personal por el nombre, código de personal asignado por el sistema.	
3. El sistema muestra el resultado de la búsqueda de personal.	3.1. Si no se encuentra ningún resultado el sistema muestra un mensaje de información que indica que no se encontró ningún elemento.

Tabla N° 29: Caso de Uso: Buscar Personal.**CASO DE USO:** Adicionar Personal.

Caso de uso:	Adicionar Personal.
Actor:	Administrador.
Descripción: Este caso de uso nos permite registrar o adicionar un miembro del personal.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Personal y el usuario selecciona la opción Adicionar Personal.	
2. El sistema muestra el formulario Adicionar Personal.	
3. El usuario llena el formulario adicionar personal.	
4. Si todos los campos son válidos el sistema almacena los datos del formulario en la tabla Persona, Personal, Cargo,	4.1. Si los datos no están llenados correctamente se

Dirección, Teléfono.	muestra el error en los campos mal llenados del formulario.
----------------------	---

Tabla N° 30 Caso de Uso: Adicionar Personal.

CASO DE USO: Modificar Personal.

Caso de uso:	Modificar Personal.	
Actor:	Administrador.	
Descripción: Este caso de uso nos permite modificar los datos de un miembro del personal registrado en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Personal y el usuario ingresa al enlace Modificar Personal.		
2. El sistema muestra el formulario Modificar Personal.		
3.- El usuario Modifica el formulario de Personal.		
4. Si todos los campos son válidos el sistema actualiza los datos del formulario en la tabla Persona, Cliente, Dirección, Teléfono, Personal, Cargo.		4.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados del formulario.

Tabla N° 31: Caso de Uso: Modificar Personal.

CASO DE USO: Dar de Baja Personal.

Caso de uso:	Dar de Baja Personal.	
Actor:	Administrador, Administrador Auxiliar, Encargado de Ventas.	
Descripción: Este caso de uso nos permite borrar un miembro registrado en el sistema.		
Curso Normal		Alternativas

1. El sistema muestra la pantalla Gestionar Personal y el usuario selecciona un miembro del personal de la lista presentada y presiona el enlace Borrar personal.	
2. El sistema muestra un mensaje de confirmación de borrado, donde el usuario presiona el botón aceptar.	
3.- Si el personal no tiene relación con una venta es borrado de la tabla Personal.	3.1. Si el personal tiene relación con una venta no puede ser borrado.

Tabla N° 32: Caso de Uso: Dar de Baja Personal

MODULO GESTIONAR VENTA

CASO DE USO: Listar Venta.

Caso de uso:	Listar Venta.
Actor:	Administrador y Encargado de Ventas.
Descripción: Este caso de uso nos permite listar las ventas registradas en el sistema.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Venta.	
2. El Usuario lista las ventas por tipo, estado y fecha de venta.	
3. El sistema actualiza la lista de ventas registradas en el sistema.	

Tabla N° 33: Caso de Uso: Listar Venta.

CASO DE USO: Buscar Venta.

Caso de uso:	Buscar Venta.
Actor:	Administrador y Encargado de Ventas.
Descripción: Este caso de uso nos permite buscar una venta registrada en el sistema.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Venta.	
2. El Usuario busca una venta por el nombre del cajero o del cliente, por ci del cajero o del cliente, por el código del cajero o del cliente.	
3. El sistema muestra el resultado de la búsqueda de la venta.	3.1. Si no se encuentra ningún resultado el sistema muestra un mensaje de información que indica que no se encontró ningún elemento.

Tabla N° 34: Caso de Uso: Buscar Venta.**CASO DE USO:** Registra Venta.

Caso de uso:	Registra Venta
Actor:	Administrador y Encargado de Ventas.
Descripción: Este caso de uso nos permite registrar una venta,	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Gestionar Venta y el usuario selecciona la opción Registrar Venta.	
2. El sistema muestra el primer formulario para el registro de la venta.	
3. El usuario llena el formulario y presiona el botón continuar.	3.1. Si los datos no están llenados correctamente se

	muestra el error en los campos mal llenados.
4. El sistema muestra una segunda pantalla para el registro o búsqueda del cliente ya registrado.	
5. El usuario seleccionar el cliente registrado y presiona el botón continuar.	
6. El sistema muestra una lista para la selección de los productos que van a ser vendidos, los cuales son seleccionados por el usuario.	
7. El sistema adiciona los datos en las tablas Venta, Detalleventa, y muestra un mensaje que dice: "La venta se realizó correctamente".	

Tabla N° 35: Caso de Uso: Registra Venta.

MODULO GESTIONAR ENTRADA PRODUCTO

CASO DE USO: Listar Entrada Producto.

Caso de uso:	Listar Entrada Producto.	
Actor:	Administrador y Encargado de Ventas.	
Descripción: Este caso de uso nos permite listar todas las entradas producto registradas en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Entrada Producto.		
2. El Usuario lista las entradas producto por estado.		
3. El sistema actualiza la lista de entradas producto registradas en el sistema.		

Tabla N° 36: Caso de Uso: Listar Entrada Producto.

CASO DE USO: Buscar Entrada Producto.

Caso de uso:	Buscar Entrada Producto.	
Actor:	Administrador y Encargado de Ventas.	
Descripción: Este caso de uso nos permite buscar una entrada producto registrada en el sistema.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Entrada Producto.		
2. El Usuario busca una entrada producto por el nombre del cajero o del cliente, por ci del cajero o del cliente, por el código del cajero o del cliente.		
3. El sistema muestra el resultado de la búsqueda de la Entrada Producto.		3.1. Si no se encuentra ningún resultado el sistema muestra un mensaje de información que indica que no se encontró ningún elemento.

Tabla N° 37: Caso de Uso: Buscar Entrada Producto.**CASO DE USO:** Registra Entrada Producto.

Caso de uso:	Registra Entrada Producto.	
Actor:	Administrador y Encargado de Ventas.	
Descripción: Este caso de uso nos permite registrar una Entrada Producto.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Gestionar Compra y el usuario selecciona la opción Registrar Entrada Producto.		
2. El sistema muestra el primer formulario para el registro de la entrada producto.		

3. El usuario llena el formulario y presiona el botón continuar.	3.1. Si los datos no están llenados correctamente se muestra el error en los campos mal llenados.
4. El sistema muestra una segunda pantalla para el registro o búsqueda del proveedor ya registrado.	
5. El usuario seleccionar el proveedor registrado y presiona el botón continuar.	
6. El sistema muestra una lista para la selección de los productos que van a ser comprados, los cuales son seleccionados por el usuario.	6.1. Si el producto a comprar es nuevo, el sistema permite la adición del mismo.
7. El sistema adiciona los datos en las tablas EntradaProducto, DetalleEntradaProducto, y muestra un mensaje que dice: "Se registró correctamente".	

Tabla N° 38: Caso de Uso: Registra Entrada Producto.

CASO DE USO: Resguardar Base de Datos.

Caso de uso:	Resguardar Base de Datos.
Actor:	Administrador.
Descripción: Este caso de uso nos permite resguardar la base de datos.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Mantenimiento y el usuario ingresa a Resguardar Base de Datos.	
2. El sistema muestra la pantalla Resguardar Base de Datos y el usuario presiona el botón Examinar.	
3. El sistema muestra una pantalla para que el usuario pueda buscar o seleccionar un destino para el resguardo.	
4. El usuario presión el botón Resguardar.	

5. El sistema resguarda la base de datos en el lugar escogido por el usuario y muestra un mensaje de confirmación de resguardo.	
---	--

Tabla N° 39: Caso de Uso: Resguardar Base de Datos.

CASO DE USO: Restaurar Base de Datos.

Caso de uso:	Restaurar Base de Datos.	
Actor:	Administrador.	
Explicación: Este caso de uso nos permite restaurar la base de datos.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Mantenimiento y el usuario ingresa a Restaurar Base de Datos.		
2. El sistema muestra la pantalla Restaurar Base de Datos y el usuario presiona el botón Examinar.		
3. El sistema muestra una pantalla para que el usuario pueda buscar o seleccionar una base de datos resguardada anterior mente.		
4. El usuario presión el botón Restaurar.		
5. El sistema verifica si la extensión del archivo (.backup) es válida.		5.1 Si la extensión no es válida muestra un mensaje de error en pantalla.
6. El sistema restaura la base de datos y muestra un mensaje de confirmación de restauro.		

Tabla N° 40: Caso de Uso: Restaurar Base de Datos.

MODULO REPORTE

CASO DE USO: Generar Reporte de Cantidad de Ventas.

Caso de uso:	Generar Reporte de Cantidad de Ventas.
Actor:	Administrador y Encargado de Ventas.
Explicación: Este caso de uso nos permite generar el reporte de cantidad de ventas anuales.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Reporte y el usuario selecciona la opción Reporte de Cantidad de Ventas.	
2. El sistema muestra la pantalla Generar Reporte.	
3. El usuario selecciona el año y presiona el botón generar.	
4. El sistema muestra una el reporte grafico generado.	
5. El usuario seleccionar el botón Imprimir.	
6. El sistema muestra la pantalla de Impresión.	
7. El usuario configura la impresión y presiona Aceptar.	
8. El Sistema realiza la impresión del reporte.	

Tabla N° 41: Caso de Uso: Generar Reporte de Cantidad de Ventas.

CASO DE USO: Generar Reporte de Recaudado en Ventas.

Caso de uso:	Generar Reporte de Recaudado en Ventas.	
Actor:	Administrador y Encargado de Ventas.	
Explicación: Este caso de uso nos permite generar el reporte de recaudado en ventas anuales.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Reporte y el usuario selecciona la opción Reporte de Recaudado en Ventas.		
2. El sistema muestra la pantalla Generar Reporte.		
3. El usuario selecciona el año y presiona el botón generar.		
4. El sistema muestra una el reporte grafico generado.		
5. El usuario seleccionar el botón Imprimir.		
6. El sistema muestra la pantalla de Impresión.		
7. El usuario configura la impresión y presiona Aceptar.		
8. El Sistema realiza la impresión del reporte.		

Tabla N° 42: Caso de Uso: Generar Reporte de Recaudado en Ventas.

CASO DE USO: Generar Reporte de Producto más Vendido.

Caso de uso:	Generar Reporte de Producto más Vendido.	
Actor:	Administrador y Encargado de Ventas.	
Explicación: Este caso de uso nos permite generar el reporte del producto más vendido en el año.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Reporte y el usuario selecciona la opción Reporte de Producto más Vendido.		
2. El sistema muestra la pantalla Generar Reporte.		
3. El usuario selecciona el año y presiona el botón generar.		
4. El sistema muestra una el reporte grafico generado.		
5. El usuario seleccionar el botón Imprimir.		
6. El sistema muestra la pantalla de Impresión.		
7. El usuario configura la impresión y presiona Aceptar.		
8. El Sistema realiza la impresión del reporte.		

Tabla N° 43: Caso de Uso: Generar Reporte de Producto más Vendido.

MODULO GESTIONAR MANTENIMIENTO

CASO DE USO: Resguardar Base de Datos.

Caso de uso:	Resguardar Base de Datos.
Actor:	Administrador.
Explicación: Este caso de uso nos permite resguardar la base de datos.	
Curso Normal	Alternativas
1. El sistema muestra la pantalla Mantenimiento y el usuario ingresa a Resguardar Base de Datos.	
2. El sistema muestra la pantalla Resguardar Base de Datos y el usuario presiona el botón Examinar.	
3. El sistema muestra una pantalla para que el usuario pueda buscar o seleccionar un destino para el resguardo.	
4. El usuario presión el botón Resguardar.	
5. El sistema resguarda la base de datos en el lugar escogido por el usuario y muestra un mensaje de confirmación de resguardo.	

Tabla N° 44: Caso de Uso: Resguardar Base de Datos.

CASO DE USO: Restaurar Base de Datos.

Caso de uso:	Restaurar Base de Datos.	
Actor:	Administrador.	
Explicación: Este caso de uso nos permite restaurar la base de datos.		
Curso Normal		Alternativas
1. El sistema muestra la pantalla Mantenimiento y el usuario ingresa a Restaurar Base de Datos.		
2. El sistema muestra la pantalla Restaurar Base de Datos y el usuario presiona el botón Examinar.		
3. El sistema muestra una pantalla para que el usuario pueda buscar o seleccionar una base de datos resguardada anterior mente.		
4. El usuario presión el botón Restaurar.		
5. El sistema verifica si la extensión del archivo (.backup) es válida.		5.1 Si la extensión no es válida muestra un mensaje de error en pantalla.
6. El sistema restaura la base de datos y muestra un mensaje de confirmación de restauro.		

Tabla N° 45: Caso de Uso: Restaurar Base de Datos.

II.1.9. MODELADO DE DIAGRAMA DE ACTIVIDADES

II.1.9.1. Introducción

El Diagrama de Actividades es uno más de los diagramas requeridos por la metodología RUP la cual estamos implementando.

II.1.9.1.1. Propósito

- Comprender la estructura y la dinámica del sistema deseado para la organización
- Identificar posibles mejoras

II.1.9.1.2. Alcance

- Describir los procesos de sistema.
- 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.

II.1.9.2. Diagrama de Actividades

Diagrama de Actividad: Ingresar al Sistema

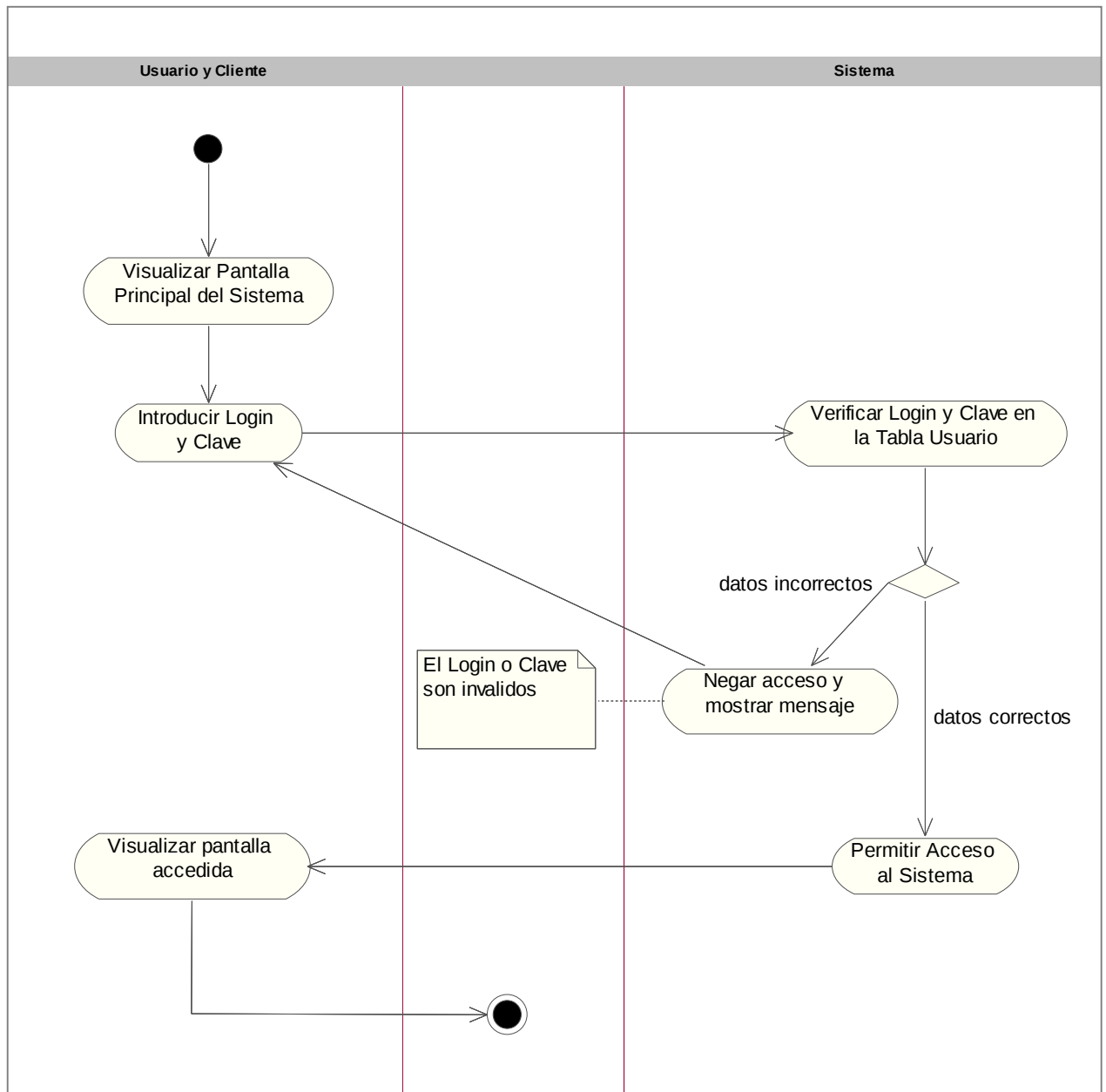


Figura N° 22: Diagrama de Actividad: Ingresar al Sistema

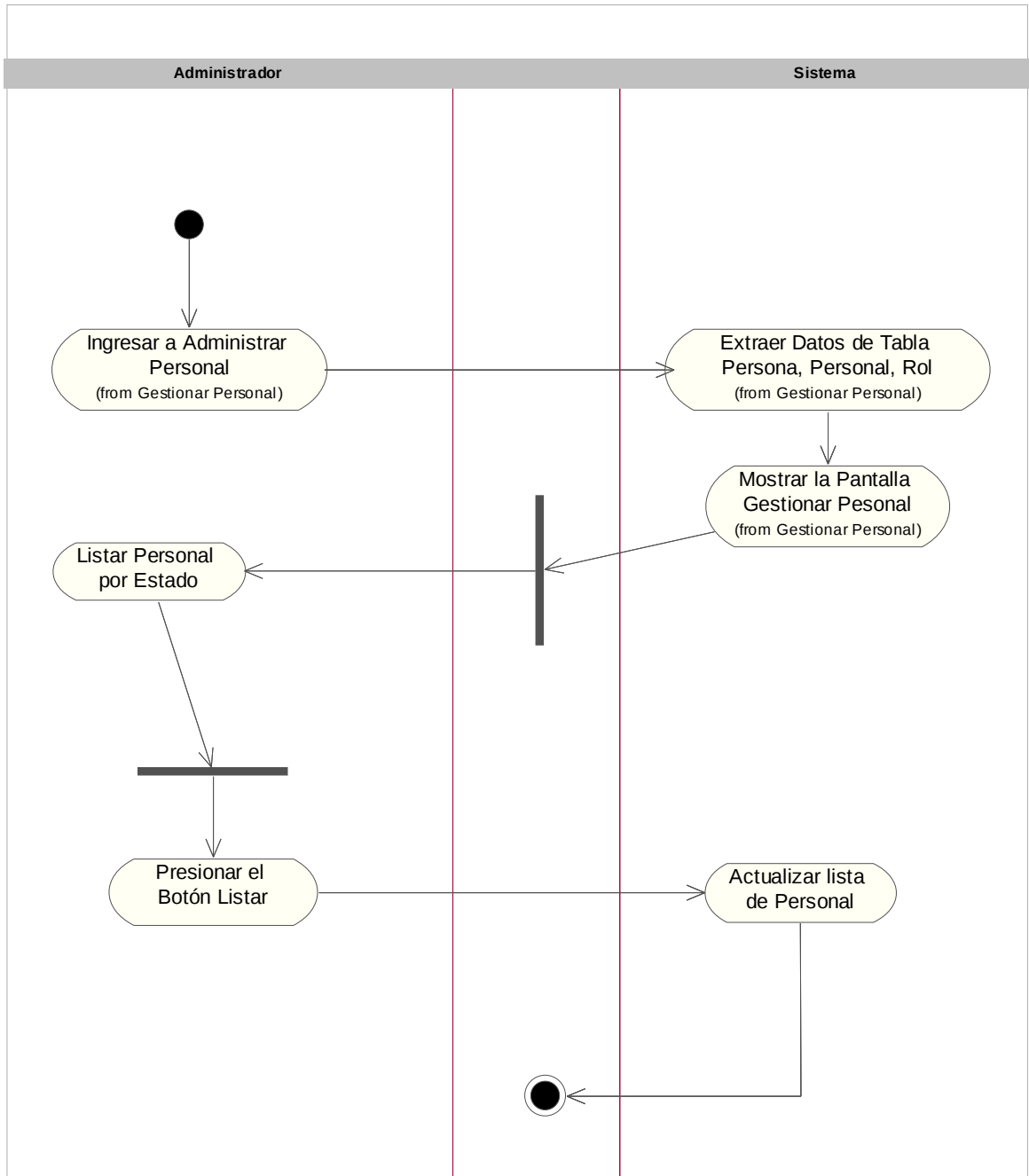
MODULO GESTIONAR PERSONAL**DIAGRAMA DE ACTIVIDAD: Listar Personal****Figura N° 23: Diagrama de Actividad: Listar Personal**

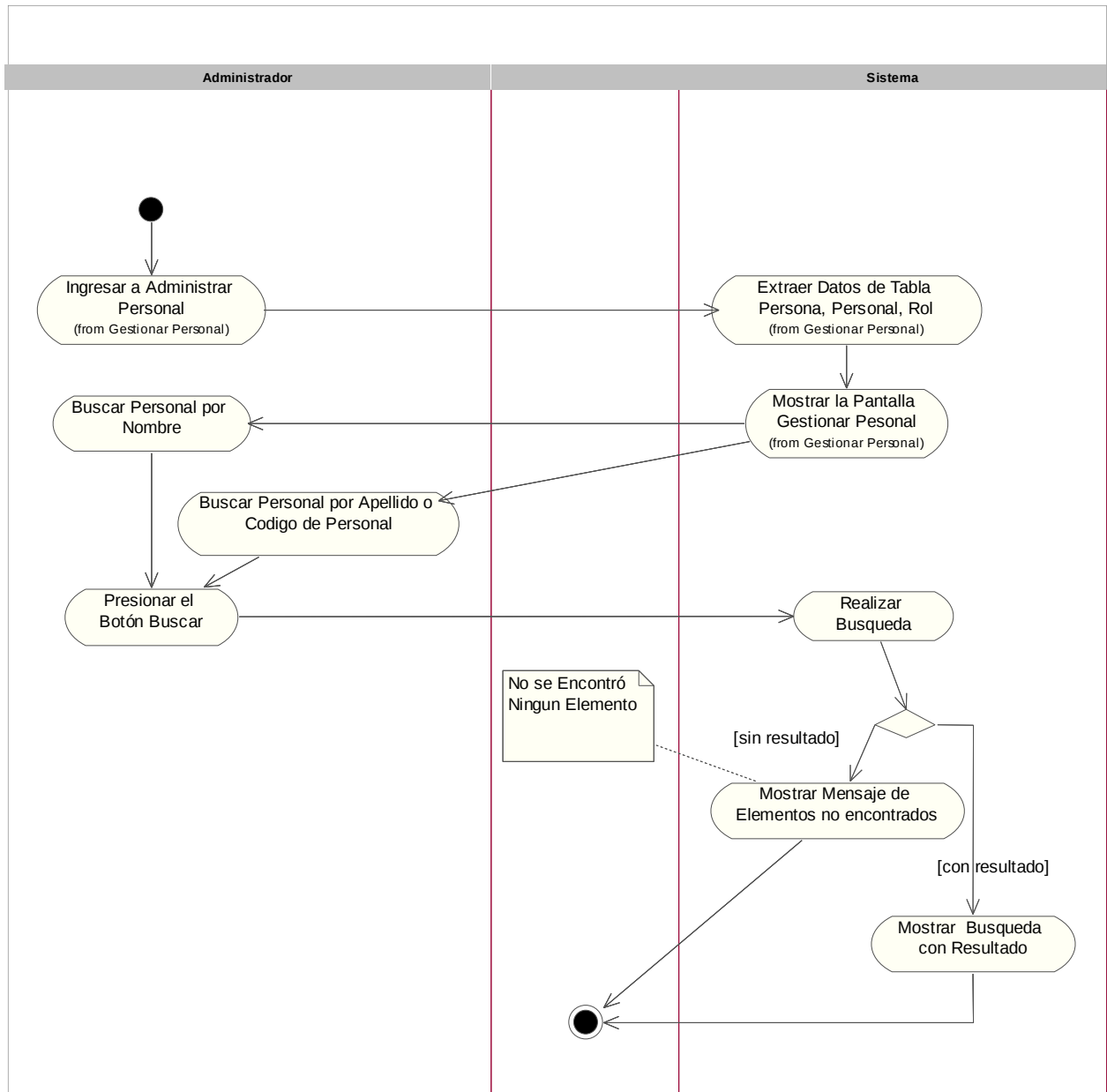
DIAGRAMA DE ACTIVIDAD: Buscar Personal**Figura N° 24: Diagrama de Actividad: Buscar Personal**

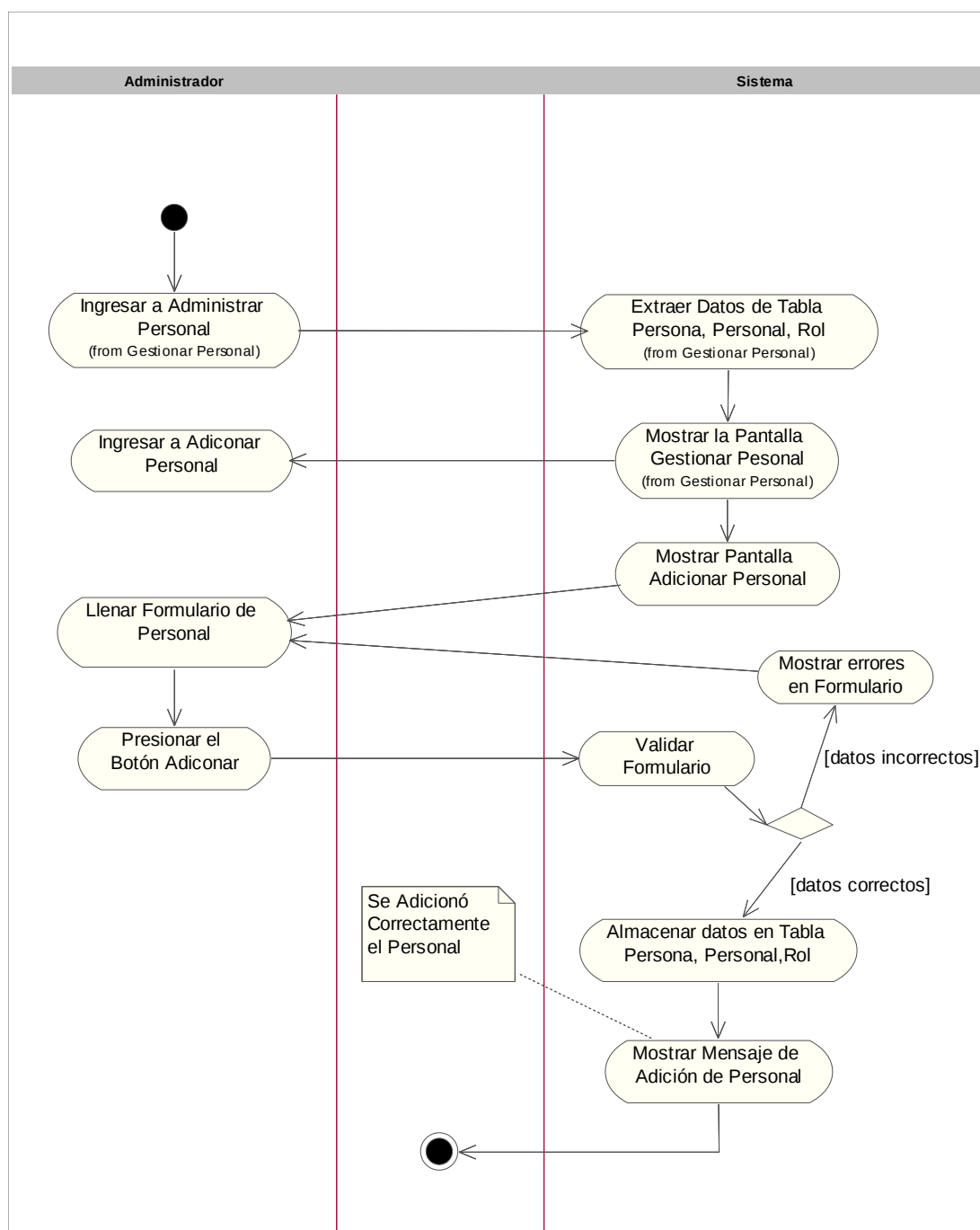
DIAGRAMA DE ACTIVIDAD: Adicionar Personal**Figura N° 25:** Diagrama de Actividad: Adicionar Personal

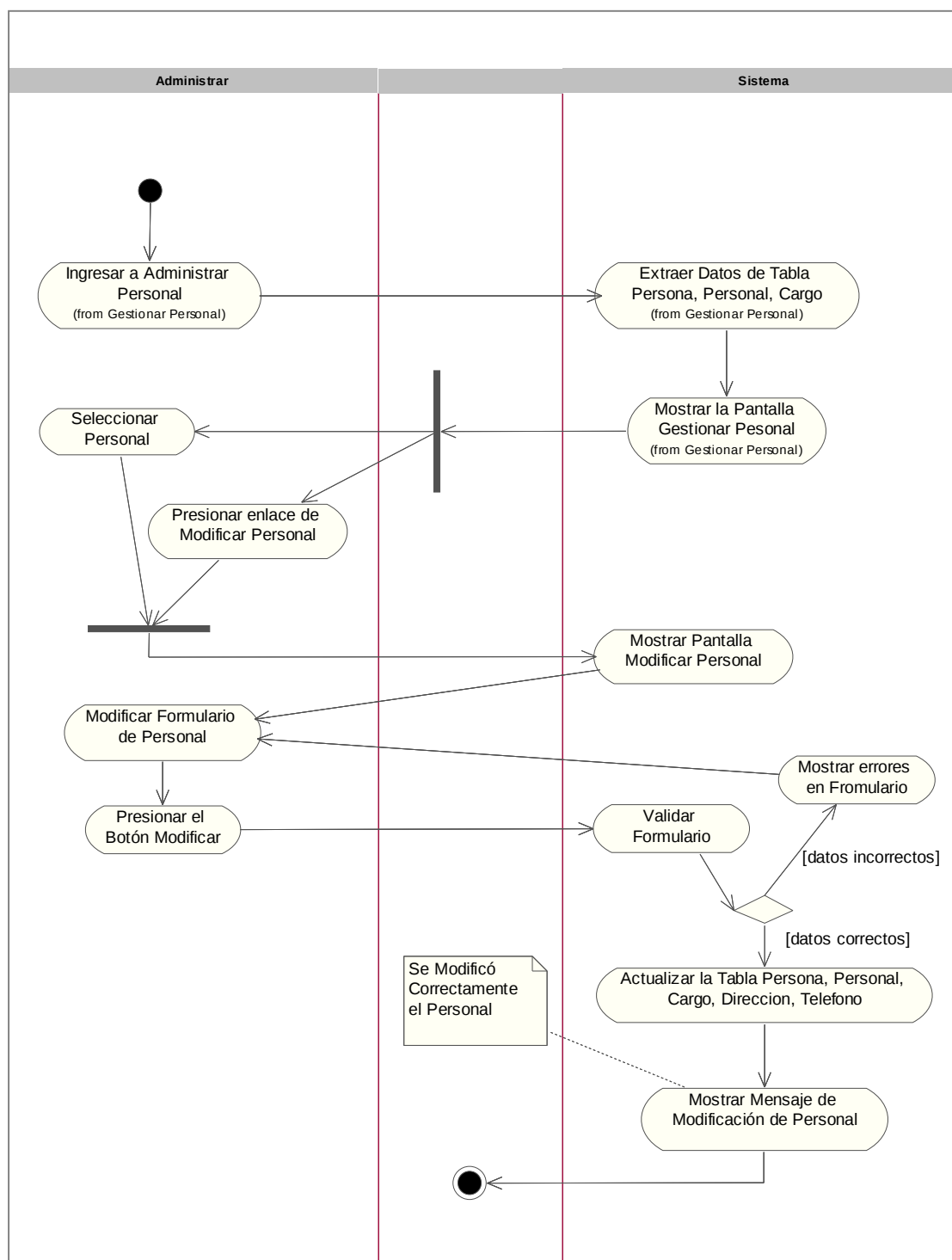
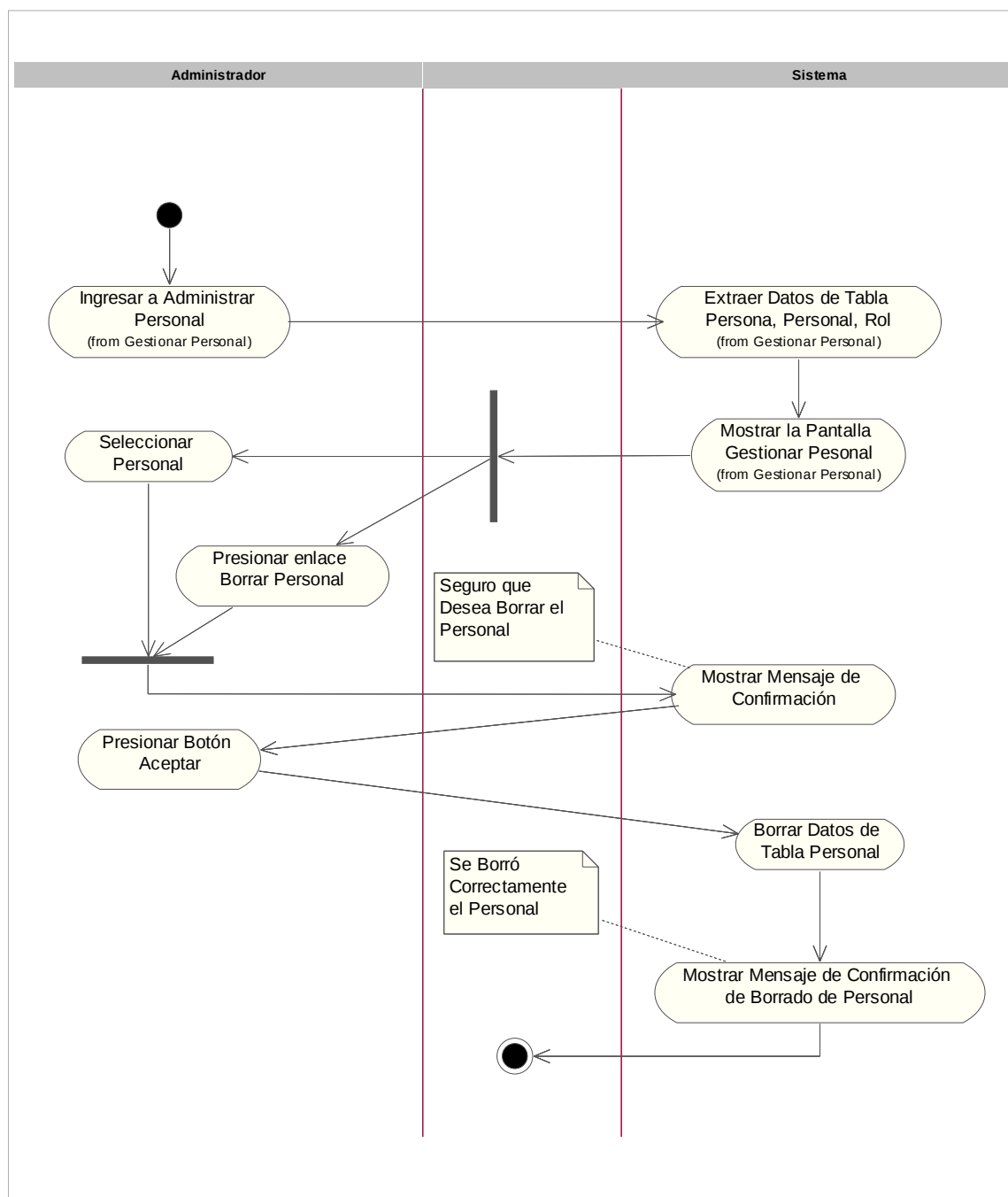
DIAGRAMA DE ACTIVIDAD: Modificar Personal**Figura N° 26: Diagrama de Actividad: Modificar Personal**

DIAGRAMA DE ACTIVIDAD: Dar de Baja Personal**Figura N° 27: Diagrama de Actividad: Dar de Baja Personal**

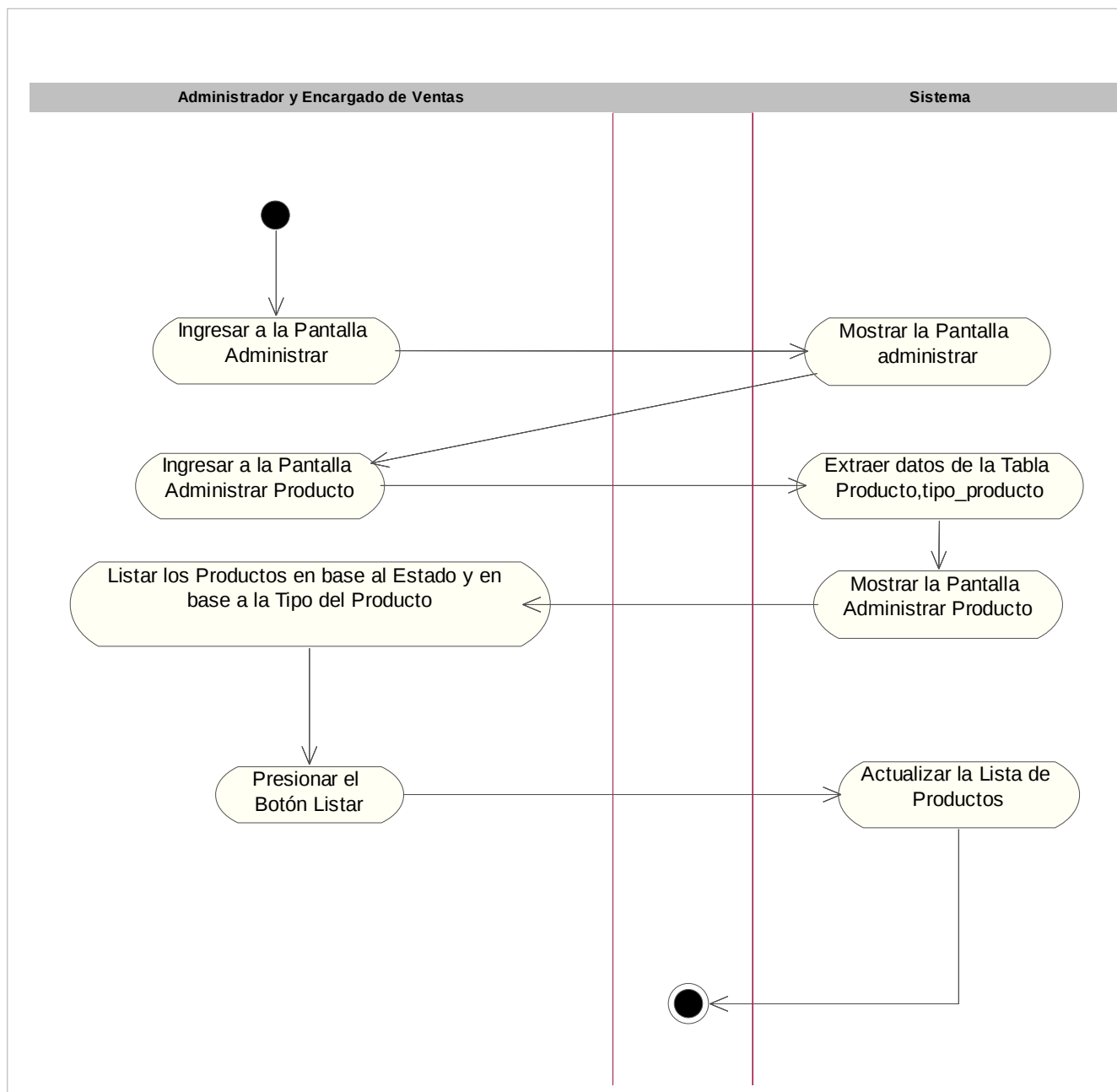
MODULO GESTIONAR PRODUCTO
DIAGRAMA DE ACTIVIDAD: Listar Producto

Figura N° 28: Diagrama de Actividad: Listar Producto

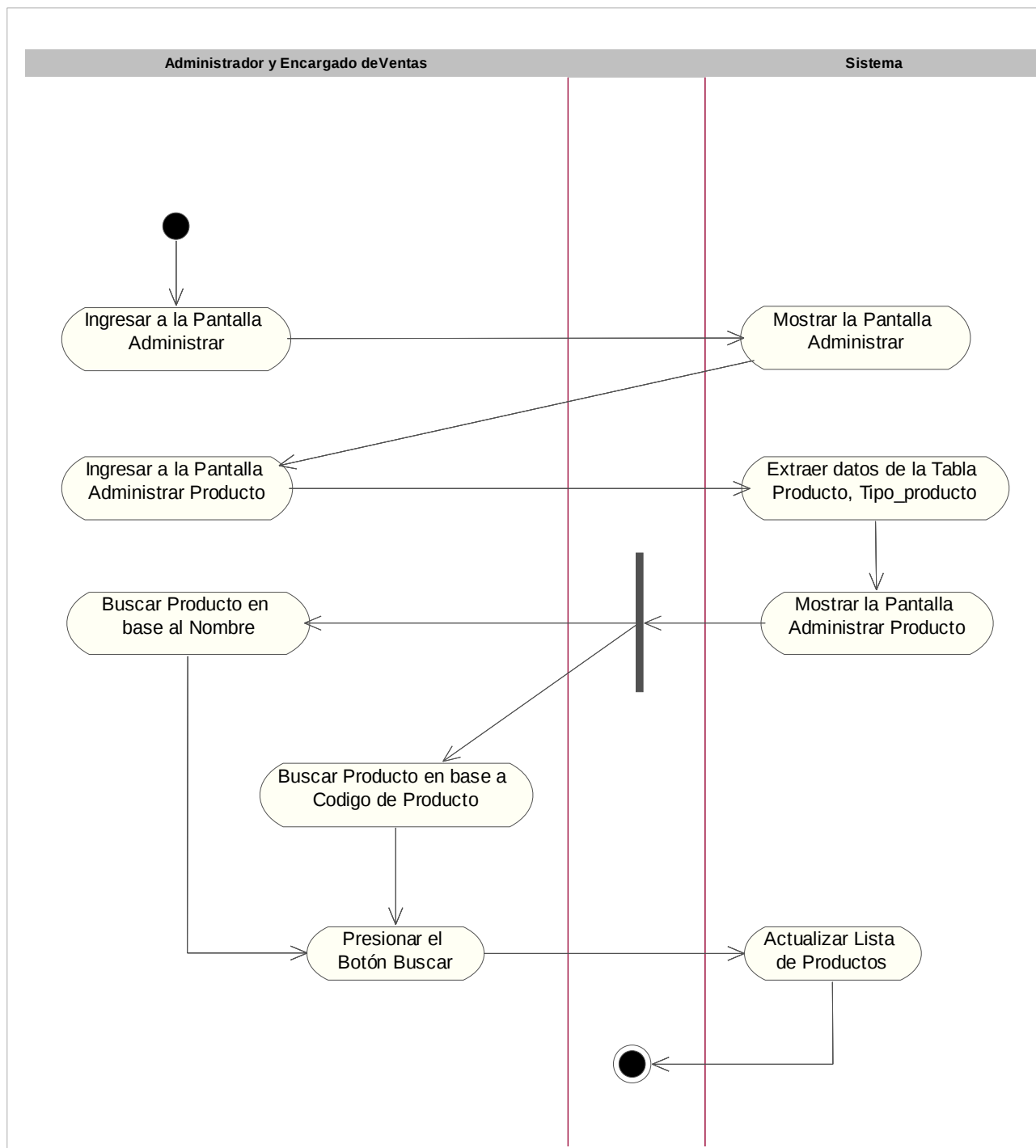
DIAGRAMA DE ACTIVIDAD: Buscar Producto**Figura N° 29: Diagrama de Actividad: Buscar Producto**

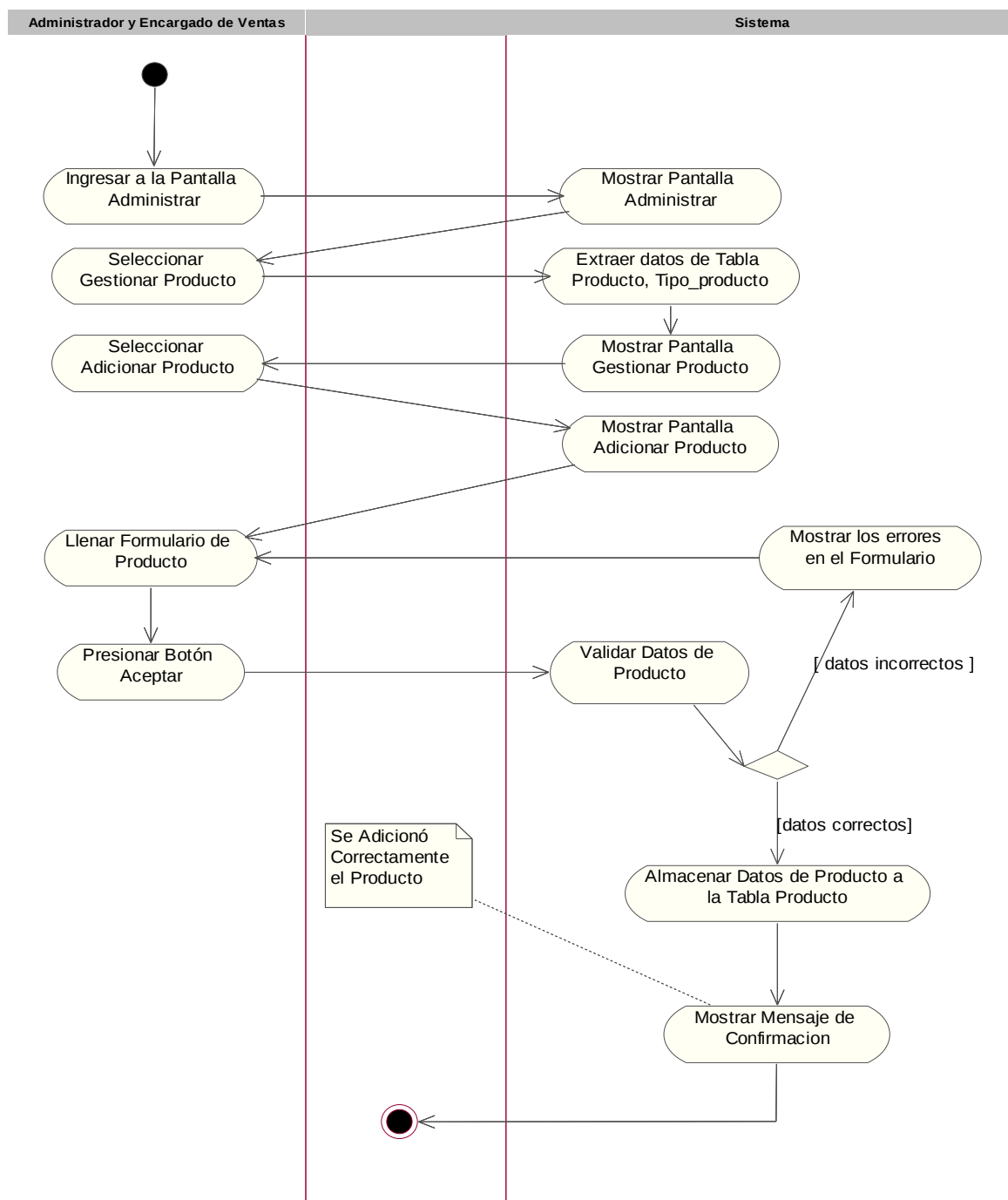
DIAGRAMA DE ACTIVIDAD: Adicionar producto**Figura N° 30:** Diagrama de Actividad: Adicionar Producto

