

I. CAPÍTULO I: EL PROYECTO

1.1 PRESENTACIÓN DEL PROYECTO

1.1.1 TÍTULO DEL PROYECTO

El título del presente proyecto es: **Mejorar la Administración de la Información de Ventas del “Comercial Figueroa”**

1.1.2 CARRERA

Ingeniería Informática

1.1.3 FACULTAD

Ciencias y Tecnologías

1.1.4 DURACIÓN DEL PROYECTO

8 meses

1.1.5 ÁREA/LÍNEA DE INVESTIGACIÓN PRIORIZADA

Desarrollo de Sistemas de Información

1.2 PERSONAL VINCULADO AL PROYECTO

1.2.1 DIRECTOR DEL PROYECTO

Gallardo Apellido Paterno	Aparicio Apellido Materno	Griselda Verónica Nombre	5047061 C.I.
Estudiante Profesión	Ing. Informática Carrera o Unidad	Ciencias y Tecnología Facultad:	
Telef. Oficina	78239544 Celular	gry537d4@gmail.com Correo electrónico	Firma

1.2.2 PARTICIPANTES DEL PROYECTO:

Categoría	Nombres y Apellidos	Profesión	C.I.	Firma
Director	Griselda Gallardo Aparicio	Estudiante	5047061	
Asesor	Lic.	Profesional		

1.2.3 EQUIPO DE TRABAJO: EMPRESA/INSTITUCIÓN

Nombre: Universidad Autónoma Juan Misael Saracho			
Dirección: Calle España / Barrio El Tejar		Teléfono Oficina:	
Nombre y Apellidos	Cargo	C.I.	Firma
Sr. Nelson Figueroa A.	Gerente General del “COMERCIAL FIGUEROA”		

1.2.4 ACTIVIDADES PREVISTAS PARA LOS INTEGRANTES DEL EQUIPO DE INVESTIGACIÓN

Responsable	Actividades

Directora Griselda Gallardo A.	El Director del proyecto es el encargado de proveer todos los recursos necesarios para el desarrollo del mismo, además de elaborar el plan de desarrollo del proyecto en el que se incluye desde el análisis, la elaboración, la difusión del sistema a los usuarios, hasta la implementación de éste. Además deberá ser el coordinador y realizar los seguimientos que se requieran para poder garantizar la fiabilidad, seguridad y por sobre todo la conformidad de los usuarios con los resultados finales, un plan de contingencia también será parte de su trabajo, de esta manera se podrá garantizar la total organización de los involucrados en el proyecto a la hora de realizar sus tareas. Sistema Web: • Determinación de Requerimientos • Desarrollo del Análisis. • Desarrollo del Diseño. • Desarrollo de la Programación. • Elaboración de un texto guía. • Realización de pruebas del Software. • Implementación del Software. Capacitación sobre el manejo del sistema a los miembros del COMERCIAL FIGUEROA. • Elaboración del material de capacitación. • Preparación del ambiente para la capacitación. • Realizar la capacitación.

1.2 DESCRIPCIÓN DEL PROYECTO

1.2.5 RESUMEN EJECUTIVO DEL PROYECTO

El presente trabajo de grado, consiste en desarrollar un sistema web que coadyuve a la gestión administrativa del **COMERCIAL FIGUEROA**.

Para lograr el diseño de este sistema web se identificó las necesidades del **COMERCIAL FIGUEROA** (comercio dedicado a la venta al contado y al crédito de

muebles y electrodomésticos), mediante la aplicación de Ingeniería de requerimientos (Norma 830), se estableció que la metodología de desarrollo será RUP y por último, el modelado se llevara a cabo a través del lenguaje UML.

Dicho comercial presenta problemas en el proceso de obtención de datos, registro de clientes, empleados, ventas y productos, todo esto, conlleva a una mala toma de decisiones.

El personal realiza un registro de datos generales del cliente, ventas, productos y precios, de forma manual, lo que conduce a veces a un registro erróneo e incluso extravió de la información. El comercial no está habilitado para brindar una respuesta óptima, en cuanto a tiempo y exactitud de la información a sus clientes.

Este proyecto contendrá un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades del comercio y también al personal encargado del mismo.

Por otra parte apoyara, al proceso de toma de decisiones, control de mercadería y lograra ventajas competitivas, a través del uso de la tecnología de información.

Este proyecto, denominado **Mejorar a la administración de la información de ventas del “COMERCIAL FIGUEROA”**, está basado en tecnologías web, contendrá un sistema con los siguientes módulos: usuario, productos, compra, venta, proveedores, clientes, empleados y reportes.

Generará:

- Registro y búsqueda de datos de clientes.
- Registro y búsqueda de datos de ventas.
- Registro y búsqueda de datos de productos.
- Registro y búsqueda de datos de empleados.
- Registro y búsqueda de datos de proveedores.
- Reporte cantidad de compras de clientes.
- Reporte de ventas.
- Reporte producto almacenado.
- Reporte Empleado.
- Reporte ganancias.

Gerente/Administrador:

Recibirá capacitación sobre el correcto uso del sistema, para que pueda realizar operaciones de vital importancia, como el registro y búsqueda de datos de las personas que son parte fundamental del comercial (empleados y clientes), así como de los productos existentes para la venta, para ello se le proveerá, de textos guías, con ilustraciones e indicaciones a cerca del manejo del sistema.

Vendedores:

Siendo indispensables para la atención de los clientes y registrar las ventas, recibirán capacitación individual sobre algunas áreas del sistema como ser: registro de ventas, clientes.

Para lograr que los vendedores se familiaricen con el sistema, se le mostrara las ventajas del mismo, con ejemplos que, le demostraran el gran ahorro de tiempo que resulta el trabajar con un sistema computarizado. Se le dotara de manuales que le ayuden a conocer mejor cada pantalla, y un pequeño curso donde se la expondrá el manejo del sistema.

1.2.6 DESCRIPCIÓN, FUNDAMENTACIÓN Y JUSTIFICACIÓN DEL PROYECTO**¿POR QUÉ?**

En la actualidad en el “**COMERCIAL FIGUEROA**”, se maneja toda la información (clientes, productos, compras, empleados y ventas) de forma manual, usando documentos (formularios) impresos, por lo cual se tiene dificultades a la hora del registro y búsqueda de los mismos, ocasionando que no se tenga una información más acorde a las necesidades que se requiere (información rápida, segura y confiable).

En cuanto a los productos, no se puede establecer de manera confiable, a que almacén pertenecen.

Los reportes, son inexistentes.

El personal del comercial, desconoce las ventajas de aplicar las NTICS a las diferentes áreas empresariales, y el “**COMERCIAL FIGUEROA**”, no cuenta con un sistema computarizado, lo cual, no permite que ellos puedan interactuar con herramientas de estas características.

¿Para qué?