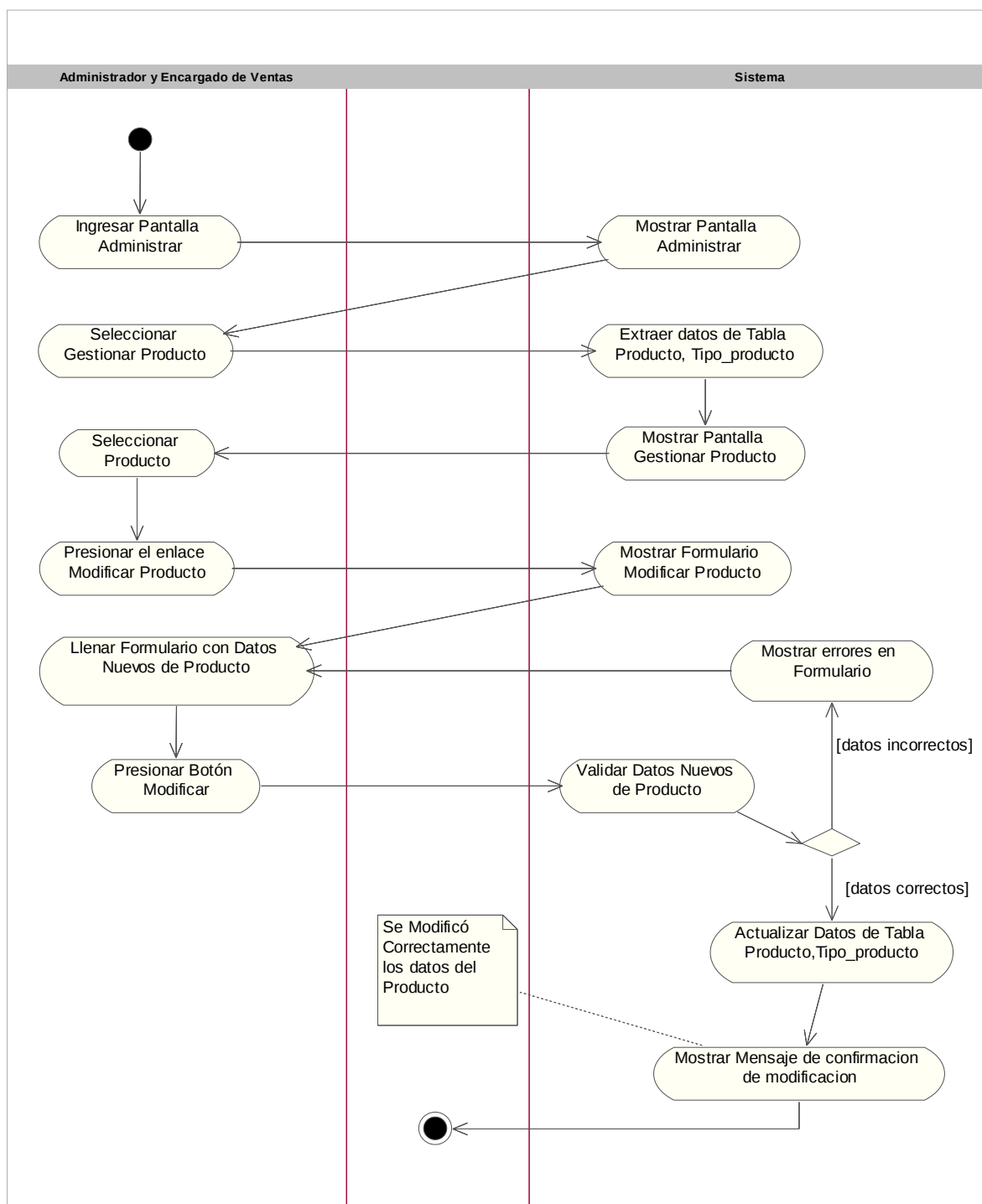
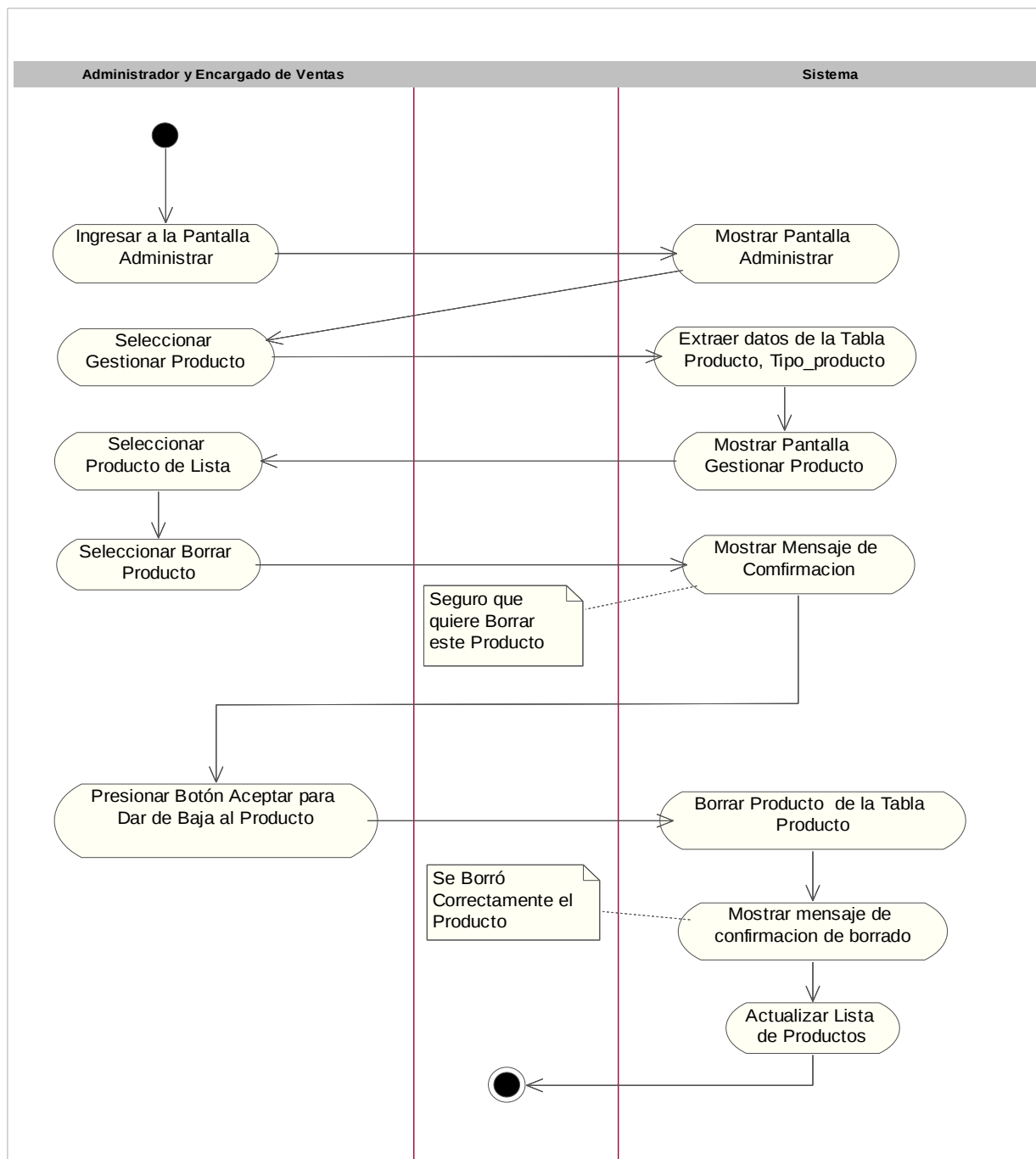
DIAGRAMA DE ACTIVIDAD: Modificar Producto**Figura N° 31:** Diagrama de Actividad: Modificar Producto

DIAGRAMA DE ACTIVIDAD: Dar de Baja Producto**Figura N° 32:** Diagrama de Actividad: Dar de Baja Producto

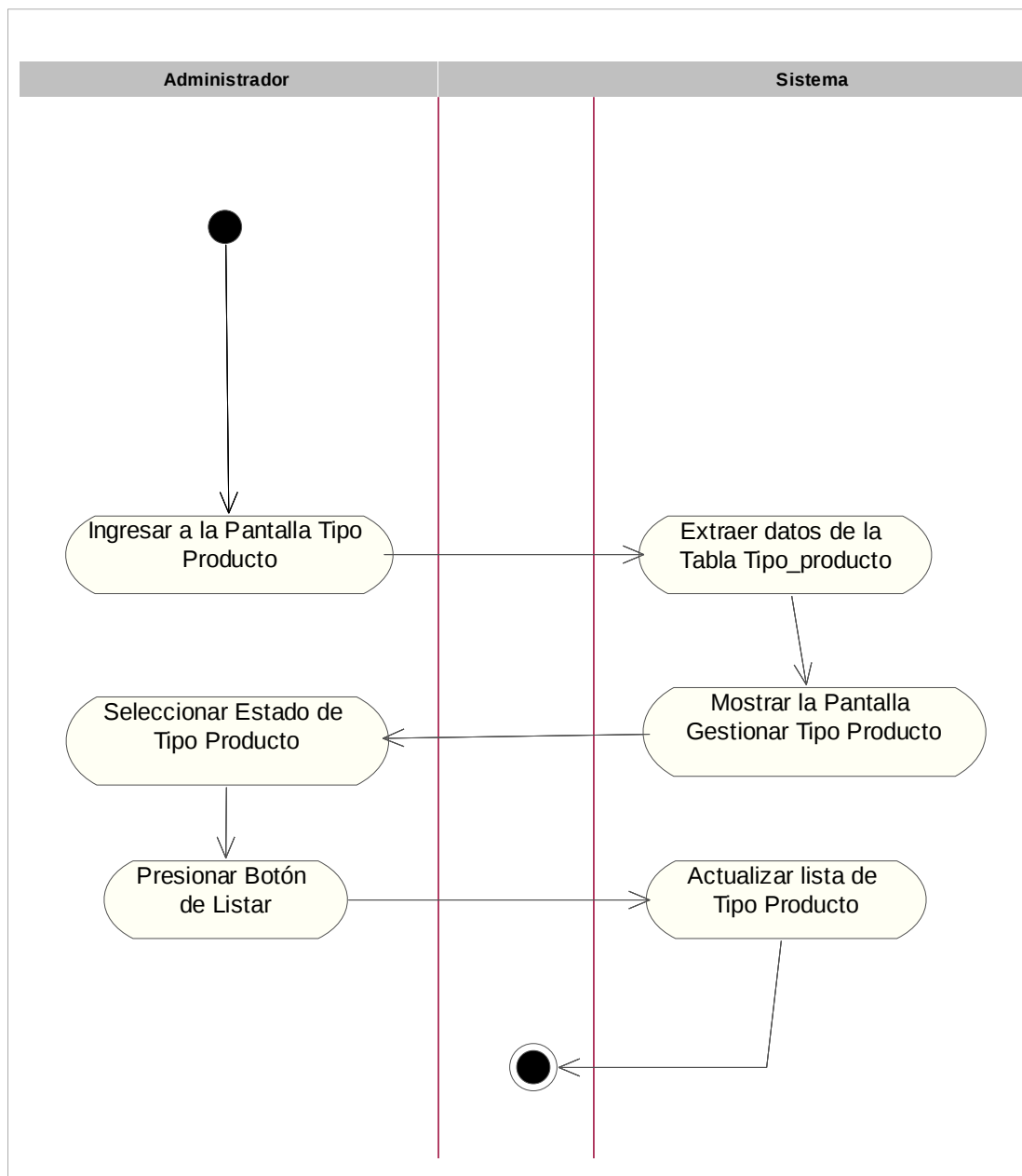
GESTIONAR TIPO PRODUCTO**DIAGRAMA DE ACTIVIDAD: Listar Tipo Producto****Figura N° 33: Diagrama de Actividad: Listar Tipo Producto**

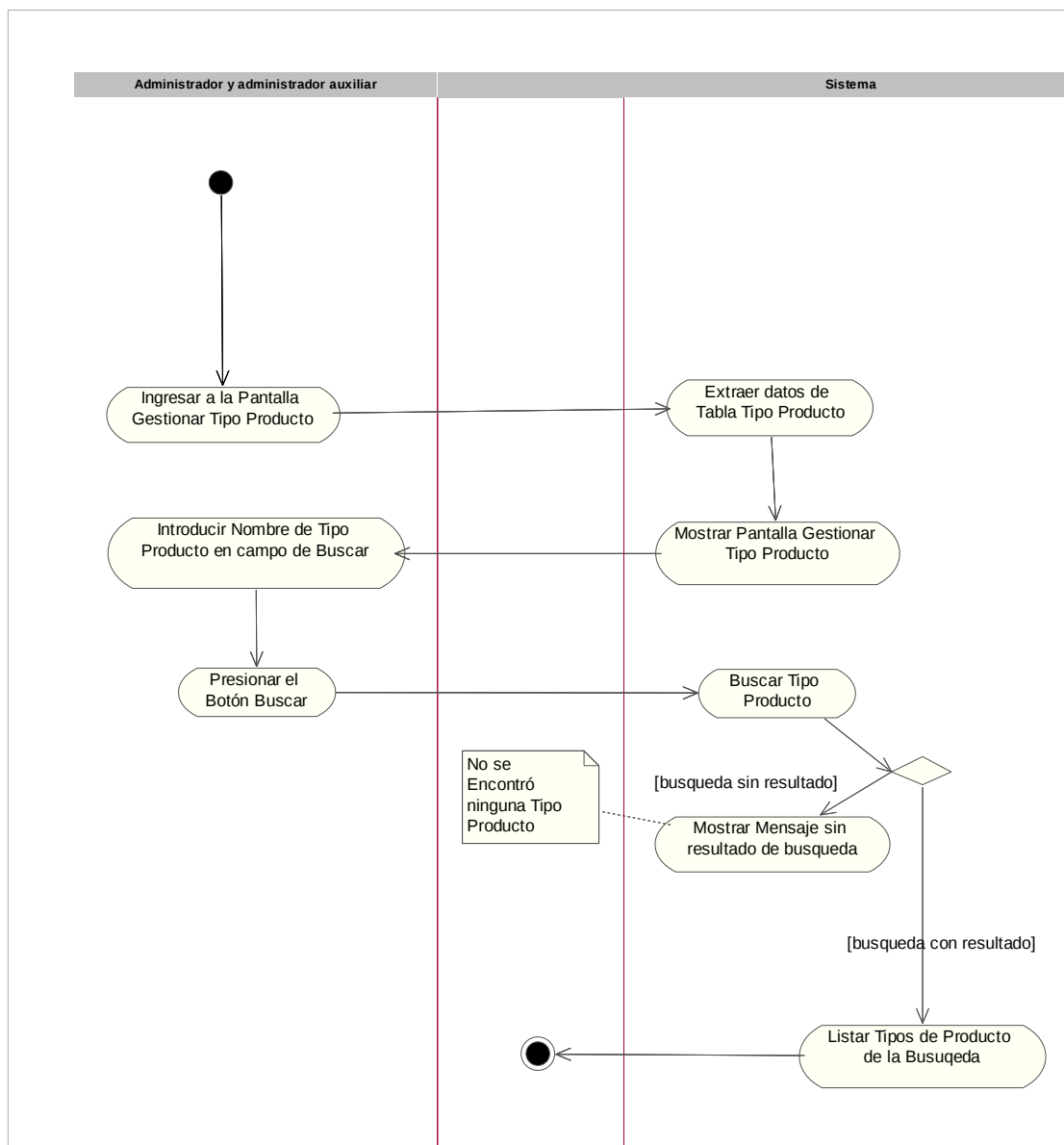
DIAGRAMA DE ACTIVIDAD: Buscar Tipo Producto**Figura N° 34: Diagrama de Actividad: Buscar Tipo Producto**

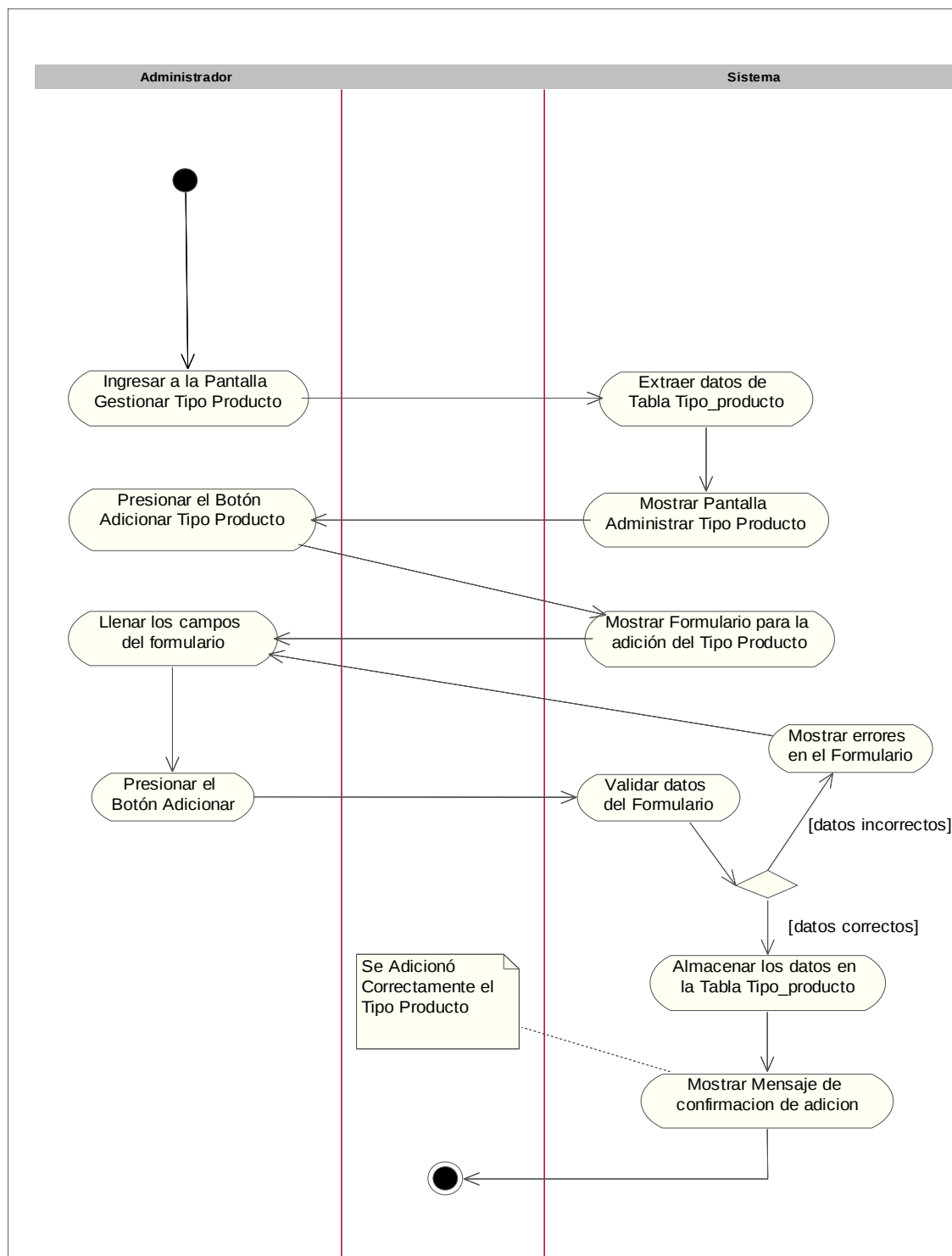
DIAGRAMA DE ACTIVIDAD: Adicionar Tipo Producto**Figura N° 35:** Diagrama de Actividad: Adicionar Tipo Producto

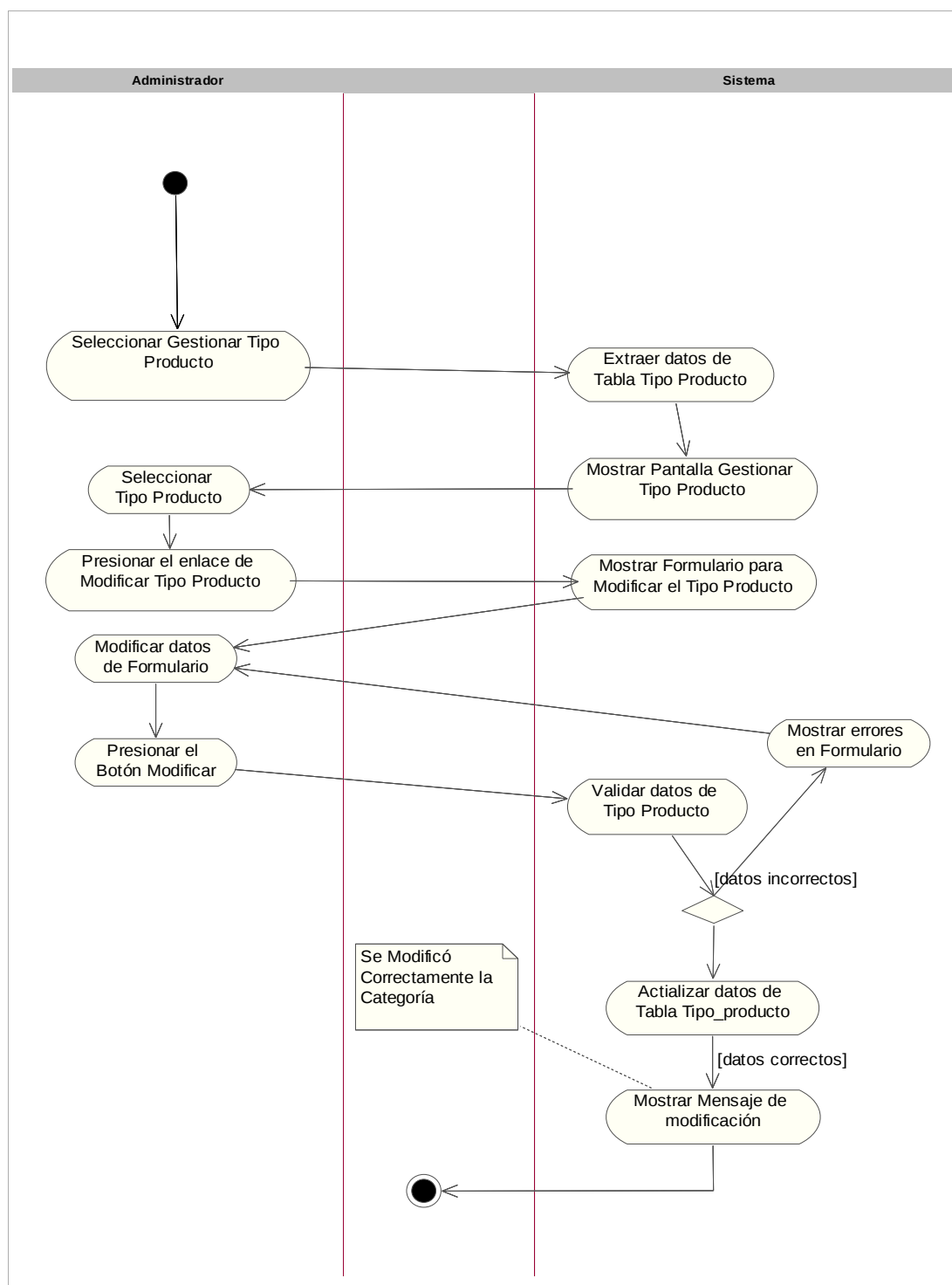
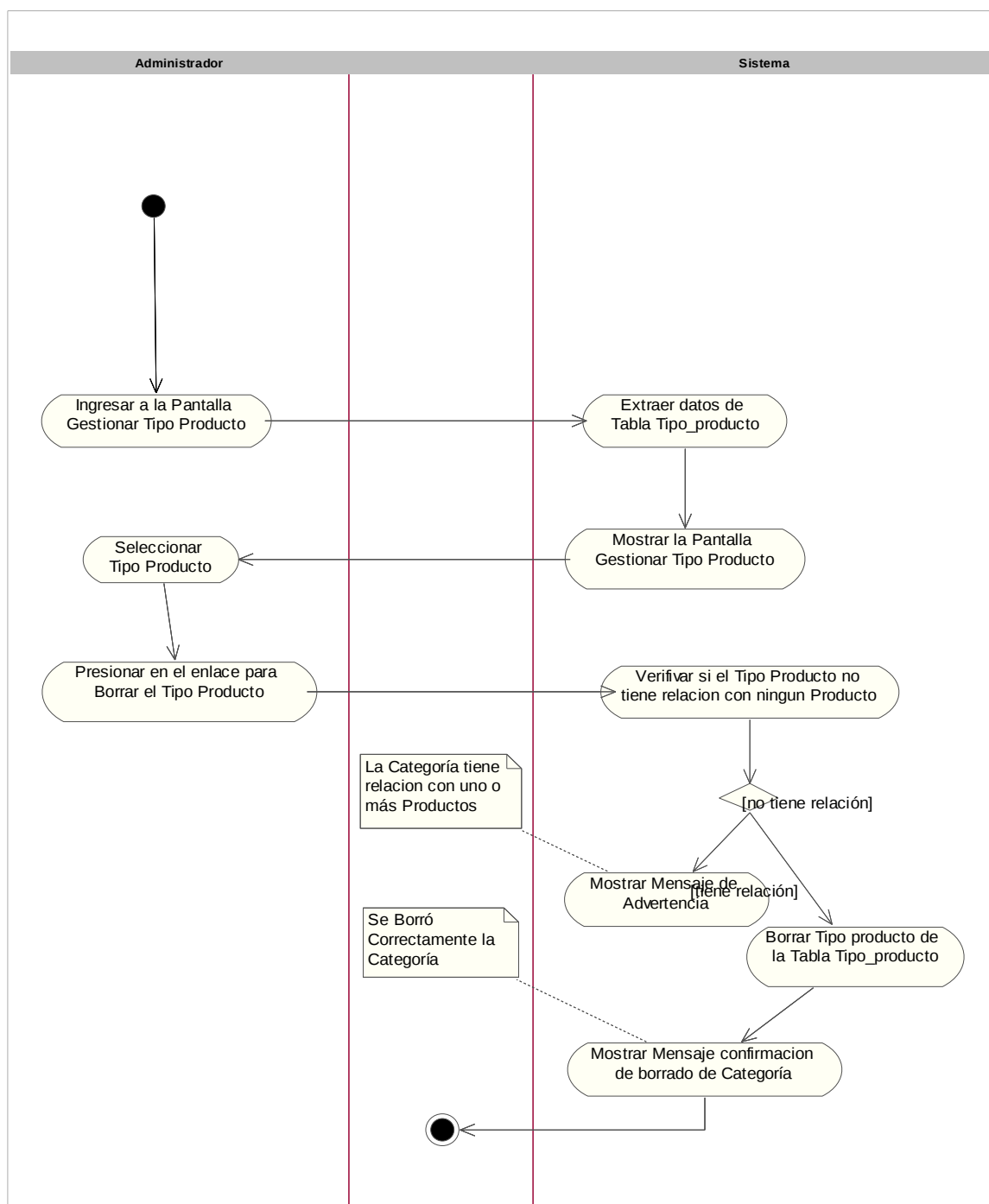
DIAGRAMA DE ACTIVIDAD: Modificar Tipo Producto**Figura N° 36:** Diagrama de Actividad: Modificar Tipo Producto

DIAGRAMA DE ACTIVIDAD: Dar de Baja Tipo Producto**Figura N° 37: Diagrama de Actividad: Dar de Baja Tipo Producto**

MODULO GESTIONAR CLIENTE

DIAGRAMA DE ACTIVIDAD: Listar Cliente

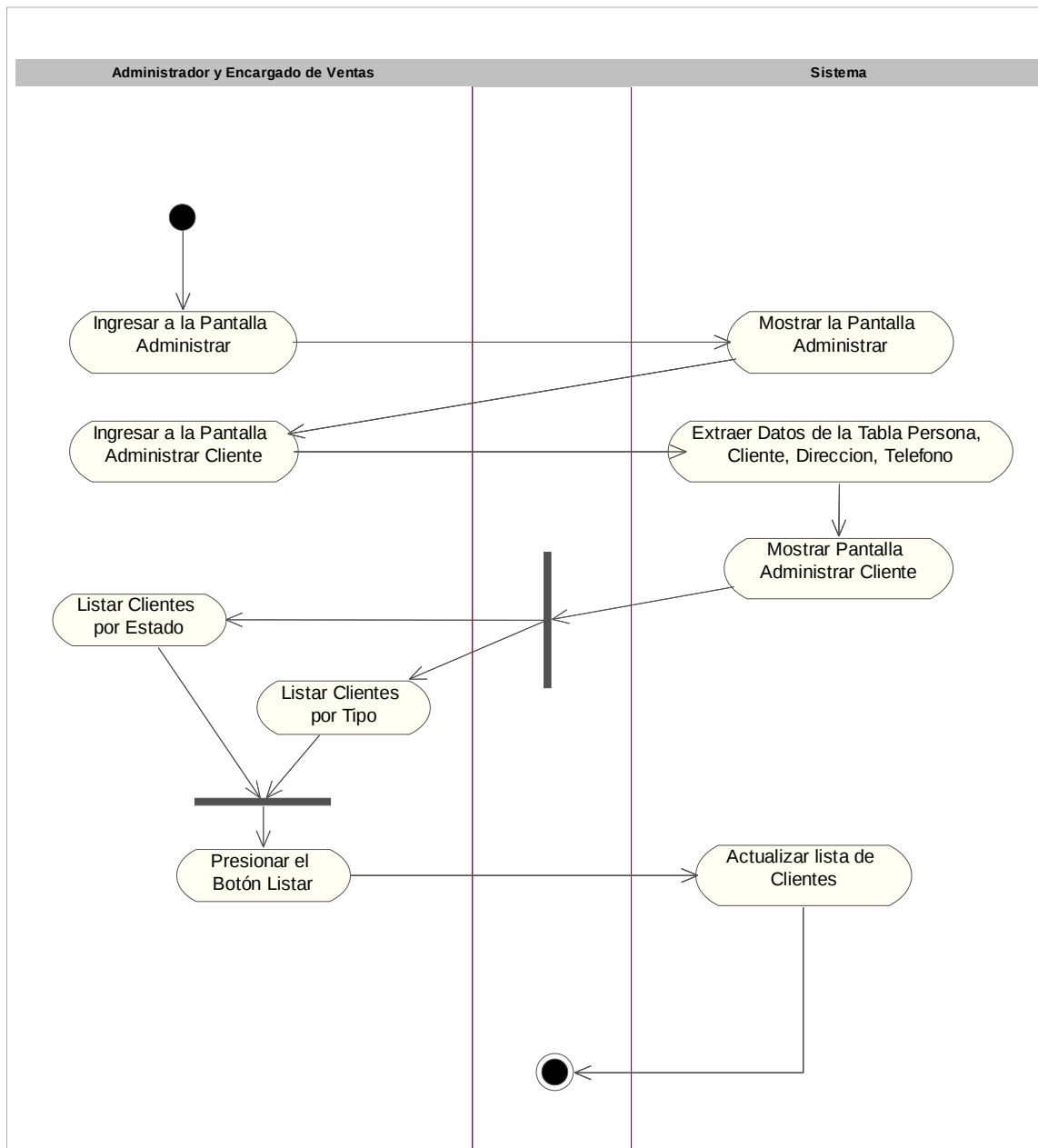


Figura N° 38: Diagrama de Actividad: Listar Cliente

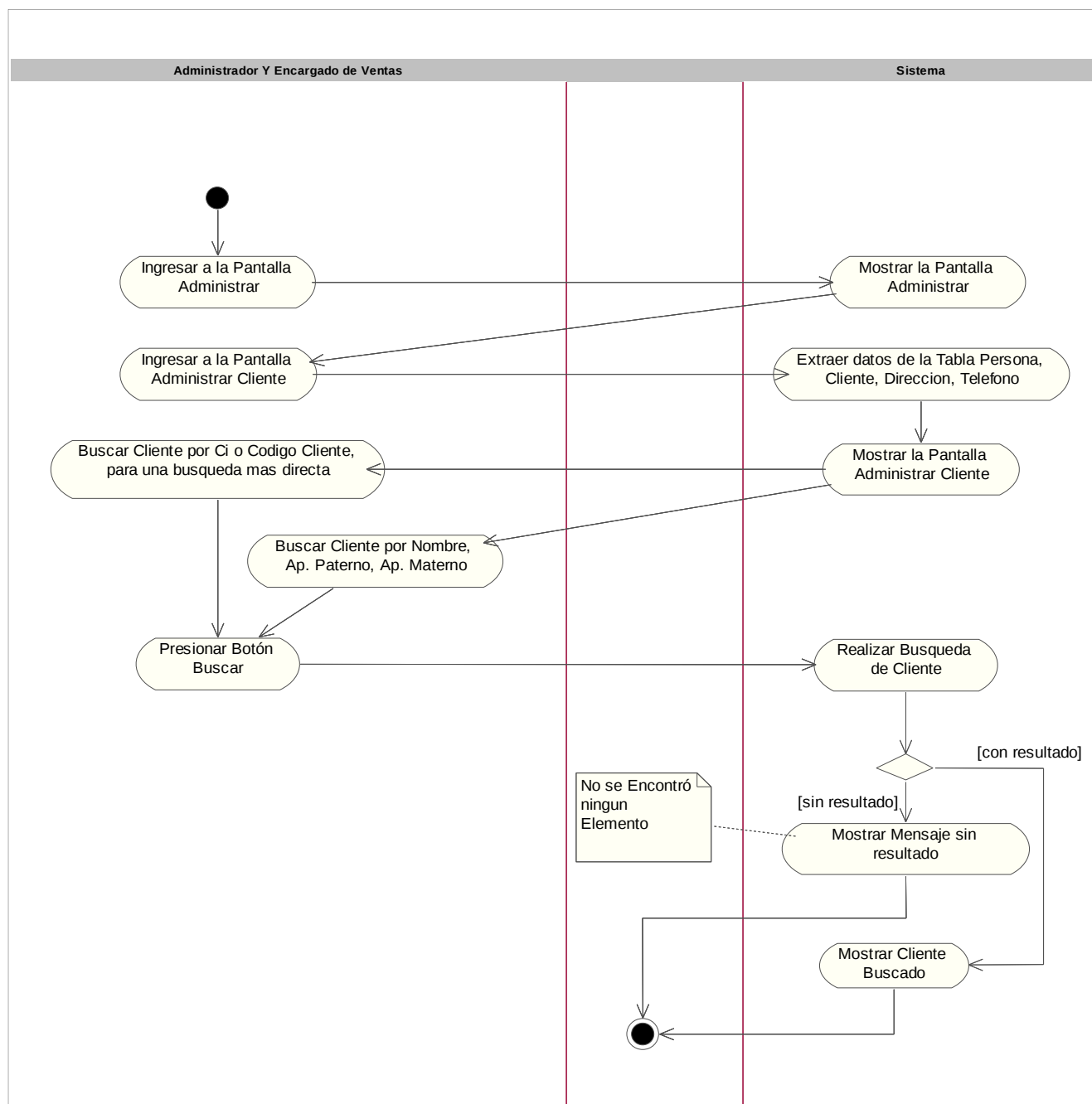
DIAGRAMA DE ACTIVIDAD: Buscar Cliente**Figura N° 39: Diagrama de Actividad: Buscar Cliente**

DIAGRAMA DE ACTIVIDAD: Adicionar Cliente

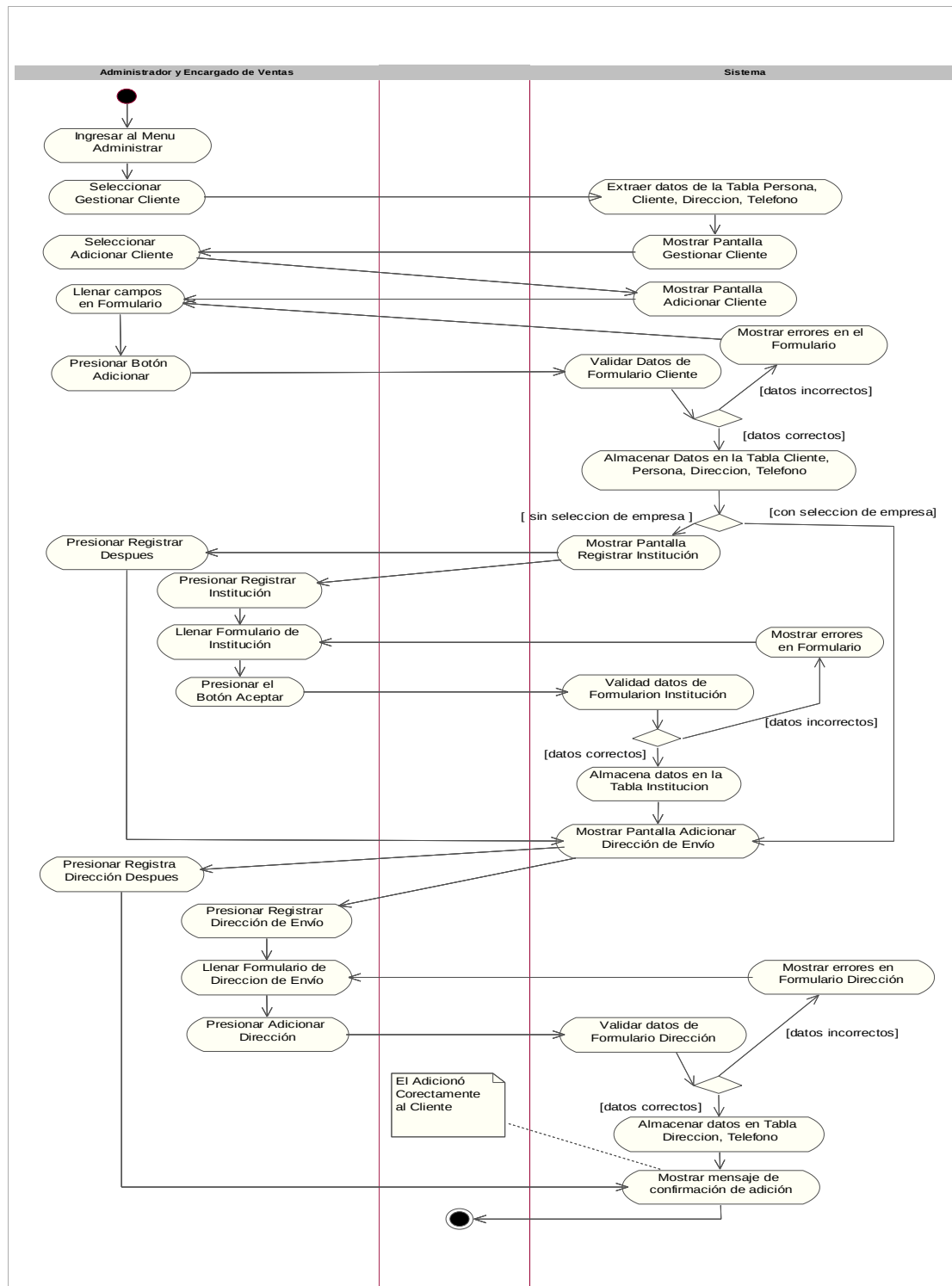


Figura N° 40: Diagrama de Actividad: Adicionar Cliente

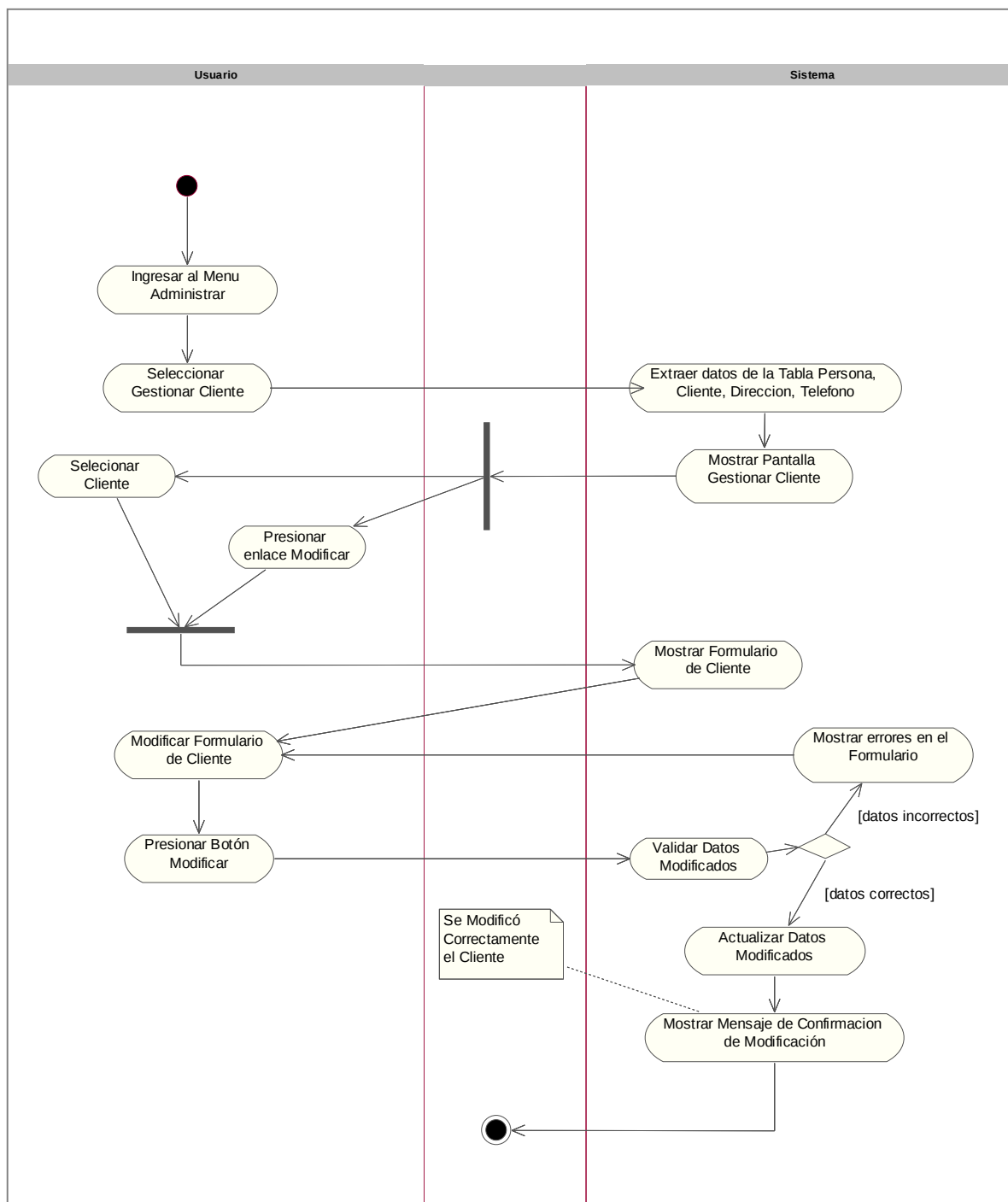
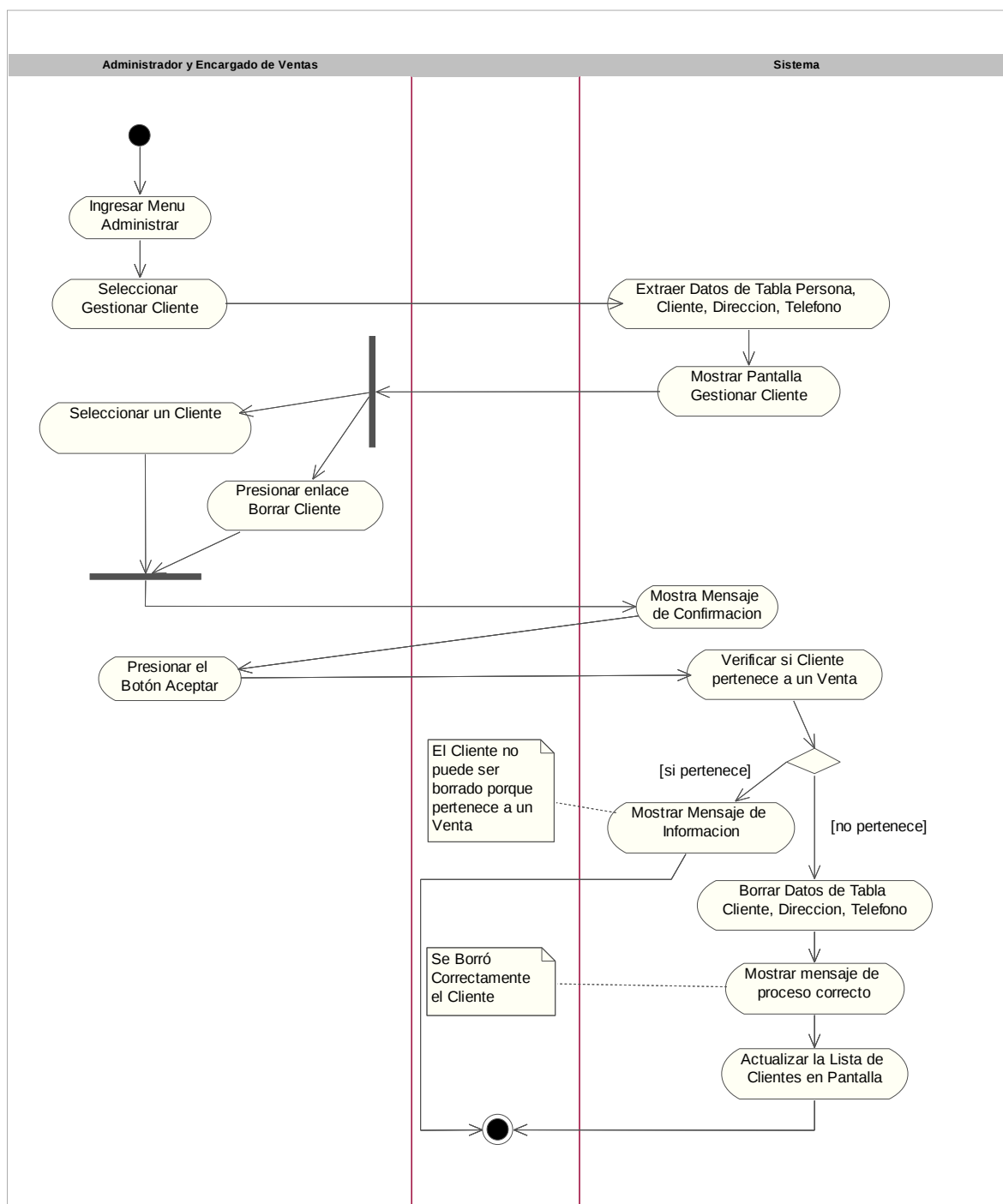
DIAGRAMA DE ACTIVIDAD: Modificar Cliente**Figura N° 41:** Diagrama de Actividad: Modificar Cliente

DIAGRAMA DE ACTIVIDAD: Dar de Baja Cliente**Figura N° 42:** Diagrama de Actividad: Dar de Baja Cliente

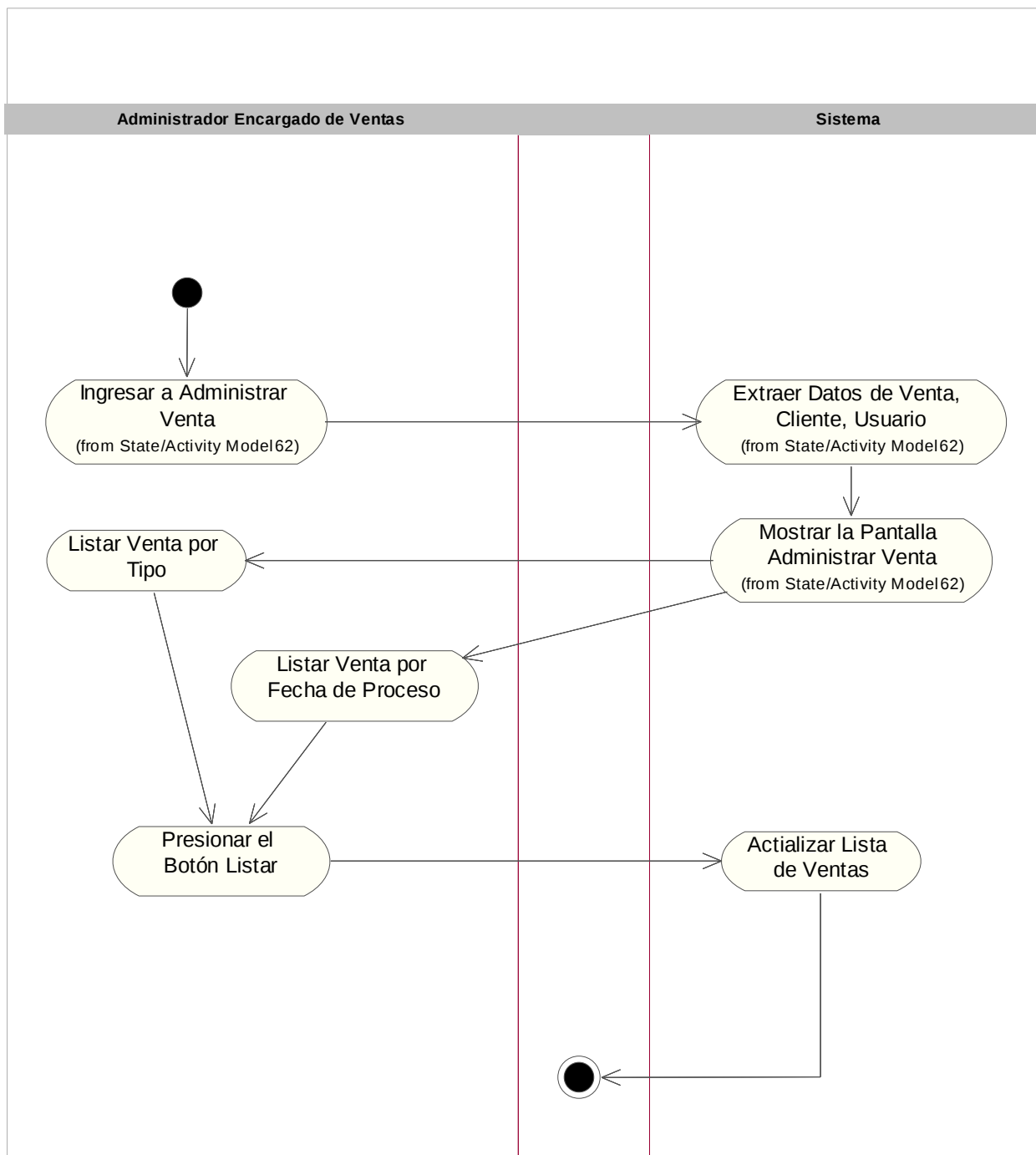
MODULO GESTIONAR VENTA**DIAGRAMA DE ACTIVIDAD:** Listar Venta**Figura N° 43:** Diagrama de Actividad: Listar Venta