Este proyecto, tiene como fin optimizar el manejo y control de información del **“COMERCIAL FIGUEROA”**, y por ende, generar varios beneficios mediante la implementación de un sistema automatizado, que permitirá el registro de datos (clientes, empleados, productos, ventas y compras), y la búsqueda de los mismos, de manera rápida y confiable.

Se podrá también acceder a información acerca de los productos almacenados.

Este sistema permitirá generar reportes.

Además, se llevara a cabo, una adecuada capacitación para el manejo del sistema computarizado y se dotara de un texto guía, donde se ilustrara, como se deben llevar a cabo las varias funcionalidades del sistema.

¿Cómo?

Se trazó un plan realizable, para mejorar la administración del **“COMERCIAL FIGUEROA”**, para ello se estableció, un estudio de requerimientos (un modelado del negocio en la actualidad, entrevistas y encuestas, para una mejor visión de cómo se maneja la información), luego se llevó a cabo el análisis, diseño y desarrollo del sistema (diagramas, programación y pruebas del sistema), además de un plan de trabajo para las capacitaciones y la creación de los textos guías, para el correcto uso del sistema.

Todo este plan, está regido, bajo un calendario, el que se deberá cumplir para obtener un sistema, que pueda amortiguar las necesidades de todos los involucrados en el comercial.

Justificación Tecnológica:

Se cuenta en nuestro departamento con la tecnología suficiente para poder llevar a cabo este tipo de proyecto, sin duda, el uso y la disponibilidad de internet es imprescindible, además de que este sistema tendrá la capacidad de evolucionar si así es necesario en un futuro.

Justificación Social:

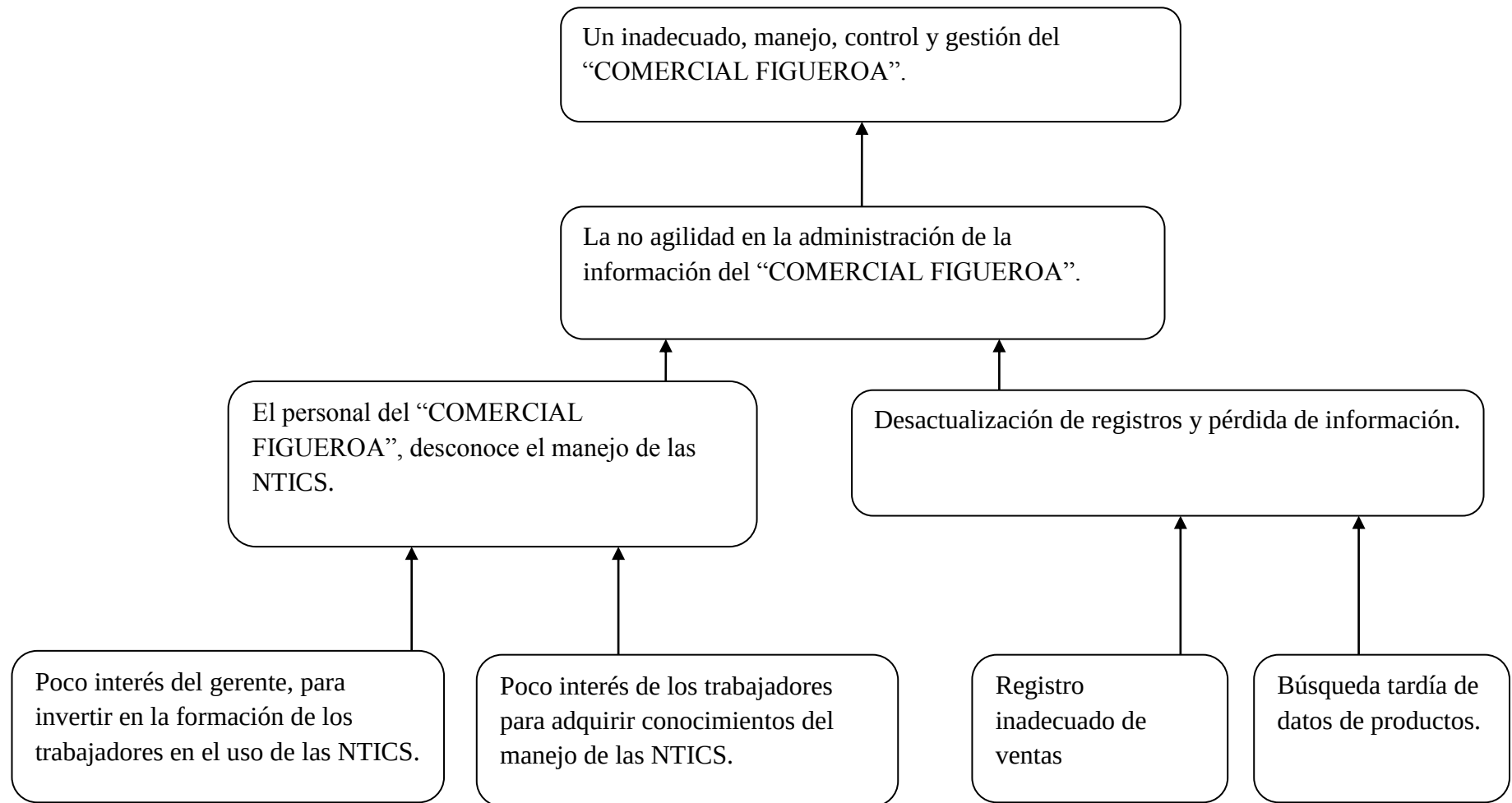
Es del conocimiento de la gran mayoría de los habitantes de nuestro departamento que es imprescindible hoy en día el tener conocimiento en cuanto al manejo de las TIC, por lo que los estudiantes de la Juan Misael Saracho Y los que específicamente cursamos la carrera de Ingeniería Informática, tenemos la obligación social de poder aportar con

nuestros conocimientos a la ciudadanía y que mejor empujando a que los habitantes de nuestro departamento, instituciones públicas, comerciantes, etc. se adhieran a la renovación tecnológica que sufrimos día a día.

Y no menos importante es la objetividad que debemos tener para poder abrirnos las puertas en la sociedad como futuros profesionales, este tipo de proyectos nos permitirá plasmar todos los conocimientos adquiridos en los años de universitario, por lo tanto será nuestra carta de presentación en el futuro.