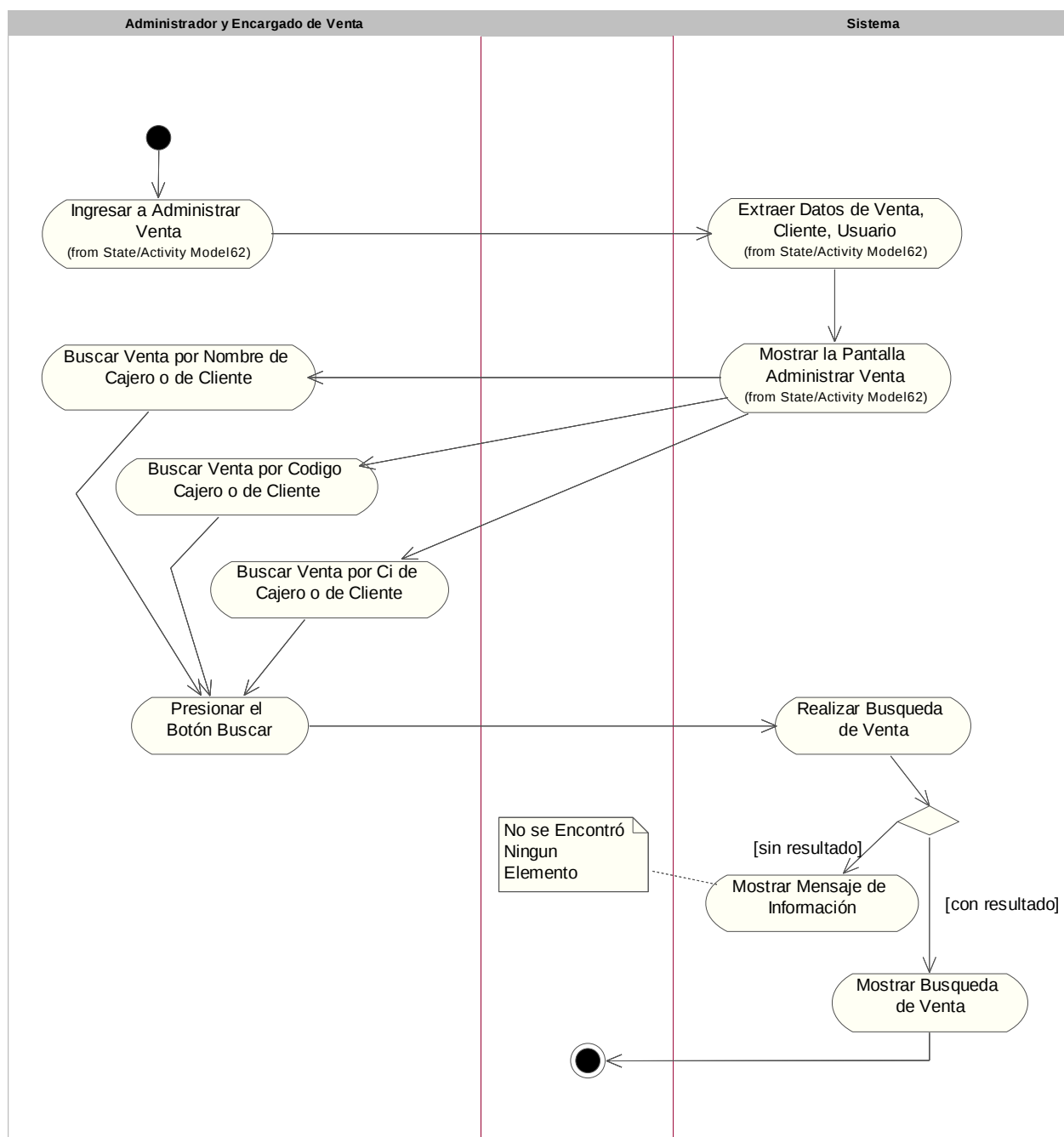
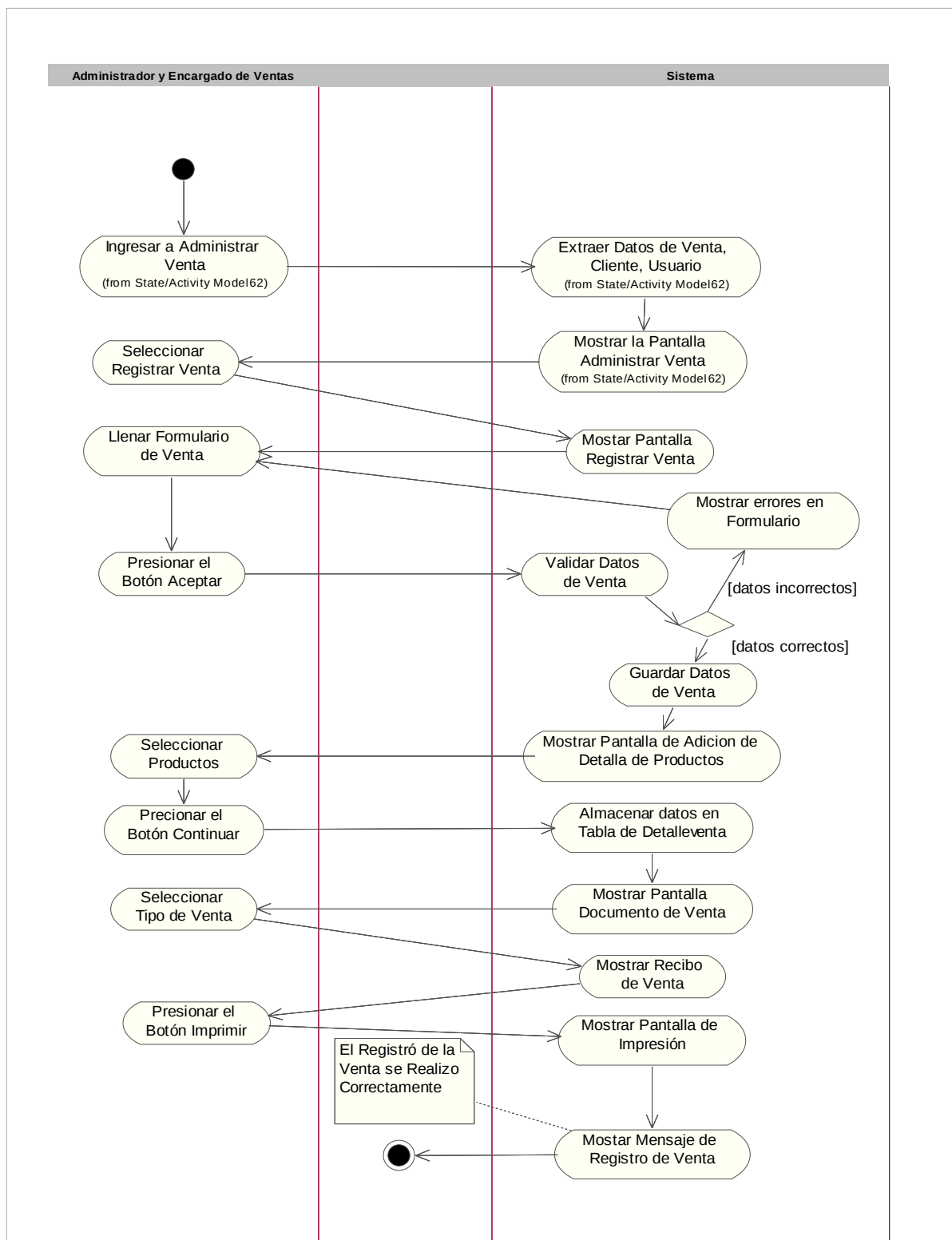
DIAGRAMA DE ACTIVIDAD: Buscar Venta**Figura N° 44:** Diagrama de Actividad: Buscar Venta

DIAGRAMA DE ACTIVIDAD: Registrar Venta**Figura N° 45: Diagrama de Actividad: Registrar Venta**

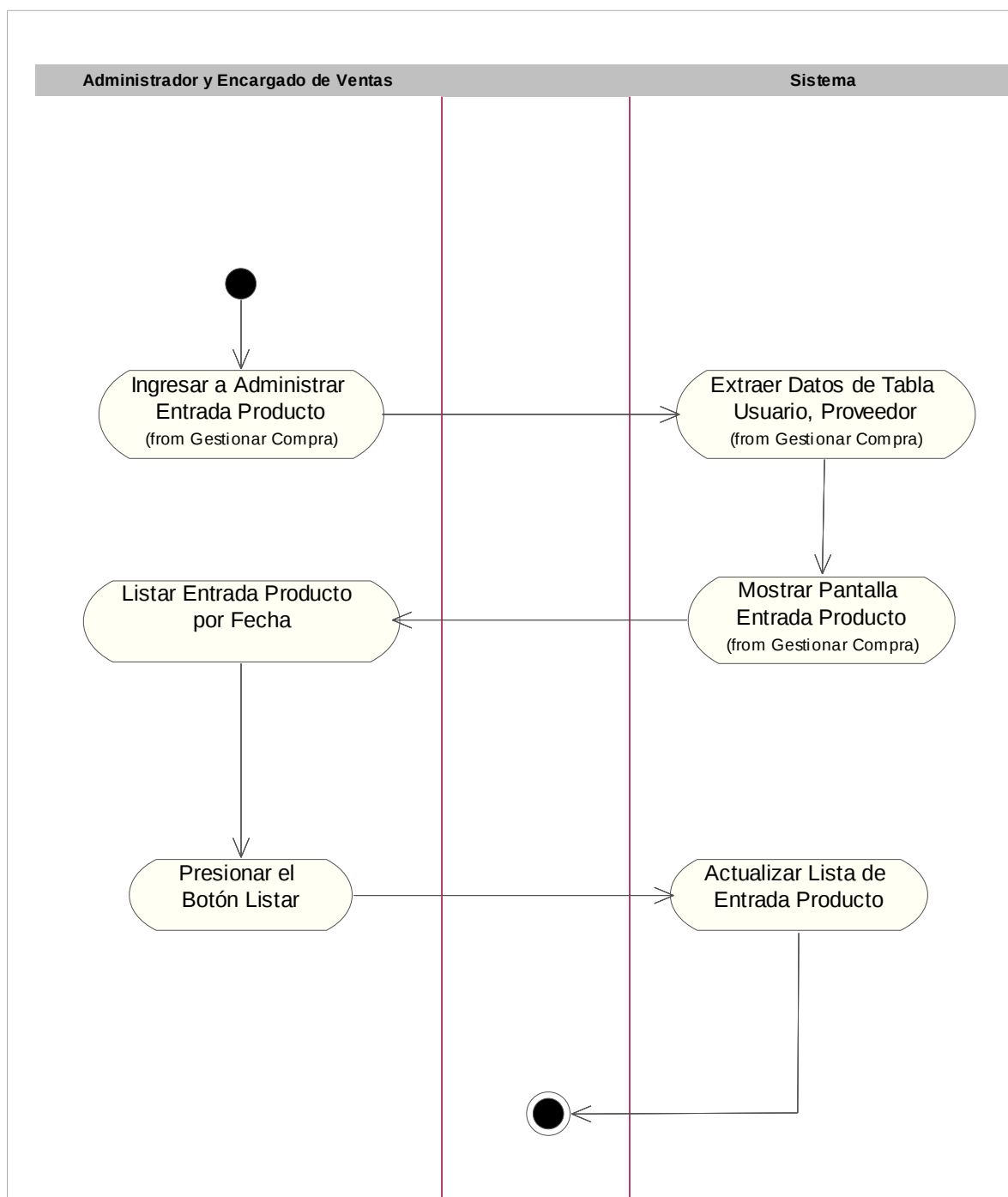
MODULO GESTIONAR ENTRADA PRODUCTO**DIAGRAMA DE ACTIVIDAD:** Listar Entrada Producto**Figura N° 46:** Diagrama de Actividad: Listar Entrada Producto

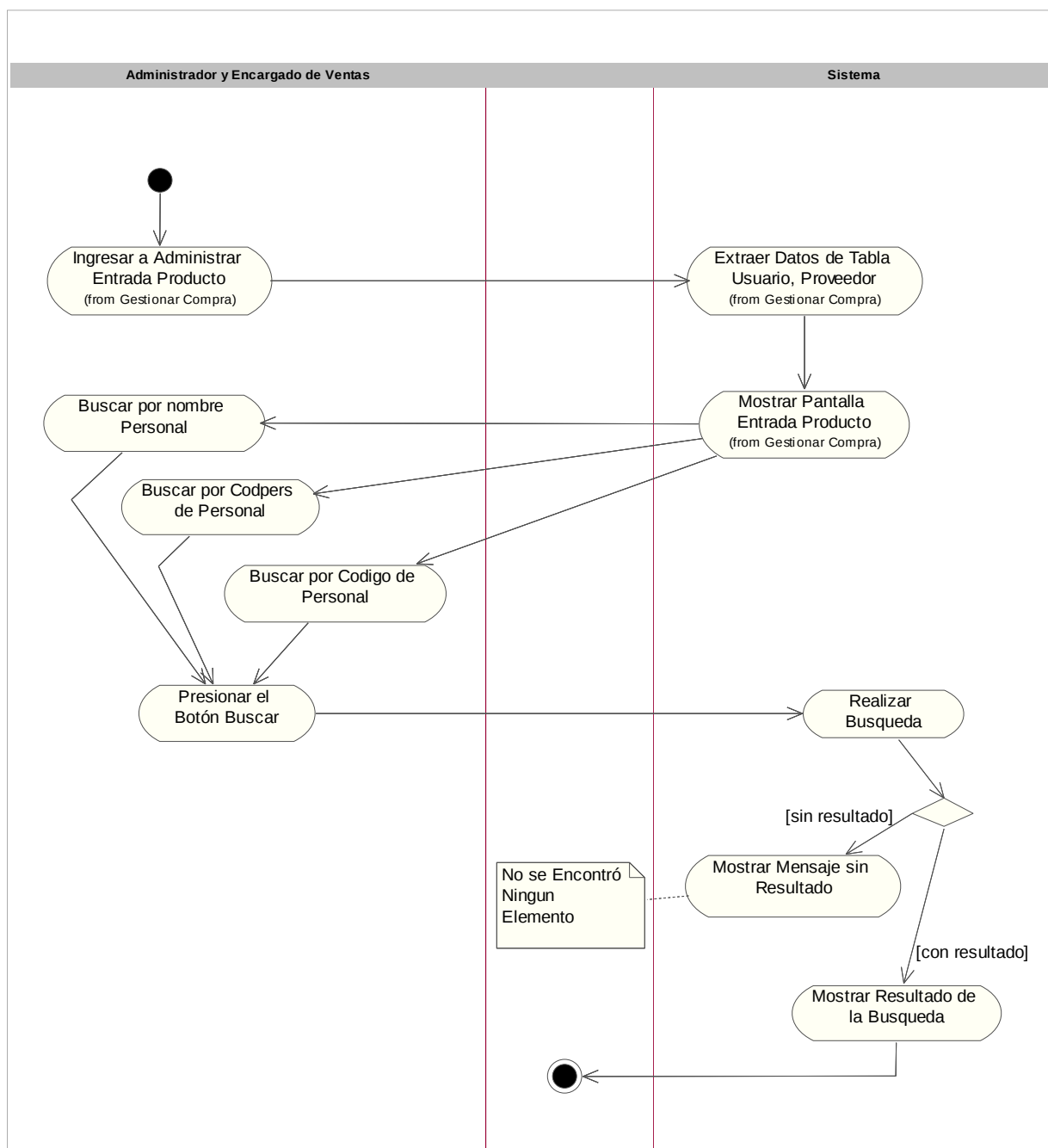
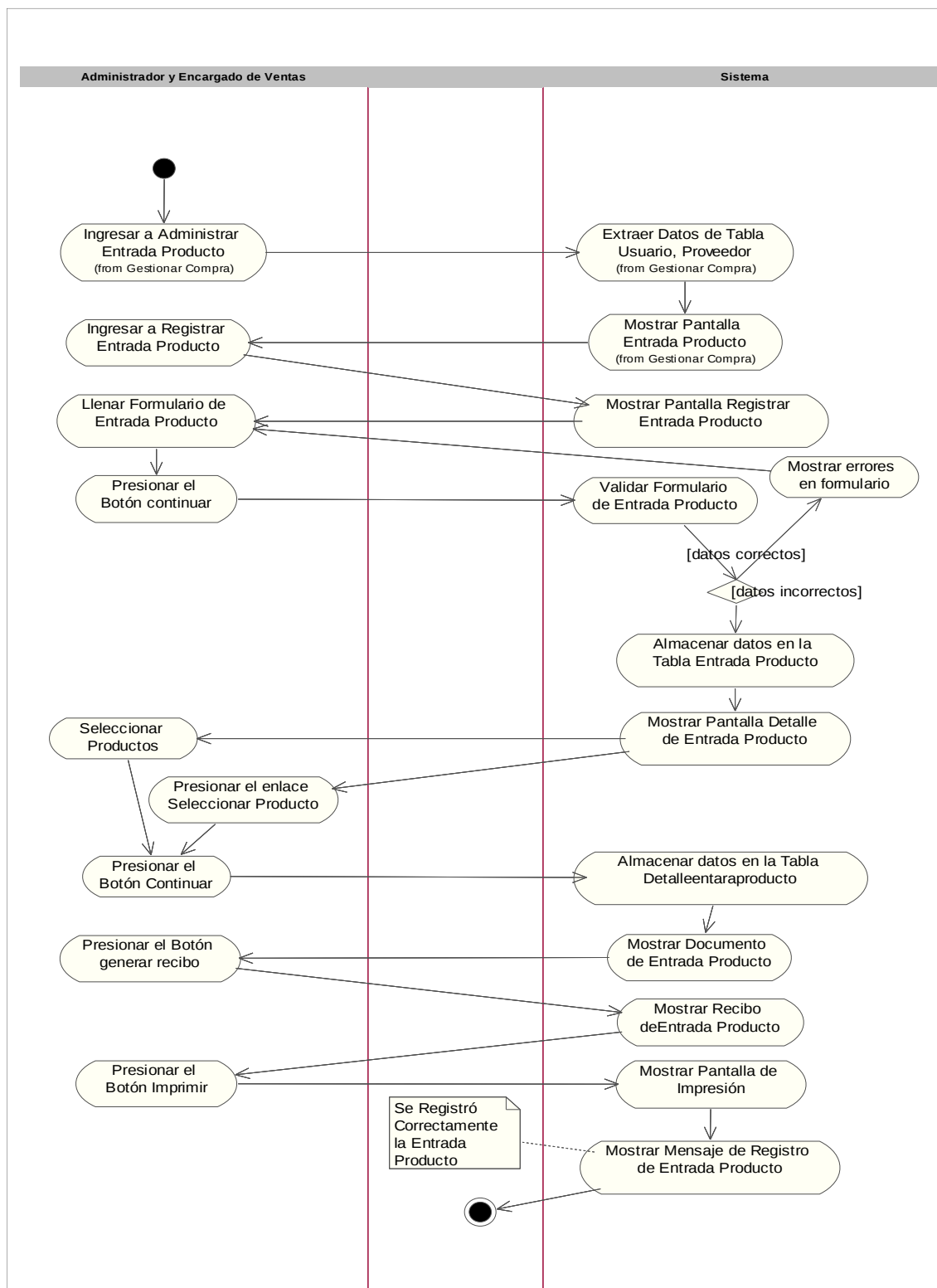
DIAGRAMA DE ACTIVIDAD: Buscar Entrada Producto**Figura N° 47: Diagrama de Actividad: Buscar Entrada Producto**

DIAGRAMA DE ACTIVIDAD: Registrar Entrada Producto**Figura N° 48:** Diagrama de Actividad: Registrar Compra

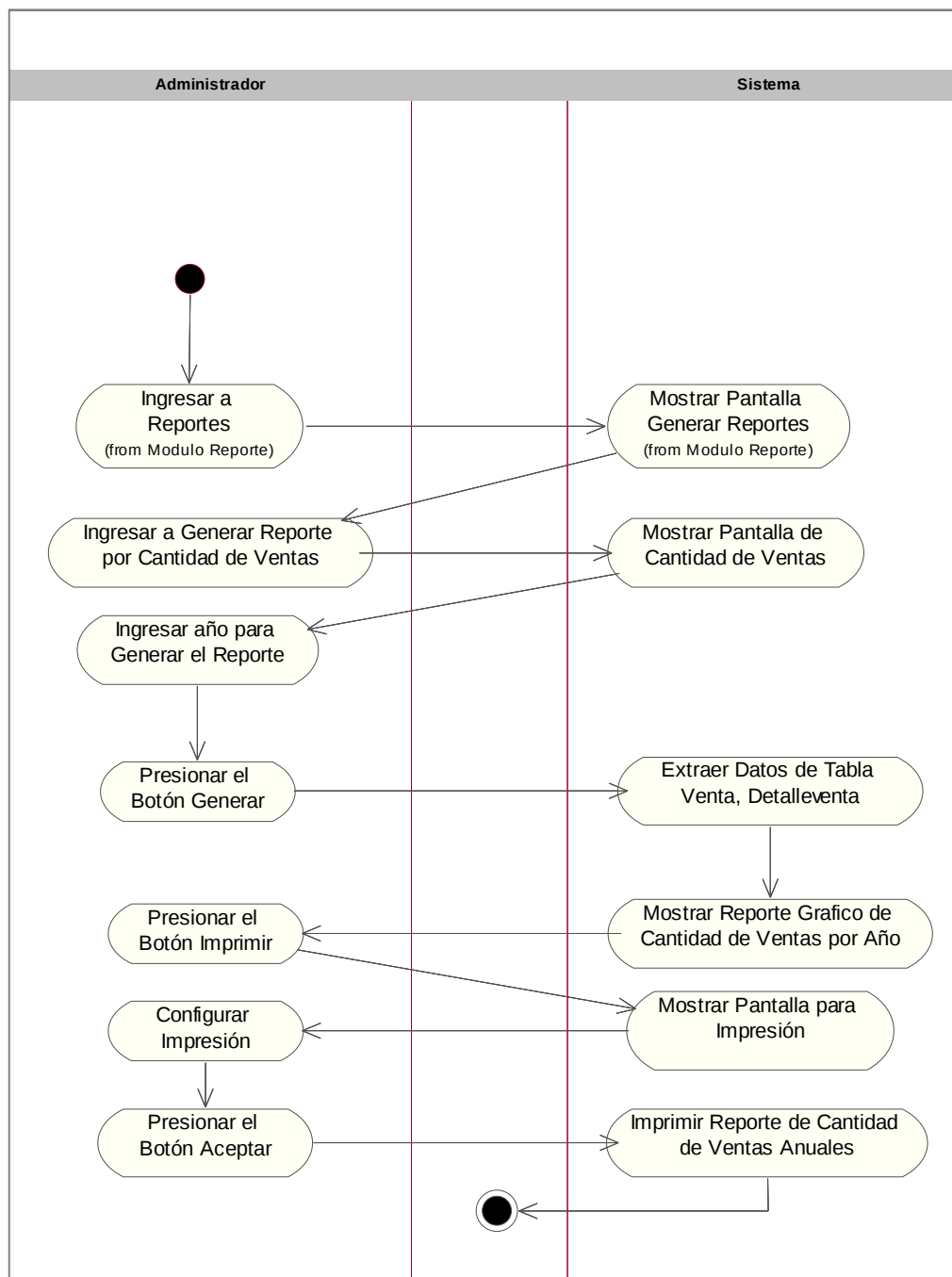
MODULO REPORTE
DIAGRAMA DE ACTIVIDAD: Generar Reporte de Cantidad de Ventas

Figura N° 49: Diagrama de Actividad: Generar Reporte de Cantidad de Ventas

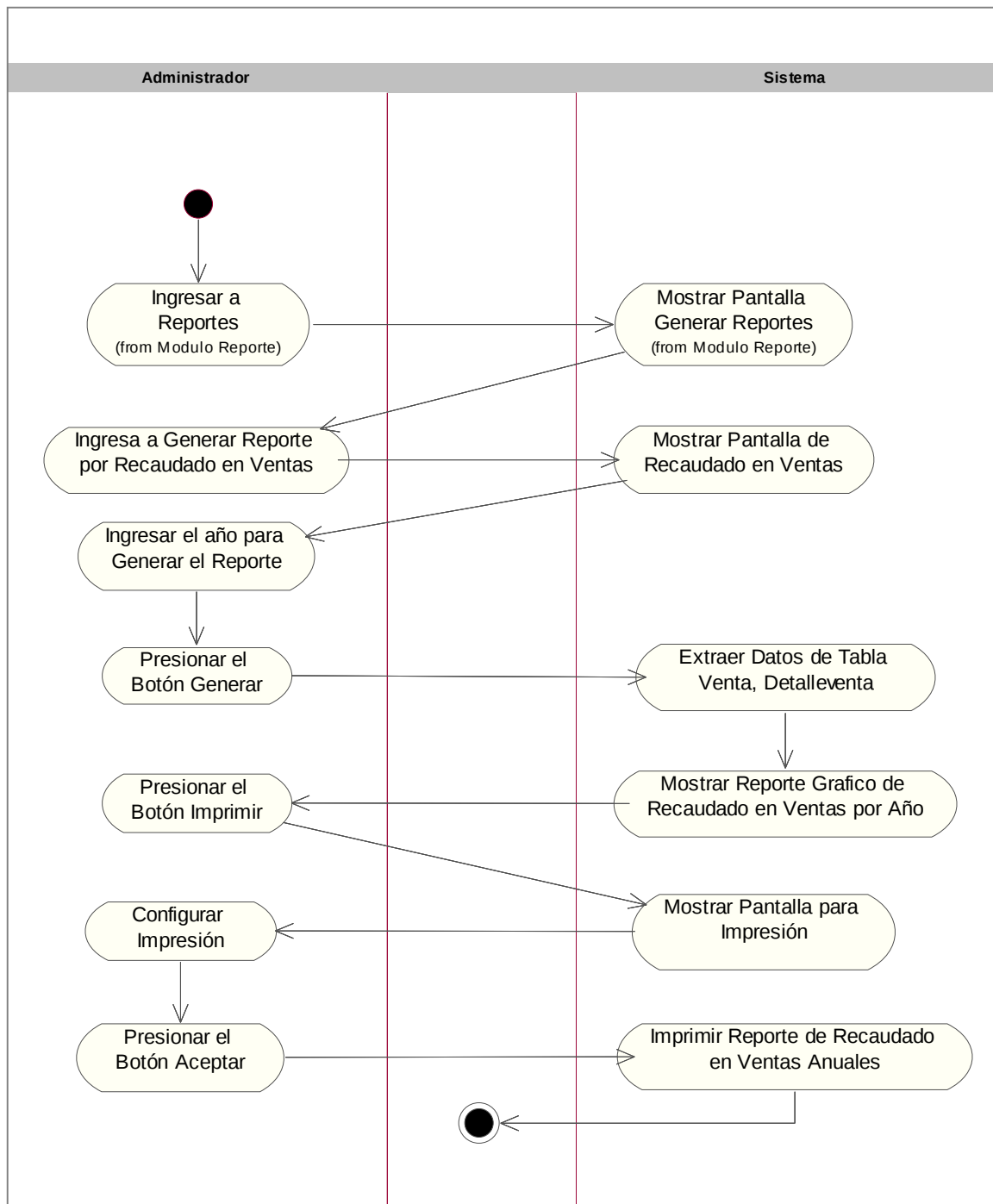
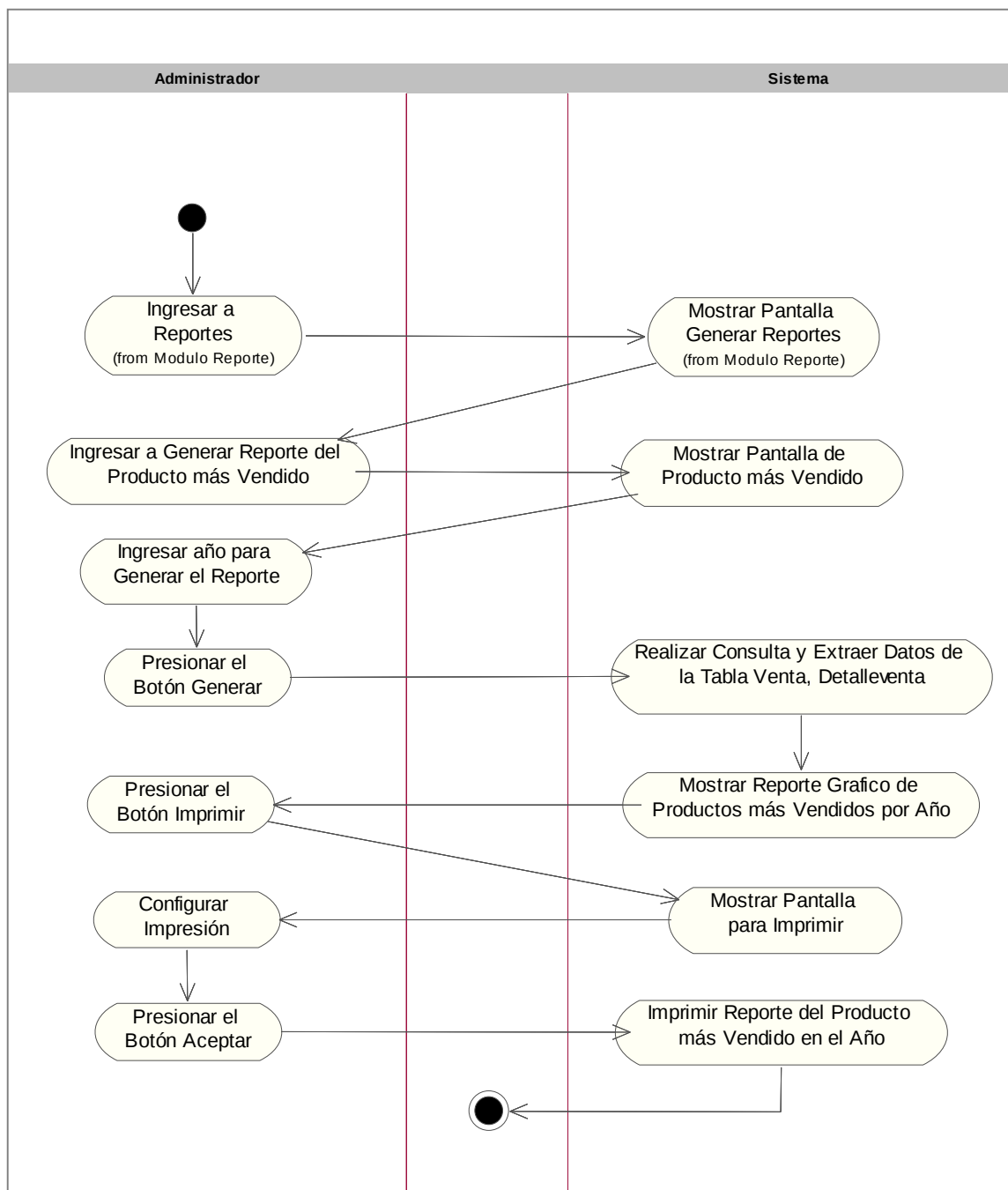
DIAGRAMA DE ACTIVIDAD: Generar Reporte de Recaudado en Ventas**Figura N° 50:** Diagrama de Actividad: Generar Reporte de Recaudado en Ventas

DIAGRAMA DE ACTIVIDAD: Generar Reporte de Producto más Vendido**Figura N° 51:** Diagrama de Actividad: Generar Reporte de Producto más Vendido

MODULO GESTIONAR MANTENIMIENTO

DIAGRAMA DE ACTIVIDAD: Resguardar Base de Datos.

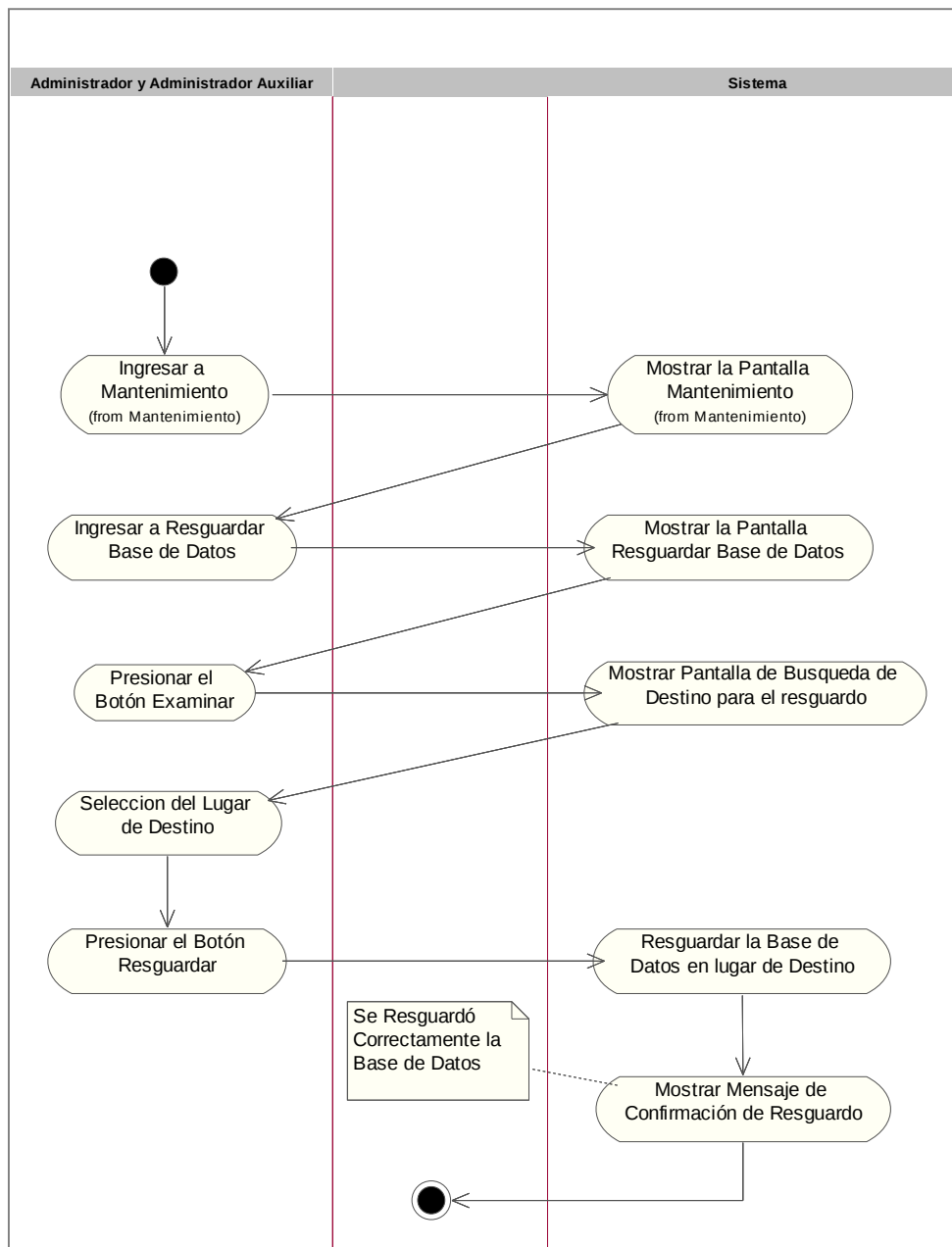
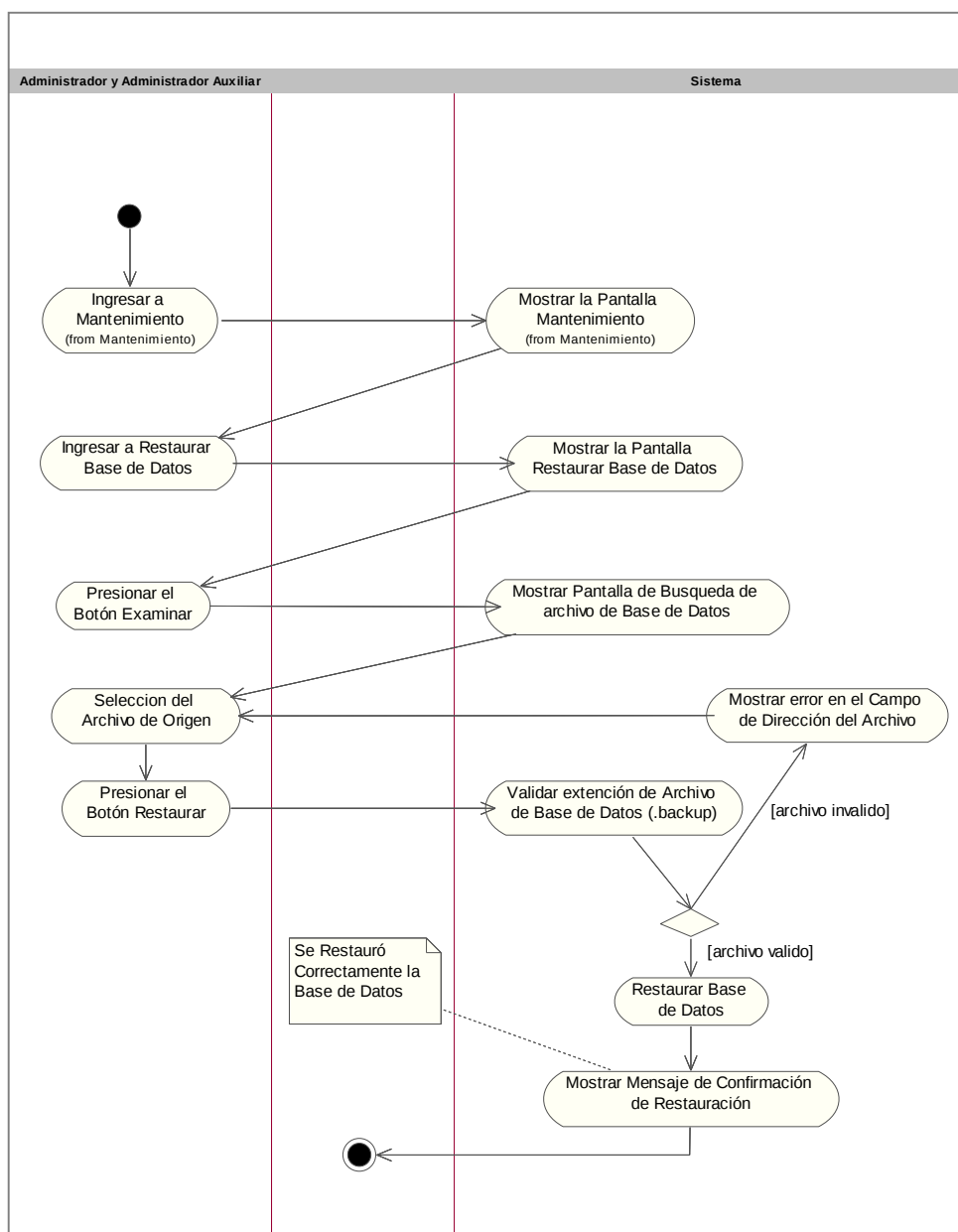


Figura N° 52: Diagrama de Actividad: Resguardar Base de Datos.

DIAGRAMA DE ACTIVIDAD: Restaurar Base de Datos.**Figura N° 53:** Diagrama de Actividad: Restaurar Base de Datos.

II.1.10. MODELADO DE DIAGRAMA DE INTERACCION

II.1.10.1. Introducción

Los diagramas de interacción consisten en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

II.1.10.1.1. Propósito

- Comprender la dinámica del sistema deseado para la organización.
- Identificar clases de análisis y diseño.

II.1.10.1.2. Alcance

- Describir la dinámica de sistema en el tiempo de vida de las clases u objetos
- Definir un ***diagrama de secuencia*** para cada caso de uso del usuario Administrador.
- Definir un ***diagrama de Colaboración*** para cada caso de uso del sistema.

II.1.10.2. Diagrama de Secuencias

Diagrama de Secuencia: Ingresar al Sistema

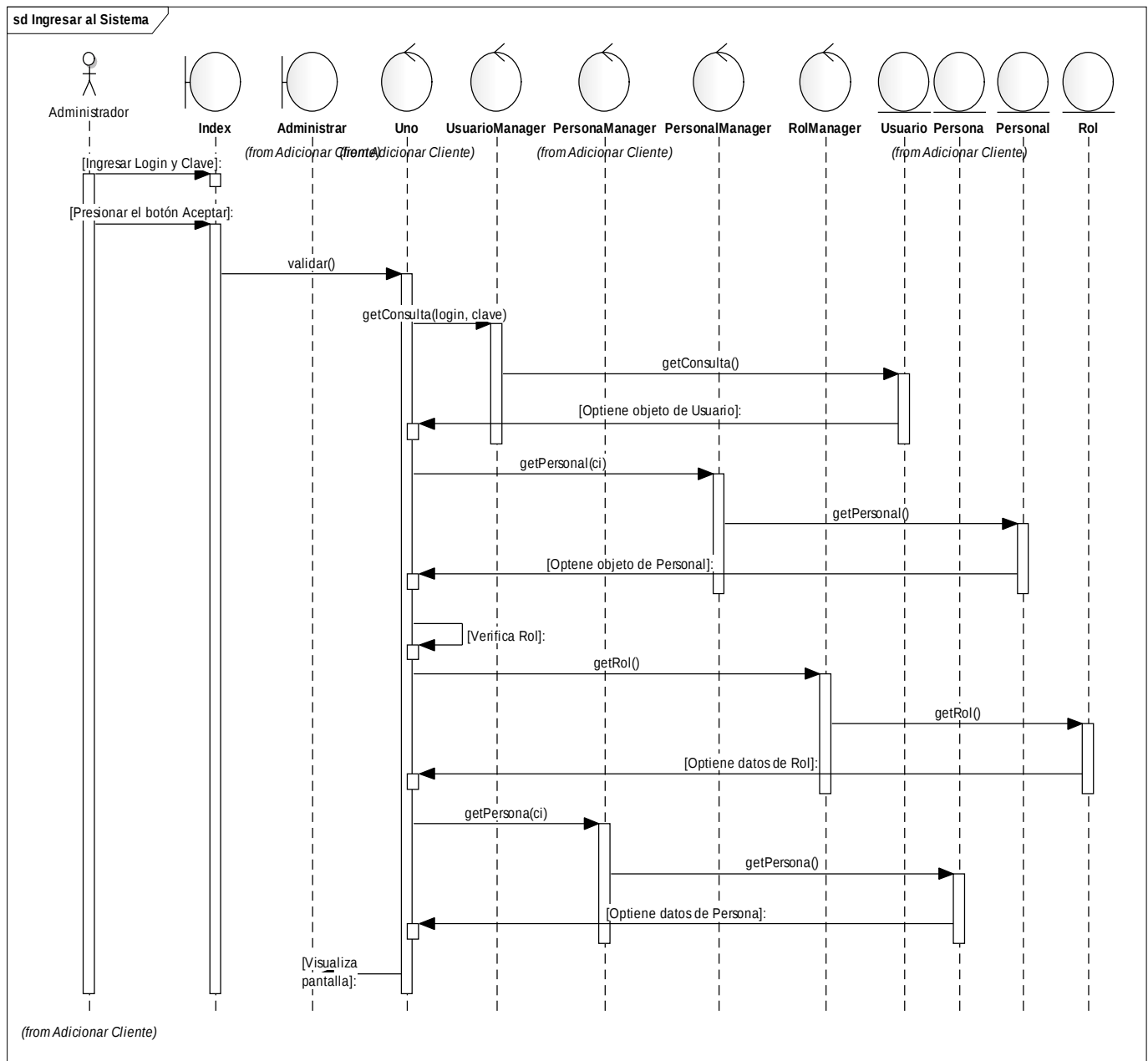


Figura N° 54: Diagrama de Secuencia: Ingresar al Sistema.