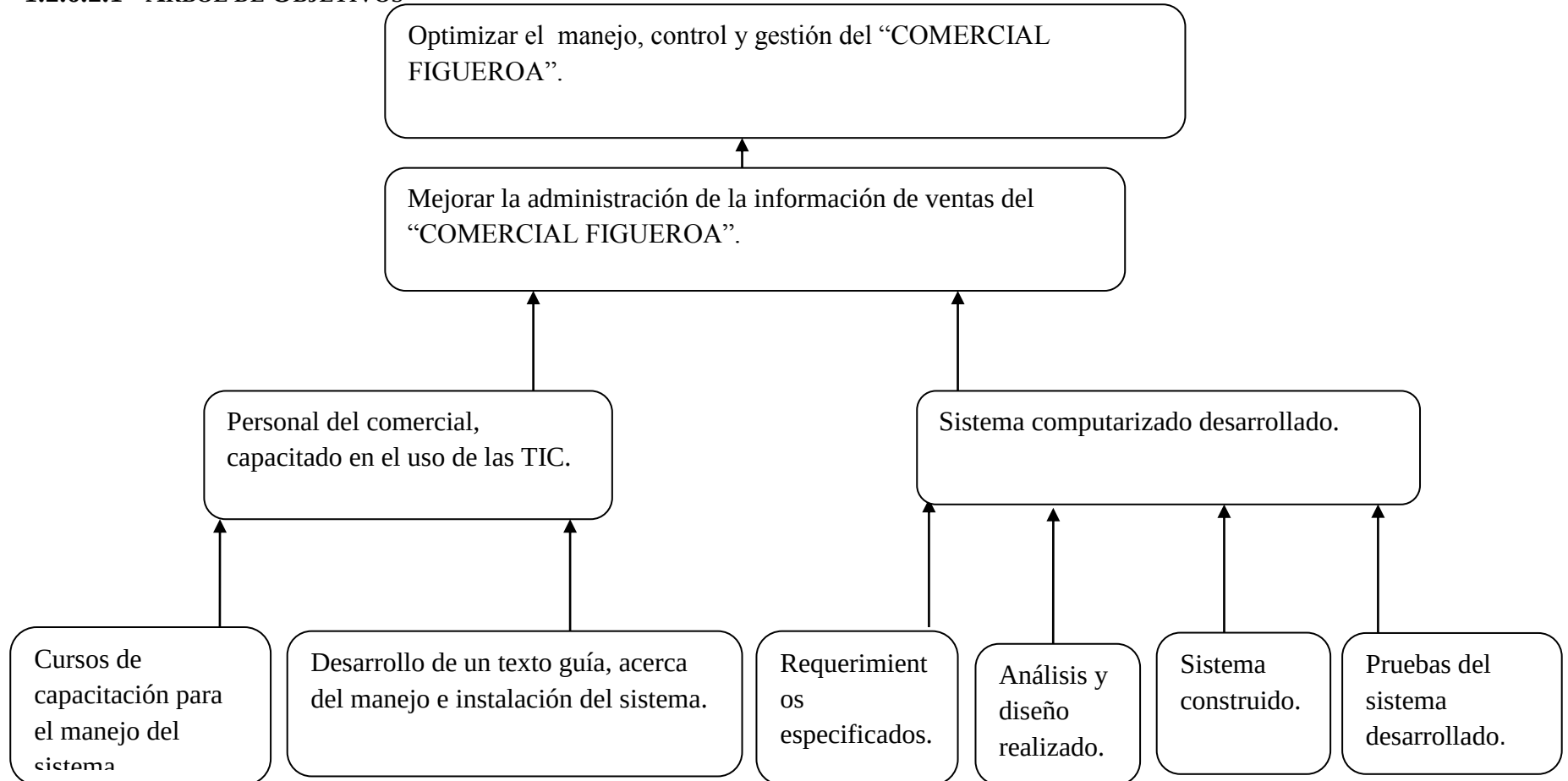
1.2.6.1 ANÁLISIS DE CAUSAS DEL PROBLEMAS

1.2.6.1.1 ÁRBOL DE PROBLEMAS



1.2.6.2 ANÁLISIS DE OBJETIVOS

1.2.6.2.1 ÁRBOL DE OBJETIVOS



1.2.6.3 SITUACIÓN PLANTEADA CON Y SIN PROYECTO

Situación sin proyecto	Situación con proyecto
* Incapacidad del gerente de poder brindar un informe de todos los productos con los que cuenta el comercial. * Desconocimiento y desmotivación por parte del personal del comercial para el manejo de las TIC. * Imposibilidad de contar con reportes de acuerdo a fechas, de determinadas ventas.	* Capacidad del gerente de poder brindar un informe de todos los productos con los que cuenta el comercial. * Conocimiento y motivación por parte del personal de la institución para el manejo de las TIC. * Posibilidad de contar con reportes de acuerdo a fechas, de determinadas ventas.

1.2.7 OBJETIVOS

1.2.7.1 OBJETIVO GENERAL

Mejorar la administración de la información de ventas del “**COMERCIAL FIGUEROA**”.

1.2.7.2 OBJETIVOS ESPECÍFICOS

- Sistema desarrollado para mejorar la administración de la información de ventas del “**COMERCIAL FIGUEROA**”.
- Capacitar al personal sobre la importancia de las NTICS y en el manejo del sistema.

1.2.8 MARCO LÓGICO DEL PROYECTO

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Optimizar la obtención, manejo, control y administración de la gestión en el “COMERCIAL FIGUEROA”.		La elaboración de un informe, a futuro, de parte del gerente general Sr. Nelson Figueroa A. expresando de conformidad con el proyecto.	Los docentes responsables de la materia, aprueban el proyecto. Se mantiene el costo presupuestario por el comercio “COMERCIAL FIGUEROA” para invertir en opciones que mejoren el proceso administrativo.
Propósito Mejorar la administración de la información de ventas del “COMERCIAL FIGUEROA”, mediante la aplicación de las NTICS.	Al finalizar la ejecución del proyecto a finales del 2014, El “COMERCIAL FIGUEROA” contara con un área de atención a sus clientes automatizada, lo cual permitirá mejorar su eficiencia en cuanto el registro de sus pagos en un 70%.	Reportes actualizados de los informes que emiten las áreas de ventas, compras y productos del “COMERCIAL FIGUEROA”. Informe del comercial, dando conformidad y	Existe voluntad del Gerente General Sr. Nelson Figueroa A., para apoyar el fortalecimiento de Información de Ventas del “COMERCIAL FIGUEROA” Empleados con

	(Eficiencia) A= 10 clientes encuestados. B= 7 dicen que el registro de sus pagos es más rápida, a partir del uso del sistema. $C=B*100/A=70\%$.	avalando la eficiencia del sistema en el seguimiento de las actividades del comercio.	disponibilidad e interés, para manejar el sistema.
Objetivos Específicos (Componentes) 1. Sistema Web desarrollado para mejorar la administración de la información, de Ventas del “COMERCIAL FIGUEROA”.	1. Al finalizar el proyecto, en diciembre del 2014, se ha desarrollado un Sistema informático de acuerdo a estándares (IEEE 830) que automatice la información del “COMERCIAL FIGUEROA”, por lo cual, la búsqueda de todos los cardex de los clientes se realiza en un tiempo mínimo.	1.1 Documento de análisis y diseño del sistema aprobado por el docente de taller III Gustavo Succi A. 1.2 Se cuenta con un informe del gerente del “COMERCIAL FIGUEROA”, expresando su conformidad.	1.3 Se cuenta con toda la disposición del Gerente General, y el personal del “COMERCIAL FIGUEROA”, en brindarnos toda la información necesaria para el desarrollo del sistema.