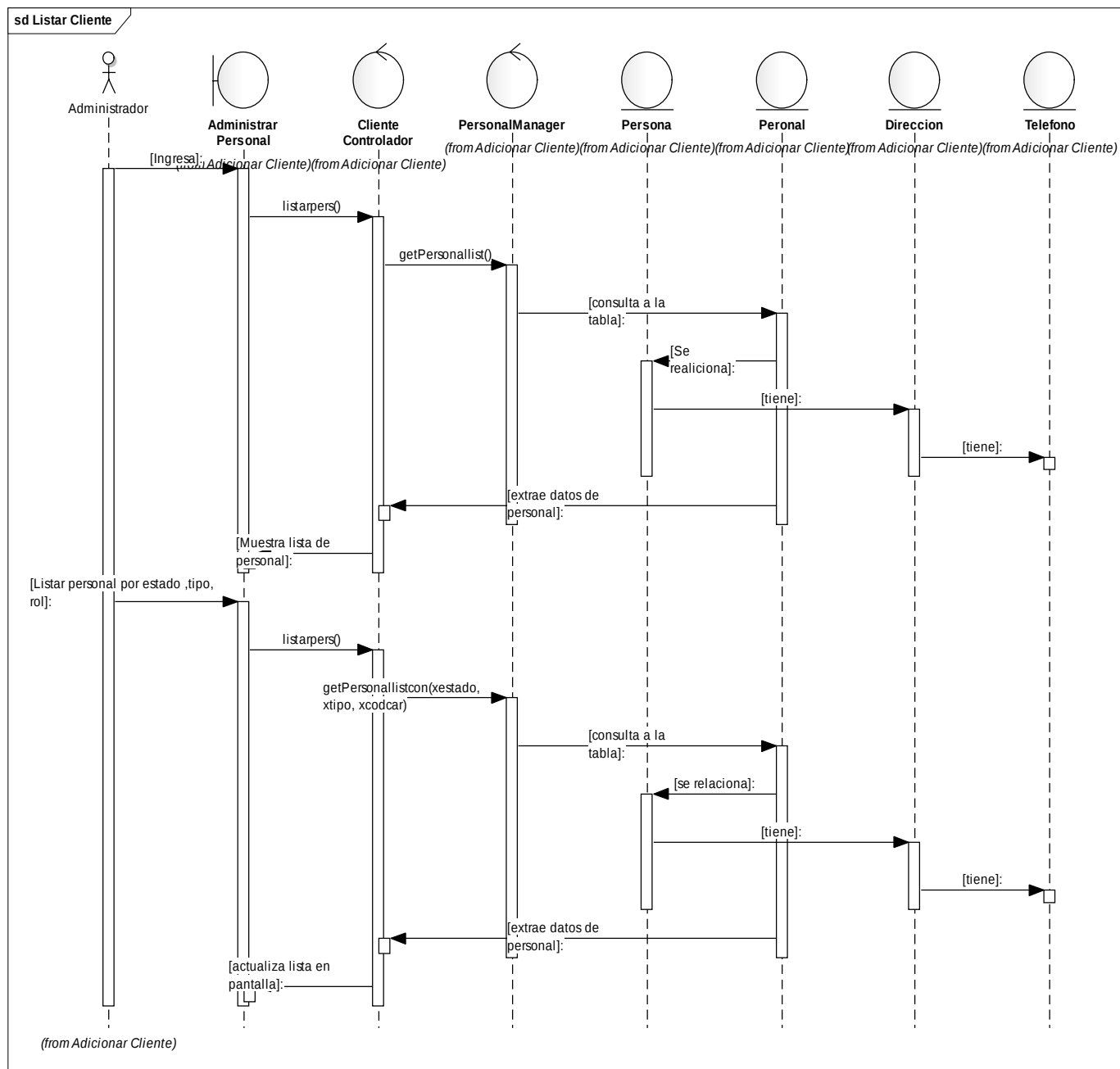
DIAGRAMA DE SECUENCIA: Listar Cliente.**Figura N° 55:** Diagrama de Secuencia: Listar Cliente

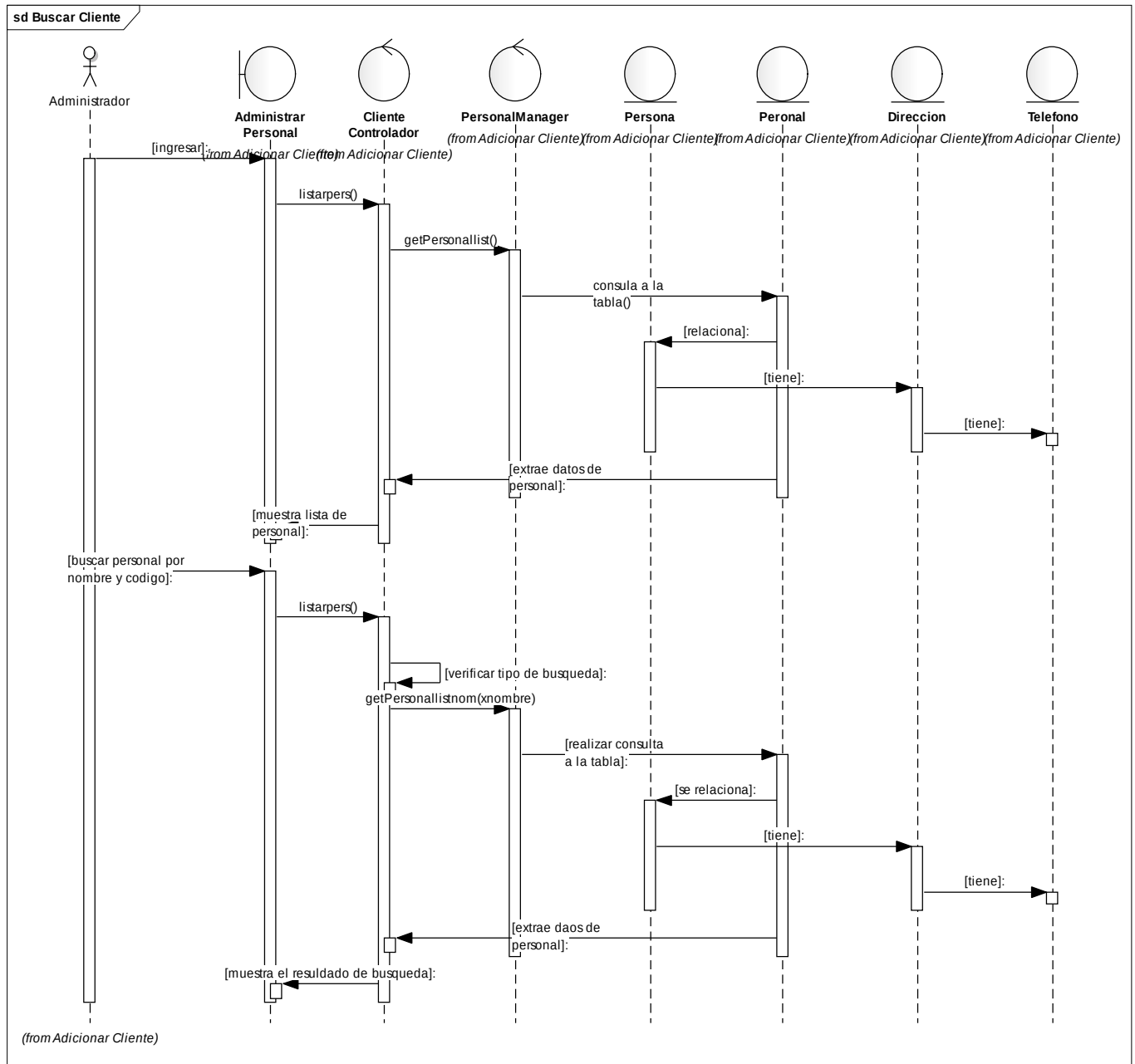
DIAGRAMA DE SECUENCIA: Buscar Cliente.**Figura N° 56: Diagrama de Secuencia: Buscar Cliente.**

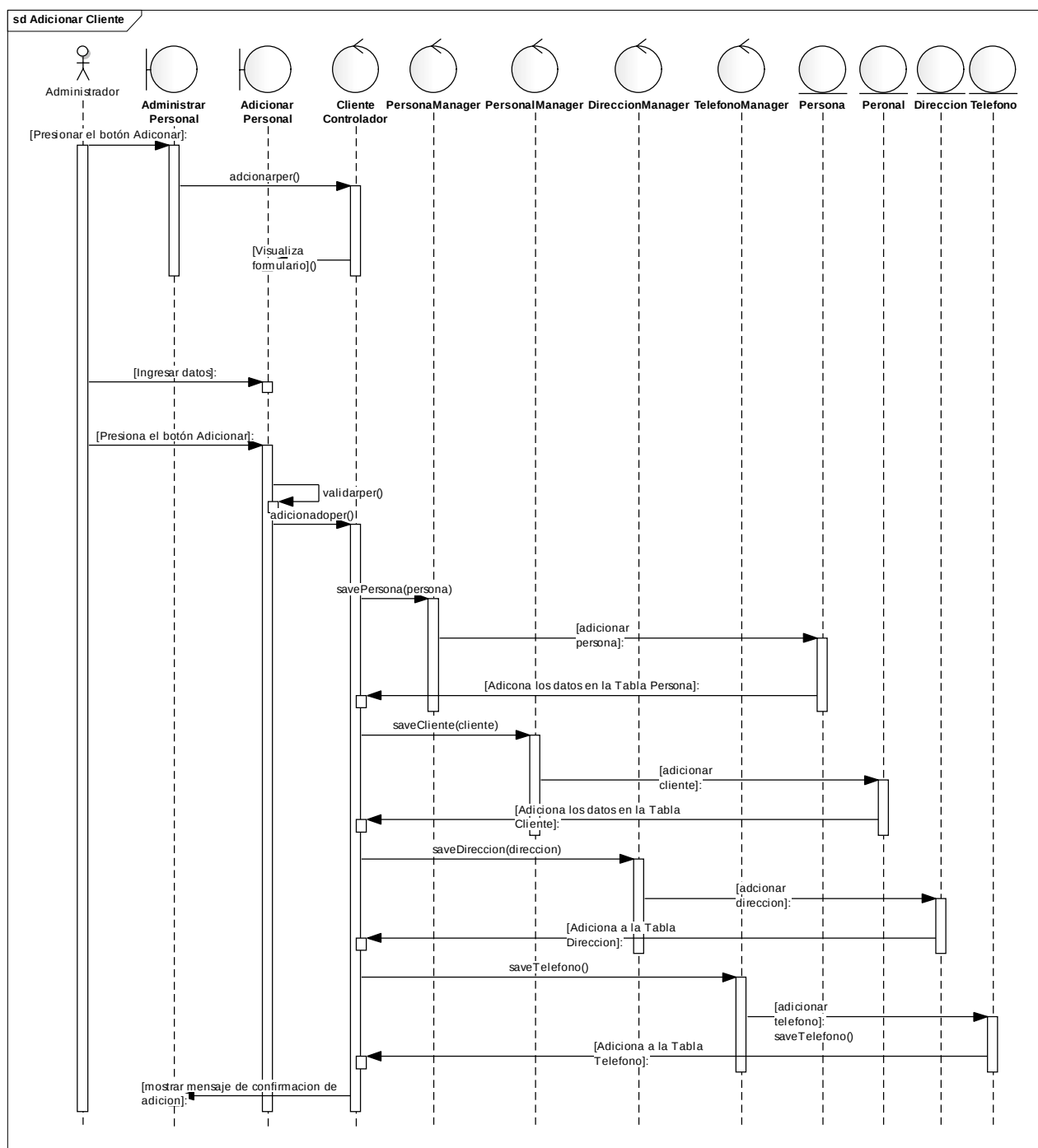
DIAGRAMA DE SECUENCIA: Adicionar Cliente.**Figura N° 57: Diagrama de Secuencia: Adicionar Cliente**

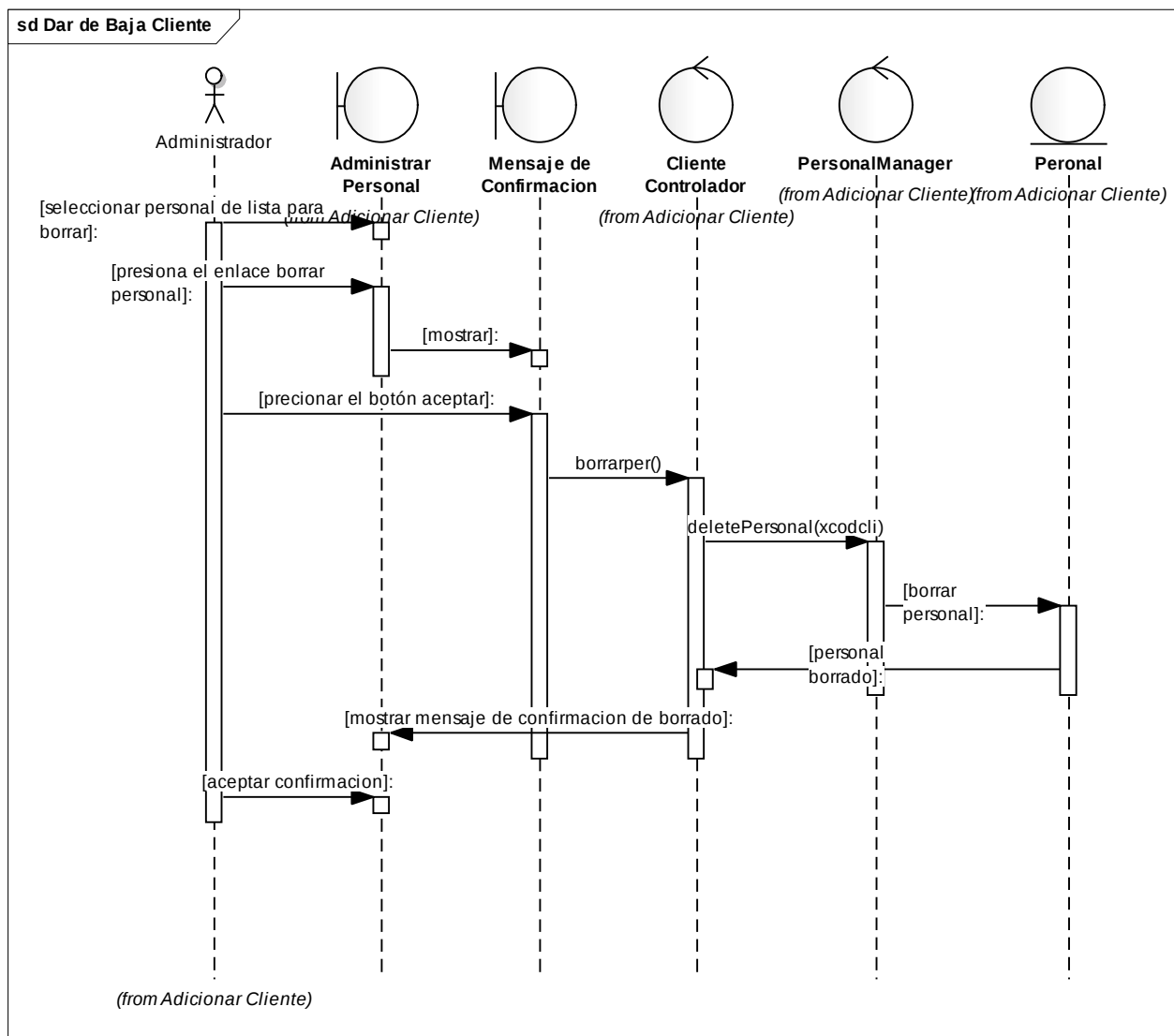
DIAGRAMA DE SECUENCIA: Dar de Baja Cliente.**Figura N° 59: Diagrama de Secuencia: Dar de Baja Cliente.**

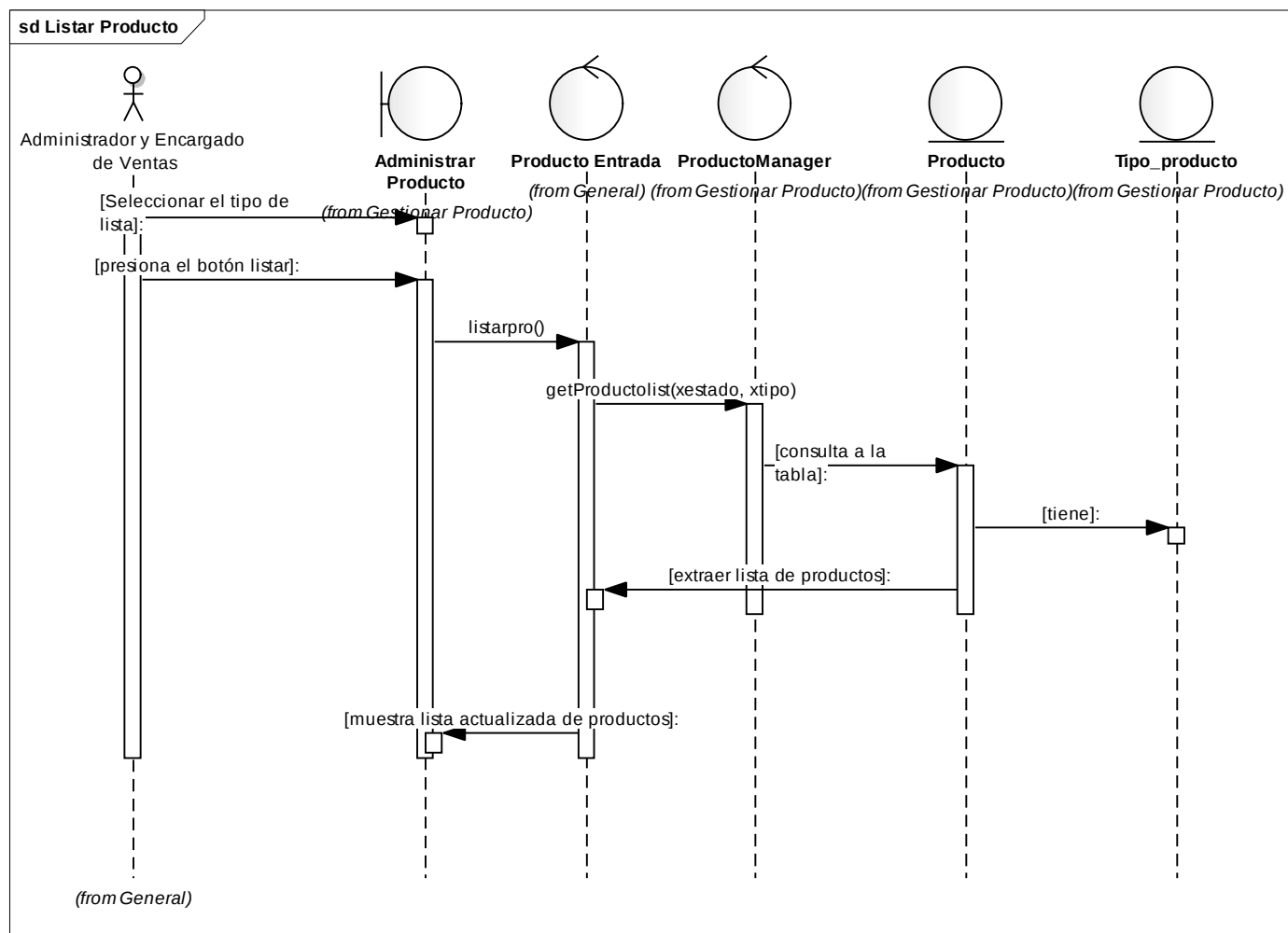
DIAGRAMA DE SECUENCIA: Listar Producto.**Figura N° 60: Diagrama de Secuencia: Listar Producto.**

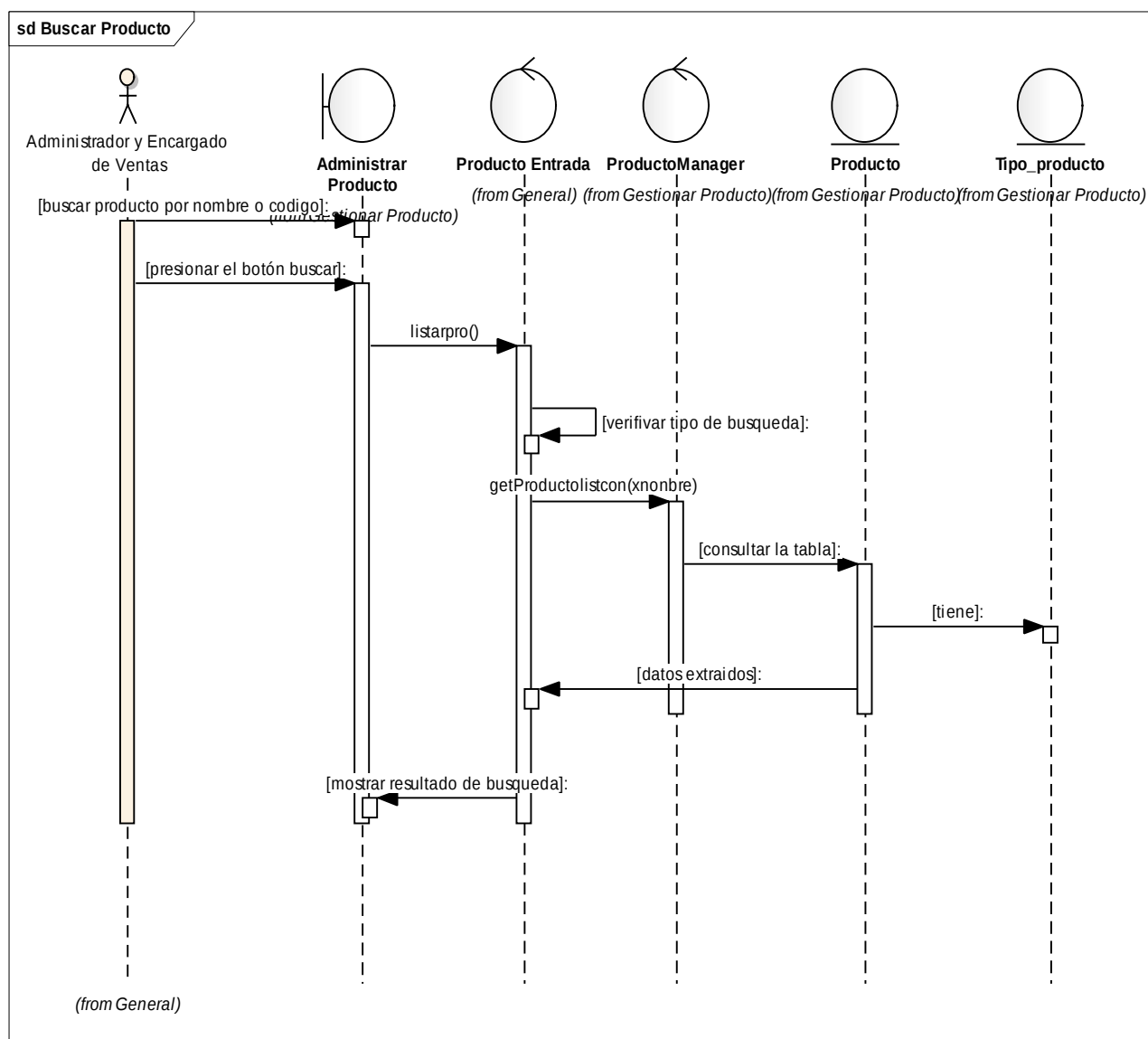
DIAGRAMA DE SECUENCIA: Buscar Producto.**Figura N° 61: Diagrama de Secuencia: Buscar Producto.**

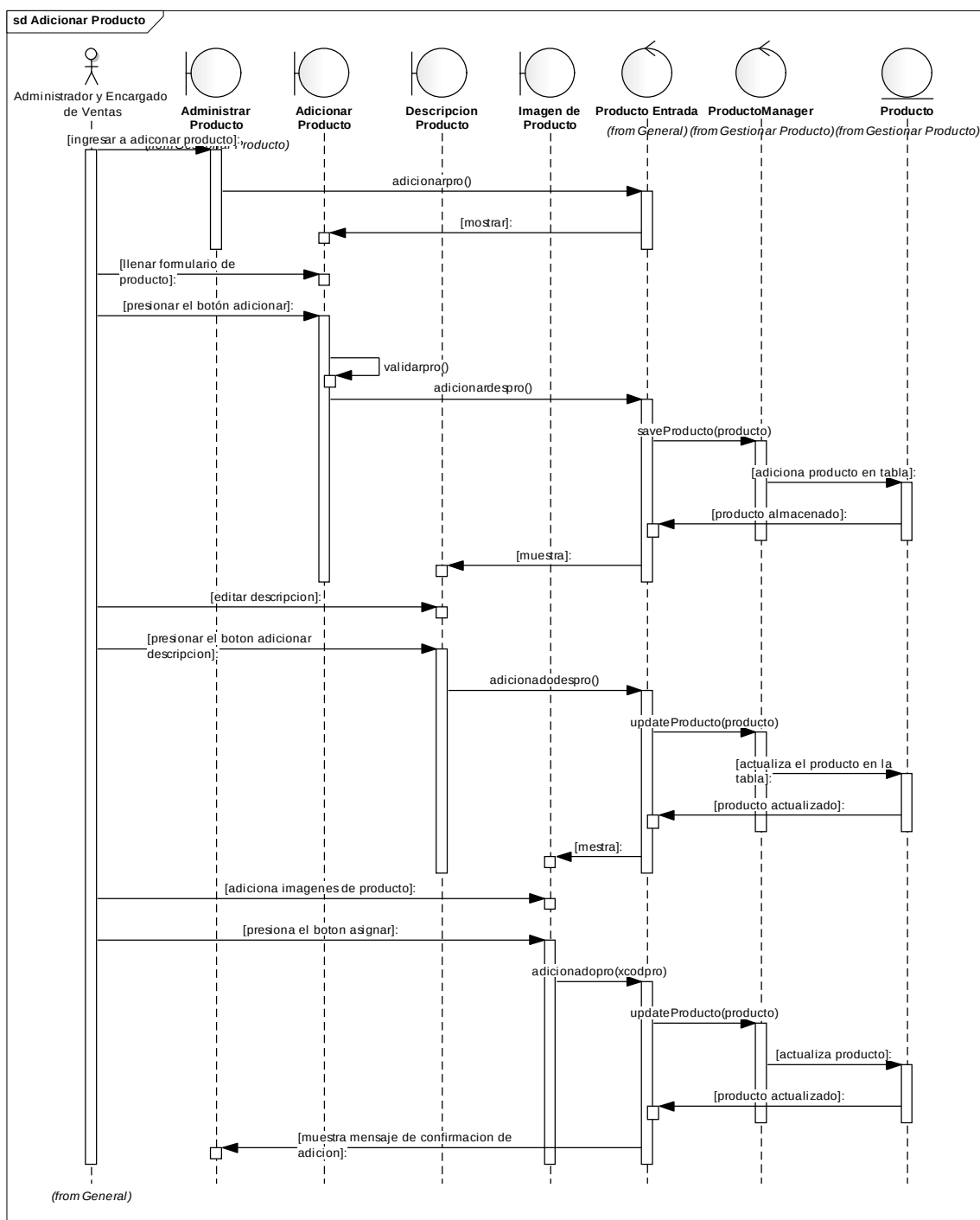
DIAGRAMA DE SECUENCIA: Adicionar Producto.**Figura N° 62: Diagrama de Secuencia: Adicionar Producto.**

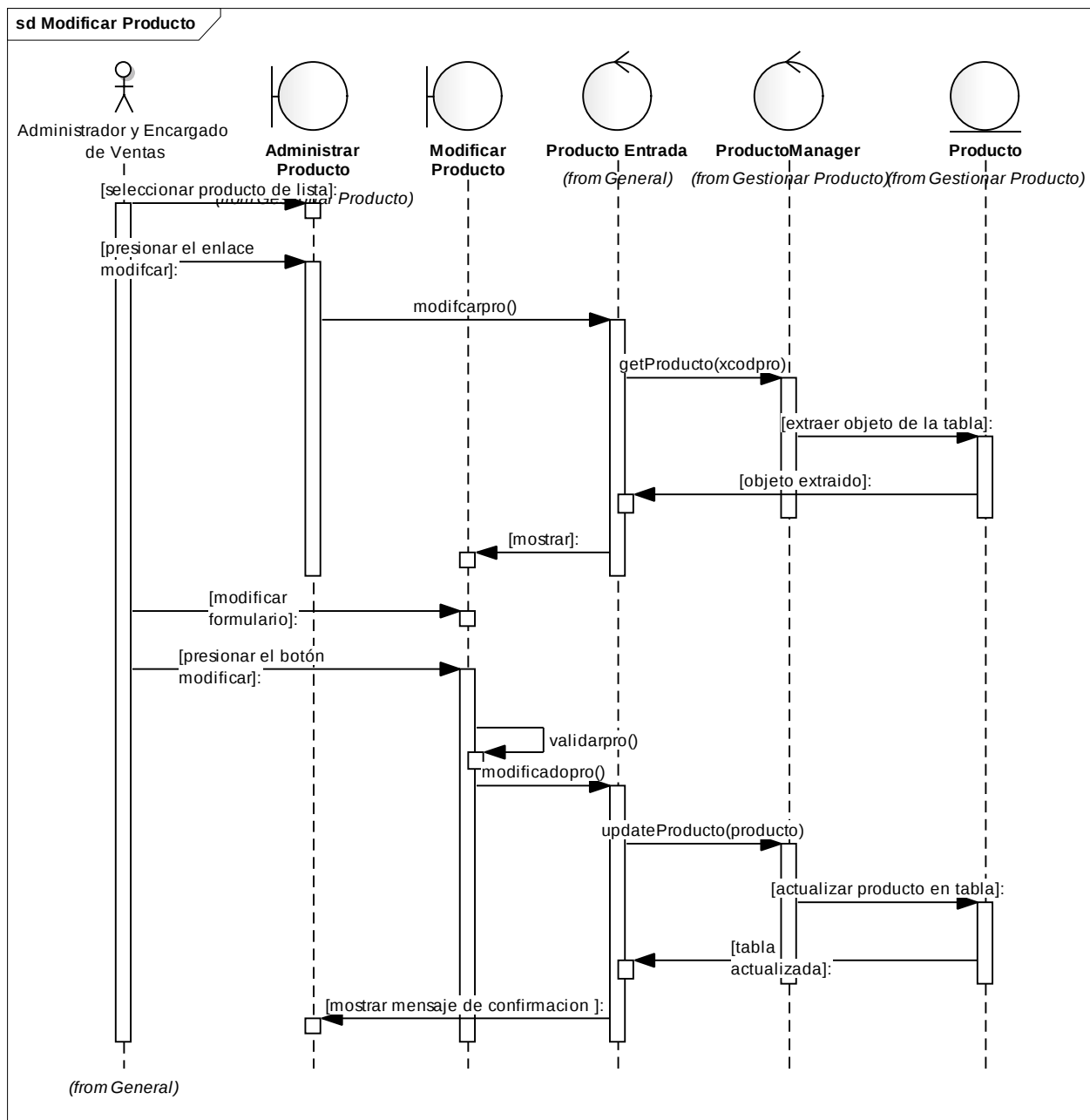
DIAGRAMA DE SECUENCIA: Modificar Producto.**Figura N° 64: Diagrama de Secuencia: Modificar Producto.**

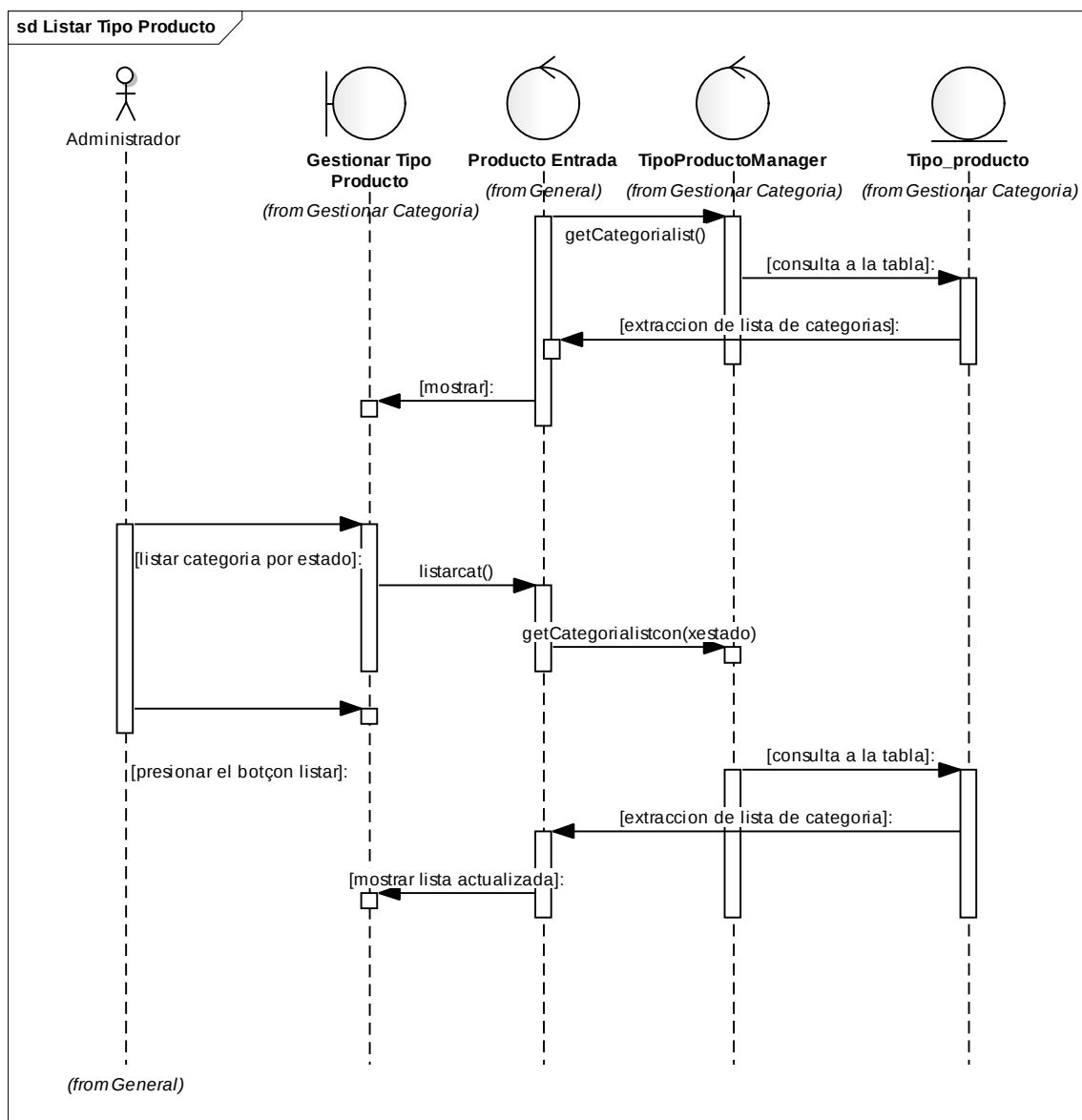
DIAGRAMA DE SECUENCIA: Listar Tipo Producto.**Figura N° 65: Diagrama de Secuencia: Listar Tipo Producto.**

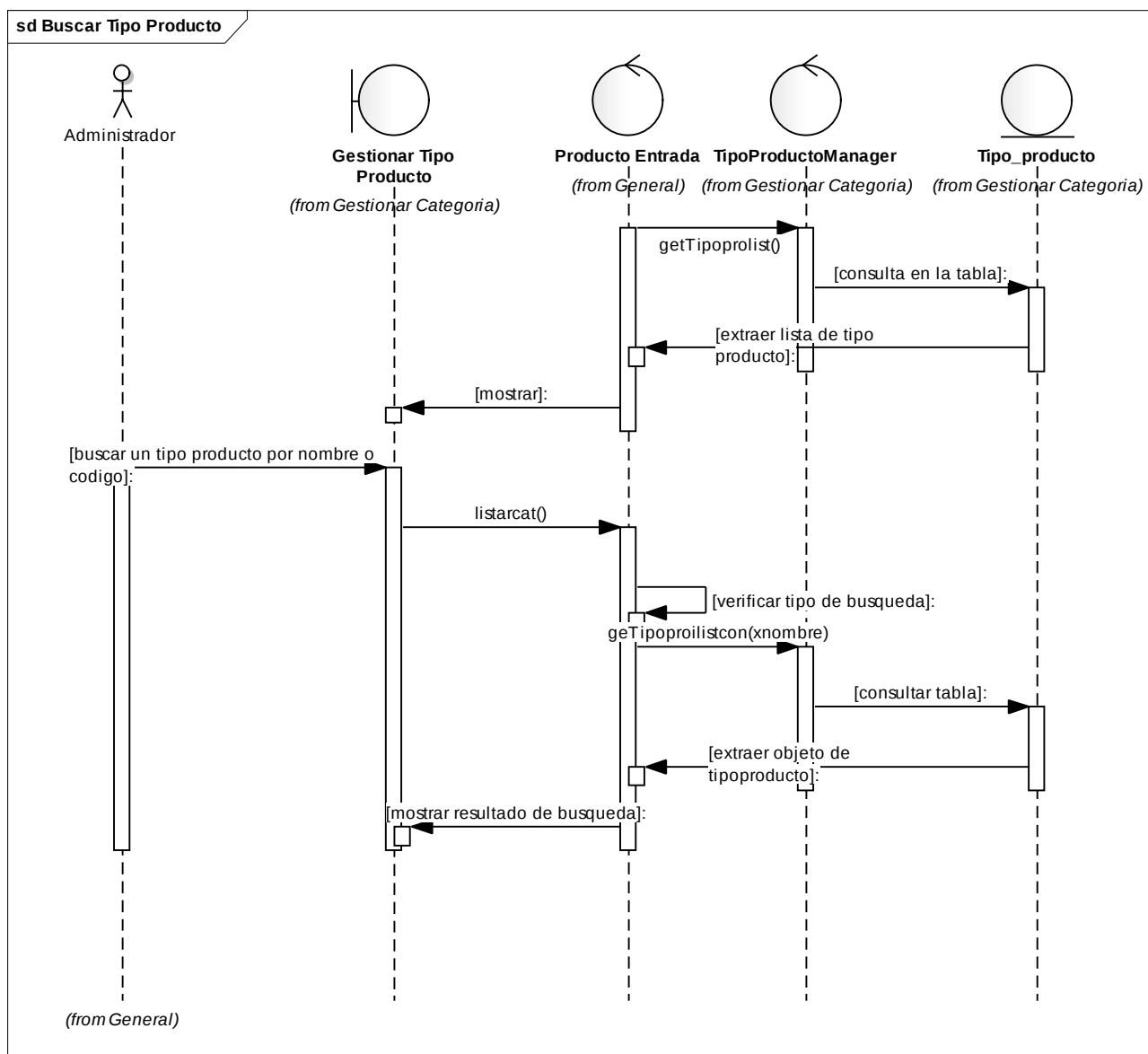
DIAGRAMA DE SECUENCIA: Buscar Tipo Producto.**Figura N° 66: Diagrama de Secuencia: Buscar Tipo Producto.**

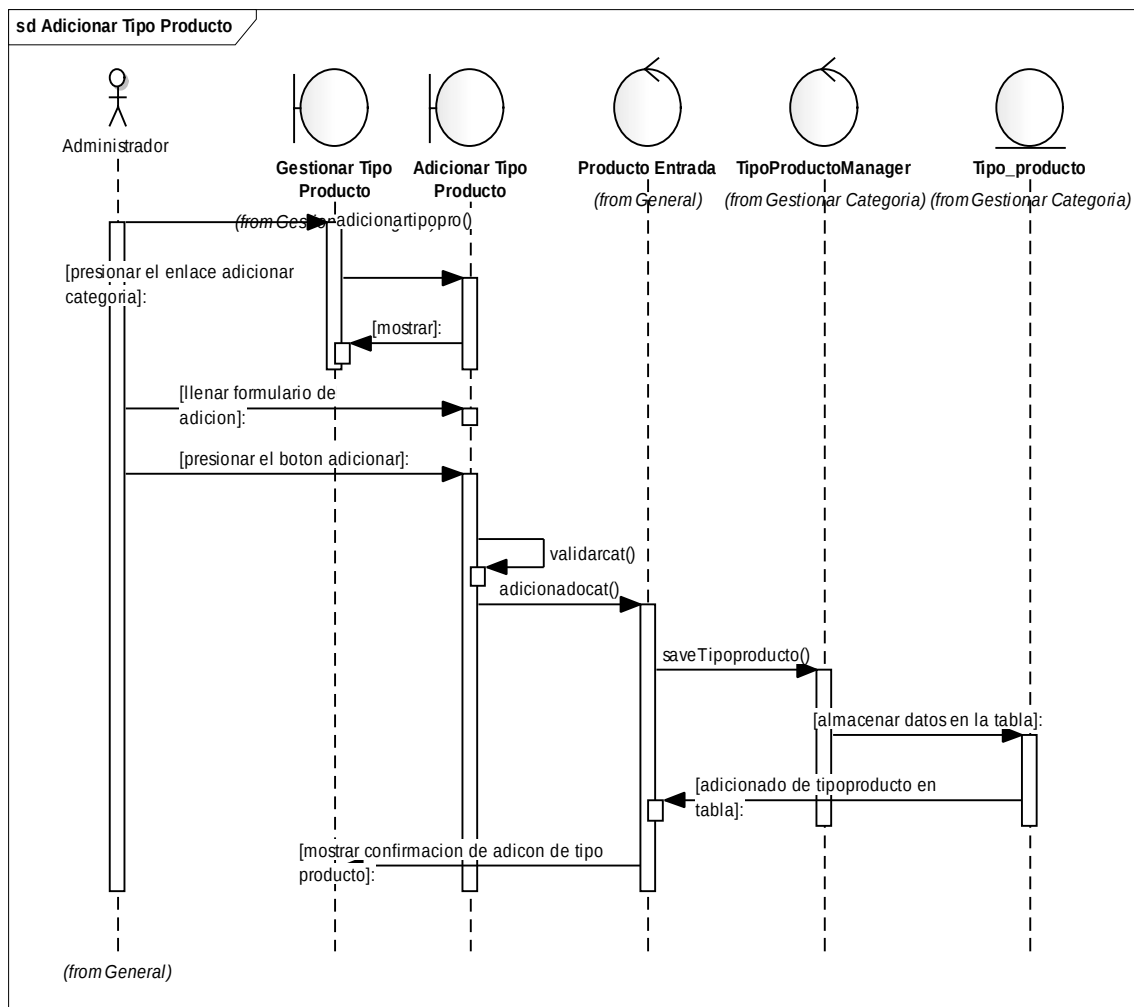
DIAGRAMA DE SECUENCIA: Adicionar Tipo Producto.**Figura N° 67: Diagrama de Secuencia: Adicionar Tipo Producto.**

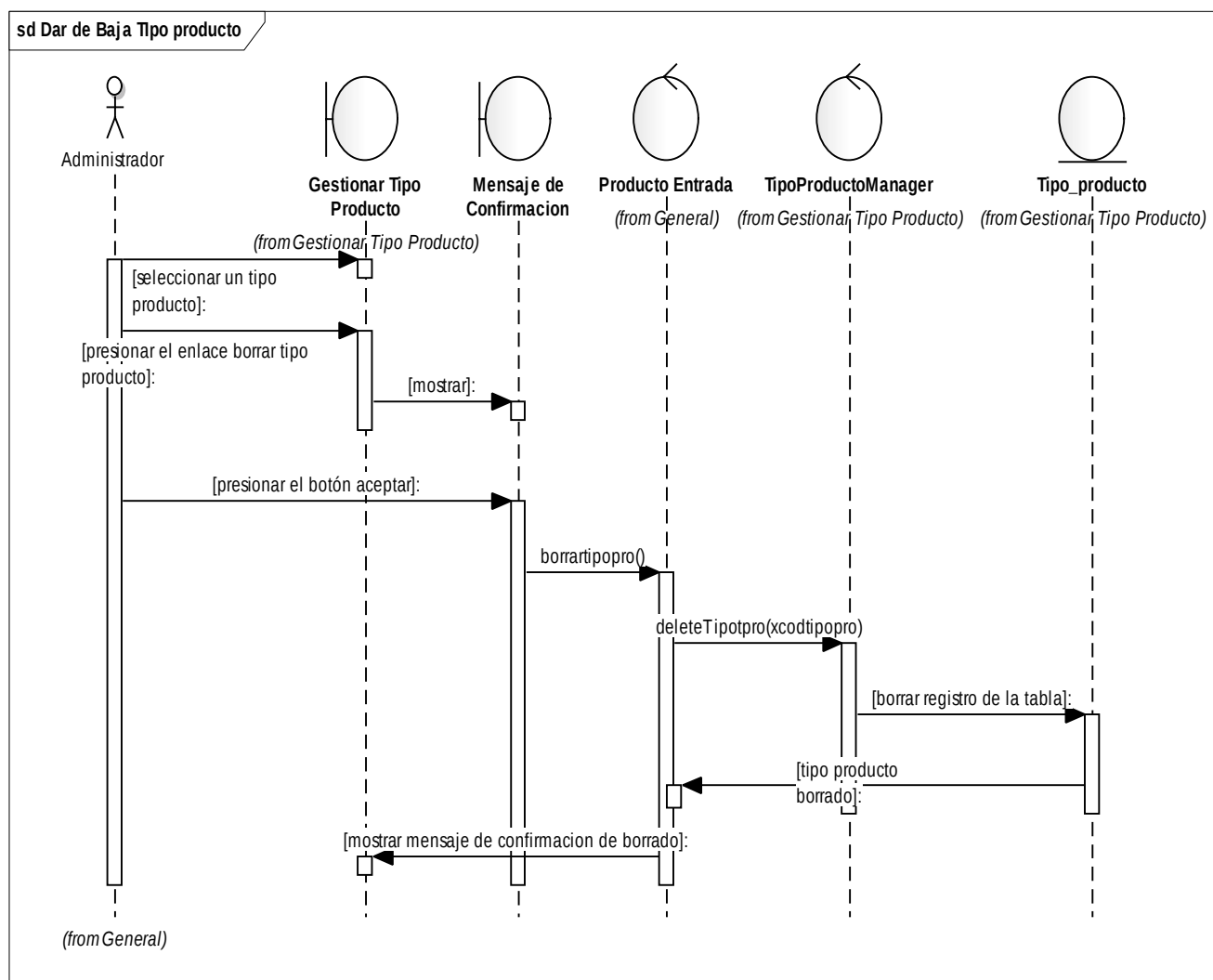
DIAGRAMA DE SECUENCIA: Dar de Baja Tipo Producto.**Figura N° 68: Diagrama de Secuencia: Dar de Baja Tipo Producto.**

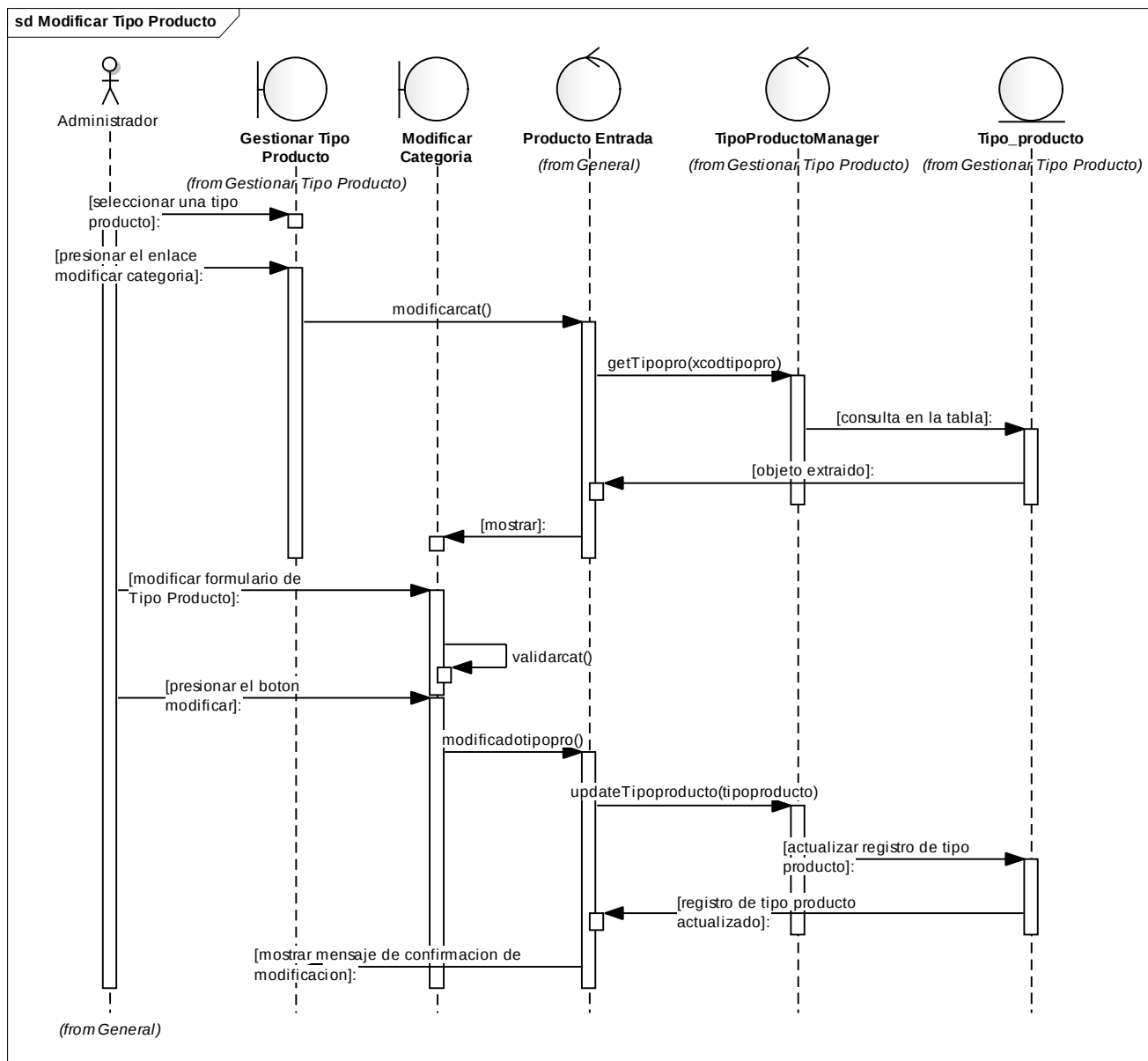
DIAGRAMA DE SECUENCIA: Modificar Categoría.**Figura N° 69: Diagrama de Secuencia: Modificar Categoría.**

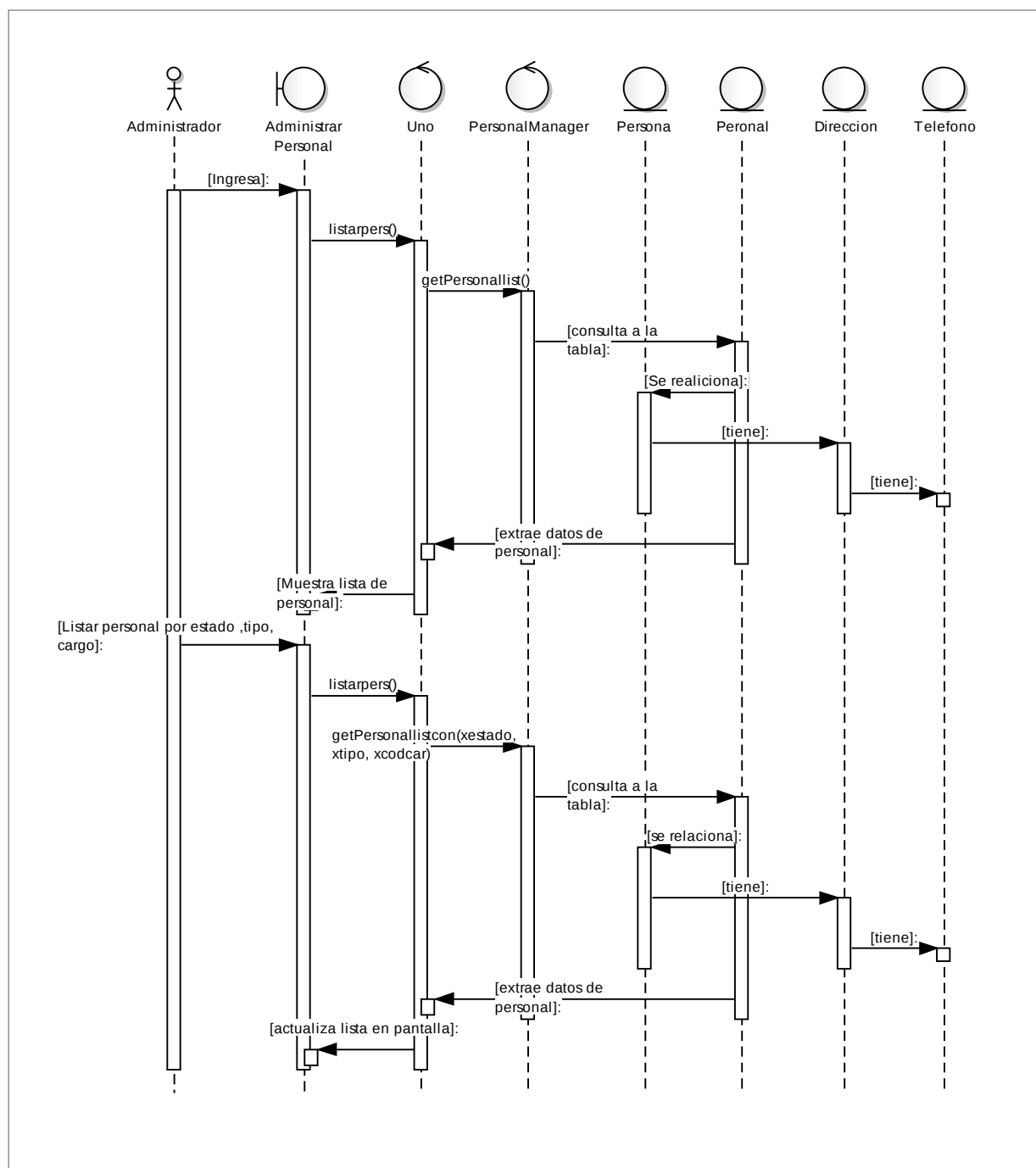
DIAGRAMA DE SECUENCIA: Listar Personal.**Figura N° 70: Diagrama de Secuencia: Listar Personal.**

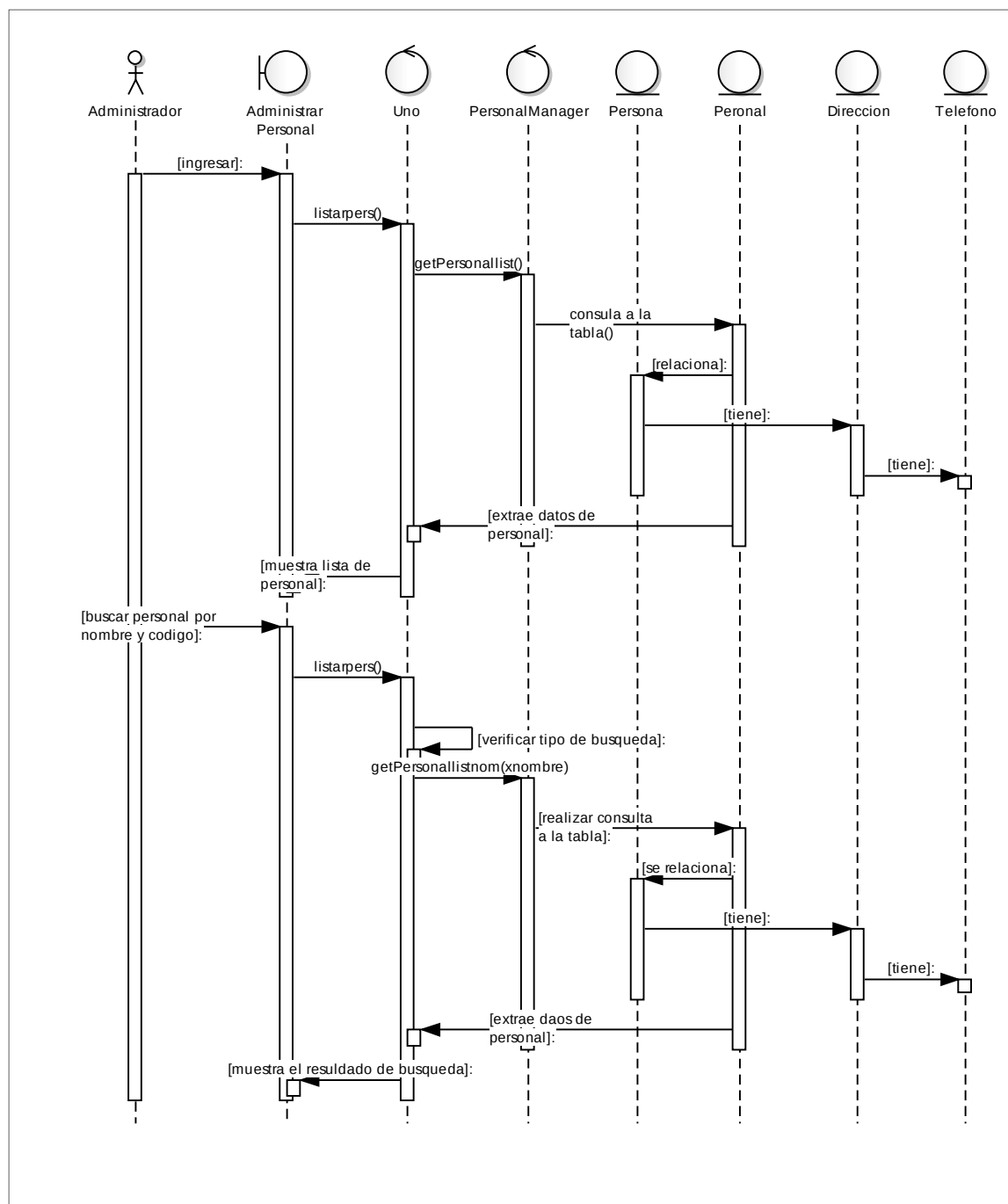
DIAGRAMA DE SECUENCIA: Buscar Personal.**Figura N° 71: Diagrama de Secuencia: Buscar Personal.**

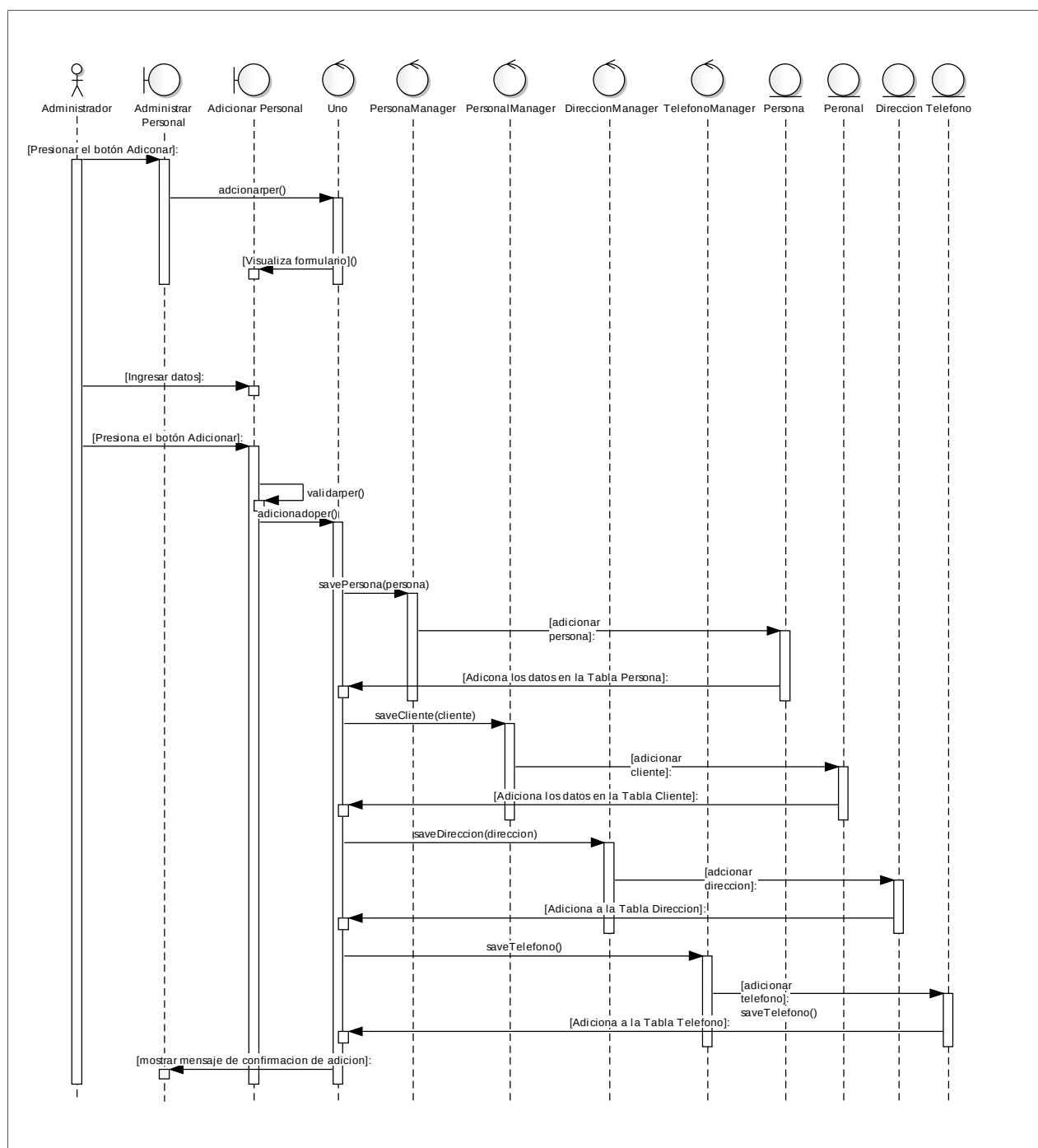
DIAGRAMA DE SECUENCIA: Adicionar Personal.**Figura N° 72: Diagrama de Secuencia: Adicionar Personal.**

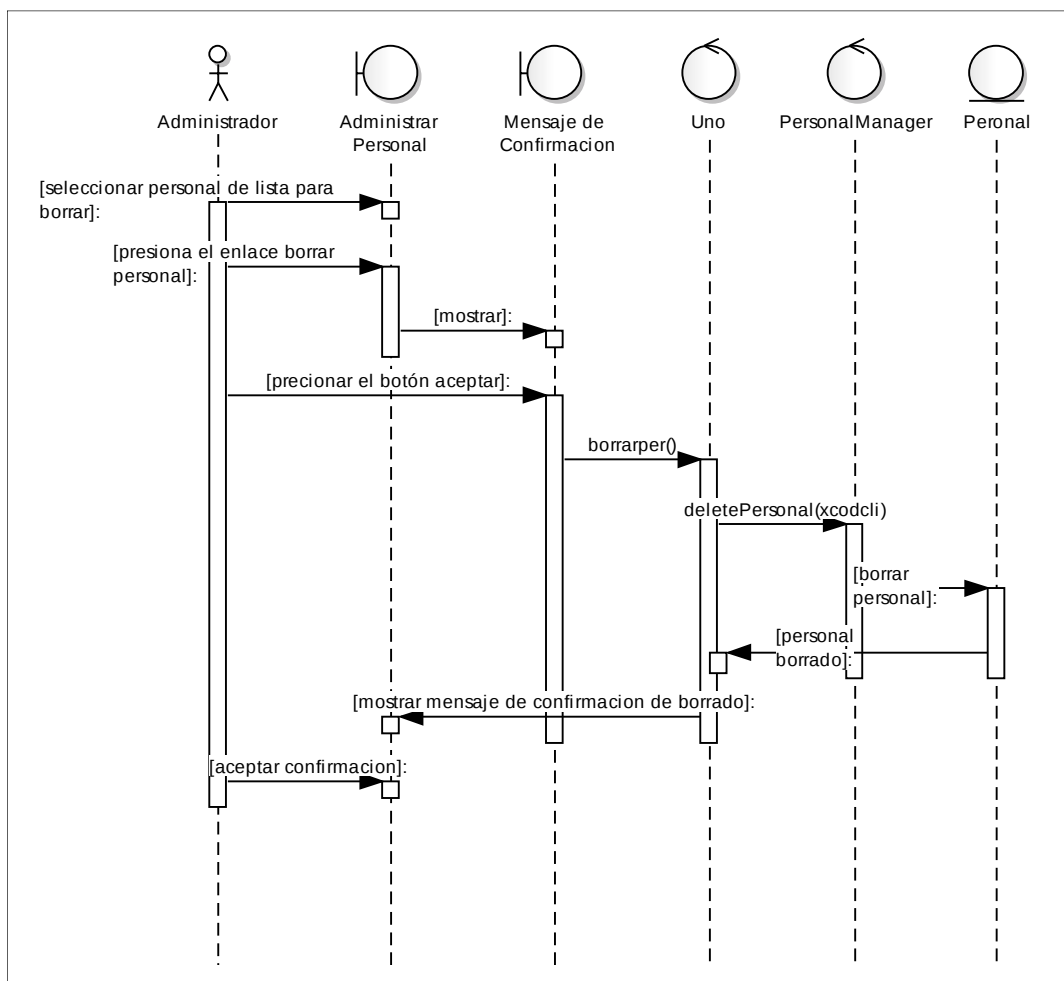
DIAGRAMA DE SECUENCIA: Dar de Baja Personal.**Figura N° 73: Diagrama de Secuencia: Dar de Baja Personal.**

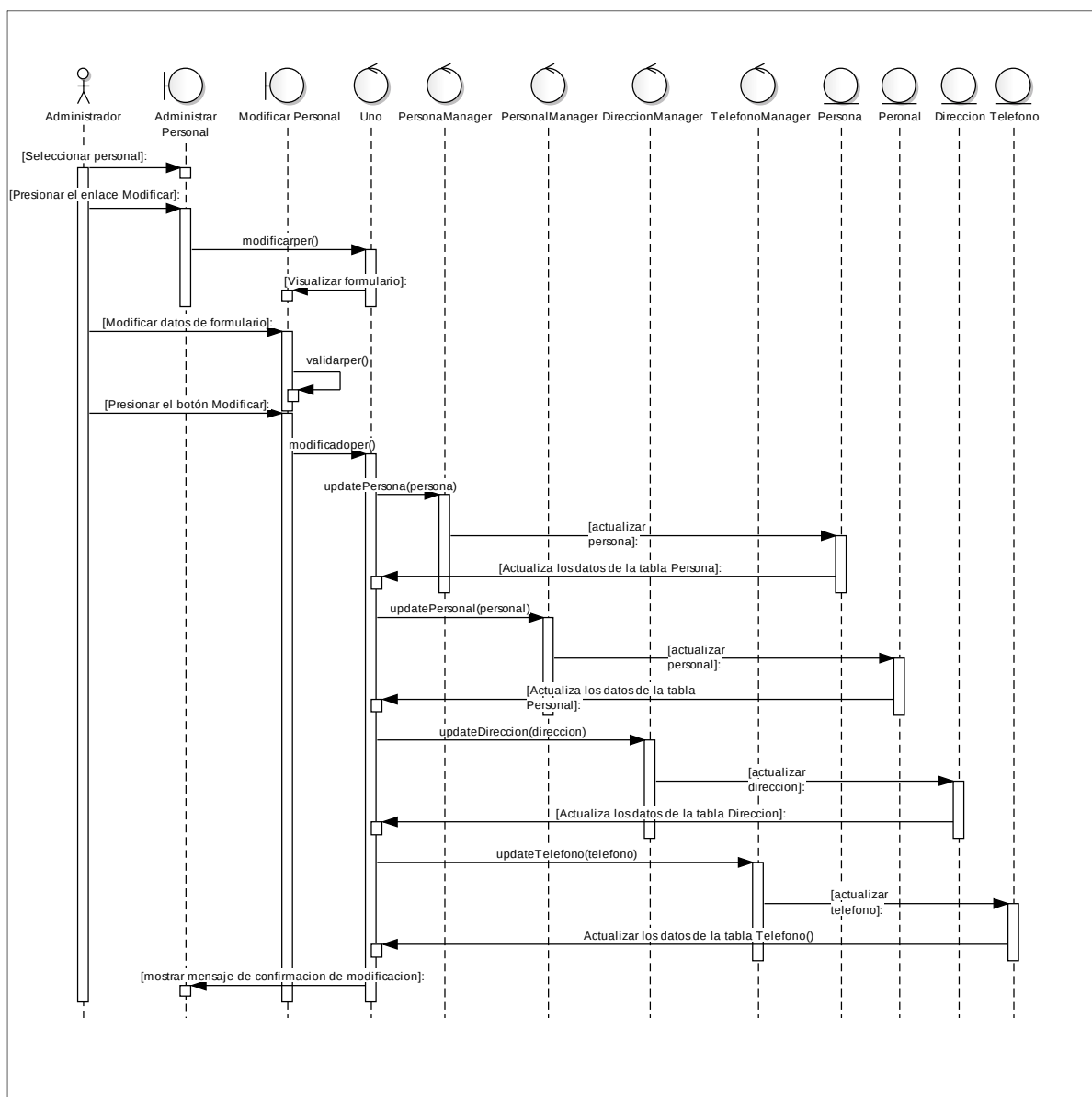
DIAGRAMA DE SECUENCIA: Modificar Personal.**Figura N° 74: Diagrama de Secuencia: Modificar Personal.**

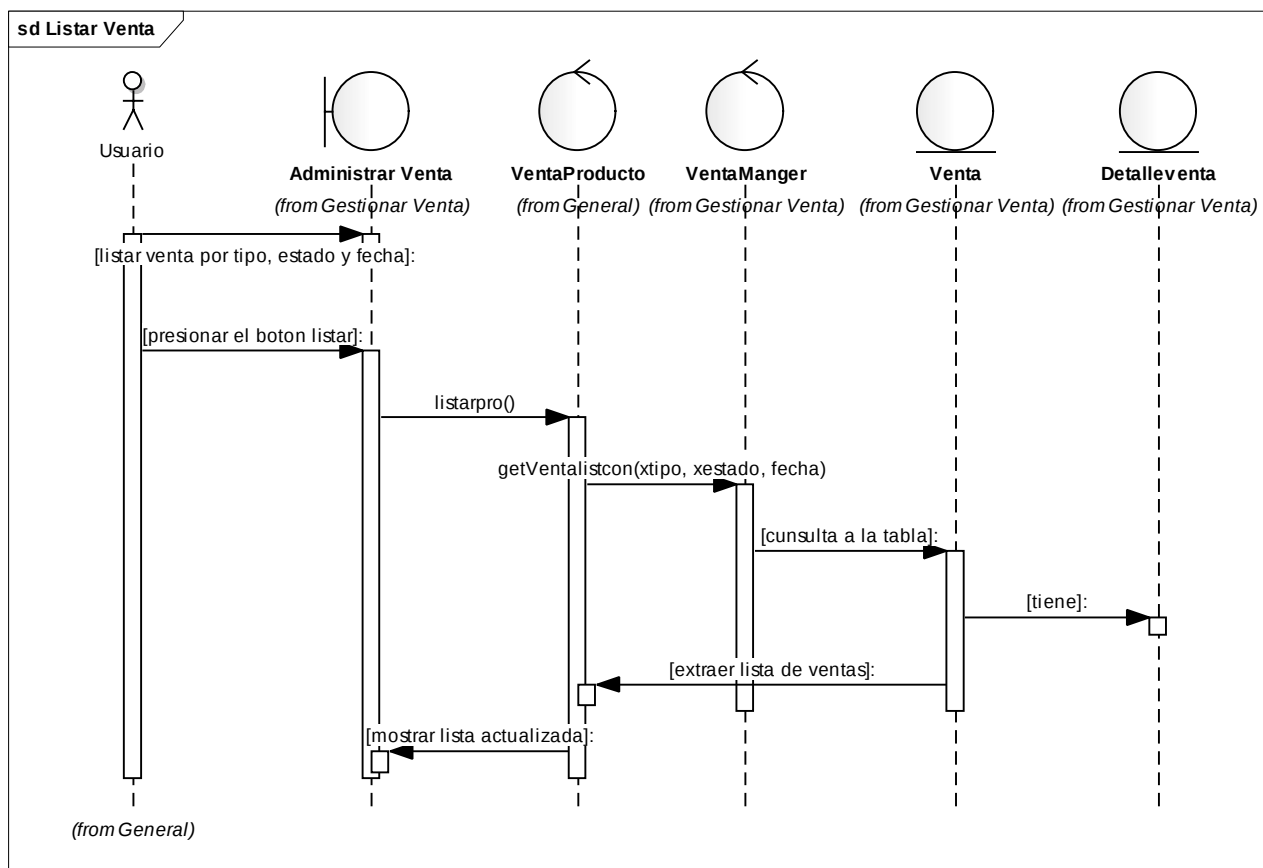
DIAGRAMA DE SECUENCIA: Listar Venta.**Figura N° 75: Diagrama de Secuencia: Listar Venta.**

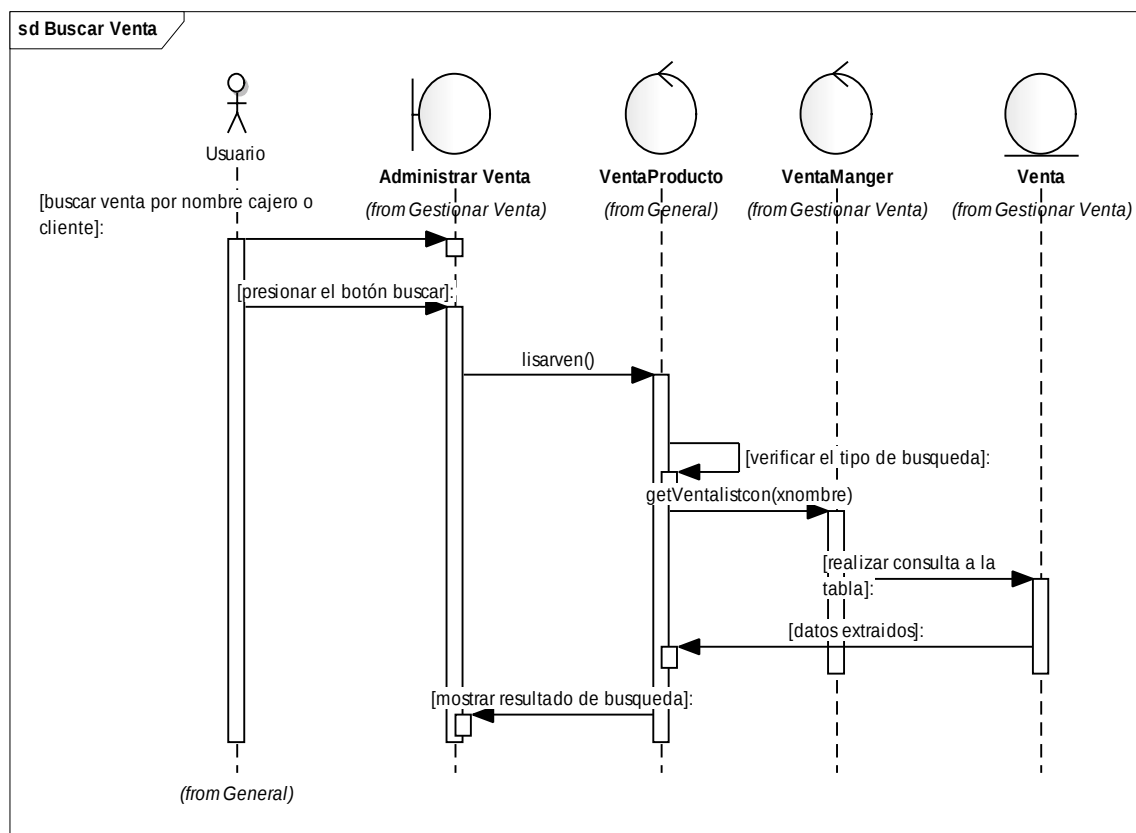
DIAGRAMA DE SECUENCIA: Buscar Venta.**Figura N° 76: Diagrama de Secuencia: Buscar Venta.**

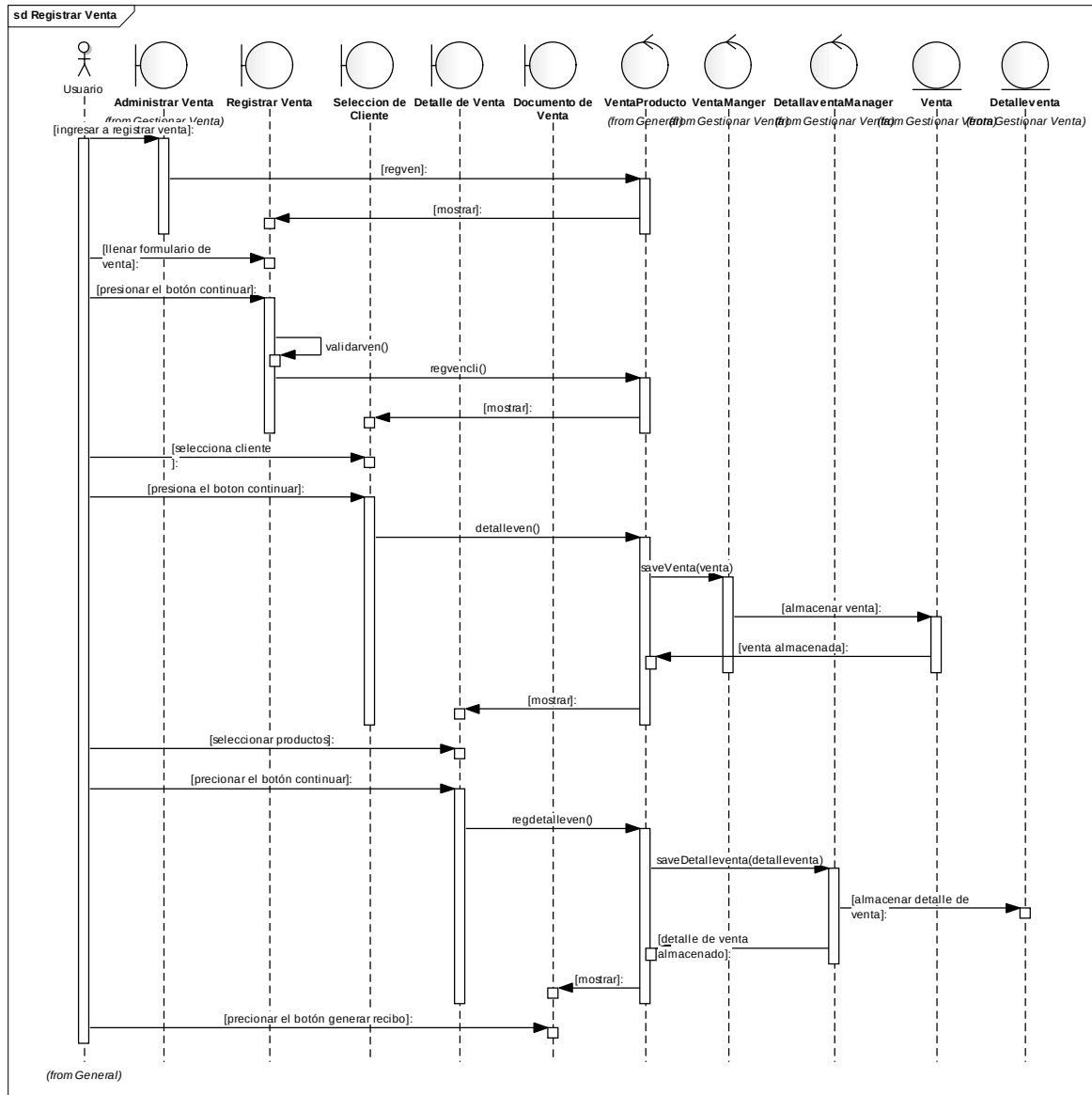
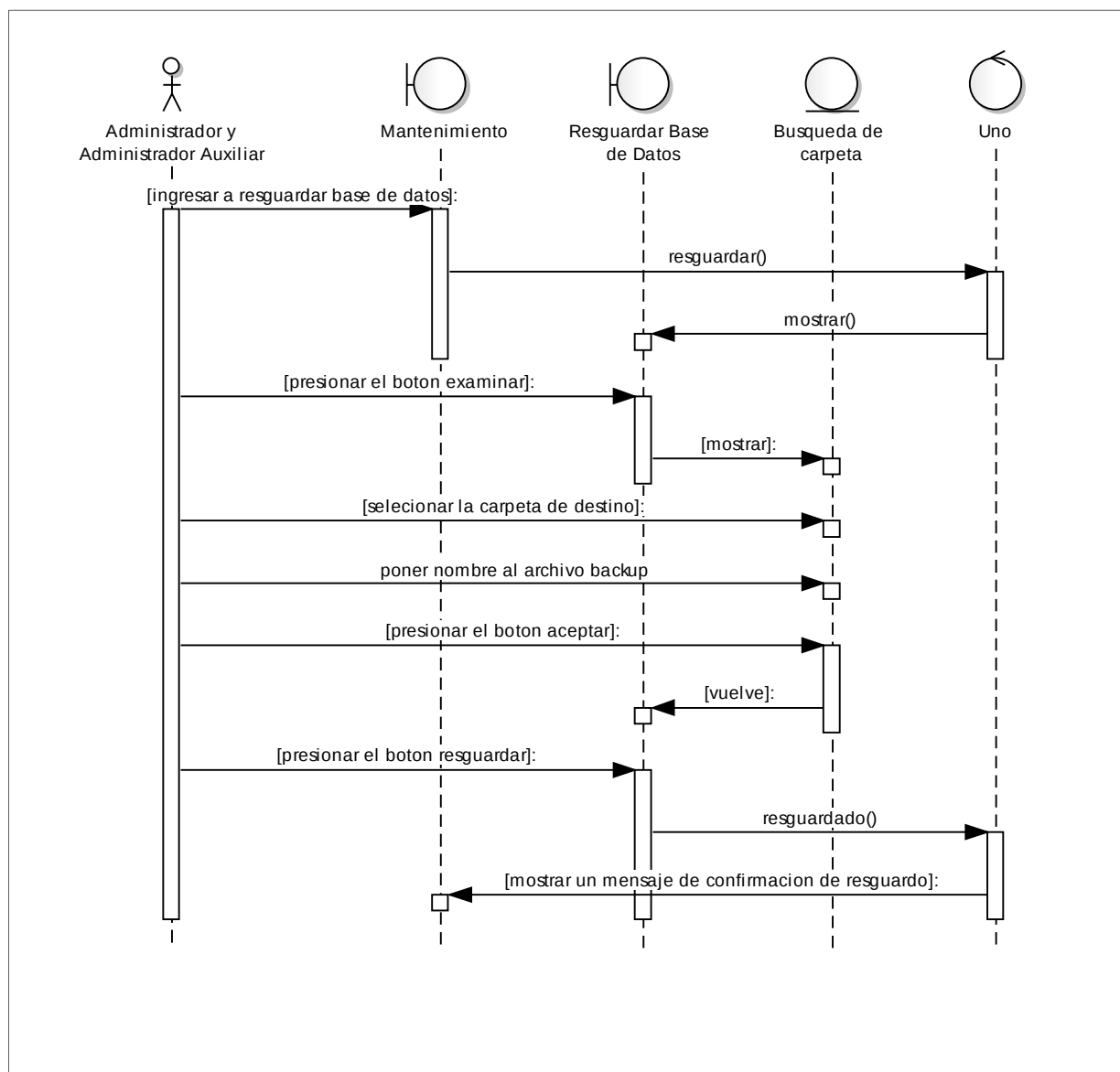
DIAGRAMA DE SECUENCIA: Registrar Venta.**Figura N° 77: Diagrama de Secuencia: Registrar Venta.**

DIAGRAMA DE SECUENCIA: Resguardar Base de Datos.**Figura N° 78:** *Diagrama de Secuencia: Resguardar Base de Datos.*

II.1.10.3. Diagrama de Colaboración

Diagrama de Colaboración: Ingresar al Sistema

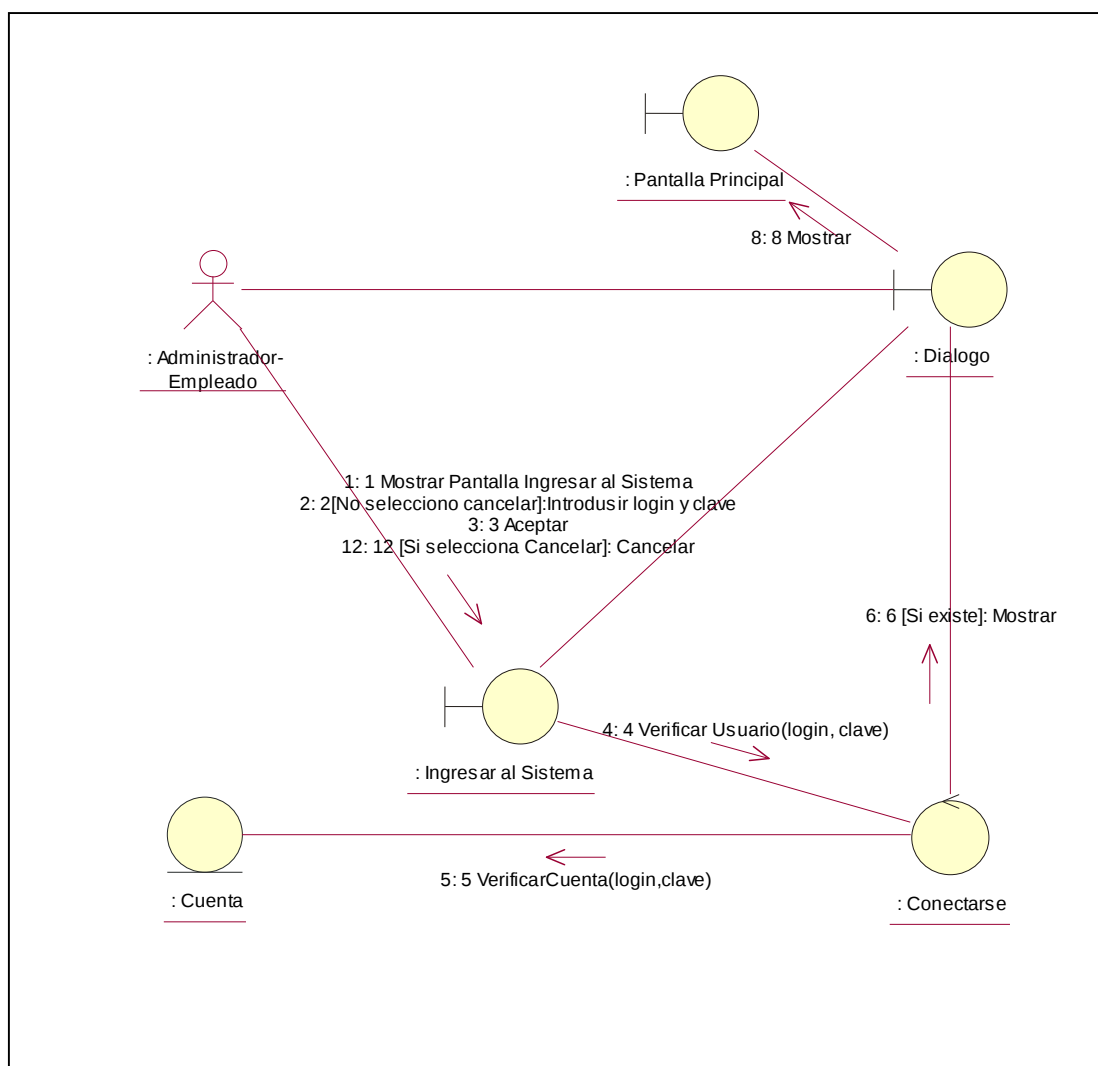


Figura N° 79: Diagrama de Colaboración: Ingresar al Sistema

Diagrama de Colaboración: Gestionar Personal

Diagrama de Colaboración: Listar Personal

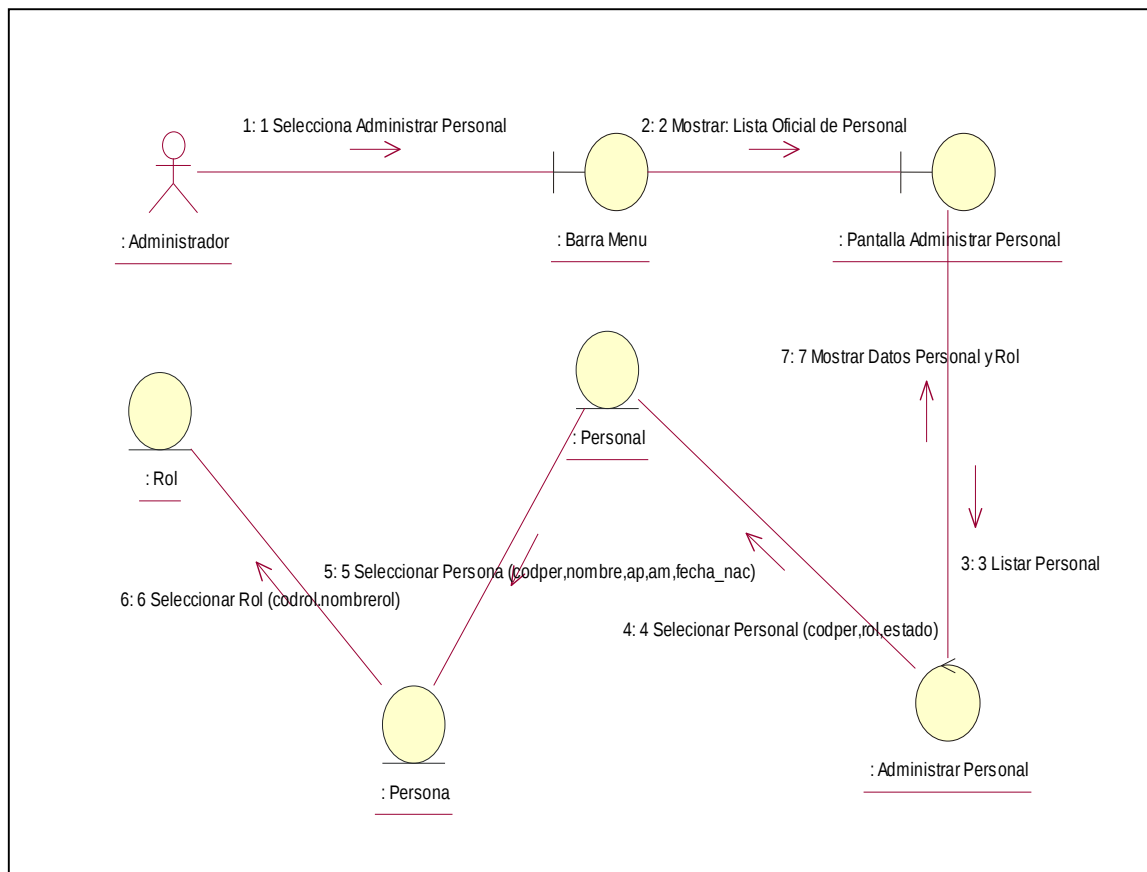


Figura N° 80: Diagrama de Colaboración: Listar Personal

Diagrama de Colaboración: Agregar Personal

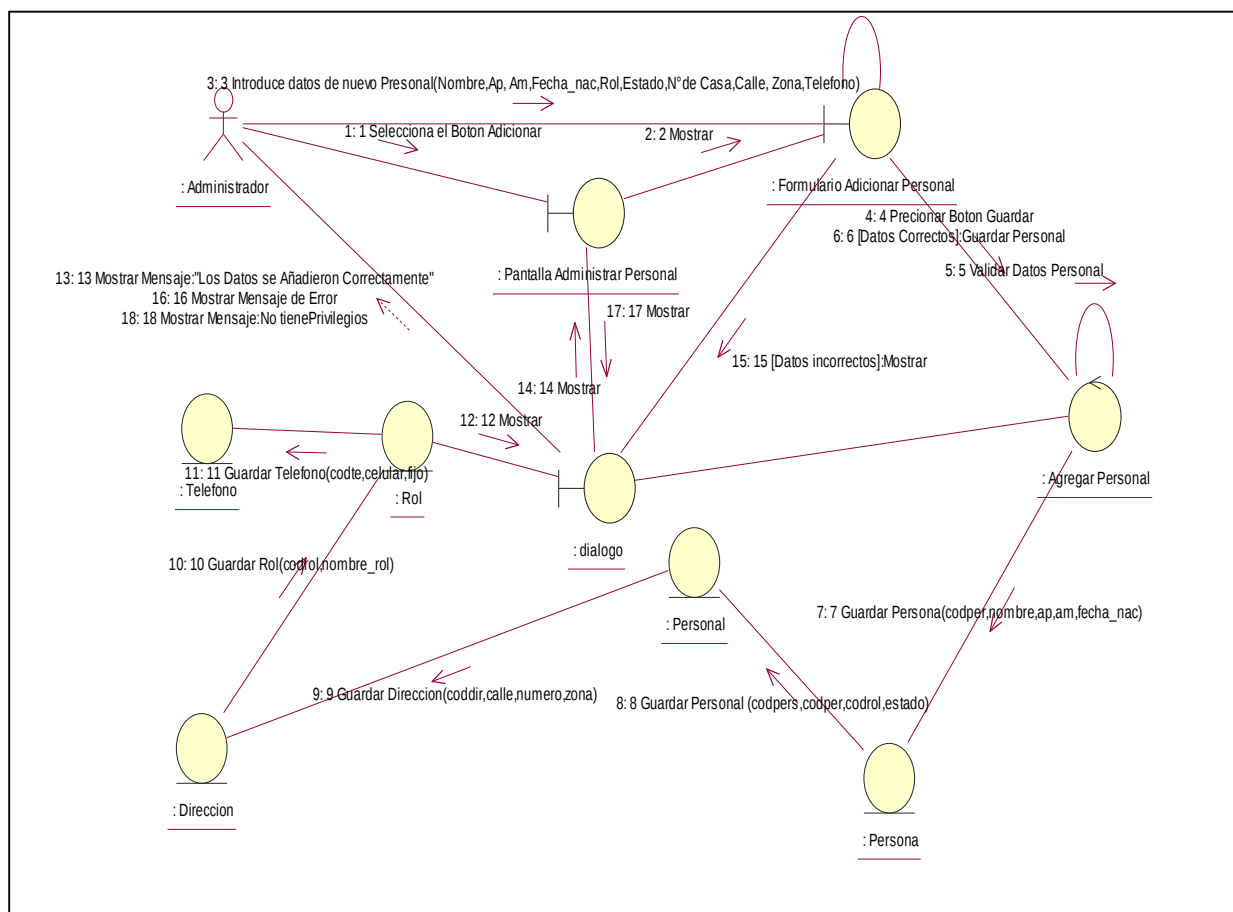


Figura N° 81: Diagrama de Colaboración: Agregar Personal

Diagrama de Colaboración: Modificar Personal

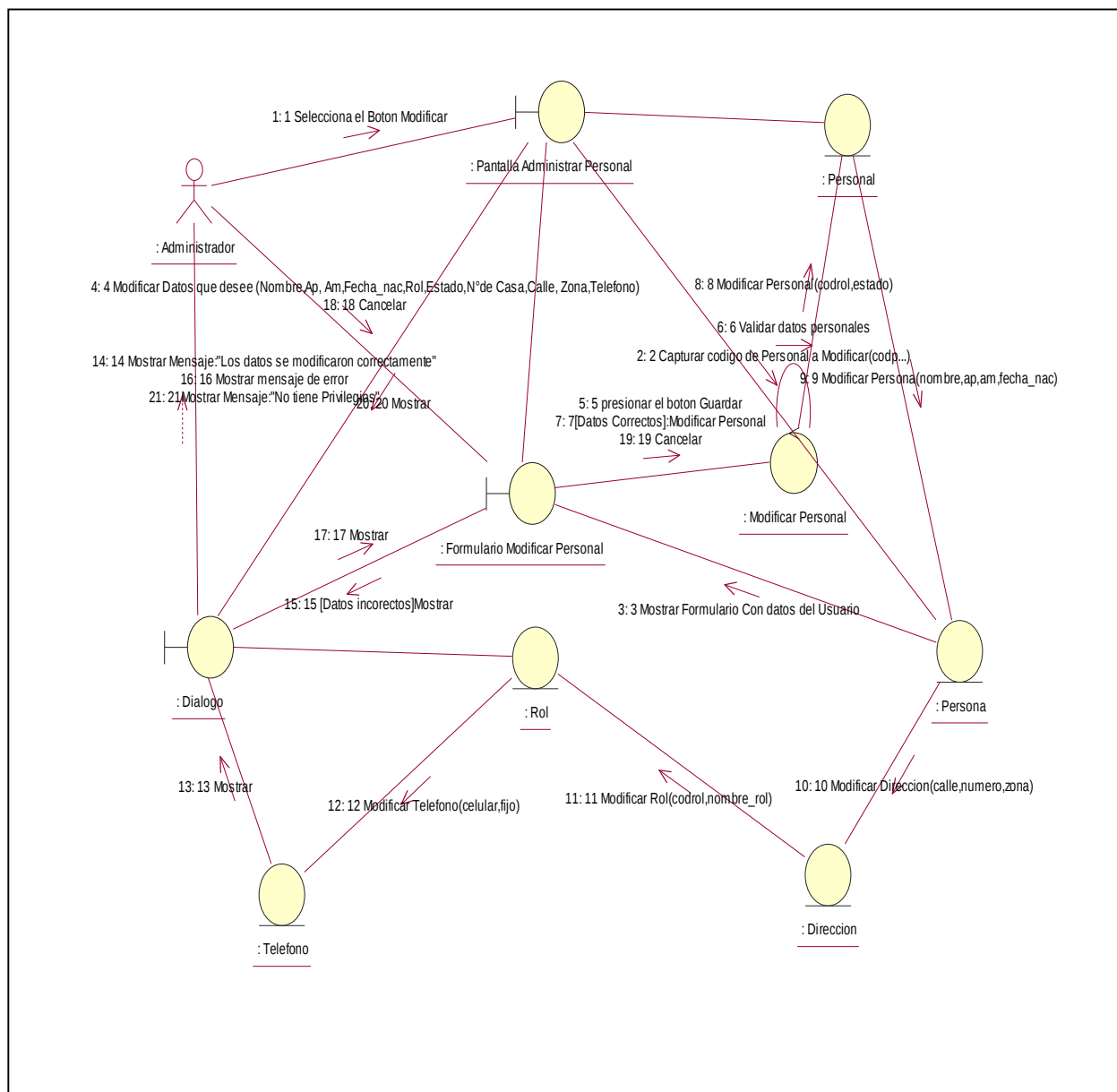


Figura N° 82: Diagrama de Colaboración: Modificar Personal

Diagrama de Colaboración: Dar de Baja Personal

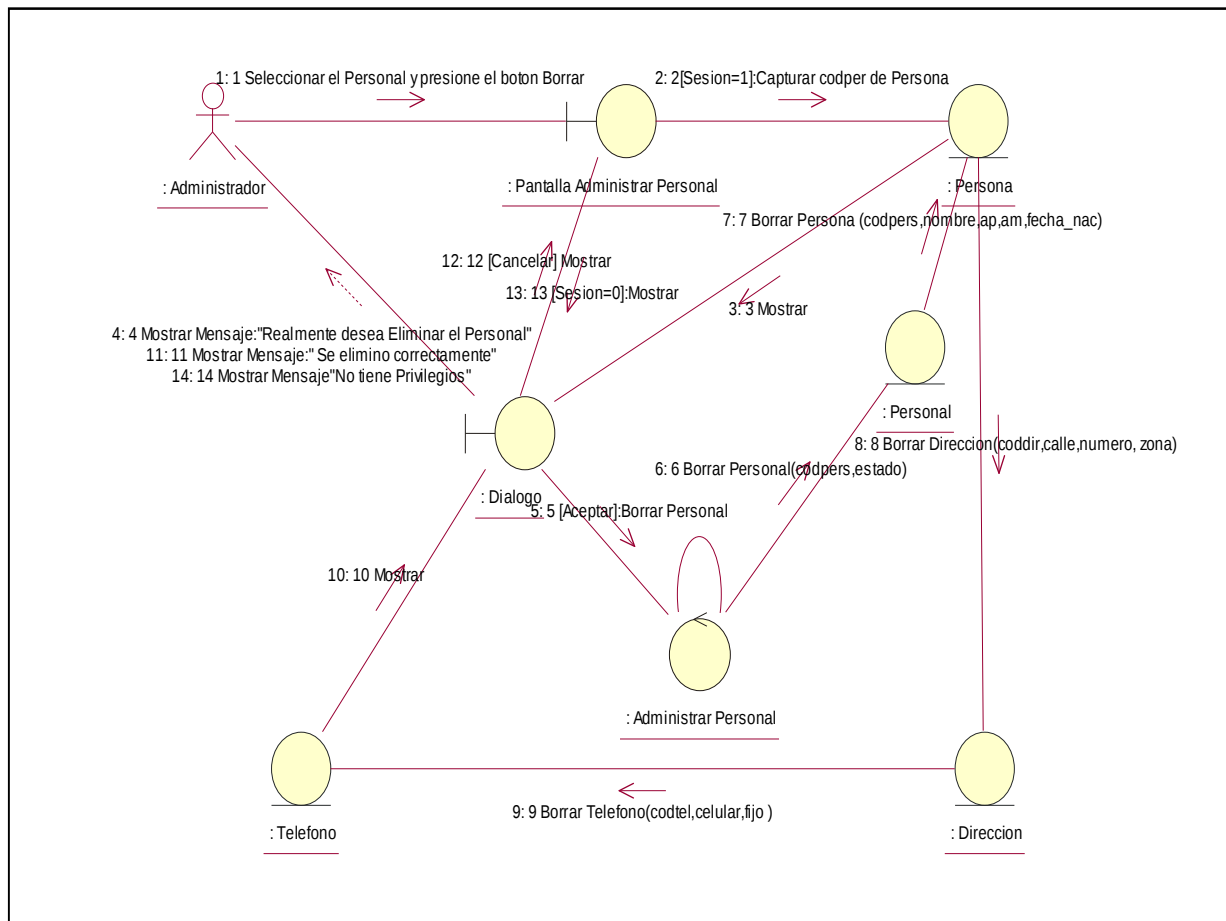


Figura N° 83: Diagrama de Colaboración: Dar de Baja Personal

Diagrama de Colaboración: Plan de Pago

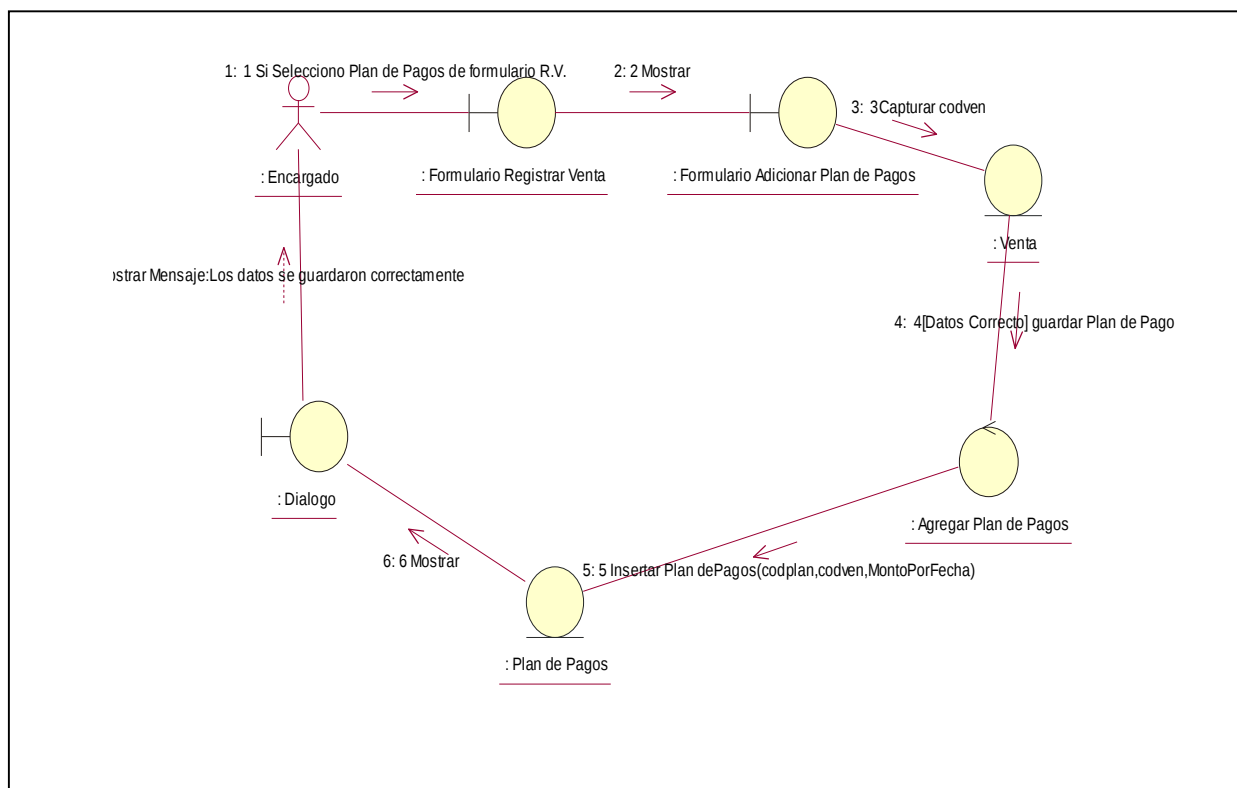


Figura N° 84: Diagrama de Colaboración: Plan de Pagos

Diagrama de Colaboración: Actualizar Plan de Pago

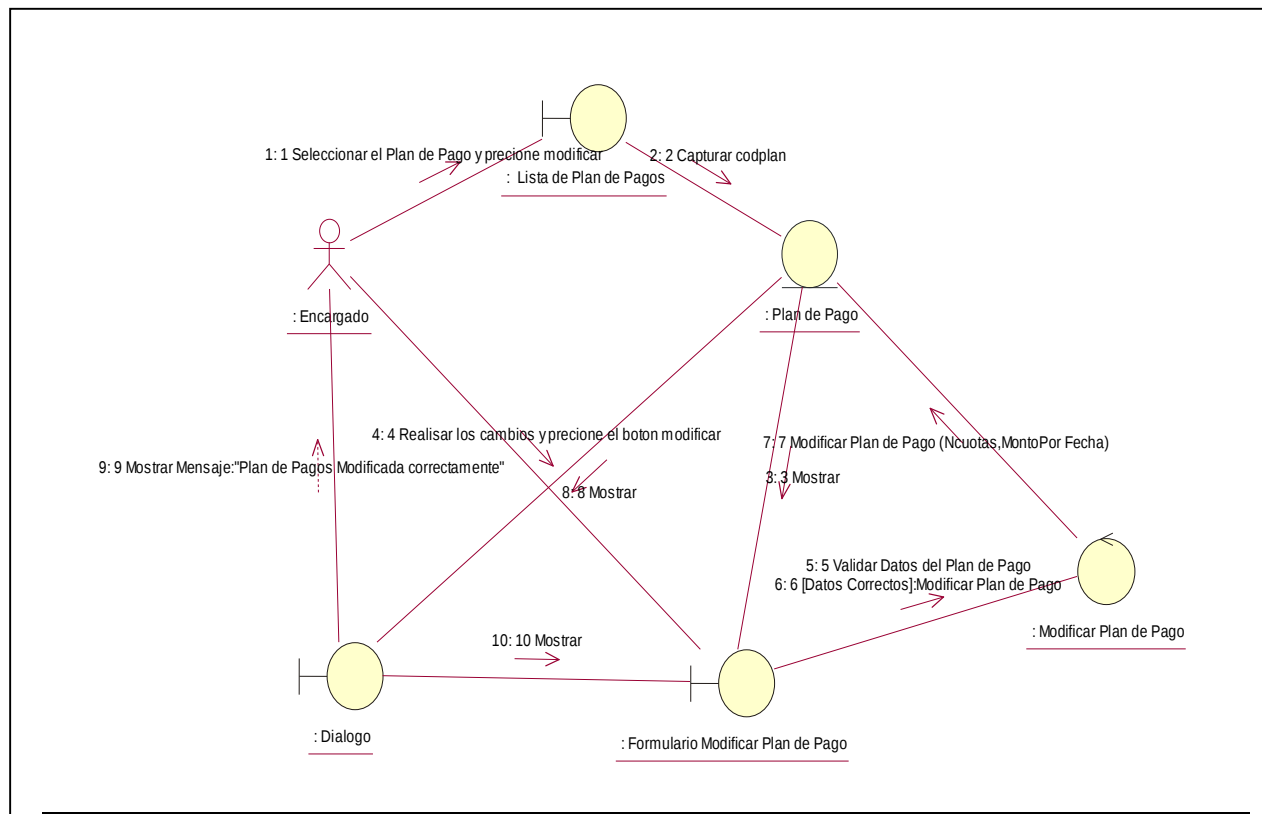


Figura N° 85: Diagrama de Colaboración: Actualizar Plan de Pago

II.1.11. DIAGRAMA DE DESPLEGUE

II.1.11.1. Introducción

El Diagrama de Despliegue es un tipo de diagrama del [Lenguaje Unificado de Modelado](#) que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones¹.