	(Calidad) Sin el uso del sistema: a fines del año 2014, un empleado tardaba en buscar, datos existentes de un cliente, aproximadamente 4 minutos. Con el uso del sistema: a principios del año 2015, la búsqueda de los datos de un cliente, se llevara a cabo 0.50 segundos aproximadamente.		
2. Capacitación sobre la importancia de las NTICS y capacitación sobre el uso del sistema, con el apoyo de textos guías para el usuario, acerca del	2. Al finalizar el proyecto, en diciembre del 2014, se ha desarrollado al menos 1 taller de capacitación a usuarios del sistema del “COMERCIAL	2.1 Registro de nombres de los participantes del taller de capacitación. 2.2 Un informe del gerente general Sr. Nelson	2.3 Los miembros involucrados en el proyecto reconocen la importancia de la capacitación.

funcionamiento del sistema.	FIGUEROA”, mejorando así, la calidad del servicio a los clientes en un 80%, en comparación con el año anterior. (Calidad) A=5 usuarios en total. B= 4 usuarios que están de acuerdo con que la capacitación cumplió su objetivo. $C = B * 100 / A = 80\%$	Figueroa, expresando su conformidad.	2.4 Se cuenta con el ambiente, equipo y material necesario para la capacitación de los usuarios. 2.5 Se elaboraran textos guías del uso del sistema, para mejorar las capacitaciones.

Actividades				
1.1. Especificación de requerimientos	Bs. 400.00		Informe detallado presentado mensualmente, por el Gerente, Sr. Nelson Figueroa A.	Desembolso mensualmente de parte del encargado de las finanzas del “COMERCIAL FIGUEROA”.
1.2. Elaboración de perfil (Marco Lógico)	Bs 100.00			
1.3. Análisis y Diseño del Sistema.	Bs. 5000.00			
1.4. Programación.	Bs. 6400.00			
1.5. Pruebas de software.	Bs. 400.00			
1.6. Elaboración de un texto guía de usuario.	Bs 200.00			
	TOTAL	Bs.		
	12500.00			
2.1. Diseñar metodología de capacitación para el personal de ventas y administrativo.	Bs. 200.00		Informe detallado de todos los gastos realizados en la capacitación por el gerente del “COMERCIAL	Desembolso de los recursos económicos, para la preparación y
2.2. Ejecución de la Capacitación.	Bs. 300.00			

	<u>TOTAL</u> <u>Bs. 500.00</u> TOTAL Bs. 13000.00	FIGUEROA”.	ejecución de la capacitación, de parte del encargado de las finanzas del “COMERCIAL FIGUEROA”.

Tabla 1: Matriz de Marco Lógico

1.2.9 METODOLOGÍA DEL TRABAJO

Las metodologías mencionadas a continuación serán utilizadas para el desarrollo del Sistema Web y sus otros componentes.

Se hará uso de las fases de cada metodología de acuerdo a los requerimientos del Sistema Web a desarrollar.

Sistema Web:

- * **Metodología RUP**
- * Etapa de Inicio
- * Etapa de Elaboración.
- * Etapa de Construcción.
- * Etapa de Transición.

Diagramas UML:

I. Análisis

- * Diagrama de Casos de Uso
- * Diagrama de actividades
- * Diagrama de Clases.
- * Diagrama de Colaboración
- * Diagrama de Componentes

Taller de capacitación para el personal de la institución:

- * Definir fechas y horarios de capacitación accesibles.
- * Elaboración del material a usar.
- * Plasmar en un texto guía, de los pasos a seguir para el acceso y el buen uso del sistema
- * Llevar a cabo la capacitación, sobre el sistema y la importancia de las NTICS.
- * Elaboración del informe final.

1.2.10 DESCRIPCIÓN Y RELACIÓN DE LAS ESTRATEGIAS CON LOS OBJETIVOS

Estrategias	Objetivos Específicos
* Llevar a cabo el diseño conceptual y arquitectónico del sistema. * Realizar la construcción del sistema. * Llevar a cabo pruebas del sistema. * Implementar el sistema.	Realizar un estudio completo de los requerimientos del usuario ya verificado.
* Definir fechas y horarios de capacitación accesibles. * Elaboración del material a usar. * Elaboración de informe final. * Plasmar en textos guías los pasos a seguir para el buen uso del sistema * Elaboración del informe final.	Llevar a cabo un taller de capacitación para el uso adecuado del sistema y sobre la importancia de las NTICS en las empresas comerciales.

1.2.11 CRONOGRAMA DE ACTIVIDADES

SICOMFIG	DURACION	INICIO	FIN
SISTEMA WEB IMPLEMENTADO	150 días	02/05/14	01/08/14
-Etapa de Inicio	12 días	03/05/14	18/05/14
-Etapa de Elaboración	24 días	19/05/14	12/06/14
-Etapa de Construcción	120 días	13/06/14	13/09/14
CURSOS DE CAPACITACION	30 días	13/09/14	13/11/14
-Elaborar temas para la capacitación	10 días	14/09/14	24/09/14
-Llevar a cabo la capacitación	15 días	25/09/14	10/11/14
-Elaboración informe final	5 días	11/11/14	15/11/14
ELABORACION DE MANUALES	12 días	15/11/14	27/11/14
-Elaboración del manual de Usuario	5 días	16/11/14	21/11/14
-Elaboración del manual de instalación	5 días	21/11/14	25/11/14
-Elaborar Informe final	2 días	25/11/14	27/11/14

Figura 1: Cronograma de actividades

1.2.12 RESULTADOS ESPERADOS

1.2.12.1 RESULTADOS DIRECTOS

- **Sistema Informático:** Para tener un mejor control y administración de la información del proceso ventas, de fácil instalación, manejo sencillo e interfaz amigable. Además contará con la seguridad adecuada de protección y control de la información.
 - **Alcances:** Automatización de información del personal, Compras, Ventas (datos, productos, búsqueda). Además el sistema permitirá realizar reportes (Compras, Ventas, Almacenes, Productos) y el registro de Plan de pagos para una venta al crédito, facturación con código de control QR (VER ANEXO 3).
 - **Limitaciones:** activos fijos, cotización de productos, cuentas bancarias, inventarios y contabilidad.
- **Capacitación en el uso del sistema.**
 - La capacitación de usuarios sobre la importancia de las TIC y uso del sistema, es parte importante para el éxito y buen funcionamiento del sistema, para lograr esto al concluir el sistema se realizará al menos un taller de capacitación al personal que hará uso del sistema.