Nodo

Un nodo es un objeto físico de ejecución que representa un recurso computacional (computador). Los nodos pueden tener estereotipos como UCP, dispositivos, y memorias.²

Sistemas cliente-servidor

Los sistemas cliente-servidor son un extremo del espectro de los sistemas distribuidos y requieren tomar decisiones sobre la conectividad de red de los clientes a los servidores y sobre la distribución física de los componentes software del sistema a través de nodos.

II.1.11.2. Propósito

- Modelar la arquitectura física de los recursos.
- Representar el despliegue con nodos.

II.1.11.3. Alcance

- Solo se Modelara la vista de despliegue estática.

¹ Diagrama de despliegue [2]

² Modelado de Objetos con UML [3]

- Representar líneas de comunicación entre nodos.

Diagrama de Despliegue

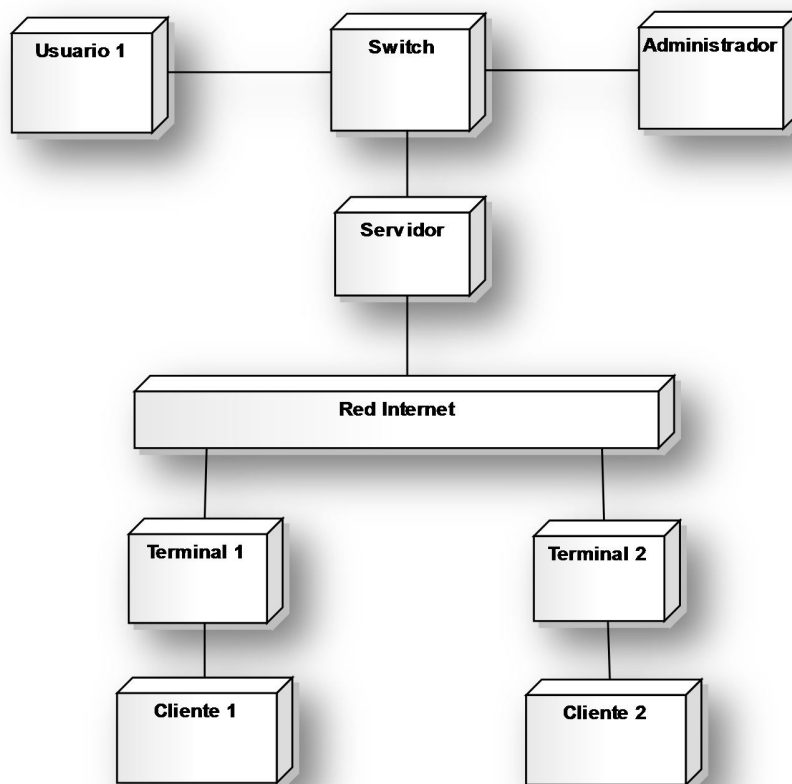


Figura N° 86: Diagrama de Despliegue

a) Arquitectura física Implementación del Sistema.

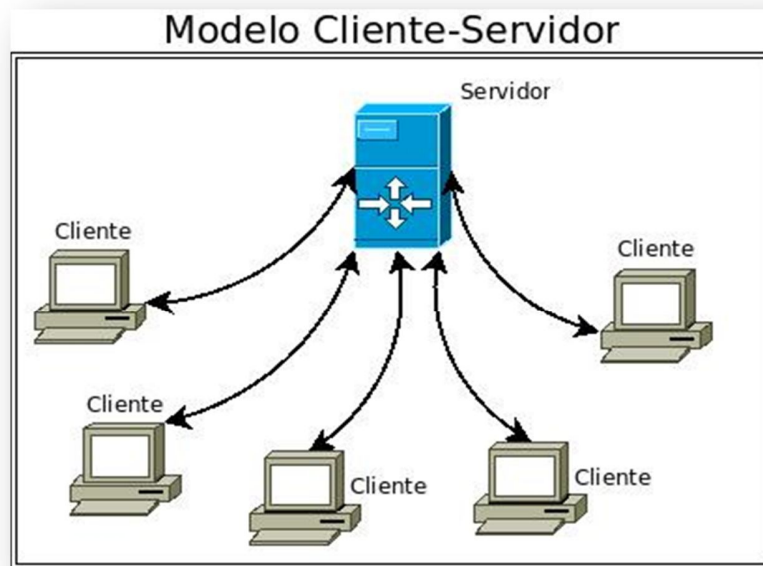


Figura N° 87: Arquitectura Física Implementación del Sistema

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

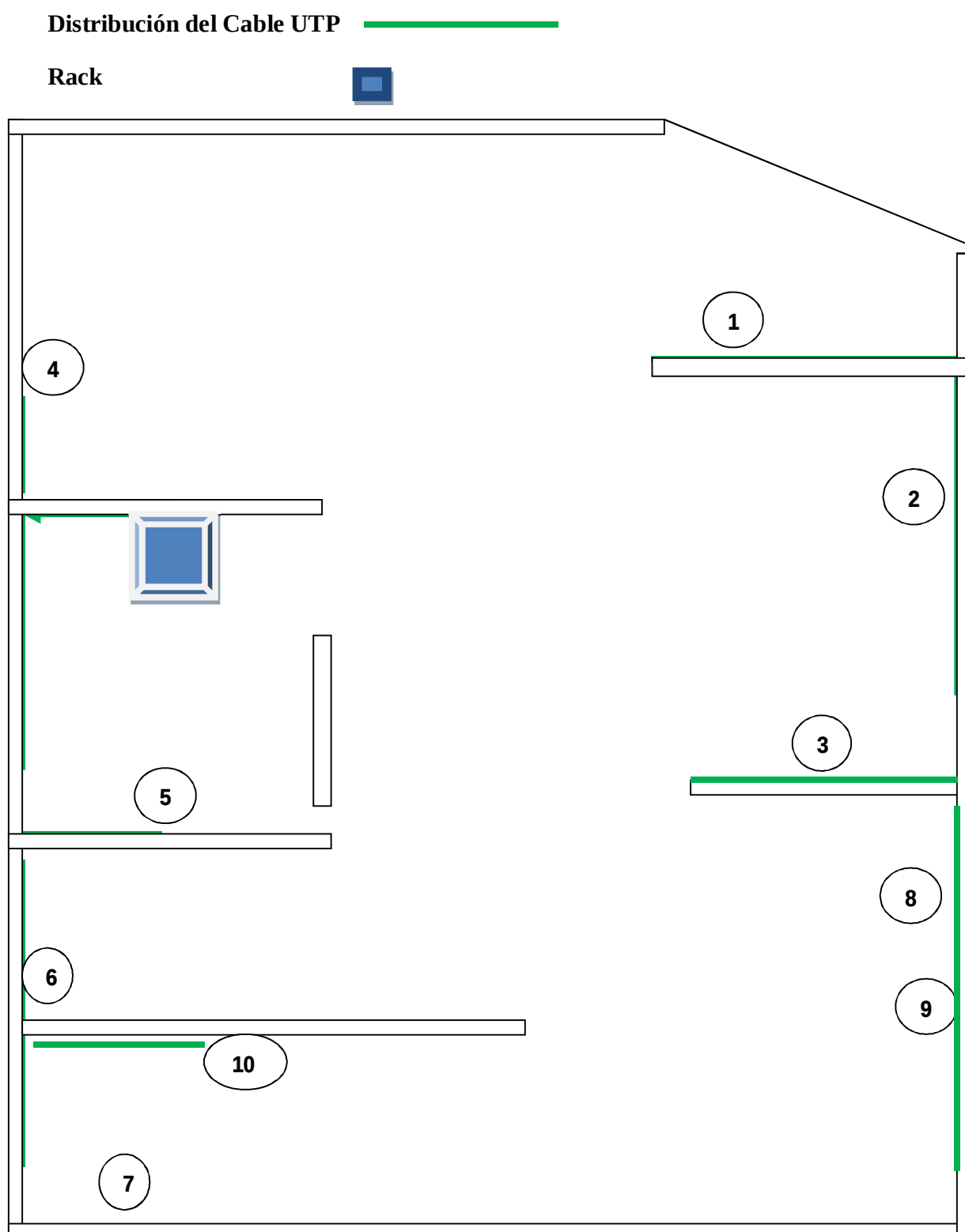


Figura N° 88: Distribución de Cable UTP

b) Modelo de Implementación del Sistema

Elemento	Características	Cantidad	Precio	Costo
Host 	Computadora con la suficiente capacidad de soportar el software Los requisitos de hardware del Servidor Web son los siguientes: • Procesador Core2duo de 3.4Ghz en adelante. • Memoria RAM de 1 GB en adelante. • Lector de CD: 4x. • Monitor: Color de 15". • Espacio libre en la unidad de disco rígido: 200 Mb aproximadamente. Conexión a Internet ADSL. O en su Defecto Conexión Intranet.	10	720 \$	7200 \$
Cable Canal 	Normal	400 mts.	1.10 \$	440 \$
Accesorios para Cable Canal 	Normal	50 mts.	0.5 \$	25 \$
Patch cord de cobre 	Normal	12 unid.	1\$	12 \$

Armario de telecomunicaciones 	Normal	1 unid	150\$	150 \$
Cable UTP Cat 5e 		200 mts	0.50\$	100 \$
SWITCH APILABLE 48 PUERTOS GIGABIT ETHERNET + 4 PUERTOS COMBRO MINIGBIC  <i>Switch o concentrador de cobre de una IDF</i>		1 unid	70\$	70 \$
Patch Panel 		1 unid	55\$	55 \$
Jack RJ-45 		30 unid	0.30\$	9 \$
Routers 		1 unid	100\$	100 \$

Conectores RJ-45 		30 unid	0.40\$	12 \$
--	--	---------	--------	--------------

Costo Total: 8061 \$

II.1.12. MODELADO DE DIAGRAMA DE CLASES

II.1.12.1. Introducción

El Modelado de Diagrama de Clases es uno más de los diagramas requeridos en la fase de Análisis/Diseño de la metodología RUP la cual estamos implementando.

II.1.12.1.1. Propósito

- Comprender la estructura del sistema deseado para la organización.
- Identificar clases de análisis y diseño.

II.1.12.1.2. Alcance

- Describir las clases y objetos de diseño del sistema en su segunda iteración
- Identificar y definir los **objetos del sistema** según los **objetivos** del sistema deseado aprobado por la organización.

II.1.12.2. Diagrama de Clases

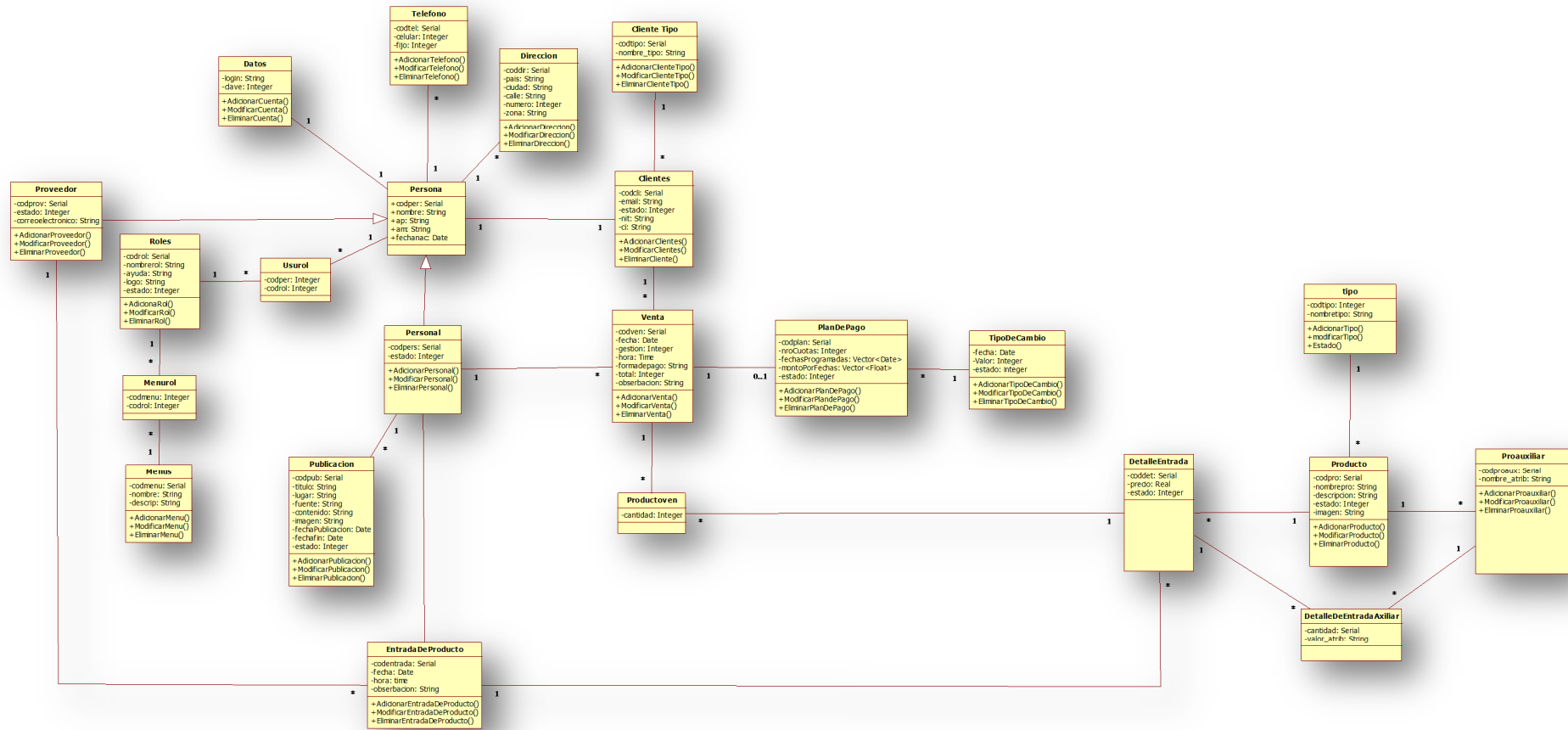


Figura N° 89: Diagrama de Clases

II.1.13. Creación de la Base de Datos

Nombre de la Base de la Base de Datos es “DigitalCorporation”

```
CREATE TABLE cliente
```

```
(
```

```
codcli serial NOT NULL,
```

```
nit character varying(50),
```

```
email character varying(100),
```

```
estado integer,
```

```
codper integer,
```

```
codtipocli integer,
```

```
CONSTRAINT cliente_pkey PRIMARY KEY (codcli ),
```

```
CONSTRAINT cliente_codper_fkey FOREIGN KEY (codper)
```

```
REFERENCES persona (codper) MATCH SIMPLE
```

```
ON UPDATE NO ACTION ON DELETE NO ACTION,
```

```
CONSTRAINT cliente_codtipocli_fkey FOREIGN KEY (codtipocli)
```

```
REFERENCES tipocliente (codtipocli) MATCH SIMPLE
```

```
ON UPDATE NO ACTION ON DELETE NO ACTION
```

```
)
```

```
CREATE TABLE cotizacion
```

```
(
```

```
codcot serial NOT NULL,
```

```
fecha date,
```

hora time without time zone,

codcli integer,

codpers integer,

CONSTRAINT cotizacion_pkey PRIMARY KEY (codcot),

CONSTRAINT cotizacion_codcli_fkey FOREIGN KEY (codcli)

REFERENCES cliente (codcli) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT cotizacion_codpers_fkey FOREIGN KEY (codpers)

REFERENCES personal (codpers) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

CREATE TABLE datos

(

codper serial NOT NULL,

login character varying(50) NOT NULL,

clave character varying(50) NOT NULL,

estado smallint NOT NULL,

CONSTRAINT datos_pkey PRIMARY KEY (codper),

CONSTRAINT datos_codper_fkey FOREIGN KEY (codper)

REFERENCES persona (codper) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

```

CREATE TABLE detalleentradaauxiliar
(
coddetaux serial NOT NULL,
valoratrib character varying(50),
coddet integer,
codproaux integer,
CONSTRAINT detalleentradaauxiliar_pkey PRIMARY KEY (coddetaux ),
CONSTRAINT detalleentradaauxiliar_coddet_fkey FOREIGN KEY (coddet)
REFERENCES detallentradaproducto (coddet) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT detalleentradaauxiliar_codproaux_fkey FOREIGN KEY (codproaux)
REFERENCES productoauxiliar (codproaux) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE detallentradaproducto
(
coddet serial NOT NULL,
codentrada integer NOT NULL,
codpro integer NOT NULL,
cant integer NOT NULL,
estado integer,
CONSTRAINT detallentradaproducto_pkey PRIMARY KEY (coddet ),

```

```
CONSTRAINT detallentradaproducto_codentrada_fkey FOREIGN KEY (codentrada)
REFERENCES entradaproducto (codentrada) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT detallentradaproducto_codpro_fkey FOREIGN KEY (codpro)
REFERENCES producto (codpro) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

```
CREATE TABLE direccion
(
coddir serial NOT NULL,
pais character varying(50),
ciudad character varying(50),
zona character varying(50),
calle character varying(50),
numero integer,
codper integer,
CONSTRAINT direccion_pkey PRIMARY KEY (coddir ),
CONSTRAINT direccion_codper_fkey FOREIGN KEY (codper)
REFERENCES persona (codper) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

```

CREATE TABLE entradaproducto
(
codentrada integer NOT NULL,
gestion character varying(4) NOT NULL,
fecha date NOT NULL,
codper integer,
estado smallint NOT NULL,
CONSTRAINT entradaproducto_pkey PRIMARY KEY (codentrada ),
CONSTRAINT entradaproducto_codper_fkey FOREIGN KEY (codper)
REFERENCES persona (codper) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE menurol
(
codmenu integer NOT NULL,
codrol integer NOT NULL,
CONSTRAINT menurol_pkey PRIMARY KEY (codmenu , codrol ),
CONSTRAINT menurol_codmenu_fkey FOREIGN KEY (codmenu)
REFERENCES menus (codmenu) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT menurol_codrol_fkey FOREIGN KEY (codrol)
REFERENCES roles (codrol) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```
CREATE TABLE menus
```

```
(  
  codmenu integer NOT NULL,  
  nombre character varying(50) NOT NULL,  
  descrip character varying(120) NOT NULL,  
  CONSTRAINT menus_pkey PRIMARY KEY (codmenu )  
)
```

```
CREATE TABLE persona
```

```
(  
  codper serial NOT NULL,  
  nombre character varying(50) NOT NULL,  
  ap character varying(50) NOT NULL,  
  am character varying(50) NOT NULL,  
  estado smallint NOT NULL,  
  ecivil smallint NOT NULL,  
  fnac date NOT NULL,  
  foto character varying(120),  
  email character varying(120),  
  CONSTRAINT persona_pkey PRIMARY KEY (codper )  
)
```

```
CREATE TABLE personal
(
codpers serial NOT NULL,
estado integer,
codper integer,
CONSTRAINT personal_pkey PRIMARY KEY (codpers ),
CONSTRAINT personal_codper_fkey FOREIGN KEY (codper)
REFERENCES persona (codper) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

```
CREATE TABLE planpagoventa
(
codplan serial NOT NULL,
fecha date,
monto real,
observacion character varying(50),
estado integer,
codven integer,
codtipocam integer,
hora character varying,
CONSTRAINT planpagoventa_pkey PRIMARY KEY (codplan ),
CONSTRAINT planpagoventa_codtipocam_fkey FOREIGN KEY (codtipocam)
REFERENCES tipodecambio (codtipocam) MATCH SIMPLE
```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT planpagoventa_codven_fkey FOREIGN KEY (codven)
        REFERENCES venta (codven) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
    )

```

```

CREATE TABLE producto
(
    codpro integer NOT NULL,
    nombre character varying(50) NOT NULL,
    descrip character varying(120) NOT NULL,
    estado smallint NOT NULL,
    foto character varying(50),
    codt integer NOT NULL,
    CONSTRAINT producto_pkey PRIMARY KEY (codpro ),
    CONSTRAINT producto_codt_fkey FOREIGN KEY (codt)
        REFERENCES tipoproducto (codt) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE productoauxiliar
(
    codproaux serial NOT NULL,

```

```

nombre_atrib character varying(50),
codpro integer,
tipo integer,
CONSTRAINT productoauxiliar_pkey PRIMARY KEY (codproaux ),
CONSTRAINT productoauxiliar_codpro_fkey FOREIGN KEY (codpro)
REFERENCES producto (codpro) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE productocot
(
codprocot serial NOT NULL,
codcot integer,
coddet integer,
precio real,
CONSTRAINT productocot_pkey PRIMARY KEY (codprocot ),
CONSTRAINT productocot_codcot_fkey FOREIGN KEY (codcot)
REFERENCES cotizacion (codcot) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT productocot_coddet_fkey FOREIGN KEY (coddet)
REFERENCES detallentradaproducto (coddet) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```
CREATE TABLE productores
(
codprores serial NOT NULL,
codres integer,
coddet integer,
precio real,
CONSTRAINT productores_pkey PRIMARY KEY (codprores ),
CONSTRAINT productores_coddet_fkey FOREIGN KEY (coddet)
REFERENCES detallentradaproducto (coddet) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT productores_codres_fkey FOREIGN KEY (codres)
REFERENCES reserva (codres) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

```
CREATE TABLE productoven
(
codproven serial NOT NULL,
codven integer,
coddet integer,
precio real,
```

```

CONSTRAINT productoven_pkey PRIMARY KEY (codproven ),
CONSTRAINT productoven_coddet_fkey FOREIGN KEY (coddet)
REFERENCES detallentradaproducto (coddet) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT productoven_codven_fkey FOREIGN KEY (codven)
REFERENCES venta (codven) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE proveedor_persona
(
codprov serial NOT NULL,
email character varying(50),
estado integer,
codper integer,
CONSTRAINT proveedor_persona_pkey PRIMARY KEY (codprov ),
CONSTRAINT proveedor_persona_codper_fkey FOREIGN KEY (codper)
REFERENCES persona (codper) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```
CREATE TABLE publicacion
(
codpub serial NOT NULL,
titulo character varying(50),
lugar character varying(50),
fuente character varying(50),
contenido character varying(120),
    fecha_publicacion date,
    fecha_fin date,
estado integer,
codpers integer,
    CONSTRAINT publicacion_pkey PRIMARY KEY (codpub ),
    CONSTRAINT publicacion_codpers_fkey FOREIGN KEY (codpers)
        REFERENCES personal (codpers) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)

CREATE TABLE roles
(
codrol integer NOT NULL,
nombre character varying(50) NOT NULL,
ayuda character varying(200) NOT NULL,
logo character varying(120),
estado smallint NOT NULL,
    CONSTRAINT roles_pkey PRIMARY KEY (codrol )
)
```

```
CREATE TABLE telefono
(
codtel serial NOT NULL,
nro character varying(50),
tipo integer,
codper integer,
CONSTRAINT telefono_pkey PRIMARY KEY (codtel ),
CONSTRAINT telefono_codper_fkey FOREIGN KEY (codper)
REFERENCES persona (codper) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

CREATE TABLE tipocliente
(
codtipocli serial NOT NULL,
nombre character varying(100),
CONSTRAINT tipocliente_pkey PRIMARY KEY (codtipocli )
)

CREATE TABLE tipodecambio
(
codtipocam serial NOT NULL,
fecha date,
valor real,
estado integer,
CONSTRAINT tipodecambio_pkey PRIMARY KEY (codtipocam )
```

)

CREATE TABLE tipoproducto

(

codt integer NOT NULL,

nombre character varying(50) NOT NULL,

estado smallint NOT NULL,

CONSTRAINT tipoproducto_pkey PRIMARY KEY (codt)

)

CREATE TABLE usurol

(

codper integer NOT NULL,

codrol integer NOT NULL,

CONSTRAINT usurol_pkey PRIMARY KEY (codper , codrol),

CONSTRAINT usurol_codper_fkey FOREIGN KEY (codper)

REFERENCES persona (codper) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT usurol_codrol_fkey FOREIGN KEY (codrol)

REFERENCES roles (codrol) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

```
CREATE TABLE venta
(
codven serial NOT NULL,
fecha date,
hora time without time zone,
formapago character varying(50),
observacion character varying(50),
codcli integer,
codpers integer,
CONSTRAINT venta_pkey PRIMARY KEY (codven ),
CONSTRAINT venta_codcli_fkey FOREIGN KEY (codcli)
REFERENCES cliente (codcli) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT venta_codpers_fkey FOREIGN KEY (codpers)
REFERENCES personal (codpers) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

II.1.14. Modelo de Datos

II.1.14.1. Introducción

A continuación se describe la creación de la base de datos del software, apuntando su característica orientada a objetos. Para expresar este modelo se utiliza un Diagrama de Clases de Objetos y luego su correspondiente implementación en el lenguaje utilizando para el desarrollo del proyecto el cual es JAVA.

Clase : Persona

```
package model.domain;
```

```
import java.util.Date;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```
public class Persona implements java.io.Serializable {
```

```
    private int codper;
```

```
    private String nombre;
```

```
    private String ap;
```

```
    private String am;
```

```
    private short estado;
```

```
    private short ecivil;
```

```
    private Date fnac;
```

```
    private String foto;
```

```
    private String email;
```

```
        private Datos datoses;

        private Set roleses = new HashSet(0);

        private Set telefonoes = new HashSet(0);

        private Set direcciones= new HashSet(0);


        public Set getDirecciones() {

            return direcciones;

        }


        public void setDirecciones(Set direcciones) {

            this.direcciones = direcciones;

        }


        public Set getTelefonoes() {

            return telefonoes;

        }


        public void setTelefonoes(Set telefonoes) {

            this.telefonoes = telefonoes;

        }


        public Persona() {

        }
```

```
public int getCodper() {  
    return codper;  
}  
  
public void setCodper(int codper) {  
    this.codper = codper;  
}  
  
public String getNombre() {  
    return this.nombre;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
  
public String getAp() {  
    return this.ap;  
}  
  
public void setAp(String ap) {  
    this.ap = ap;  
}  
  
public String getAm() {  
    return this.am;  
}
```

```
public void setAm(String am) {  
    this.am = am;  
}  
  
public short getEstado() {  
    return this.estado;  
}  
  
public void setEstado(short estado) {  
    this.estado = estado;  
}  
  
public short getEcivil() {  
    return this.ecivil;  
}  
  
public void setEcivil(short ecivil) {  
    this.ecivil = ecivil;  
}  
  
public Date getFnac() {  
    return this.fnac;  
}  
  
public void setFnac(Date fnac) {  
    this.fnac = fnac;  
}
```

```
public String getFoto() {  
    return this.foto;  
}  
  
public void setFoto(String foto) {  
    this.foto = foto;  
}  
  
public String getEmail() {  
    return this.email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public Set getRoleses() {  
    return this.roleses;  
}  
  
public void setRoleses(Set roleses) {  
    this.roleses = roleses;  
}  
  
    public Datos getDatoses() {  
        return datoses;  
    }  
  
    public void setDatoses(Datos datoses) {
```

```
        this.datoses = datoses;
    }

}
```

Clase : Producto

```
package model.domain;

import java.util.HashSet;
import java.util.Set;

public class Producto implements java.io.Serializable {

    private int codpro;
    private Tipoproducto tipoproducto;
    private String nombre;
    private String descrip;
    private short estado;
    private String foto;
```

```
private Set productoauxiliars = new HashSet(0);
```

```
private Set detalleentradas = new HashSet(0);
```

```
public Producto() {  
    }
```

```
public int getCodpro() {  
    return this.codpro;  
    }
```

```
public void setCodpro(int codpro) {  
    this.codpro = codpro;  
    }
```

```
public Tipoproducto getTipoproducto() {  
    return this.tipoproducto;  
    }
```

```
public void setTipoproducto(Tipoproducto tipoproducto) {  
    this.tipoproducto = tipoproducto;  
    }
```

```
public String getNombre() {  
    return this.nombre;  
    }
```

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}
```

```
public String getDescrip() {  
    return this.descrip;  
}
```

```
public void setDescrip(String descrip) {  
    this.descrip = descrip;  
}
```

```
public short getEstado() {  
    return this.estado;  
}
```

```
public void setEstado(short estado) {  
    this.estado = estado;  
}
```

```
public String getFoto() {  
    return this.foto;  
}
```

```
public void setFoto(String foto) {  
    this.foto = foto;  
}
```

```
public Set getDetalleentradas() {  
    return detalleentradas;  
}  
  
public void setDetalleentradas(Set detalleentradas) {  
    this.detalleentradas = detalleentradas;  
}  
  
public Set getProductoauxiliars() {  
    return productoauxiliars;  
}  
  
public void setProductoauxiliars(Set productoauxiliars) {  
    this.productoauxiliars = productoauxiliars;  
}  
}
```

Clase : Entradaproducto

```
package model.domain;  
  
import java.util.Date;  
import java.util.HashSet;
```

```
import java.util.Set;

public class Entradaproducto implements java.io.Serializable {

    private int codentrada;
    private Persona usuarios;
    private String gestion;
    private Date fecha;
    private short estado;
    private Set detallentradaproductos = new HashSet(0);

    public Entradaproducto() {
        }

    public int getCodentrada() {
        return this.codentrada;
    }

    public void setCodentrada(int codentrada) {
        this.codentrada = codentrada;
    }

    public Persona getUsuarios() {
```

```
return this.usuarios;
```

```
}
```

```
public void setUsuarios(Persona usuarios) {
```

```
    this.usuarios = usuarios;
```

```
}
```

```
public String getGestion() {
```

```
return this.gestion;
```

```
}
```

```
public void setGestion(String gestion) {
```

```
    this.gestion = gestion;
```

```
}
```

```
public Date getFecha() {
```

```
return this.fecha;
```

```
}
```

```
public void setFecha(Date fecha) {
```

```
    this.fecha = fecha;
```

```
}
```

```
public short getEstado() {
```

```
return this.estado;
```

```
}
```

```
public void setEstado(short estado) {
```

```
        this.estado = estado;
    }

    public Set getDetallentradaproductos() {
        return this.detallentradaproductos;
    }

    public void setDetallentradaproductos(Set detallentradaproductos) {
        this.detallentradaproductos = detallentradaproductos;
    }

}
```

Clase : Detallentradaproducto

```
package model.domain;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```
public class Detallentradaproducto implements java.io.Serializable {
```

```
private int coddet;  
private Producto producto;  
private Entradaproducto entradaproducto;  
private int cant;  
private Set productovens = new HashSet(0);  
private Set valoresatrib = new HashSet(0);  
private Set valoresatribdatos = new HashSet(0);  
private int estado;
```

```
public int getEstado() {  
    return estado;  
}
```

```
public void setEstado(int estado) {  
    this.estado = estado;  
}
```

```
public Set getValoresatribdatos() {  
    return valoresatribdatos;  
}
```

```
public void setValoresatribdatos(Set valoresatribdatos) {  
    this.valoresatribdatos = valoresatribdatos;  
}
```

```
public Set getValoresatrib() {  
    return valoresatrib;  
}
```

```
public void setValoresatrib(Set valoresatrib) {  
    this.valoresatrib = valoresatrib;  
}
```

```
public Detallentradaproducto() {  
}
```

```
public Producto getProducto() {  
    return this.producto;  
}
```

```
public void setProducto(Producto producto) {
```

```
        this.producto = producto;
    }

    public Entradaproducto getEntradaproducto() {
        return this.entradaproducto;
    }

    public void setEntradaproducto(Entradaproducto entradaproducto) {
        this.entradaproducto = entradaproducto;
    }

    public int getCant() {
        return this.cant;
    }

    public void setCant(int cant) {
        this.cant = cant;
    }

    public int getCoddet() {
        return coddet;
    }

    public void setCoddet(int coddet) {
        this.coddet = coddet;
    }
}
```

```
public Set getProductovens() {  
    return productovens;  
}  
  
public void setProductovens(Set productovens) {  
    this.productovens = productovens;  
}  
  
}  
  
  
Clase : Detalleentradaauxiliar  
  
package model.domain;  
  
public class Detalleentradaauxiliar implements java.io.Serializable {  
  
    private int coddetaux;  
    private Productoauxiliar productoauxiliar;
```

```
private Detallentradaproducto detallentradaproducto;

private String valoratrib;

public Detallentradaproducto getDetallentradaproducto() {
    return detallentradaproducto;
}

public void setDetallentradaproducto(Detallentradaproducto detallentradaproducto)
{
    this.detallentradaproducto = detallentradaproducto;
}

public String getValoratrib() {
    return valoratrib;
}

public void setValoratrib(String valoratrib) {
    this.valoratrib = valoratrib;
}

public Detalleentradaauxiliar() {
}

public Detalleentradaauxiliar(int coddetaux) {
    this.coddetaux = coddetaux;
}
```

```
}

public int getCoddetaux() {
    return this.coddetaux;
}

public void setCoddetaux(int coddetaux) {
    this.coddetaux = coddetaux;
}

public Productoauxiliar getProductoauxiliar() {
    return this.productoauxiliar;
}

public void setProductoauxiliar(Productoauxiliar productoauxiliar) {
    this.productoauxiliar = productoauxiliar;
}
}
```

Clase : Venta

```
package model.domain;
```

```
import java.util.Date;
import java.util.HashSet;
import java.util.Set;

public class Venta implements java.io.Serializable {

    private int codven;
    private Cliente cliente;
    private Personal personal;
    private Date fecha;
    private Date hora;
    private String formapago;
    private String observacion;
    private Set productovens = new HashSet(0);
    //private Set planpagoventas = new HashSet(0);

    public Venta() {
    }

    public Venta(int codven) {
        this.codven = codven;
    }
}
```

```
public int getCodven() {  
    return this.codven;  
}
```

```
public void setCodven(int codven) {  
    this.codven = codven;  
}
```

```
public Cliente getCliente() {  
    return this.cliente;  
}
```

```
public void setCliente(Cliente cliente) {  
    this.cliente = cliente;  
}
```

```
public Personal getPersonal() {  
    return this.personal;  
}
```

```
public void setPersonal(Personal personal) {  
    this.personal = personal;  
}
```

```
}
```

```
public Date getFecha() {  
    return this.fecha;  
}
```

```
public void setFecha(Date fecha) {  
    this.fecha = fecha;  
}
```

```
public Date getHora() {  
    return this.hora;  
}
```

```
public void setHora(Date hora) {  
    this.hora = hora;  
}
```

```
public String getFormapago() {  
    return this.formapago;  
}
```

```
public void setFormapago(String formapago) {  
    this.formapago = formapago;  
}
```

```
public String getObservacion() {  
    return this.observacion;  
}  
  
public void setObservacion(String observacion) {  
    this.observacion = observacion;  
}  
  
public Set getProductovens() {  
    return productovens;  
}  
  
public void setProductovens(Set productovens) {  
    this.productovens = productovens;  
}  
}
```

Clase : Planpagoventa

```
package model.domain;
```

```
import java.util.Date;

public class Planpagoventa implements java.io.Serializable {

    private int codplan;
    private Venta venta;
    private Tipodecambio tipodecambio;
    private Date fecha;
    private String hora;
    private double monto;
    private String observacion;
    private int estado;

    public Planpagoventa() {
    }

    public int getCodplan() {
        return codplan;
    }

    public void setCodplan(int codplan) {
        this.codplan = codplan;
    }
}
```

```
public int getEstado() {  
    return estado;  
}
```

```
public void setEstado(int estado) {  
    this.estado = estado;  
}
```

```
public Date getFecha() {  
    return fecha;  
}
```

```
public void setFecha(Date fecha) {  
    this.fecha = fecha;  
}
```

```
public String getHora() {  
    return hora;  
}
```

```
public void setHora(String hora) {  
    this.hora = hora;  
}
```

```
public double getMonto() {  
    return monto;  
}
```

```
public void setMonto(double monto) {  
    this.monto = monto;  
}
```

```
public String getObservacion() {  
    return observacion;  
}
```

```
public void setObservacion(String observacion) {  
    this.observacion = observacion;  
}
```

```
public Tipodecambio getTipodecambio() {  
    return tipodecambio;  
}
```

```
public void setTipodecambio(Tipodecambio tipodecambio) {  
    this.tipodecambio = tipodecambio;  
}
```

```
public Venta getVenta() {
```

```
        return venta;
    }

    public void setVenta(Venta venta) {
        this.venta = venta;
    }
}
```

Clase : Productoauxiliar

```
package model.domain;

import java.util.HashSet;
import java.util.Set;

public class Productoauxiliar implements java.io.Serializable {

    private int codproaux;
    private Producto producto;
```

```
private String nombre_atrib;  
private Set detalleentradaauxiliars = new HashSet(0);  
private int tipo;
```

```
public int getTipo() {  
    return tipo;  
}
```

```
public void setTipo(int tipo) {  
    this.tipo = tipo;  
}
```

```
public Productoauxiliar() {  
}
```

```
public Productoauxiliar(int codproaux) {  
    this.codproaux = codproaux;  
}
```

```
public int getCodproaux() {  
    return this.codproaux;  
}
```

```
public void setCodproaux(int codproaux) {  
    this.codproaux = codproaux;  
}
```

```
public Producto getProducto() {  
    return this.producto;  
}
```

```
public void setProducto(Producto producto) {  
    this.producto = producto;  
}
```

```
public Set getDetalleentradaauxiliars() {  
    return this.detalleentradaauxiliars;  
}
```

```
public void setDetalleentradaauxiliars(Set detalleentradaauxiliars) {  
    this.detalleentradaauxiliars = detalleentradaauxiliars;  
}
```

```
public String getNombre_atrib() {  
    return nombre_atrib;  
}
```

```
public void setNombre_atrib(String nombre_atrib) {
```

```
        this.nombre_atrib = nombre_atrib;
    }

}
```

Clase : Productoven

```
package model.domain;
```

```
public class Productoven implements java.io.Serializable {
```

```
    private int codproven;
```

```
    private Venta venta;
```

```
    private Detallentradaproducto detallentradaproducto;
```

```
    private float precio;
```

```
    public float getPrecio() {
```

```
        return precio;
```

```
    }
```

```
    public void setPrecio(float precio) {
```

```
        this.precio = precio;
    }

    public Productoven() {
    }

    public Productoven(int codproven) {
        this.codproven = codproven;
    }

    public int getCodproven() {
        return this.codproven;
    }

    public void setCodproven(int codproven) {
        this.codproven = codproven;
    }

    public Venta getVenta() {
        return this.venta;
    }

    public void setVenta(Venta venta) {
        this.venta = venta;
    }
}
```

```
}
```

```
public Detallentradaproducto getDetallentradaproducto() {  
    return detallentradaproducto;  
}
```

```
public void setDetallentradaproducto(Detallentradaproducto detallentradaproducto)  
{  
    this.detallentradaproducto = detallentradaproducto;  
}  
  
}
```

Clase : Roles

```
package model.domain;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```
public class Roles implements java.io.Serializable {
```

```
    private int codrol;
```

```
    private String nombre;
```

```
    private String ayuda;
```

```
    private String logo;
```

```
    private short estado;
```

```
    private Set personaes = new HashSet(0);
```

```
    private Set menuss = new HashSet(0);
```

```
    public Set getMenuss() {
```

```
        return menuss;
```

```
    }
```

```
    public void setMenuss(Set menuss) {
```

```
        this.menuss = menuss;
```

```
    }
```

```
    public Set getPersonaes() {
```

```
        return personaes;
```

```
}
```

```
public void setPersonaes(Set personaes) {  
    this.personaes = personaes;  
}
```

```
public Roles() {  
}
```

```
public Roles(int codm, String nombre, short estado) {  
    this.codrol = codm;  
    this.nombre = nombre;  
    this.estado = estado;  
}
```

```
public int getCodrol() {  
    return this.codrol;  
}
```

```
public void setCodrol(int codm) {  
    this.codrol = codm;  
}
```

```
public String getNombre() {  
    return this.nombre;  
}
```

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}
```

```
public String getAyuda() {  
    return this.ayuda;  
}
```

```
public void setAyuda(String ayuda) {  
    this.ayuda = ayuda;  
}
```

```
public String getLogo() {  
    return this.logo;  
}
```

```
public void setLogo(String logo) {  
    this.logo = logo;  
}
```

```
public short getEstado() {  
    return this.estado;  
}
```

```
public void setEstado(short estado) {  
    this.estado = estado;  
}  
  
}
```

Clase : Menus

```
package model.domain;  
  
import java.util.HashSet;  
import java.util.Set;  
public class Menus implements java.io.Serializable {  
  
    private int codmenu;  
    private String nombre;  
    private String descrip;  
    private Set roleses = new HashSet(0);  
  
    public Menus() {  
        }  
    }
```

```
public Menus(int codmenu, String nombre) {  
    this.codmenu = codmenu;  
    this.nombre = nombre;  
}
```

```
public int getCodmenu() {  
    return this.codmenu;  
}
```

```
public void setCodmenu(int codmenu) {  
    this.codmenu = codmenu;  
}
```

```
public String getNombre() {  
    return this.nombre;  
}
```

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}
```

```
public String getDescrip() {  
    return this.descrip;  
}
```

```
public void setDescrip(String descrip) {
```

```
        this.descrip = descrip;
    }

    public Set getRoleses() {
        return this.roleses;
    }

    public void setRoleses(Set roleses) {
        this.roleses = roleses;
    }

}
```

Clase : Menurol

```
package model.domain;
```

```
import java.util.HashSet;
import java.util.Set;
```

```
public class Menurol implements java.io.Serializable {
```

```
    private MenurolId id;
```

```
    private Roles roles;
```

```
    private Menus menus;
```

```
    public Menus getMenu() {
```

```
        return menus;
```

```
    }
```

```
    public void setMenu(Menus menu) {
```

```
        this.menus = menu;
```

```
    }
```

```
    public Menurol() {
```

```
    }
```

```
    public Menurol(MenurolId id, Roles roles,
```

```
        Menus menu) {
```

```
        this.id = id;
```

```
        this.roles = roles;
```

```
        this.menus = menu;
```

```
    }
```

```
public MenurolId getId() {  
    return id;  
}  
  
public void setId(MenurolId id) {  
    this.id = id;  
}  
  
public Roles getRoles() {  
    return roles;  
}  
  
public void setRoles(Roles roles) {  
    this.roles = roles;  
}  
  
}
```

Clase : Datos

```
package model.domain;
```

```
public class Datos implements java.io.Serializable {
```

```
    private int codper;
```

```
    private Persona persona;
```

```
    private String login;
```

```
    private String clave;
```

```
    private short estado;
```

```
    public Datos() {
```

```
    }
```

```
    public int getCodper() {
```

```
        return codper;
```

```
    }
```

```
    public void setCodper(int codper) {
```

```
        this.codper = codper;
```

```
    }
```

```
    public String getLogin() {
```

```
        return this.login;
```

```
    }
```

```
    public void setLogin(String login) {
```

```
        this.login = login;
```

```
    }  
  
    public String getClave() {  
        return this.clave;  
    }  
  
    public void setClave(String clave) {  
        this.clave = clave;  
    }  
  
    public short getEstado() {  
        return this.estado;  
    }  
  
    public void setEstado(short estado) {  
        this.estado = estado;  
    }  
  
    public Persona getPersona() {  
        return persona;  
    }  
  
    public void setPersona(Persona persona) {  
        this.persona = persona;  
    }  
  
}
```

II.1.15. PROPOTIPOS DE INTERFAZ DE USUARIO

II.1.15.1. Introducción

Se trata de prototipos que permiten al usuario hacerse una idea sobre las interfaces que proveerá el sistema.

II.1.15.1.1. Propósito

Presentar los prototipos de Pantallas para que el usuario tenga una idea de la interfaz que presentara el sistema.

II.1.15.1.2. Alcance

Mostrar los Prototipos de Pantallas, solamente el diseño que adoptarán todas.

II.1.15.2. Prototipo de Pantallas

Pantalla Principal

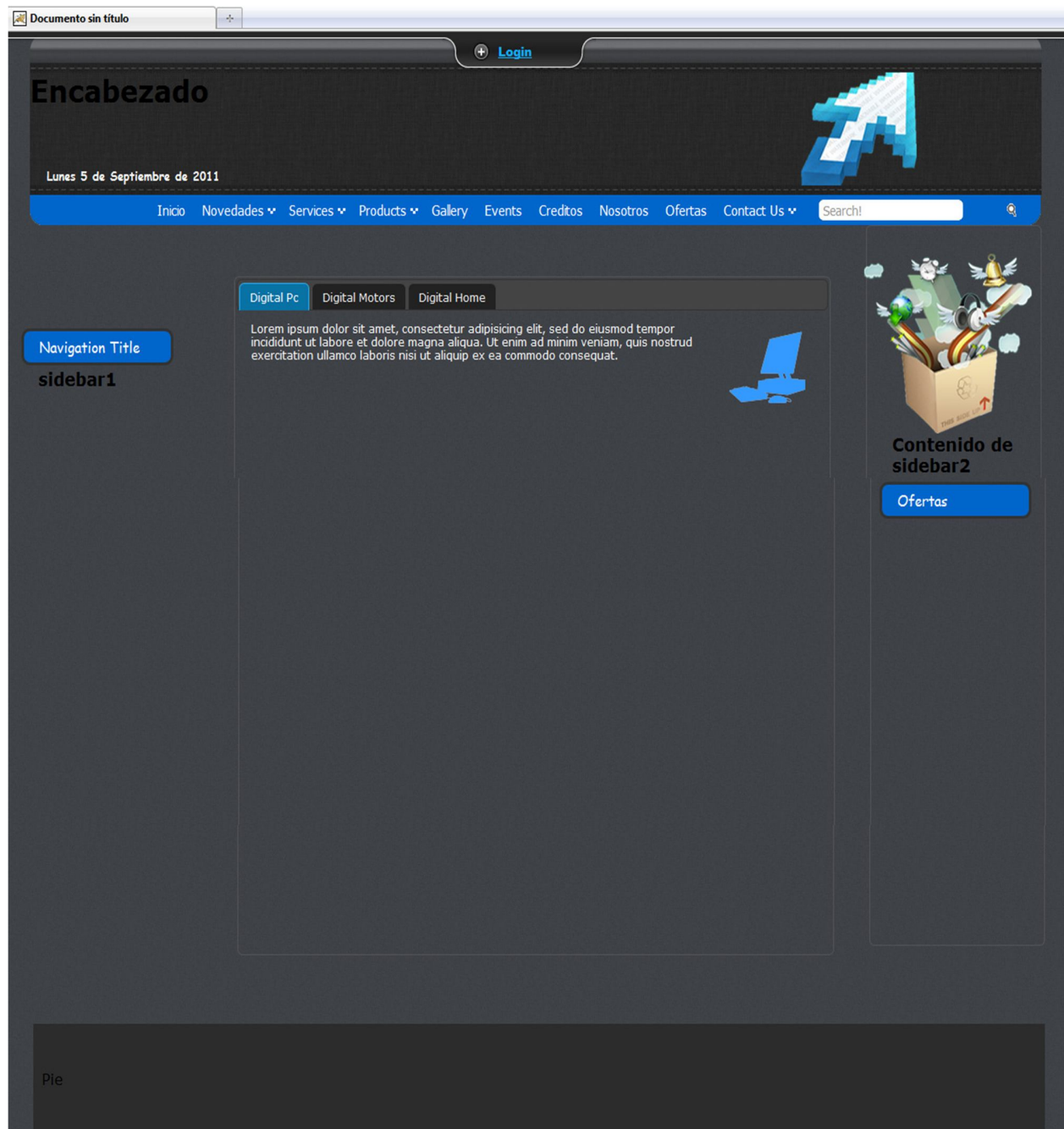


Figura N° 90: Pantalla Principal

Pantalla Presentación

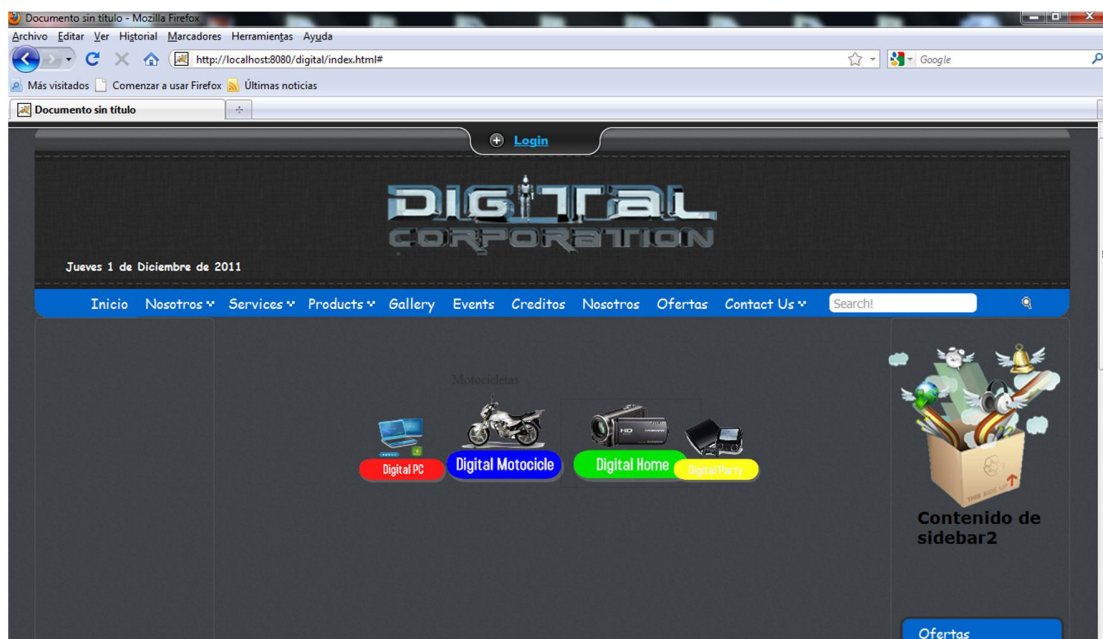


Figura N° 91: Pantalla Presentación

Pantalla Misión

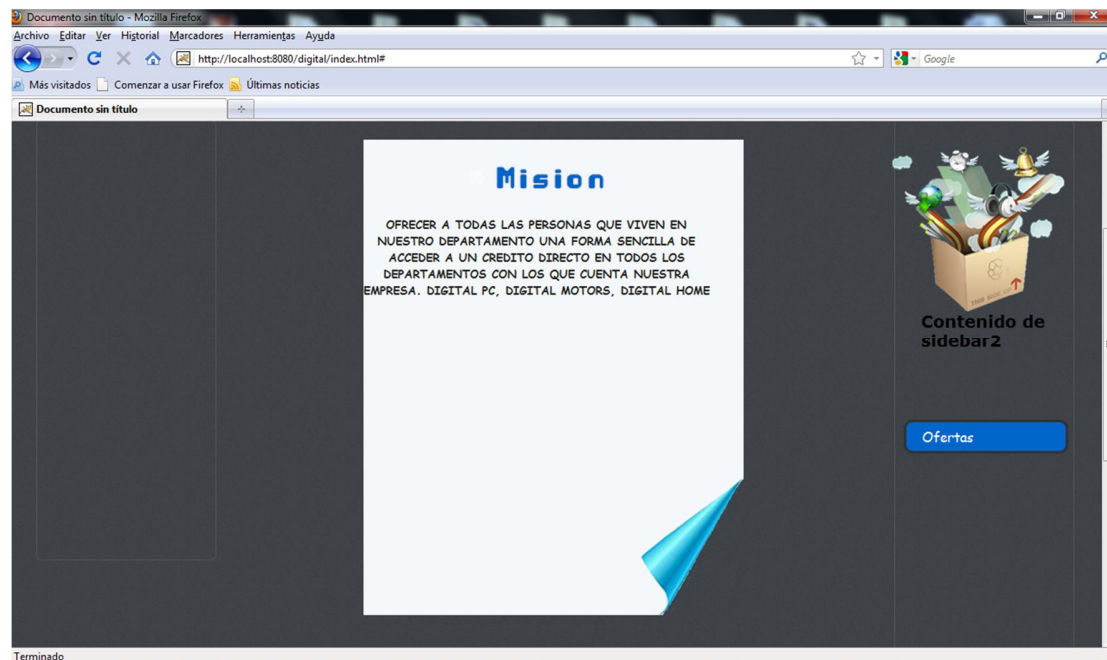


Figura N° 92: Pantalla Misión

Pantalla Visión

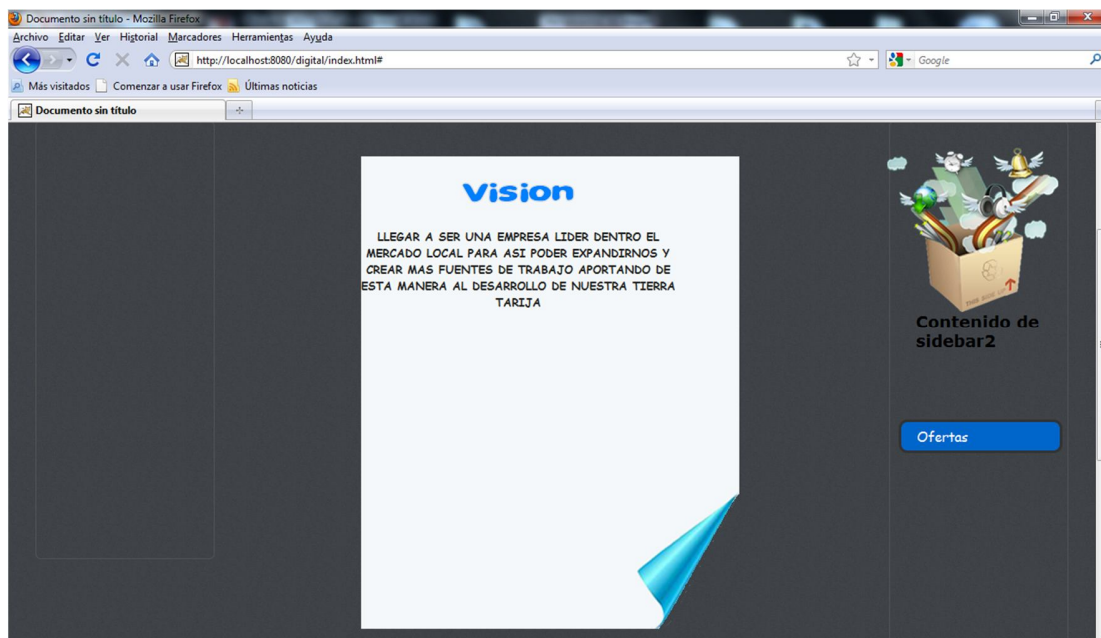


Figura N° 93: Pantalla Visión

Pantalla Introducción

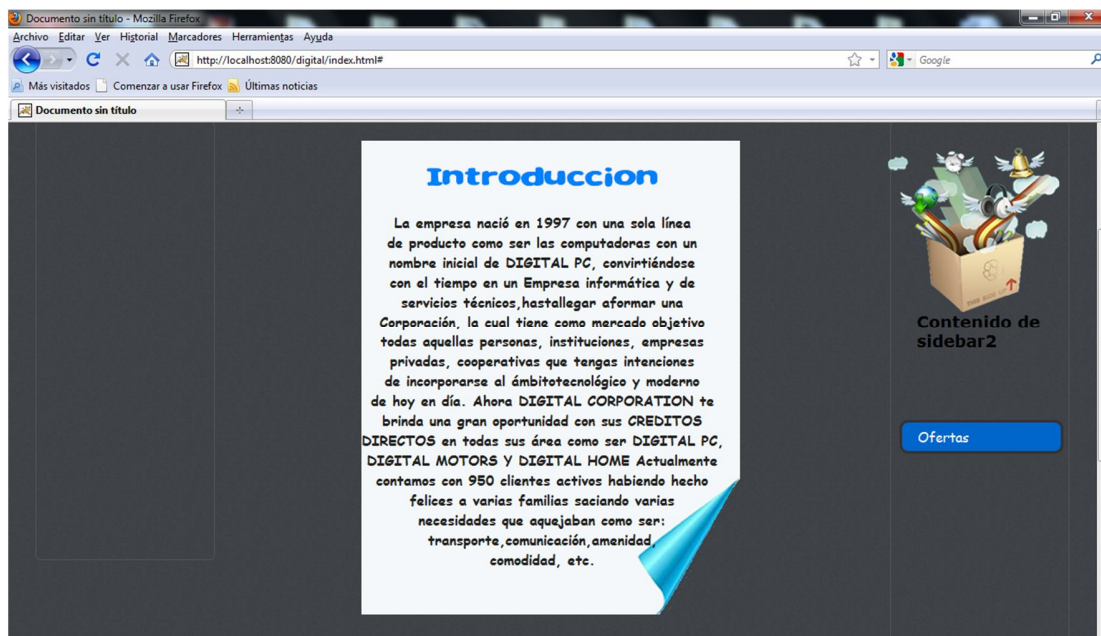


Figura N° 94: Pantalla Introducción

Pantalla Galería de Productos

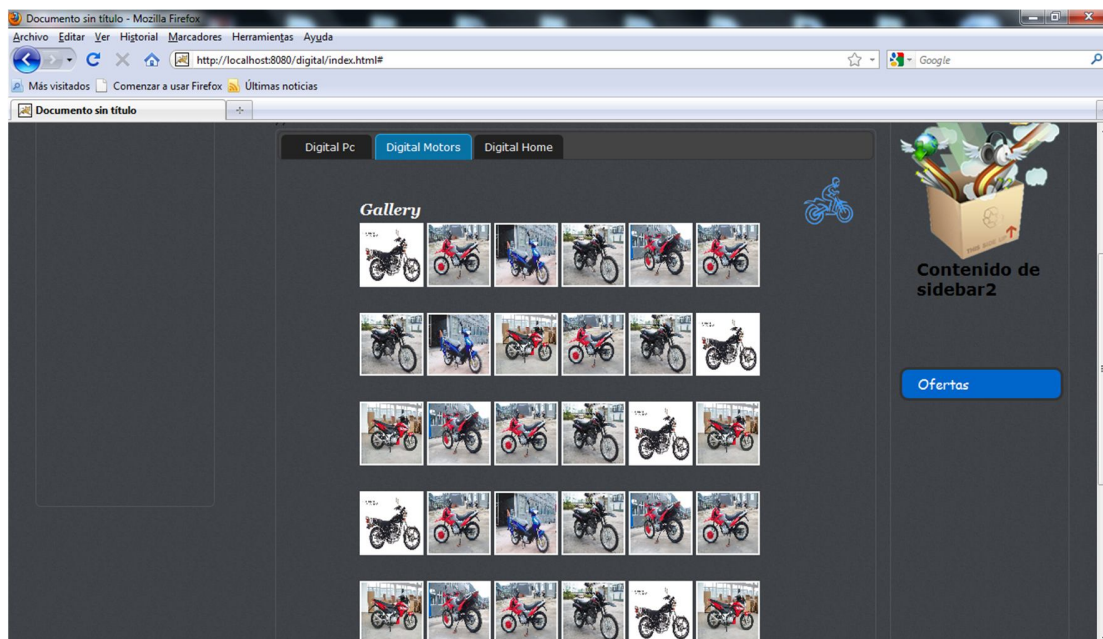


Figura N° 95: Pantalla Galería de Productos

Pantalla de Ingreso (Login)

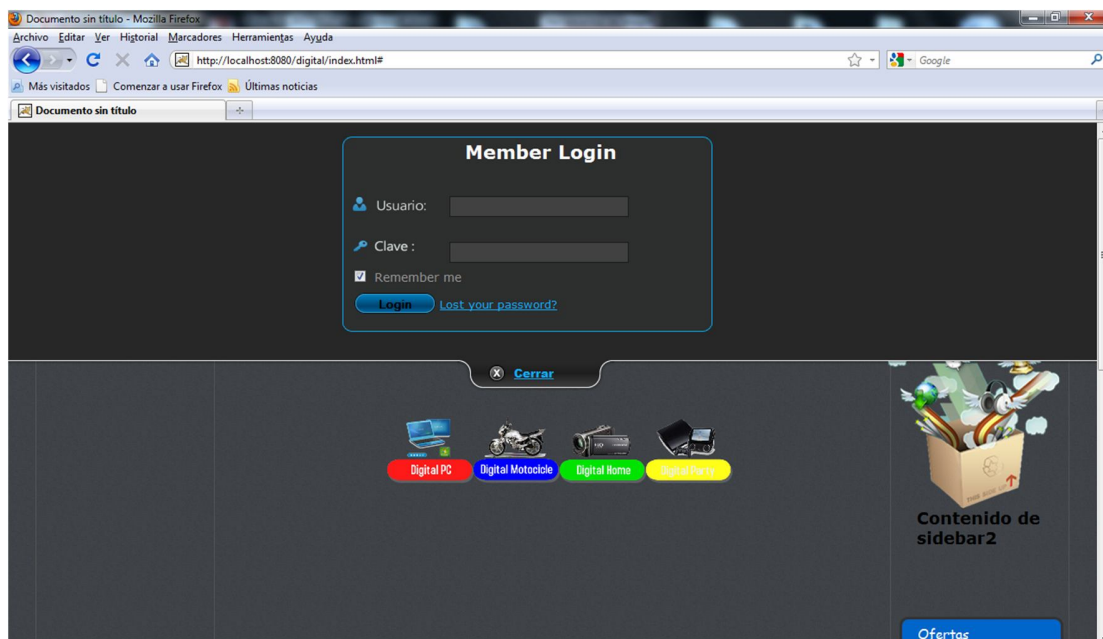


Figura N° 96: Pantalla Ingreso

Pantalla de Datos Ingresados (Login)

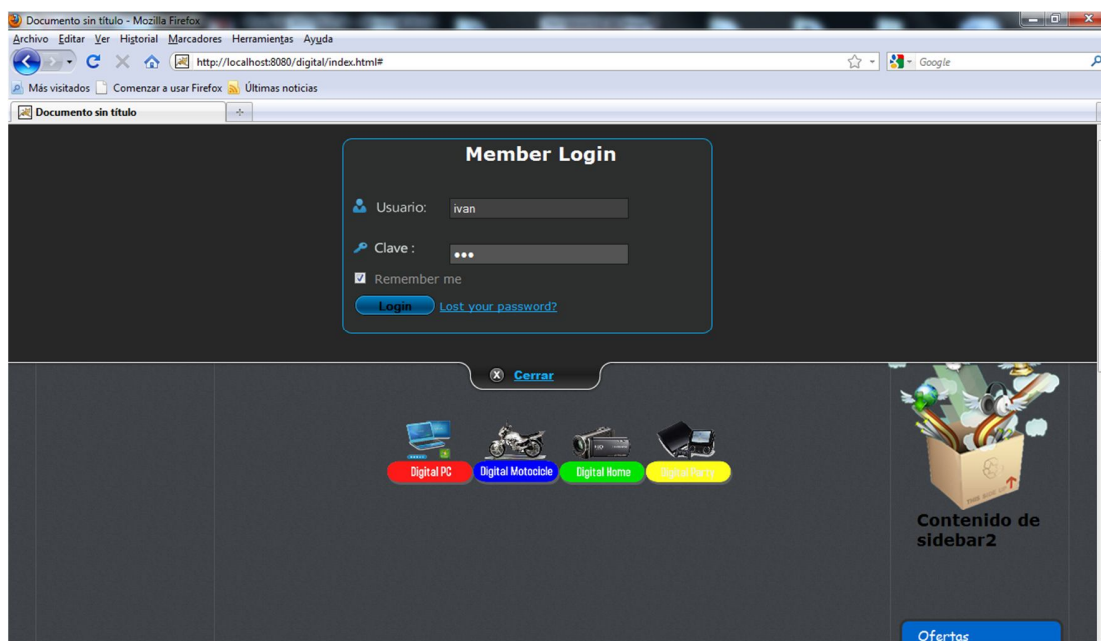


Figura N° 97: Pantalla Datos Ingresados

Pantalla de Usuario Aceptado (Login)

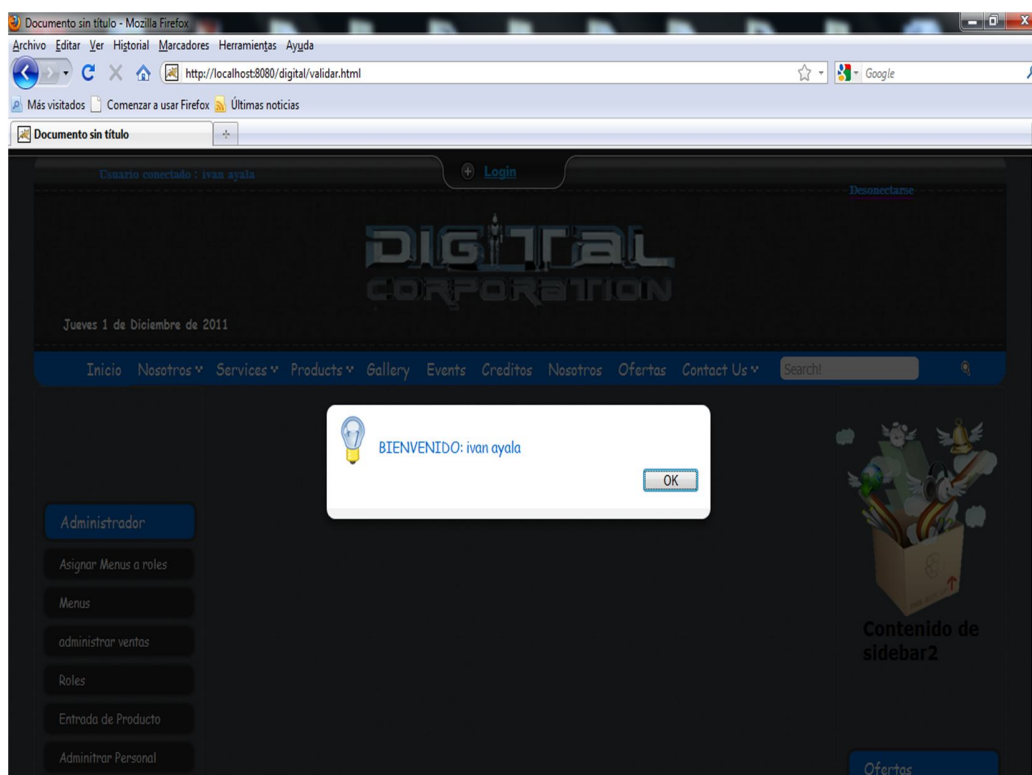


Figura N° 98 Pantalla Usuario Aceptado

Menú del Administrador

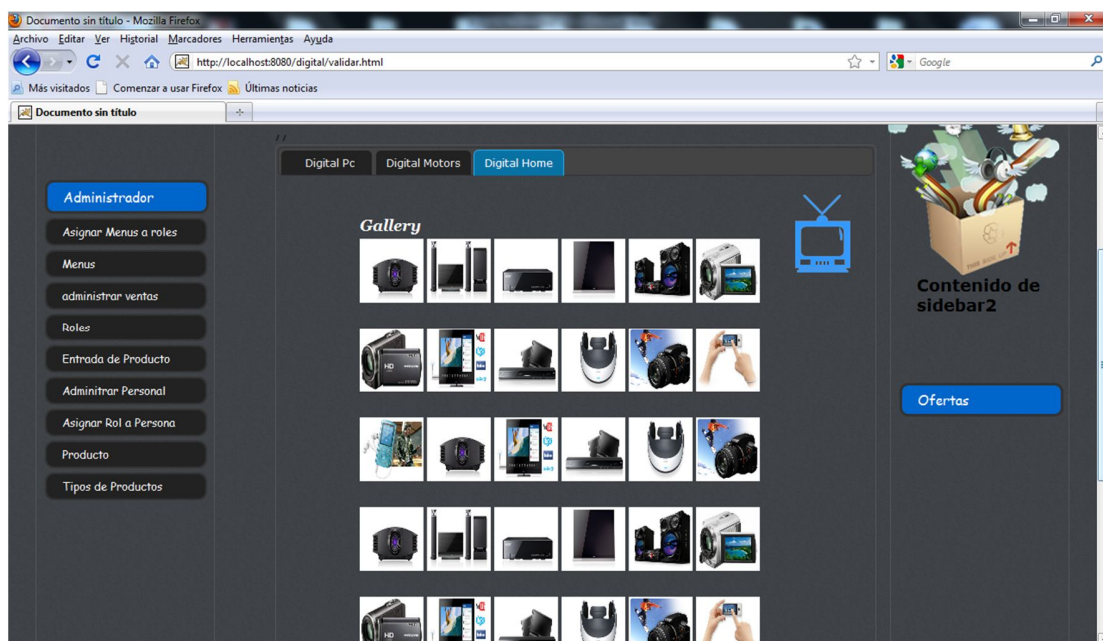


Figura N° 99: Pantalla Menú del Administrador

Gestionar Personal

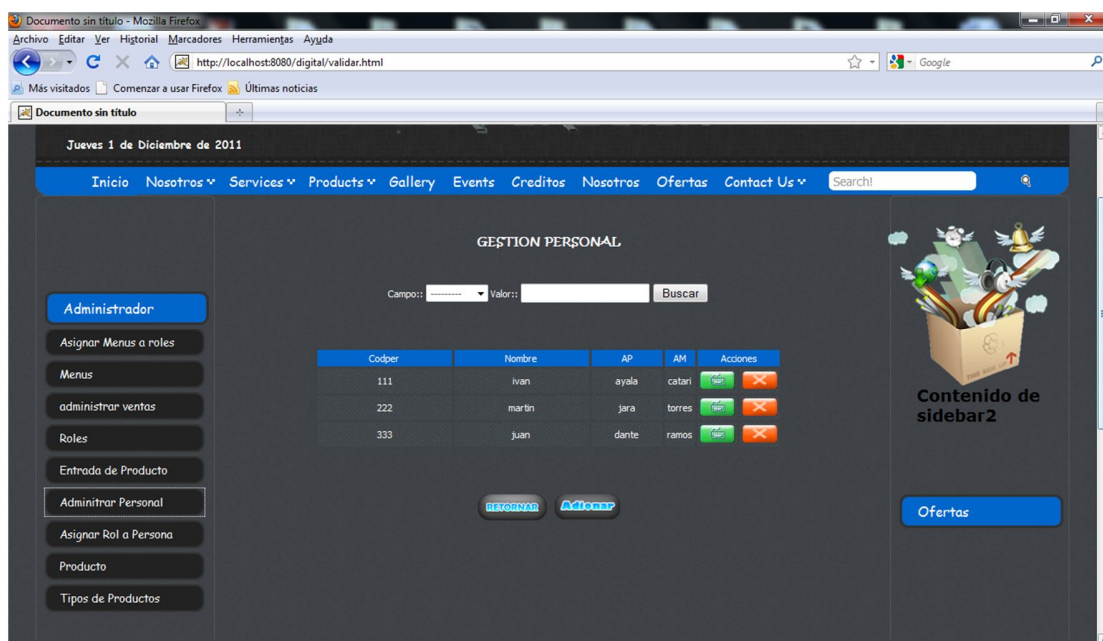
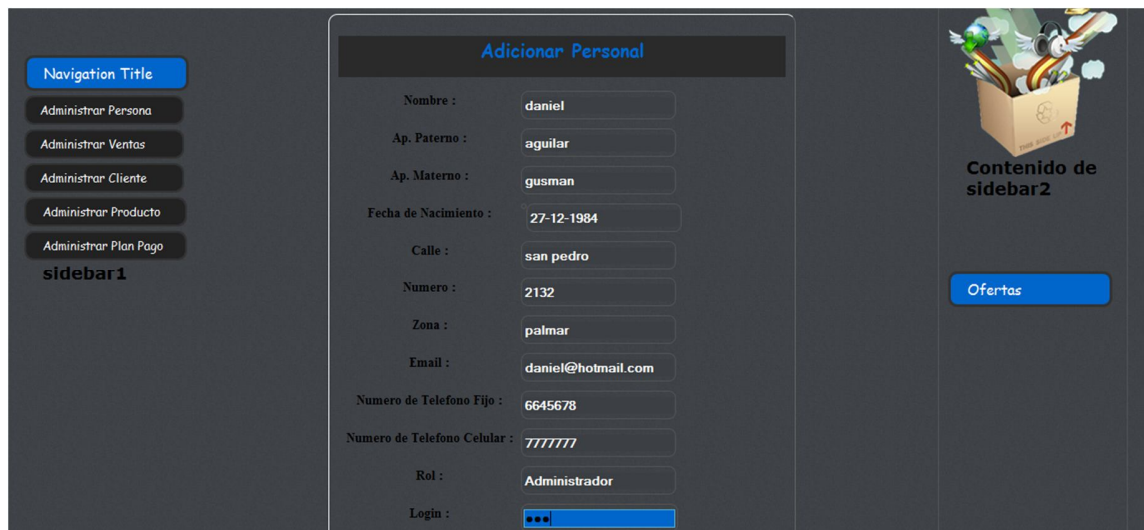


Figura N° 100: Pantalla Gestionar Personal

Adicionar Nuevo Personal



The screenshot shows a web application interface for adding a new person. On the left is a sidebar with a navigation menu. The main area contains a form titled 'Adicionar Personal' with various input fields. On the right is another sidebar with a box icon and a button.

Navigation Title

- Administrar Persona
- Administrar Ventas
- Administrar Cliente
- Administrar Producto
- Administrar Plan Pago

sidebar1

Adicionar Personal

Nombre : daniel

Ap. Paterno : aguilar

Ap. Materno : gusman

Fecha de Nacimiento : 27-12-1984

Calle : san pedro

Numero : 2132

Zona : palmar

Email : daniel@hotmail.com

Numero de Telefono Fijo : 6645678

Numero de Telefono Celular : 777777

Rol : Administrador

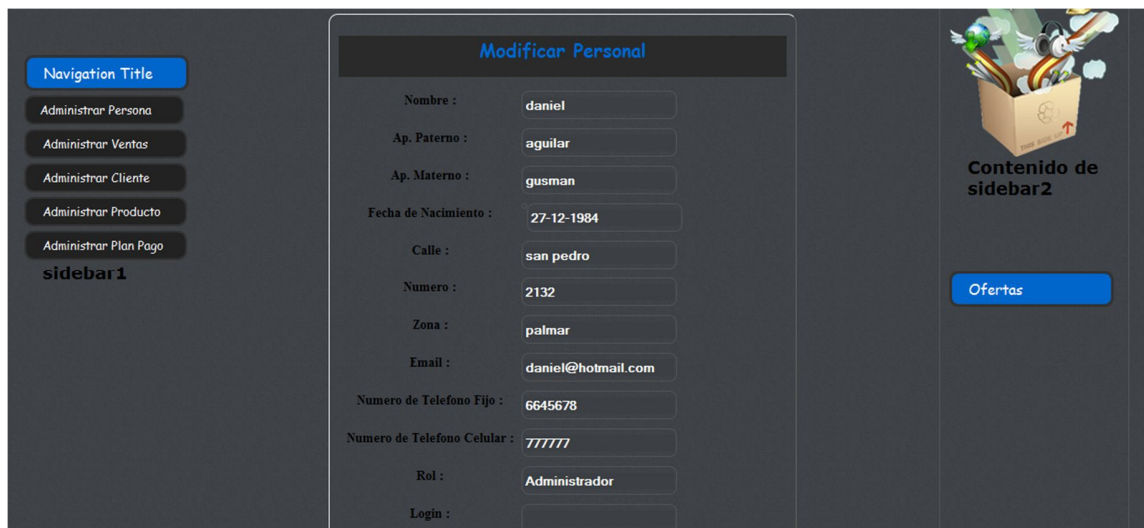
Login : ●●●

Contenido de sidebar2

Ofertas

Figura N° 101: Pantalla Adicionar Nuevo Personal

Modificar Personal



The screenshot shows a web application interface for modifying a person's details. The layout is similar to the previous form, but the title and the 'Login' field are different.

Navigation Title

- Administrar Persona
- Administrar Ventas
- Administrar Cliente
- Administrar Producto
- Administrar Plan Pago

sidebar1

Modificar Personal

Nombre : daniel

Ap. Paterno : aguilar

Ap. Materno : gusman

Fecha de Nacimiento : 27-12-1984

Calle : san pedro

Numero : 2132

Zona : palmar

Email : daniel@hotmail.com

Numero de Telefono Fijo : 6645678

Numero de Telefono Celular : 777777

Rol : Administrador

Login :

Contenido de sidebar2

Ofertas

Figura N° 102: Pantalla Modificar Personal

Eliminar Personal

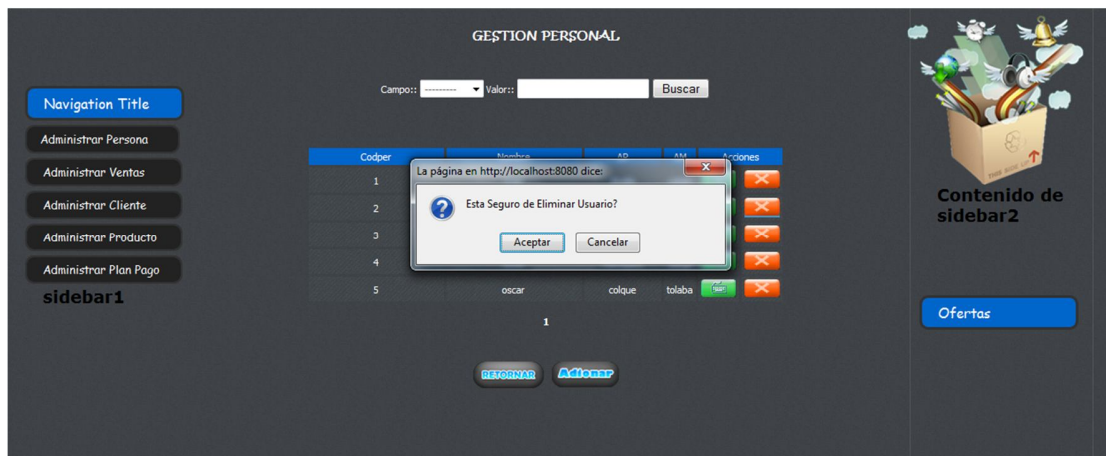


Figura N° 103: Pantalla Eliminar Personal

Gestión Roles

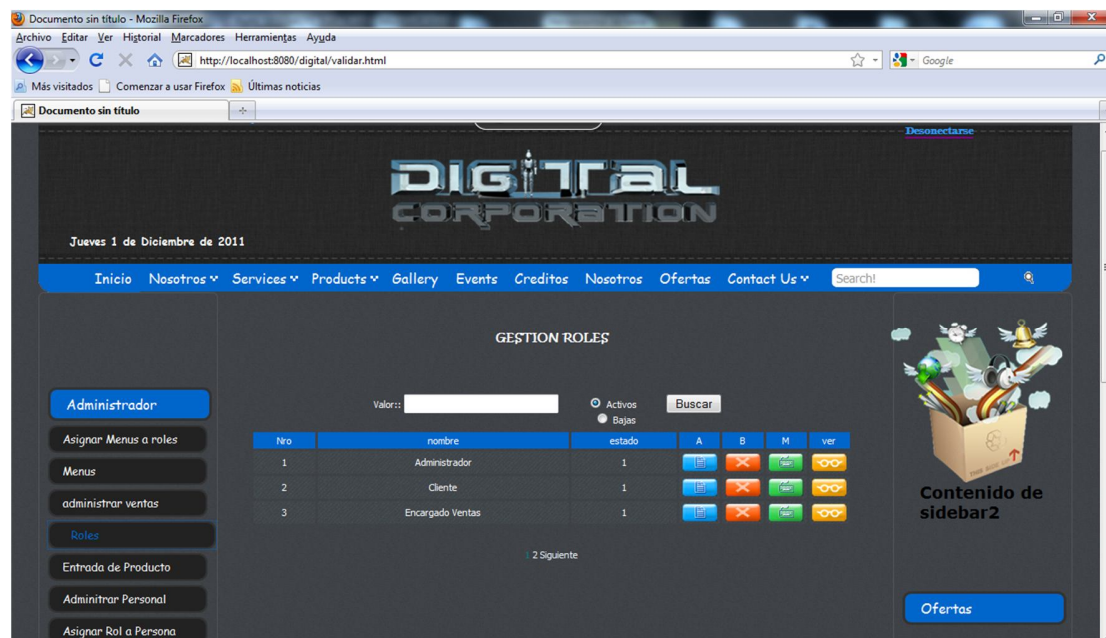


Figura N° 104: Pantalla Gestionar Roles

Adicionar Nuevo Rol

Documento sin título - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost:8080/digital/validar.html

Más visitados Comenzar a usar Firefox Últimas noticias

Documento sin título

Jueves 1 de Diciembre de 2011

Inicio Nosotros Services Products Gallery Events Credits Nosotros Ofertas Contact Us Search!

Administrador

Asignar Menus a roles

Menus

administrar ventas

Roles

Entrada de Producto

Adminitrar Personal

Asignar Rol a Persona

Producto

Tipos de Productos

..ADICIONAR NUEVO ROL..

codigo: 123

NOMBRE

Ayuda:

foto

Examinar...

adicionar cancelar

Contenido de sidebar2

Ofertas

Figura N° 105: Pantalla Adicionar Nuevo Rol

Asignar Menús al Rol

Documento sin título - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost:8080/digital/validar.html

Más visitados Comenzar a usar Firefox Últimas noticias

Documento sin título

Jueves 1 de Diciembre de 2011

Inicio Nosotros Services Products Gallery Events Credits Nosotros Ofertas Contact Us Search!

Administrador

Asignar Menus a roles

Menus

administrar ventas

Roles

Entrada de Producto

Adminitrar Personal

Asignar Rol a Persona

Producto

Tipos de Productos

..Menus..

Buscar: ROL filtro:

Nro	Nombre	A	B
1	Administrador		
2	Cliente		
3	Encargado Ventas		
4	Gerente ventas		

Contenido de sidebar2

Ofertas

Figura N° 106: Pantalla Asignar Menús al Rol

Gestión Menús

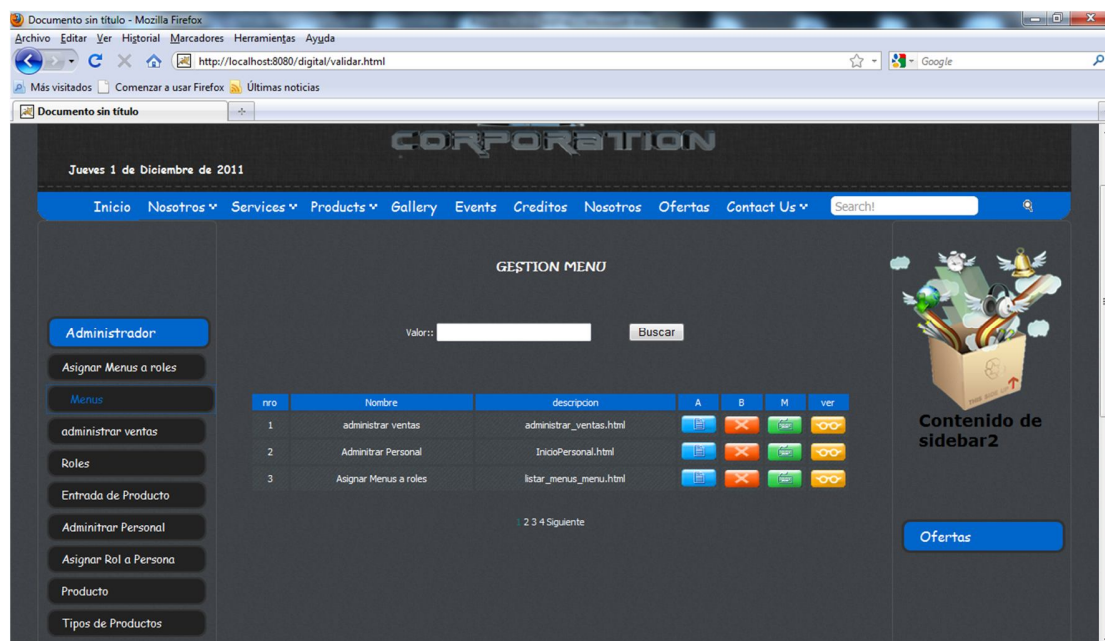


Figura N° 107: Pantalla Gestionar Menus

Pantalla Gestión Tipo Producto

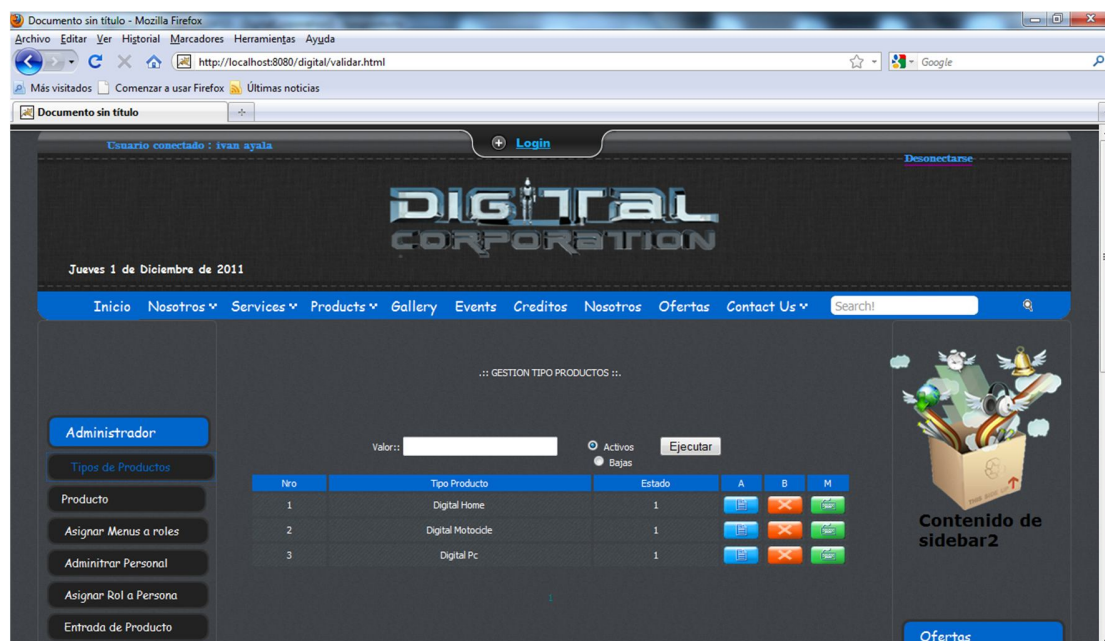


Figura N° 108: Pantalla Gestión Tipo Producto

Adicionar Nuevo Tipo de Producto

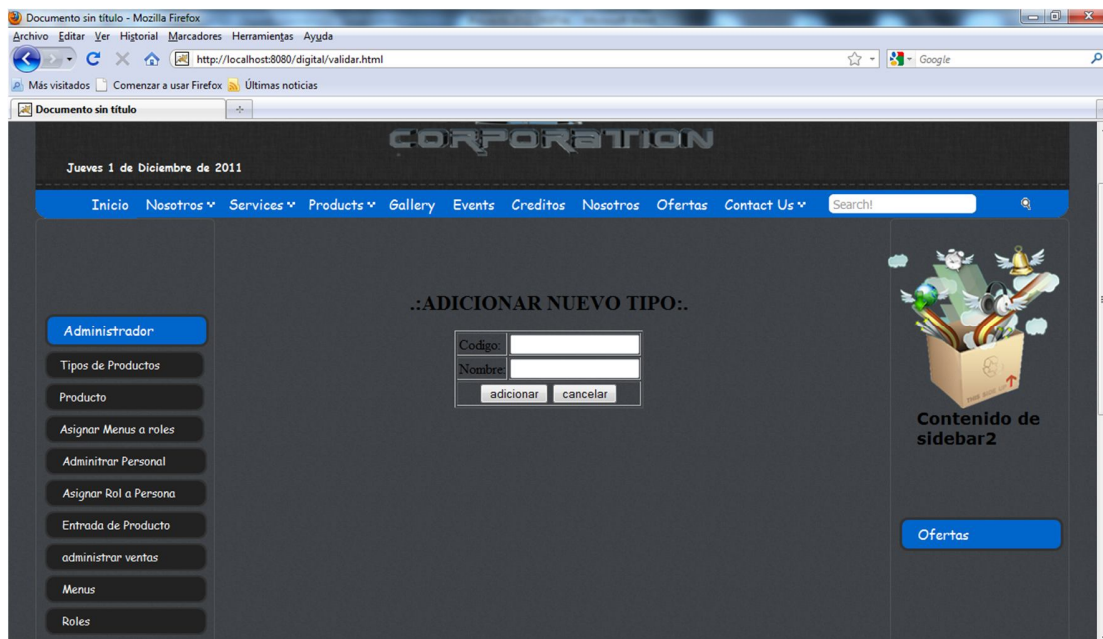


Figura N° 109: Pantalla Adicionar Nuevo Tipo de Producto

Modificar Tipo de Producto

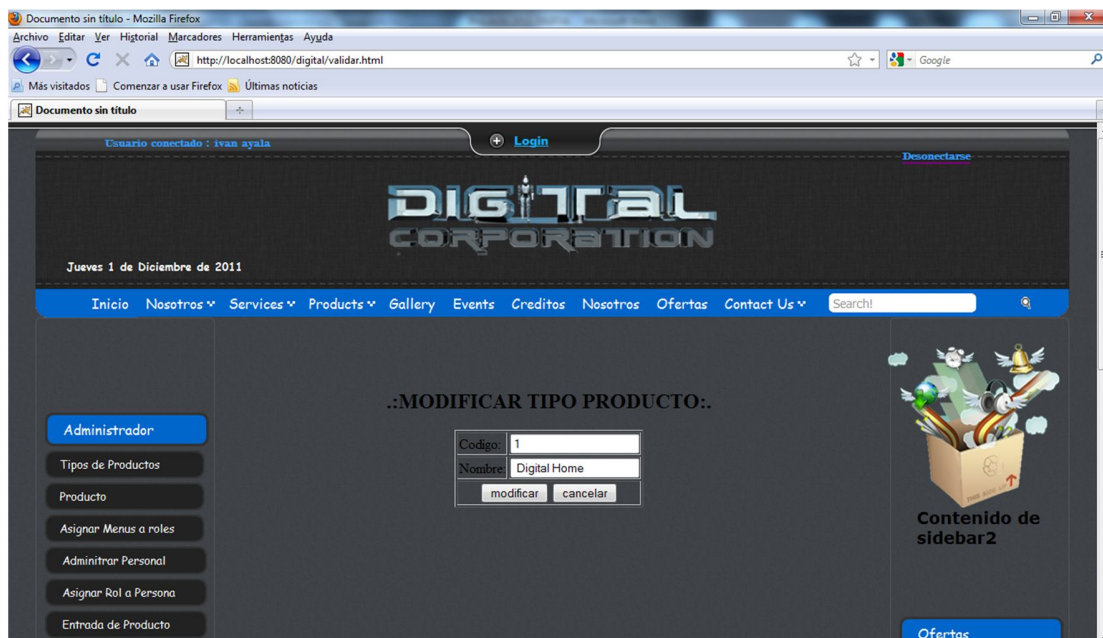


Figura N° 110: Pantalla Modificar Tipo de Producto

Pantalla Gestión Producto



Figura N° 111: Pantalla Gestión Producto

Adicionar Nuevo Producto

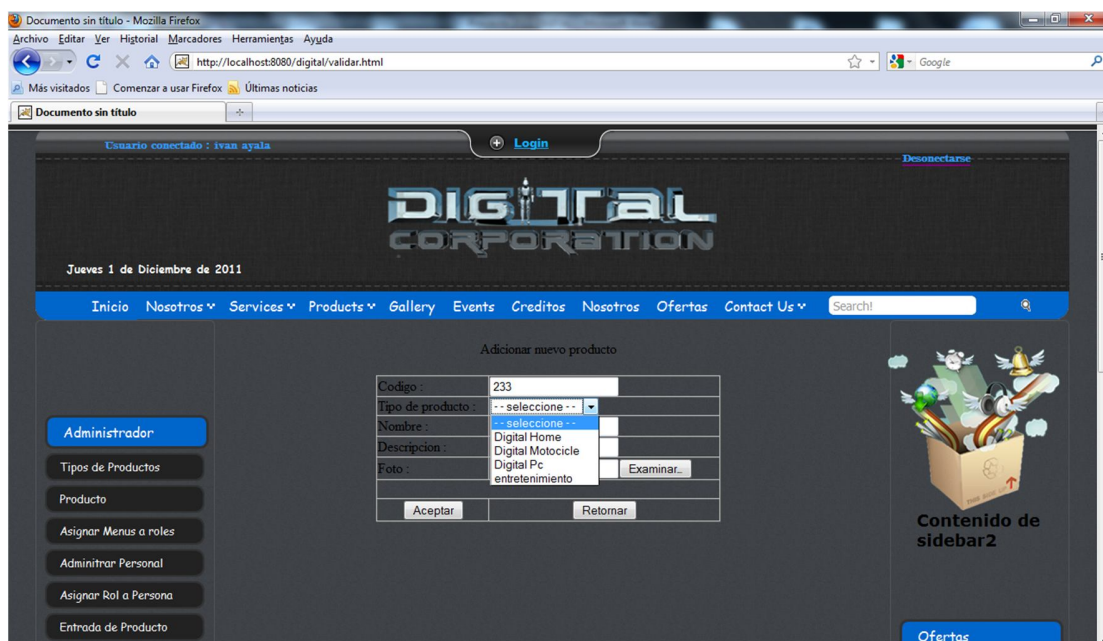


Figura N° 112: Pantalla Adicionar Nuevo Producto

Modificar Producto

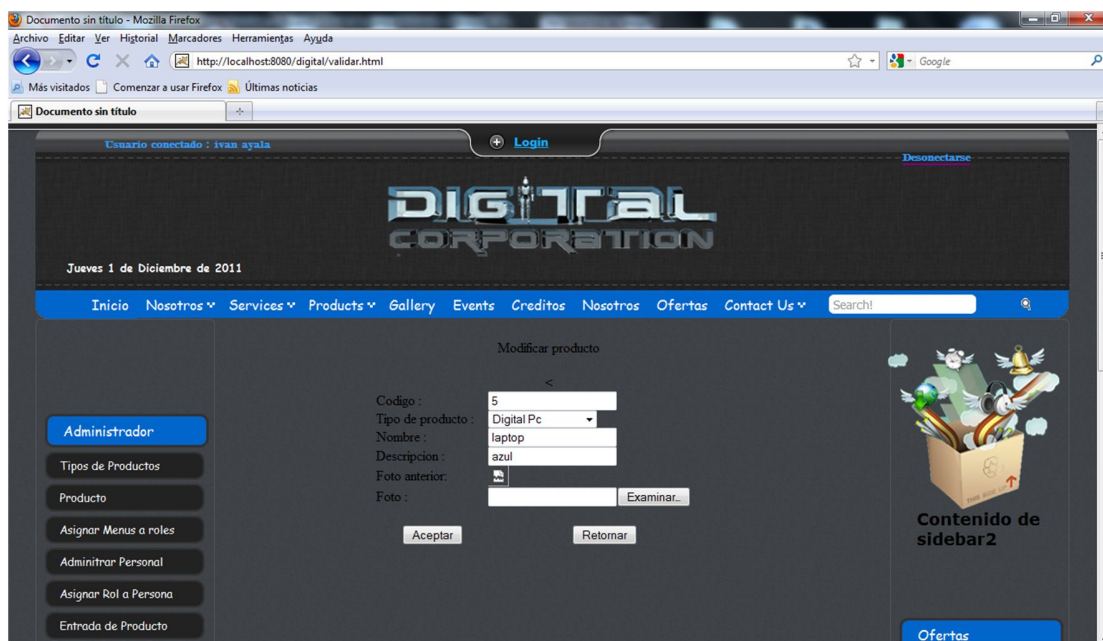


Figura N° 113: Pantalla Modificar Producto

Ver Producto

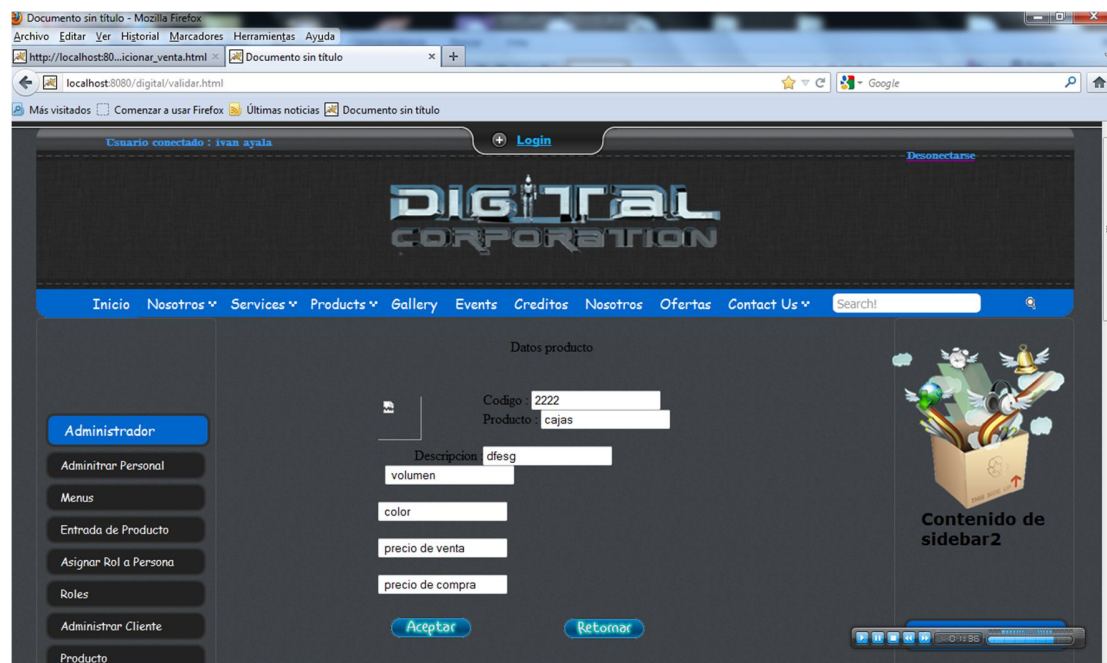


Figura N° 114: Pantalla Ver Producto

Pantalla Gestión Entrada Producto

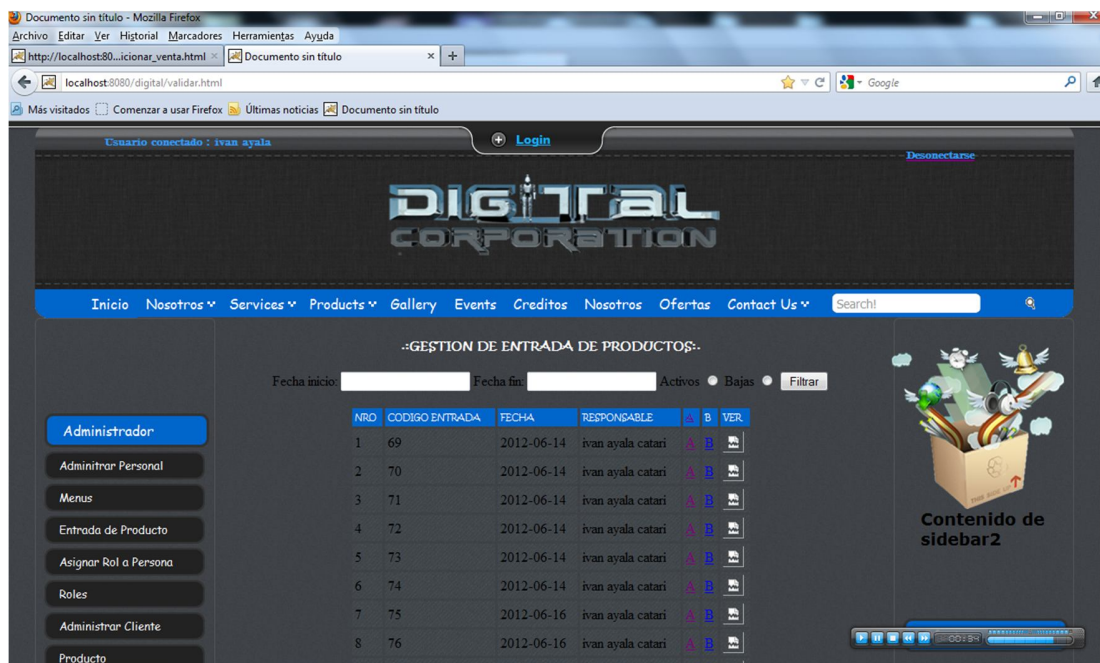


Figura N° 115: Pantalla Gestión Entrada Producto

Adicionar Entrada Producto

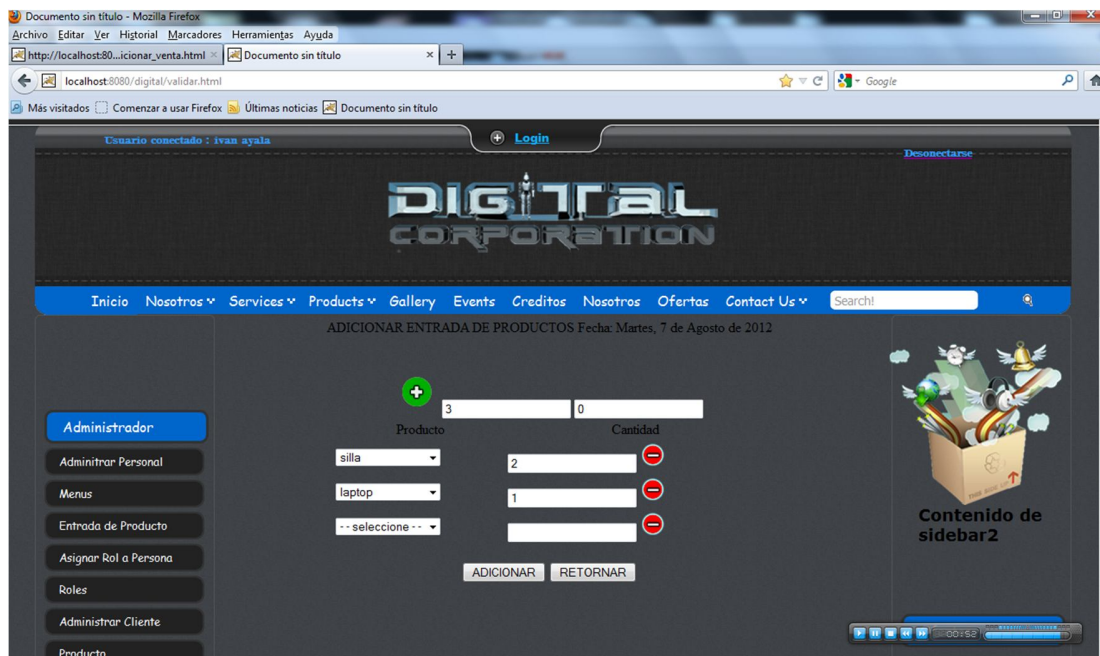


Figura N° 116: Pantalla Adicionar Entrada Producto

Registró De Entrada De Productos

Documento sin título - Mozilla Firefox
 Archivo Editar Ver Historial Marcadores Herramientas Ayuda
 http://localhost:8080/...iccionar_venta.html x +
 localhost:8080/digital/validar.html
 Más visitados Comenzar a usar Firefox Últimas noticias Documento sin título

CORPORATION

Inicio Nosotros Services Products Gallery Events Credits Nosotros Ofertas Contact Us Search!

REGISTRO DE ENTRADA DE PRODUCTOS
 Fecha: 2012-08-07 Codigoo: 108
 Responsable: ayala catari ivan

DETALLE:

Nro	Producto
1.-	lapiz : cantidad = 1 8 1 color 115 precio_venta 114 precio 113 3
2.-	Estabilizador : cantidad = 1 9 1 precio de venta 123 voltaje 120 precio de compra 122 marca 121 4 2

Administrador
 Roles
 administrar ventaproducto
 Menus
 Producto
 Entrada de Producto
 Asignar Rol a Persona
 Administrar Personal
 Tipos de Productos
 Asignar Menus a roles

Contenido de sidebar2
 Ofertas

Aceptar

Figura N° 117: Pantalla Registro de Entrada de Productos

Pantalla Gestión Venta

Documento sin título - Mozilla Firefox
 Archivo Editar Ver Historial Marcadores Herramientas Ayuda
 http://localhost:8080/...iccionar_venta.html x +
 localhost:8080/digital/validar.html
 Más visitados Comenzar a usar Firefox Últimas noticias Documento sin título

DIGITAL CORPORATION

Inicio Nosotros Services Products Gallery Events Credits Nosotros Ofertas Contact Us Search!

GESTIÓN DE PRODUCTOS
 Fecha inicio: Fecha fin: Activos Bajas Filtrar
 Adicionar

NRO	CODIGO Venta	FECHA	RESPONSABLE	B	VER
-----	--------------	-------	-------------	---	-----

Administrador
 Roles
 administrar ventaproducto
 Menus
 Producto
 Entrada de Producto
 Asignar Rol a Persona

Contenido de sidebar2

Figura N° 118: Pantalla Gestión Venta

Registro De Venta

Figura N° 119: Pantalla Registro de Venta

Registro De Venta su Plan de Pagos

NRO	FECHA DE PAGO	MONTO	ESTADO
1	7/7/2012	9.16	
2	7/8/2012	9.16	
3	7/9/2012	9.16	
4	7/10/2012	9.16	
5	7/11/2012	9.16	

Figura N° 120: Pantalla Registro de Venta su Plan de Pagos

II.1.16 PRUEBA DE CAJA BLANCA

Function Buscar (patron,tamaño,porcentaje)

□ Esta función calcula la diferencia entre dos palabras(texto , patrón) y verifica si cumple con el porcentaje introducido