1.2.13 TRANSFERENCIA DE RESULTADOS

Se realizará un convenio entre la U.A.J.M.S. y el “COMERCIAL FIGUEROA”.

a) Medios y estrategias para la transferencia de resultados.

- ✓ Proporcionar un CD de instalación del sistema a los usuarios.
- ✓ Proporcionar un texto guía para el usuario y de instalación del sistema.

b) Grupo de beneficiarios de los resultados

- ✓ El gerente general del “COMERCIAL FIGUEROA”
- ✓ Todos los clientes del comercial.
- ✓ Empleados del comercial.

II. CAPITULO: COMPONENTES

2.1 COMPONENTES 1: SISTEMA WEB DESARROLLADO PARA MEJORAR LA ADMINISTRACIÓN DE LA INFORMACIÓN DEL “COMERCIAL FIGUEROA”

2.1.1 MARCO TEÓRICO

2.1.1.1 ANTECEDENTES

Desde el inicio de la humanidad, y a lo largo de toda su historia, el hombre ha ido evolucionando con la ayuda de herramientas, que le sean útiles en el transcurso de su vida diaria.

Dada la complejidad del mundo actual, con el manejo inmenso de conocimientos e información propia de esta época de crecimiento tecnológico es indispensable contar con herramientas que permitan manejar información con eficiencia, eficacia e integridad, esas herramientas son los sistemas informáticos basados en el computador con sistemas capaces de ofrecernos la información en forma rápida, ordenada, concreta y segura, etc.

La información es parte fundamental de toda empresa u organización, para tener un alto nivel de competitividad y posibilidades de desarrollo.

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

A modo de resumen de este antecedente de marco teórico, puede decirse que los sistemas de gestión de información, en su definición más general, se refieren al conjunto de todos los componentes necesarios que se interrelacionan, con el objetivo de tramitar y facilitar la información sobre el tema de interés para su consumo en cualquier medio, momento y lugar.

Del “COMERCIAL FIGUEROA” podemos decir que es una empresa familiar, que lleva 15 años en nuestra ciudad de manera activa. Como toda empresa empezó desde

muy poco, pudiendo con una medianamente lograda gestión, lograr contar hoy en día con 3 sucursales en nuestro departamento, 20 empleados entre los cuales existen fijos y remplazos, el gran aporte que este comercial hace a la sociedad es muy importante, ya que permite que las personas que no cuentan con la solvencia suficiente para adquirir cualquier tipo de electrodoméstico, que es necesario en su hogar, éste se los facilita, haciendo que el pago de dicho artículo, pueda hacerse efectivo, en un determinado transcurso de tiempo y el pago por el mismo, de acuerdo a la posibilidad de aporte de dinero, que los ingresos de los clientes se lo permitan. De ésta manera, es que el comercial no solo cuenta con electrodomésticos, también tiene como productos para el cliente otra variedad como ser mueblería.

Debido al crecimiento del comercial presenta problemas en el proceso del registro de los productos, por lo consecuente, una mala toma de decisiones.

El personal realiza un registro de datos generales del cliente, ventas, productos, precios y compras, de forma manual lo que conduce a veces a un registro erróneo e incluso el extravió de la información.

2.1.1.2 NOTACIÓN DEL MODELADO

2.1.1.2.1 UML

2.1.1.2.1.1 DEFINICIÓN

UML es un lenguaje de modelado visual que permite especificar, visualizar, construir y documentar los componentes que forman parte de un sistema de software orientado a proceso.

2.1.1.2.1.2 CARACTERÍSTICAS

- Es una especificación basada en Booch, Roumbaught y Coad-Yourdon.
- Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
- Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder la

información, para que los usuarios y desarrolladores comprendan las características de la aplicación.

- Se quiere convertir en un lenguaje estándar, con el que sea posible, modelar todos los componentes de una aplicación, sin embargo no pretende definir un modelo de desarrollo sino únicamente un lenguaje de modelado.
- **UML** ofrece los siguientes diagramas para modelar sistemas.
- **Diagramas de Casos de Uso** Descripción de los procesos entre el usuario y el sistema.
- **Diagramas de Secuencia** para modelar el paso de mensajes entre proceso
- **Diagramas de Colaboración** para modelar interacciones entre proceso
- **Diagramas de Estado** para modelar el comportamiento de los casos de uso, proceso y operaciones.
- **Diagramas de Actividad** para modelar el comportamiento de los casos de uso, proceso u operaciones.
- **Diagramas de Clases** para modelar la estructura estática de las clases en el sistema.
- **Diagramas de Proceso** para modelar la estructura estática del proceso en el sistema.
- **Diagramas de Implementación** para modelar la distribución del sistema.

2.1.1.3 METODOLOGÍA DE DESARROLLO

2.1.1.3.1 METODOLOGÍA RUP (RATIONAL UNIFIED PROCESS)

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto, con 3 características esenciales, está dirigido por los **Casos de Uso**: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la **arquitectura**: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es

iterativo e incremental: dividiéndose el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Utiliza UML (Lenguaje unificado de modelado), para preparar todos los esquemas de un sistema de software.

2.1.1.3.1.1 CARACTERÍSTICAS ESPECIALES

- **Dirigido por casos de uso**

Los casos de Uso, representan los requisitos funcionales del sistema, actuando como guía para la arquitectura del sistema, (que a su vez influye en la selección de los casos de uso), y como integrador de los flujos de trabajo (requisitos, análisis y diseño, implementación y pruebas, ver figura).

Los casos de Uso, representan los requisitos funcionales del sistema, actuando como guía para la arquitectura del sistema, (que a su vez influye en la selección de los casos de uso), y como integrador de los flujos de trabajo (requisitos, análisis y diseño, implementación y pruebas, ver figura).

- **Centrado En La Arquitectura**

La arquitectura ayuda a los desarrolladores y clientes comprender el sistema, contribuyendo a organizar el desarrollo en subsistemas e interfaces bien definidas.

En las fases iniciales lo que se hace es ir consolidando la arquitectura y se va modificando dependiendo de las necesidades del proyecto.

La arquitectura de un producto Software se describe mediante vistas de diseño del sistema en construcción, incluyendo los aspectos estáticos y dinámicos más significativos del sistema.

El producto Software tiene tanto una función (casos de uso), como una forma (arquitectura), que evolucionan en paralelo, consolidando y/o modificando la arquitectura dependiendo de las necesidades del proyecto, hasta conseguir una arquitectura robusta en las fases finales del proyecto.

- **Iterativo e Incremental**

Para el desarrollo de un producto software de gran esfuerzo, es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración, un recorrido más o menos completo a lo largo de todos los flujos de trabajo

fundamentales, de la que se obtiene una versión interna, que resulta en un incremento al crecimiento del producto software.

- **Vida Del Proceso Unificado**

El proceso Unificado se repite a lo largo de una serie de ciclos. Cada ciclo produce una nueva versión del sistema, y cada versión es un producto preparado para su entrega.

Cada ciclo se divide en cuatro fases: y cada una de estas fases tiene que pasar por flujos de trabajo y éstos son cinco, y a esto se le denomina iteración (ver figura).

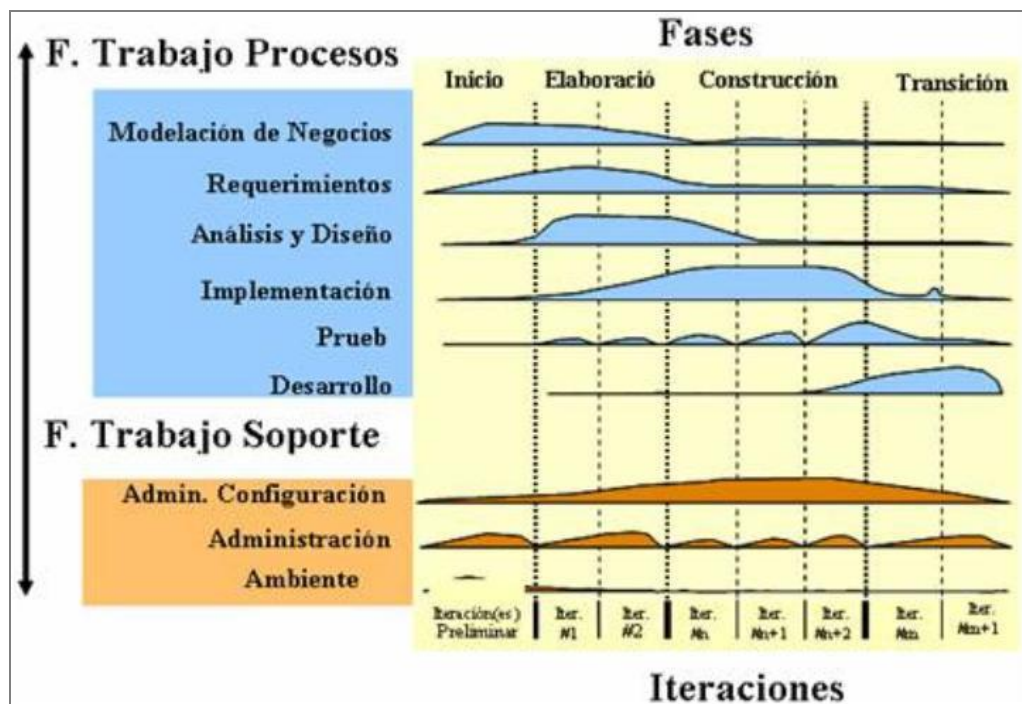


Figura 1: Vida del Proceso Unificado

2.1.1.3.1.2 FACES DE RUP

a. fase de inicio

Fase de Inicio (Inspección y Concepción) Se hace un plan de fases, donde se identifican los principales casos de uso y se identifican los riesgos. Se concreta la idea, la visión del producto, como se enmarca en el negocio, el alcance del proyecto.

b. Fase de elaboración

Se realiza el plan de proyecto, donde se completan los casos de uso y se mitigan los riesgos. Se planifica las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura. Esta fase ayuda a planificar con gran precisión la fase de construcción, concentrándose en el análisis y diseño del sistema hasta poder conseguir en sus últimas iteraciones algún prototipo que muestre el funcionamiento del sistema.

c. fase de construcción

Se basa en la elaboración de un producto totalmente operativo y en la elaboración del manual de usuario. Construir el producto, la arquitectura y los planes, hasta que el producto está listo para ser enviado a la comunidad de usuarios.

d. fase de transición

Se realiza la instalación del producto en el cliente y se procede al entrenamiento de los usuarios. Realizar la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente quede satisfecho, por tanto en esta fase suelen ocurrir cambios.

2.1.1.3.1.3 DIAGRAMAS UML UTILIZADOS

2.1.1.3.1.3.1 DIAGRAMAS DE CASO DE USO

Sirve para describir las interacciones del sistema con su entorno, identificando los actores, que representan las diferentes actividades desempeñadas por los usuarios del sistema y los casos de uso corresponden a la funcionalidad que el sistema ofrece a los usuarios, en su interacción externa.

2.1.1.3.1.3.2 DIAGRAMA DE CLASES

Describe los tipos de proceso que hay en el sistema y las diversas clases de relaciones estáticas que existen entre ellos. En el diagrama de clases definamos las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización.

Es el diagrama principal para el análisis y diseño, presentando las clases de sistema con sus relaciones estructurales y de herencia, incluyendo definición de atributos, especificando también operaciones, métodos, interfaces y dependencias.

2.1.1.3.1.3.3 DIAGRAMA DE ACTIVIDADES

Representación una serie de acciones dentro de uno o varios hilos de proceso condicionadas por unos nodos de control.

Define los flujos de trabajo de una organización, modela operaciones complejas, formaliza escenarios de un Caso de Uso, escenarios de un grupo relacionado con el CU, establece una concurrencia de procesos.

2.1.1.3.1.3.4 DIAGRAMA DE SECUENCIA

Un diagrama de secuencia muestra una interacción ordenada de proceso según la secuencia temporal de eventos o mensajes.

2.1.1.3.1.3.5 DIAGRAMA DE COLABORACIÓN

Un diagrama de colaboración muestra como las instancias específicas de las clases trabajan juntas para alcanzar un objetivo común. Describe el intercambio de mensaje entre proceso y las relaciones entre los mismos. Los diagramas de componentes pueden ser utilizados para modelar sistemas de software de cualquier tamaño y complejidad, nos permite especificar un componente como una unidad modular con interfaces bien definidas.

2.1.1.3.1.3.6 DIAGRAMA DE COMPONENTES

Un componente de software es una parte física de un sistema, como puede ser un módulo, una base de datos, un programa ejecutable, etc. Se puede considerar que un componente es la materialización de una o más clases.

2.1.1.3.1.3.7 DIAGRAMA DE PAQUETES

Muestra como el sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido

2.1.1.3.1.3.8 DIAGRAMA DE DESPLIEGUE

Muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue.

2.1.1.4 HERRAMIENTAS DE CONSTRUCCIÓN DEL SOFTWARE

2.1.1.4.1 VISUAL PARADIGM 6.8 FOR UML

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a proceso, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Herramienta CASE que da soporte al modelado visual con UML 2.1

2.1.1.4.2 MYSQL 8.0

Sistema de base de datos operacional MySQL es hoy en día uno de los más importantes en lo que hace al diseño y programación de base de datos de tipo relacional. Cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.