```
{
  n = lenght (patron );
  m = length (texto );
  Por = round (porcentaje )/100);
  If (por > m) {
    If (m > n) {
      Filas = m; columnas = n;
      for (i=0; i<=filas ; i++) {
        For (j=0; j<=columnas ; j++) {
          if (vectorTexto [i]== vectorPatron [j])
            Matriz [i][j]=1
          Else
            Matriz [i][j]=0
        }
      }
    }
    else {
      filas = n; columnas = m;
      for (i=0; i<=filas ; i++) {
        For (j=0; j<=columnas ; j++) {
          if (vectorPatron [i]== vectorTexto [j])
            Matriz [i][j]=1
          Else
            Matriz [i][j]=0
        }
      }
      for(i=0; i<=filas ; i++) {
        for(j=1; j<=columnas ; j++) {
          If Matriz [i][j]= 1
            fmat ++;
            col = 1;
            x = i; y = j;
            while (((x<=filas )&&(y<=columnas ))) {
              if (Matriz [x][y]== 1){
                matPos [fmat ][col ]=x; //matriz de posiciones
                col ++;
                x ++; y ++; a = x;
              }
            }
            else {
              if (x == filas ){
                y ++; x =a;
              }
              else
                x ++;
            }
          }
          vector [f]= col -1; //vector de columnas
        }
      }
      aux =0;
      for (i=0; i<=fmat ; i++) {
        for(k=0; k<=filas ; j++) {
          if (matPos [i][aux ]<> k)
            e ++;
          aux ++;
        }
      }
      if (e<= porcentaje )
        return true ;
      else
        return false ;
    }
  }
  else
    returnr false ;
}
```

- **Grafo de Flujo**

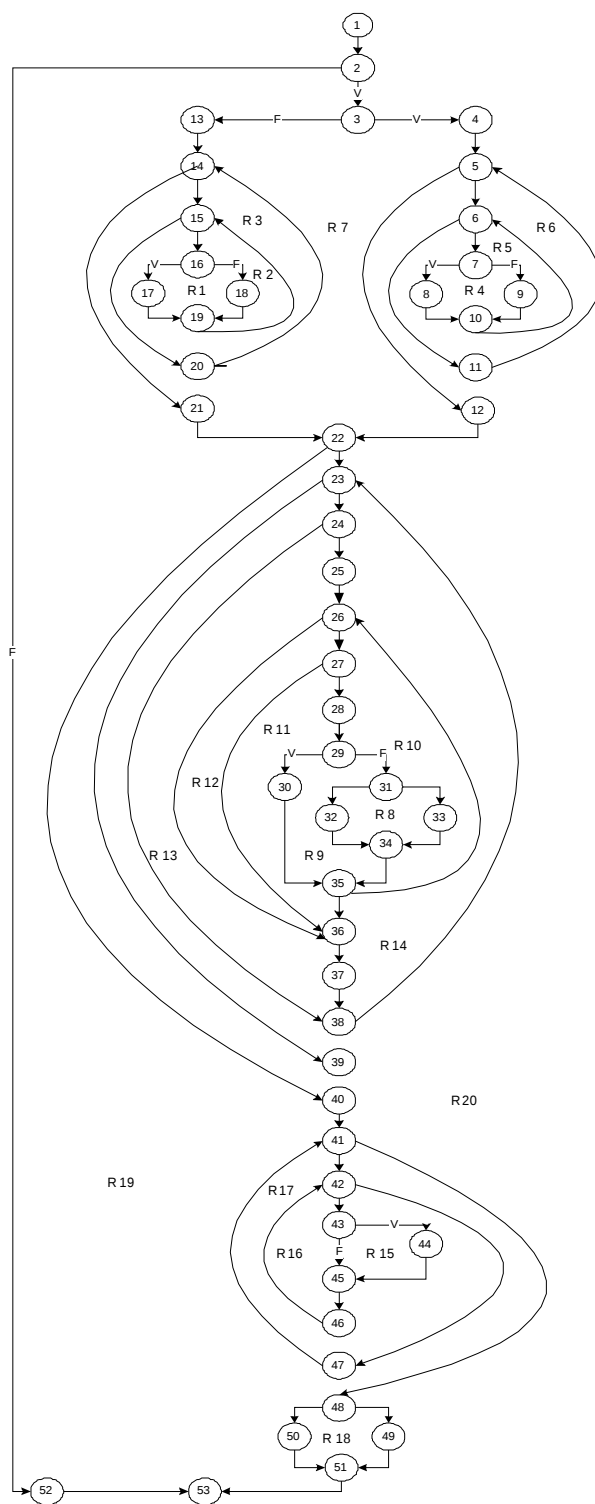


Figura. 121. Grafo de flujo

- **Complejidad Ciclomática:**

$$V(G) = 20 \text{ regiones}$$

$$V(G) = 71 \text{ aristas} - 53 \text{ nodos} + 2 = 20$$

$$V(g) = 19 \text{ nodos prediado} + 1 = 20$$

- **Caminos Linealmente Independientes**

Camino 1: 1-2-3-4-5-12-22-40-41-48-50-51-53

Camino 2: 1-2-3-4-5-6-11-5-12-22-40-41-42-43-51-53

Camino 3: 1-2-3-4-5-6-7-8-10-6-11-5-12-22-40-41-42-43-51-53

Camino 4: 1-2-3-4-5-6-7-9-10-6-11-5-12-22-40-41-48-50-51-53

Camino 5: 1-2-3-13-14-21-22-40-41-48-50-51-53

Camino 6: 1-2-3-13-14-15-20-14-21-22-40-41-48-50-51-53

Camino 7: 1-2-3-13-14-15-16-17-19-15-20-14-21-22-40-41-48-50-51-53

Camino 8: 1-2-3-13-14-15-16-18-19-15-20-14-21-22-40-41-48-50-51-53

Camino 9: 1-2-3-13-14-21-22-23-38-22-40-41-42-43-51-53

Camino 10: 1-2-3-13-14-21-22-23-24-38-23-38-22-39-40-47-49-50-53

Camino 11: 1-2 -3-13-14-21-22-23-24-25-26-36-37-38-23-39-22-40-41-48-50-51-53

Camino 12 : 1-2 -3-13-14-21-22-23-24-25-26-27-36-37-38-23-39-22-40-41-42-50-51-53

Camino 13 : 1-2 -3-13-14-21-22-23-24-25-26-27-28-29-31-32-34-35-26-36-37-38-23-39-22-40-41-42-50-51-53

Camino 14 : 1-2 -3-13-14-21-22-23-24-25-26-27-28-29-35-26-27-28-29-31-32-34-35-26-36-37-38-23-39-22-40-41-42-50-51-53

Camino 15: 1-2 -3-13-14-21-22-23-24-25-26-27-28-29-31-33-34-35-26-36-37-38-23-39-22-40-41-42-50-51-53

Camino 16: 1-2-3-13-14-21-22-23-24-25-26-27-28-30-32-33-34-25-35-37-23-38-22-39-40-41-42-47-41-48-49-50-51-53

Camino 17: 1-2-3-13-14-21-22-23-24-25-26-27-28-30-32-33-34-25-35-37-23-38-22-39-40-41-42-43-45-46-42-47-41-48-49-50-51-53

Camino 18: 1-2-3-13-14-21-22-23-24-25-26-27-28-30-32-33-34-25-35-37-23-38-22-39-40-41-42-43-44-45-46-42-47-41-48-49-50-51-53

Camino 19: 1-2-3-13-14-21-22-23-24-25-26-27-28-30-32-33-34-25-35-37-23-38-22-39-40-41-42-43-44-45-46-42-47-41-48-49-51-53

Camino 20: 1-2-52-53

- **Casos de Prueba**

Caso de Prueba para el Camino 1:

Valores de Entrada:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $por \geq m$ Cuando el valor de m (tamaño texto) sea mayor o igual a n (tamaño patrón)

Asigna a filas = m y columnas = n

Valor (i) = donde $filas < i < 0$ (nodo 5)

Valor (i) = donde $filas < i < 0$ (nodo 22)

Valor (i) = donde $fmatriz < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > porcentaje$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 2,3,4, 10,11,13,15,16,17,18

Caso de Prueba para el Camino 2:

Valores de Entrada:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = mayor al tamaño del texto m Cuando el valor de m (tamaño texto) sea mayor o igual a n (tamaño patrón)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq filas$ (nodo 5)

Valor (j)= donde $columnas < j < 0$ (nodo 6)

Valor (i) = donde $i > \text{filas}$ (nodo 5)

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre dos palabras sea mayor al porcentaje. Devuelve False

Nota : no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 3,4, 10,11,13,15,16,17,18

Caso de Prueba para el Camino 3:

Valores de Entrada:

Valor (por) = mayor al tamaño del texto m

Cuando el valor de m (tamaño texto) sea mayor o igual a n (tamaño patrón)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 5)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 6)

Donde el texto[i] sea igual a patrón [j]

Valor (j)= donde $j > \text{columnas}$

Valor (i) = donde $i > \text{filas}$

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Valor (e) = mayor al porcentaje calculado

Resultados Esperados: la diferencia entre dos palabras sea mayor al porcentaje. Devuelve False

Nota : no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 4, 10,11,13,15,16,17,18

Caso de Prueba para el Camino 4:

Valor (por) = mayor al tamaño del texto m

Cuando el valor de m (tamaño texto) sea mayor o igual a n (tamaño patrón)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 5)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 6)

Donde el texto[i] sea diferente al patrón [j]

Valor (j) = donde $j > \text{columnas}$

Valor (i) = donde $i > \text{filas}$

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Valor (e) = mayor al porcentaje calculado

Resultados Esperados: la diferencia entre dos palabras sea mayor al porcentaje. Devuelve False

Nota : no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 10,11,13,15,16,17,18

Caso de Prueba para el Camino 5:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $\text{filas} < i < 0$ (nodo 14)

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 6,7,8, 10,11,13,15,16,17,18

Caso de Prueba para el Camino 6:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $por \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $\text{columnas} < j < 0$ (nodo 15)

Valor (i) = donde $i > \text{filas}$ (14)

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 7,8, 10,11,13,15,16,17,18

Caso de Prueba para el Camino 7:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $por \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando Patrón[i] sea igual al Texto[j]

Valor (j) = donde $j > \text{columnas}$ (nodo 15)

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 10,11,13,15,16,17,18

Caso de Prueba para el Camino 8:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a $\text{filas} = m$ y $\text{columnas} = n$

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando $\text{Patrón}[i]$ sea diferente al $\text{Texto}[j]$

Valor (j) = donde $j > \text{columnas}$ (nodo 15)

Valor (i) = donde $i > \text{filas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $\text{columnas} < j < 0$ (nodo 23)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 10,11,13,15,16,17,18

Caso de Prueba para el Camino 9:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (j)= donde $j > \text{columnas}$ (nodo 15)

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $\text{columnas} < j < 0$ (nodo 23)

Valor (i) = donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 10,11,13,15,16,17,18

Caso de Prueba para el Camino 10:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (j) = donde $j > \text{columnas}$ (nodo 15)

Valor (i) = donde $i > \text{filas}$ (nodo 16)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Cuando matriz[i][j] sea diferente de 1

Valor (j) = donde $j > \text{columnas}$ (nodo 23)

Valor (i) = donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 11, 13, 15, 16, 17,18

Caso de Prueba para el Camino 11:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (j)= donde $j > \text{columnas}$ (nodo 15)

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Cuando $\text{matriz}[i][j]$ sea igual a 1

Valor (a) = donde $a > \text{filas}$ nodo (26)

Valor (j) = donde $\text{columnas} < j < 0$ (nodo 23)

Valor (i) = donde $\text{filas} < i < 0$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 13, 15, 16, 17, 18

Caso de Prueba para el Camino 12:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a $\text{filas} = m$ y $\text{columnas} = n$

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando $\text{Patrón}[i]$ sea diferente al $\text{Texto}[j]$

Valor (j) = donde $j > \text{columnas}$ (nodo 15)

Valor (i) = donde $i > \text{filas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Cuando $\text{matriz}[i][j]$ sea igual a 1

Valor (x) = donde $x \leq \text{filas}$ (nodo 26)

Valor (y) = donde $y > \text{columnas}$ (nodo 27)

Valor (x)= donde $x >$ filas (nodo 26)

Valor (j)= donde $j >$ columnas (nodo 23)

Valor (i)=donde $i >$ filas (nodo 22)

Valor (i) = donde $fmatriz < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e >$ porcentaje

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 13, 15, 16, 17,18

Caso de Prueba para el Camino 13:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $por \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq$ filas (nodo 14)

Valor (j)= donde $0 < j \leq$ columnas (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (i)= donde $i >$ filas (nodo 14)

Valor (j)= donde $j >$ columnas (nodo 14)

Valor (i) = donde $0 < i \leq$ filas (nodo 22)

Valor (j) = donde $0 < j \leq$ columnas (nodo 23)

Valor (x)= donde $x \leq$ filas (nodo 26)

Valor (y)= donde $y \leq$ columnas (nodo 27)

Cuando $matriz[i][j]$ sea diferente 1 (nodo 29)

Cuando x es igual de filas

Valor (y)= donde $y >$ columnas (nodo 27)

Valor (j)=donde $j > \text{columnas}$ (nodo 23)

Valor (i)= donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 15, 16, 17,18

Caso de Prueba para el Camino 14:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a $\text{filas} = m$ y $\text{columnas} = n$

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando $\text{Patrón}[i]$ sea diferente al $\text{Texto}[j]$

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (j)= donde $j > \text{columnas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Valor (x)= donde $x \leq \text{filas}$ (nodo 26)

Valor (y)= donde $y \leq \text{columnas}$ (nodo 27)

Cuando $\text{matriz}[i][j]$ es 1 (nodo 29)

Valor (y)= donde $y > \text{columnas}$ (nodo 27)

Valor (j)=donde $j > \text{columnas}$ (nodo 23)

Valor (i)= donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $\text{fmatriz} < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 15, 16, 17,18

Caso de Prueba para el Camino 15:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (j)= donde $j > \text{columnas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Valor (x)= donde $x \leq \text{filas}$ (nodo 26)

Valor (y)= donde $y \leq \text{columnas}$ (nodo 27)

Cuando $\text{matriz}[i][j]$ sea diferente 1 (nodo 29)

Cuando x es diferente de filas

Valor (y)= donde $y > \text{columnas}$ (nodo 27)

Valor (j)=donde $j > \text{columnas}$ (nodo 23)

Valor (i)= donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $fmatriz < i < 0$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 16,17,18.

Caso de Prueba para el Camino 16:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a filas = m y columnas = n

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (j)= donde $j > \text{columnas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Valor (x)= donde $x \leq \text{filas}$ (nodo 26)

Valor (y)= donde $y \leq \text{columnas}$ (nodo 27)

Cuando $matriz[i][j]$ sea diferente 1 (nodo 29)

Cuando x es diferente de filas

Valor (y)= donde $y > \text{columnas}$ (nodo 27)

Valor (j)=donde $j > \text{columnas}$ (nodo 23)

Valor (i)= donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $0 < i \leq fmatriz$ (nodo 41)

Valor (k)= donde $filas < k < 0$ (nodo 42)

Valor (i)= donde $i > fmatriz$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > porcentaje$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 17,18

Caso de Prueba para el Camino 17:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $por \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a $filas = m$ y $columnas = n$

Valor (i) = donde $0 < i \leq filas$ (nodo 14)

Valor (j)= donde $0 < j \leq columnas$ (nodo 15)

Cuando Patrón[i] sea diferente al Texto[j]

Valor (i)= donde $i > filas$ (nodo 14)

Valor (j)= donde $j > columnas$ (nodo 14)

Valor (i) = donde $0 < i \leq filas$ (nodo 22)

Valor (j) = donde $0 < j \leq columnas$ (nodo 23)

Valor (x)= donde $x \leq filas$ (nodo 26)

Valor (y)= donde $y \leq columnas$ (nodo 27)

Cuando $matriz[i][j]$ sea diferente 1 (nodo 29)

Cuando x es diferente de filas

Valor (y)= donde $y > columnas$ (nodo 27)

Valor (j)=donde $j > columnas$ (nodo 23)

Valor (i)= donde $i > filas$ (nodo 22)

Valor (i) = donde $0 < i \leq \text{fmatriz}$ (nodo 41)

Valor (k)= donde $0 < k \leq \text{filas}$ (nodo 42)

Cuando $\text{matos}[i][\text{aux}]$ sea igual a k

Valor (k)= donde $k > \text{filas}$ (nodo 42)

Valor (i)= donde $i > \text{fmatriz}$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Nota: no se puede probar por si sola debe ser aprobada como parte de la prueba de los caminos 18

Caso de Prueba para el Camino 18:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a $\text{filas} = m$ y $\text{columnas} = n$

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando $\text{Patrón}[i]$ sea diferente al $\text{Texto}[j]$

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (j)= donde $j > \text{columnas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Valor (x)= donde $x \leq \text{filas}$ (nodo 26)

Valor (y)= donde $y \leq \text{columnas}$ (nodo 27)

Cuando $\text{matriz}[i][j]$ sea diferente 1 (nodo 29)

Cuando x es diferente de filas

Valor (y)= donde $y > \text{columnas}$ (nodo 27)

Valor (j)=donde $j > \text{columnas}$ (nodo 23)

Valor (i)= donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $0 < i \leq \text{fmatriz}$ (nodo 41)

Valor (k)= donde $0 < k \leq \text{filas}$ (nodo 42)

Cuando $\text{matPos}[i][\text{aux}]$ sea diferente a k

Valor (k)= donde $k > \text{filas}$ (nodo 42)

Valor (i)= donde $i > \text{fmatriz}$ (nodo 41)

Cuando el número de errores es mayor al porcentaje calculado

Valor (e) = donde $e > \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea mayor al porcentaje. Devuelve False

Caso de Prueba para el Camino 19:

El valor del porcentaje calculado debe ser mayor al tamaño del texto Valor (por) = donde $\text{por} \geq m$ Cuando el valor de n (tamaño patrón) sea mayor o igual a m (tamaño texto)

Asigna a $\text{filas} = m$ y $\text{columnas} = n$

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 14)

Valor (j)= donde $0 < j \leq \text{columnas}$ (nodo 15)

Cuando $\text{Patrón}[i]$ sea diferente al $\text{Texto}[j]$

Valor (i)= donde $i > \text{filas}$ (nodo 14)

Valor (j)= donde $j > \text{columnas}$ (nodo 14)

Valor (i) = donde $0 < i \leq \text{filas}$ (nodo 22)

Valor (j) = donde $0 < j \leq \text{columnas}$ (nodo 23)

Valor (x)= donde $x \leq \text{filas}$ (nodo 26)

Valor (y)= donde $y \leq \text{columnas}$ (nodo 27)

Cuando $\text{matriz}[i][j]$ sea diferente 1 (nodo 29)

Cuando x es diferente de filas

Valor (y)= donde $y > \text{columnas}$ (nodo 27)

Valor (j)=donde $j > \text{columnas}$ (nodo 23)

Valor (i)= donde $i > \text{filas}$ (nodo 22)

Valor (i) = donde $0 < i \leq \text{fmatriz}$ (nodo 41)

Valor (k)= donde $0 < k \leq \text{filas}$ (nodo 42)

Cuando $\text{matPos}[i][\text{aux}]$ sea diferente a k

Valor (k)= donde $k > \text{filas}$ (nodo 42)

Valor (i)= donde $i > \text{fmatriz}$ (nodo 41)

Cuando el número de errores es menor al porcentaje calculado

Valor (e) = donde $e < \text{porcentaje}$

Resultados Esperados:

La diferencia entre el texto y el patrón sea menor al porcentaje. Devuelve True

Caso de Prueba para el Camino 20:

El valor del porcentaje calculado debe ser menor al tamaño del texto

Valor (por) = donde $\text{por} < m$

Resultados Esperados:

El tamaño del texto es menor al número del porcentaje calculado entonces devuelve

False.

8.2. PRUEBAS DE EJECUCIÓN

Se presenta un análisis de pruebas de ejecución del buscador respecto al tiempo de respuesta.

Los valores de entrada para la ejecución son:

- Tamaño del Patrón, medido en el número de caracteres
- Tamaño del Texto, medido en Kbytes
- Número de errores permitidos

Valores de salida:

- Número de resultados, cadenas emparejadas dado un error
- Tiempo de ejecución, medido en segundos y milisegundos

El comportamiento de los resultados obtenidos se muestra en la siguiente tabla:

Tamaño Patrón	Tamaño del Texto	Errores	Resultados	Tiempo de Ejecución
Mayor	Mayor-Mediano	Mayor	Menor	Mediano
Mayor	Mayor-Mediano-Menor	Menor-Mediano	Menor	Menor
Mayor	Menor	Mayor-Mediano-Menor	Menor	Menor
Mayor	Mayor-Mediano	Mediano	Menor	Menor

Mediano	Mayor	Mayor	Mayor	Mayor
Mediano	Mayor	Mediano	Mayor	Mediano
Mediano	Mayor	Menor	Menor	Mediano
Mediano	Mediano	Mayor	Mediano	Mediano
Mediano	Mediano	Mediano	Mayor	Mayor
Mediano	Mediano	Menor	Menor	Mediano
Menor	Mayor	Mayor	Mayor	Mayor

Dónde:

Variables	Mayor	Mediano	Menor
Tamaño patrón (caracteres)	[20,30]	[10,19]	[1,9]
Tamaño de Texto(Kb)	[3072,5000]	[1086,3071]	[0,1085]
Errores (caracteres)	$(n/2)+1$	[1,n/2]	[0]
Resultados	[16,..]	[6,15]	[0,5]
Tiempo de ejecución	[2.01,..]	[1.01,2.00]	[0.00,1.00]

II.1.17 Medios de Verificación del Componente 1: Sistema web de gestión de procesos administrativos desarrollado.

- ✓ **Informe otorgado por la Gerencia de DIGITAL CORPORATION sobre la Culminación de la Actividad denominada “Análisis y Diseño del Sistema Web”**



Tarija lunes 1 de octubre del 2012

Señores:
Docentes de taller III

Ref. Proyecto

De mi mayor consideración:

El motivo de la presente, es hacerle conocer que como gerencia de la empresa DIGITAL CORPORATION . Doy aprobada la etapa de Análisis y Diseño del Proyecto titulado

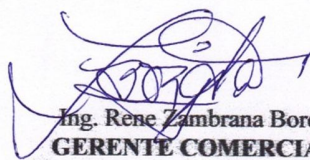
**"MEJORAMIENTO DE LA GESTION ADMINISTRATIVA DE LA EMPRESA
"DIGITAL CORPORATION", CON PLATAFORMA WEB"**

Elaborado por el universitario Iván Ayala Catari. Que viene trabajando desde abril en esta tarea y a la cual se le otorga toda la información necesaria.

Habiendo controlado la evolución de cada una de las pantallas diseñadas por el tenista, doy aprobada esta etapa del proyecto. Sin dejar de lado futuras modificaciones que mejore aún más la eficiencia del sistema al igual que mi trabajo.

Sin otra particular saludo a ustedes atentamente.

Atentamente


Ing. Rene Zambrana Borda
**GERENTE COMERCIAL
DIGITAL CORPORATION**

DIGITAL CORPORATION
Dirección: Calle General Trigo 294 / Alejandro del Carpio 383
Tel: 04-6644361 Tel/Fax: 04-6112303 E-mail: digitpc@hotmail.es elmolinetja.es

II.2.Componente 2**II. 2. COMPONENTE 2****TALLERES DE CAPACITACIÓN EN EL USO
DEL SISTEMA WEB, IMPLEMENTADO.**

Índice

II.2.1.	<i>Introducción</i>	116
II.2.2.	<i>Planteamiento del Problema</i>	116
II.2.3.	<i>Objetivo</i>	116
II.2.4.	<i>Alcances y Limitaciones</i>	117
II.2.5.	<i>Especificaciones de la Capacitación</i>	117
II.2.5.1.	<i>Marco Teórico</i>	117
II.2.5.1.1.	<i>Tendencia Pedagógica</i>	117
II.2.5.1.2.	<i>Temario</i>	118
II.2.5.1.3.	<i>Medios de Enseñanza</i>	118
II.2.5.1.3.	<i>Metodología</i>	118
II.2.5.1.3.	<i>Capacitacion</i>	118
II.2.6.	<i>Conclusiones</i>	¡Error! Marcador no definido.
II.2.7.	<i>Recomendaciones</i>	¡Error! Marcador no definido.
II.2.8.	<i>Medios de Verificacion</i>	¡Error! Marcador no definido.

II.2.1. Introducción

Para la implementación de un Sistema dentro de la Empresa DIGITAL CORPORATION es imprescindible capacitar a los involucrados con el manejo del Sistema. Lo cual lograra que se pueda sacar los máximos beneficios al Sistema y se reducirá enormemente los riesgos, errores que pueda surgir al momento de interactuar con el sistema.

En el caso de la Empresa DIGITAL CORPORATION se realizó la capacitación del personal que está involucrado directamente con el sistema, estos son:

Administrador del Sistema

Clientes

Gerente Propietario

OTROS

II.2.2. Planteamiento del Problema

Al no contar La Empresa DIGITAL CORPORATION con un Sistema de Gestión y Administrativo (Web) adecuado se observó que el personal no cuenta con un conocimiento suficientes como para manejar el sistema.

II.2.3. Objetivo

El objetivo principal de la capacitación del personal de la institución es lograr que los usuarios que interactuaran a diario con el sistema cuenten con una preparación adecuada que les ayudara a manejar correctamente el sistema lo cual se traducirá directamente en obtener beneficios del sistema que es el principal objetivo en sí del presente proyecto.

II.2.4. Alcances y Limitaciones

Se capacito no solo al personal involucrado al sistema sino también a otros empleados que también trabajan dentro de la institución esto por tratarse de emergencias o cambios de personal dentro de la empresa.

II.2.5. Especificaciones de la Capacitación

Para la ejecución de la capacitación se elaboró un manual de usuario con el objetivo de que dicho manual sea usado en caso de surgir alguna duda sobre el manejo del sistema. Dicho manual cuenta con imágenes impresas de las pantallas del sistema y los posibles mensajes que podrían salir al usuario.

II.2.5.1. Marco Teórico

II.2.5.1.1. Tendencia Pedagógica

Constructivismo.- El constructivismo es una corriente de la didáctica que se basa en la teoría del conocimiento constructivista. Postula la necesidad de entregar al alumno herramientas que le permitan crear sus propios procedimientos para resolver una situación problemática, lo cual implica que sus ideas se modifiquen y siga aprendiendo. El constructivismo en el ámbito educativo propone un paradigma en donde el proceso de enseñanza-aprendizaje se percibe y se lleva a cabo como proceso dinámico, participativo e interactivo del sujeto, de modo que el conocimiento

sea una auténtica construcción operada por la persona que aprende (por el «sujeto cognoscente»).

Se considera al alumno como poseedor de conocimientos que le pertenecen, en base a los cuales habrá de construir nuevos saberes. No pone la base genética y hereditaria en una posición superior o por encima de los saberes. Es decir, a partir de los conocimientos previos de los educandos, el docente guía para que los estudiantes logren construir conocimientos nuevos y significativos, siendo ellos los actores principales de su propio aprendizaje. Un sistema educativo que adopta el constructivismo como línea psicopedagógica se orienta a llevar a cabo un cambio educativo en todos los niveles.

II.2.5.1.2. Temario

- 1.- Introducción a las TIC (tecnologías de la información y la comunicación TIC o bien NTIC para *Nuevas Tecnologías de la Información y de la Comunicación* o IT para «*Information Technology*»).**
- 2.- Capacitación sobre el uso del Sistema Web para la Empresa DIGITAL CORPORATION.**

II.2.5.1.3. Medios de Enseñanza

- **Presentación Electrónica (PDF)**
- **Guías de Laboratorio (Impresiones)**

II.2.6. Metodología

Metodología de Enseñanza Socializada

Con esta metodología se pretende realizar una enseñanza dirigida al gerente y personal de la tienda, con lo que se busca una integración social sin descuidar la individualización.

Para llevarlas a cabo se desarrolló 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 capacitado, éste 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:

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

Estas técnicas se las realiza para que los participantes puedan interactuar con el sistema web de una manera adecuada.

II.2.7. Capacitación

La capacitación fue realizada en la empresa DIGITAL CORPORATION por el universitario Ivan Ayala Catari. Para la capacitación se vía positivas, que permitieron realizar la capacitación al usuario de forma adecuada.

Fecha	Horas	Lugar	Modulo	Personal Capacitado
2-08-2012	1 hora y media	Empresa DIGITAL CORPORATIO N.	- Gestionar Producto. - Gestionar Venta.	Ing. Miguel Angel Martinez.
2-09-2012	1 hora y media	Empresa DIGITAL CORPORATIO N.	- Gestionar Producto. - Gestionar Venta.	Sr. Adriana de Martinez (Contador).
6-09-2012	2 horas	Empresa DIGITAL CORPORATIO N.	- Gestionar Cliente. - Gestionar Personal.	Ing.Rene Zambrana Borda
13-09-2012	Media hora	Empresa DIGITAL CORPORATIO N.	- Gestionar Cliente. - Reportes.	Ing.Rene Zambrana Borda

Tabla II.130 Cronograma de Capacitación

II.2.8. Conclusiones

- II. Durante la capacitación se hizo evidente el interés, la entrega y la atención del personal en recibir la capacitación. Poniendo de esta manera de relieve que la incorporación del nuevo sistema les facilitara el trabajo relacionado con la gestión de usuarios (registro de usuarios), gestión de los procesos administrativos.
- II. Otro rasgo de la que sobresalió durante la capacitación es que el uso de tecnologías de información y comunicación. Dentro de la institución estas tecnologías son de verdadera importancia ya que con ellas se puede aumentar la productividad y el desempeño del personal de la institución.

II.2.9. Recomendaciones

Las experiencias vividas durante la capacitación me enseñaron algunas lecciones que vale la pena tener en cuenta a la hora de realizar una capacitación al personal de la institución, las mismas son las siguientes:

Organizar los manuales de usuario de manera ordenada, clara, y sencilla en lo referente al funcionamiento del sistema.

Si los involucrados en el sistema son pocos es mejor realizar la capacitación en forma individual pues es más fácil conocer de cerca las necesidades de cada usuario.

Mostrar al usuario que el sistema reducirá altamente los porcentajes de error que se tenía en el sistema manual.

Inculcar en el usuario el uso de manual de usuario en caso de surgir alguna duda.

II.2.10. Medios de Verificación de componente 2: “Capacitación del personal involucrado”

- ✓ **Fotografías tomadas el día de la Presentación del Sistema en el Acto que realizó la Empresa DIGITAL CORPORATION.**
- ✓ **Certificados de la Capacitación.**







124





UAMIS
www.uamis.edu.co

Universidad Autónoma "Juan Misael Saracho"

Facultad de Ciencias y Tecnología

Departamento de Ing. Informática

SISTEMA WEB PARA LA EMPRESA DIGITAL CORPORATION

CERTIFICADO DE CAPACITACION

A: **Ing. René Zambrana Borda**

Confiere el presente:

Por cuanto ha cumplido con los requisitos de asistencia y participación en el proceso de enseñanza en el transcurso de la capacitación del sistema llevado a cabo en octubre de 2012.



Departamento de Informática
Mag. Lic. S. Efraín Torrijos I.
Directora DPTI, INFORMATICA
Y TECNOLOGIA
FAC. de Ciencias y Tecnología



Univ. Juan Ayala Catari
Director del Proyecto

CAPÍTULO III

CONCLUSIONES Y RECOMENDACIONES

III.1. Conclusiones

Luego de haber concluido el presente proyecto denominado “Mejora de la Gestion de los Procesos Administrativos Via Web de DIGITAL CORPORATION. se tiene como fin el incorporar tecnologías de información y comunicación TICs en la misma se llegó a las siguientes conclusiones.

La metodología que se uso para interpretar el ciclo de vida del sistema, nos ha permitido efectuar un sistema que cubra las necesidades de la la empresa DIGITAL CORPORATION. Comenzando por los casos de uso que nos marcaron las necesidades que el sistema debería cumplir para cada uno de los actores pasando por el diagrama de secuencia que nos ayudó a implementar en el sistema los métodos y mensajes que se implementarían en el sistema.

Pasando al modelo de datos se hizo el modelo de entidad-relación por ser un modelo más amigable que el modelo semántico y sobre todo porque se hizo uso de herramientas que modelan estos datos y nos permiten migrar directamente a la base de datos postgres facilitándonos el trabajo y ahorrándonos tiempo.

Dentro de la construcción del sistema se utilizó el lenguaje de programación java el cual tiene una cantidad considerable de IDEs para su manipulación. En nuestro caso hicimos uso de Eclipse Europa porque facilita la conexión con la base de datos y presenta una interfaz cómoda para la programación.

Una de las características más importantes de un sistema es la capacidad de procesamiento del sistema y la velocidad de la misma, es por ello que tratamos con nuestro mayor esfuerzo hacer que este sistema sea lo más rápido posible algo difícil de lograr en un sistema de este tipo por la complejidad que presenta sus reportes contables, sin embargo se logró llegar a una velocidad de procesamiento razonable.

Como conclusión se puede decir que el proyecto ha sido un instrumento que nos ha ayudado a probar los conocimientos logrados en estos años pues cada componente ya sea de diseño o cualquier otro ha significado replantearnos una vez más los conocimientos adquiridos.

III.2. Recomendaciones

Antes de desarrollar un sistema como el presente es necesario previamente fortalecer los conocimientos en estos campos pues la construcción debe sujetarse a normas que establece la institución y a normas que están establecidas por organismos estatales.

Así también como la construcción debe ser cuidadosa en los cálculos por tratarse de un sistema que maneja dinero, la manipulación del sistema tiene que ser hecha con suma cautela por tratarse de dinero que debe responderse ante una autoridad establecida por la institución.

Se recomienda establecer mecanismos y métodos eficaces con enfoque activo hacia seguridad para ser implementados al sistema.

En caso de que existan cambios del personal de la institución involucrados con el manejo del sistema, se recomienda asignar un presupuesto para la capacitación del nuevo personal para que cuenten con la preparación adecuada en el uso del sistema.

Para el desarrollo del sistema de ventas se recomienda manejar con mucho cuidado y responsabilidad los datos, ya que se manipulan datos de personas, costos y precios.

Se recomienda contar con un Web Hosting, para el alojamiento del sitio web de la empresa, en una plataforma de conectividad e infraestructura de hardware y software necesario, de alta disponibilidad rendimiento. Que cuente con un Data Center seguro, confiable y atendido por especialistas, sin tener que invertir en servidores ni incurrir en gastos operacionales.