2.1.1.4.3 iREPORT-NB-3.5.2

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

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

2.1.1.4.4 APACHE TOMCAT 7.0

Tomcat es un contenedor de servlets que se utiliza en la Referencia oficial de la implementación para Java Servlet y JavaServer Pages (JSP). Las especificaciones Java Servlet y JavaServer Pages son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java Community Process). Apache Tomcat es desarrollado en un entorno abierto y participativo, bajo la licencia de Apache Software-License. Para simplificar, podríamos decir que Apache Tomcat (o Jakarta Tomcat) es un software desarrollado con Java (con lo cual puede funcionar en cualquier sistema operativo, con su máquina virtual java correspondiente) que sirve como servidor web con soporte de servlets y JSPs.

2.1.1.5 TÉCNICA

2.1.1.5.1 HTML

El HTML, acrónimo inglés de Hypertext Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML es una aplicación del estándar ISO 8879:1986(SGML), que se formalizó en 1990 con la aparición de la Word Wide Web.

2.1.1.5.2 CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

2.1.1.5.3 JQUERY

jQuery es una biblioteca o framework de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX en sistemas web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

jQuery usa las licencias MIT y GPL permitiendo su uso en proyectos libres y privados.

2.1.1.5.4 JAVA

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

Java es un lenguaje de proceso, independiente de la plataforma. Algunas características notables:

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

2.1.1.5.5 MODELO VISTA CONTROLADOR

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

Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

- **Definición de las partes**

El **Modelo** es el Proceso que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La **Vista** es el Proceso que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El **Controlador** es el Proceso que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

2.1.1.5.6 SPRING

Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia de OpenSource. Se basa en una configuración a base de javabeans bastante simple. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. Es interesante el uso de programación orientada a aspectos (IoC). Tiene plantillas que permiten un más fácil uso de Hibernate, iBatis, JDBC..., se integra "de fábrica" con Quartz, Velocity, Freemarker, Struts, Webwork2 y tienen un plugin para eclipse.

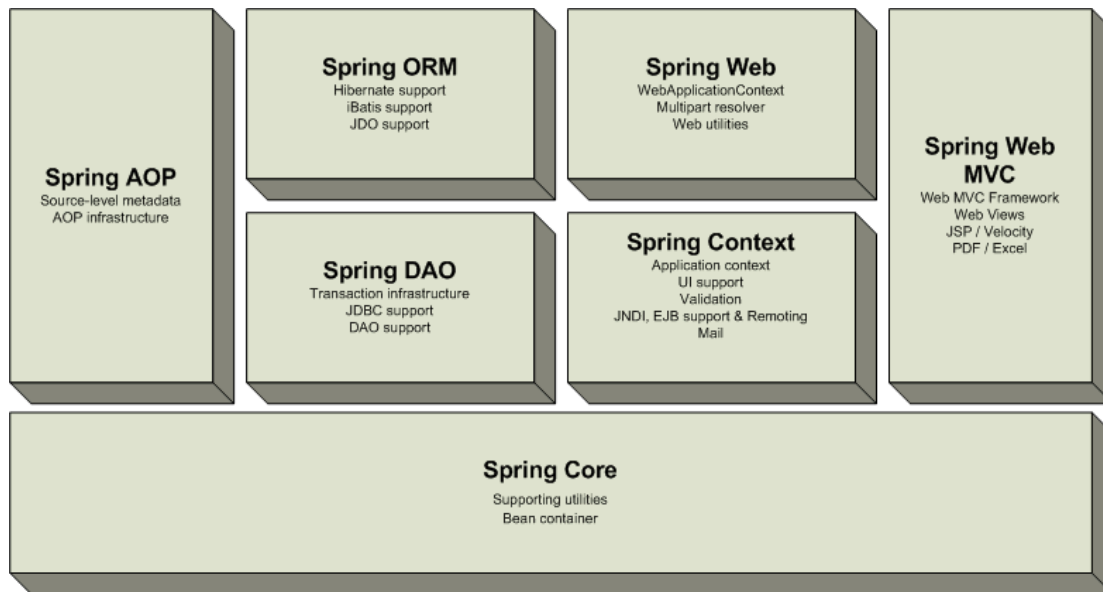


Figura 2: Arquitectura de Spring

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

2.1.1.5.7 HIBERNATE

Trabajar con software orientado a proceso y bases de datos relacionales puede hacernos invertir mucho tiempo en los entornos actuales. Para solucionar este problema nace Hibernate es una herramienta que realiza el mapping entre el mundo orientado a proceso de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a proceso y viceversa.

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

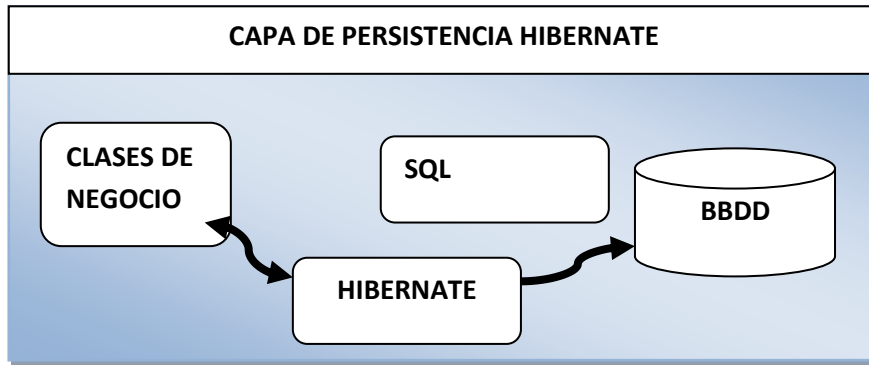


Figura 3: Hibernate

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

Características:

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

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

2.1.1.6 BASE DE DATOS

2.1.1.6.1 COMPONENTES PRINCIPALES DE UNA BASE DE DATOS

Los principales componentes de una base de datos son:

- **Datos.** Los datos son la Base de Datos propiamente dicha.
- **Hardware.** Se refiere a los dispositivos de almacenamiento en donde reside la base de datos.
- **Software.** Está constituido por un conjunto de programas que se conoce como Sistema de Gestor de Base de Datos (DBMS), manejando éste todas las solicitudes formuladas por los usuarios a la base de datos.
- **Usuarios.** 3 tipos: El programador de aplicaciones, El usuario Final y el Administrador de la Base de Datos quien se encarga del control general del Sistema de Base de Datos.

2.1.1.7 GLOSARIO

Gestión: Se refiere a la acción de administrar, ordenar, disponer, organizar, en especial la organización o unidad.

Dato: Información dispuesta de manera adecuada para su tratamiento por un computador.

Información: Se define como información a conocimiento públicamente registrable, datos almacenados. Dentro de una organización es un recurso fundamental para la toma de decisiones, como base para elaborar soluciones alternativas a un problema determinado, además es el componente integral y esencial de la educación superior en todos los campos del conocimiento.

Lenguaje de modelado: Es la notación (en su mayoría gráfica) que utilizan los métodos para expresar los diseños.

Método: Es un proceso disciplinado para generar un conjunto de modelos que describen varios aspectos de un sistema de software en desarrollo, utilizando alguna notación bien definida.

Metodología: Es una colección de métodos aplicados a lo largo del ciclo de vida del desarrollo de software y unificados por alguna aproximación general o filosófica.

Notación: Es un conjunto de diagramas normalizados que posibilita al analista o desarrollador el describir el comportamiento del sistema (análisis) y los detalles de una arquitectura (diseño) de forma no ambigua.

Meta modelo: Es la descripción de un modelo.

Modelo lógico: Define la arquitectura del sistema desde el punto de vista de las abstracciones principales y mecanismos.

Proceso: Son los pasos que se aconsejan dar para realizar un diseño.

Software: Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números y texto y también incluyen representación de información de audio, video e imágenes. La descripción de software en un libro de texto podría tomar la forma siguiente: el software es (1) instrucciones que cuando se ejecuten proporcionan la función y el rendimiento deseados, (2) estructuras de datos que permiten a los programas manipular adecuadamente la información, y (3) documentos que describen la operación y el uso de programas.

Sistema: Conjunto de instrucciones ejecutables, escritas con un lenguaje de programación que permite al usuario cumplir una función específica o realizar una tarea concreta. Una hoja de cálculo, una base de datos, un procesador de textos, etc., son programas.

2.1.2 PLAN DE DESARROLLO DE SOFTWARE

2.1.2.1 INTRODUCCIÓN

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

El proyecto ha sido propuesto por la Universitaria Griselda Gallardo Aparicio, basado en la Metodología RUP (Rational Unified Process).

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

2.1.2.1.1 PROPÓSITO

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

Los usuarios del Plan de Desarrollo del Software son:

- Solamente el jefe del proyecto, que lo utiliza para organizar la agenda y necesidades de recursos y para realizar su seguimiento.

2.1.2.1.2 ALCANCE

Esta herramienta proporciona una idea global para desarrollo del software, nos permite realizar nuestro plan de trabajo que coadyuve al cumplimiento de los objetivos fijados.

2.1.2.1.3 RESUMEN

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

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

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

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

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

2.1.2.2 VISTA GENERAL DEL PROYECTO

2.1.2.2.1 PROPÓSITO, ALCANCE Y OBJETIVOS

La información que a continuación se incluye ha sido extraída de las diferentes reuniones y entrevistas que se han realizado con el personal del “**COMERCIAL FIGUEROA**”.

2.1.2.2.1.1 PROPÓSITO

La presente propuesta de sistema tiene por Proceso hacer más eficiente las actividades dentro del Área Administrativa, a través de un: Registro de empleados, asignar un producto a un almacén, registro de ventas, la obtención de reportes cuando se lo requiera, además de otorgar a los usuarios un registro adecuado de la información: Teniendo toda información almacenada en una Base de Datos a la que solo podrá acceder personal autorizado, para consultar los datos personales de los clientes y proveedores, brindando de esta manera total seguridad a toda la información guardada. Las propuestas mencionadas con anterioridad serán ejecutadas en el “**COMERCIAL FIGUEROA**”.

2.1.2.2.1.2 OBJETIVOS

Objetivo General

Diseñar y desarrollar un sistema informático para mejorar la administración de la información de ventas del “**COMERCIAL FIGUEROA**”, mediante las TICS.

Objetivos Específicos

Según el estudio previamente realizado se planteo los siguientes objetivos:

- Obtener mayor seguridad y confidencialidad en el manejo de información de Ventas e información confidencial de los clientes.
- Llegar a brindar un sistema rápido y eficiente para la manipulación de registros.
- Conseguir mayor organización en el registro de los empleados.
- Obtener una mayor organización de los registros de productos en los almacenes.
- Obtener un sistema que permita obtener reportes de forma rápida y sencilla.
- Otorgar a los usuarios total seguridad en el resguardo de su información, permitiendo el acceso sólo a usuarios autorizados.
- Crear una base de datos de acuerdo a las necesidades de la empresa.

- Diseñar una interfaz amigable para usuario.

2.1.2.2.1.3 ALCANCE

Administrar

- Administrar Empleado (teléfonos, Dirección , cargos, horarios)
- Administrar Usuario (acceso).
- Administrar Clientes.
- Administrar Cargos.
- Administrar Venta.
- Administrar Compra.
- Administrar Proveedor.
- Administrar Producto.
- Administrar Almacén.

Reportes

- Reporte cantidad de compras de clientes.
- Reporte producto almacenado.
- Reporte ventas.
- Reporte recaudaciones.

2.1.2.2.2 SUPOSICIONES Y RESTRICCIONES

2.1.2.2.2.1 SUPOSICIONES

- Información confiable.
- Obtención de información rápida.
- Registro de Compras confiable.
- Registro de clientes nuevos confiable y fácil,
- Facilidad en búsqueda de clientes ya existentes.
- Reducción de errores en la realización de registros de Ventas al Crédito.
- Registro de Ventas confiable.

2.1.2.2.2 RESTRICCIONES

Establecemos las siguientes restricciones para el sistema:

- El sistema será restringido, sólo usuarios que tengan una cuenta habilitada podrán acceder al sistema.
- Para la manipulación de la base de datos, sólo podrán acceder el personal autorizado.
- Implementación del sistema solo en la plataforma Windows
- Deberá contar como mínimo con un sistema operativo Windows XP.
- Deberá contar como mínimo con un equipo Pentium IV.
- El administrador deberá contar con el conocimiento necesario para el manejo del sistema.
- No soporta módulo contable.

2.1.2.2.3 ENTREGABLES DEL PROYECTO

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto. Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son Proceso 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.

- **Plan de Desarrollo del Software**

Es el presente documento.

- **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.

- **Modelo de Proceso del Negocio**

El modelado de proceso del negocio es la base para comprender mejor la operación de una organización, permite realizar una visualización clara de los flujos de trabajo entre los actores de la organización.

- **Glosario**

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

- **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.

- **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.

- **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.

- **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.

- **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.

- **Modelo de Datos**

Previendo que la persistencia de la información del sistema será soportada por una base de datos entidad-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.).

- **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).

- **Modelo de Despliegue**

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

- **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.

- **Plan de Iteración**

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.

- **Evaluación de Iteraciones**

Este documento incluye la evaluación de los resultados de cada iteración, el grado en el cual se han conseguido los objetivos de la iteración, las lecciones aprendidas y los cambios a ser realizados.

- **Lista de Riesgos**

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

- **Manual de Instalación**

Este documento incluye las instrucciones para realizar la instalación adecuada y correcta del producto. Ver anexo (**Manual de Instalación y usuario**).

- **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. Ver anexo (**Manual de Instalación y usuario**).

2.1.2.2.3.1 PLAN DE DESARROLLO DEL SOFTWARE

El presente documento.

2.1.2.2.3.2 MODELO CASOS DE USO DEL NEGOCIO

Modelo Casos de Uso del Negocio: Cliente

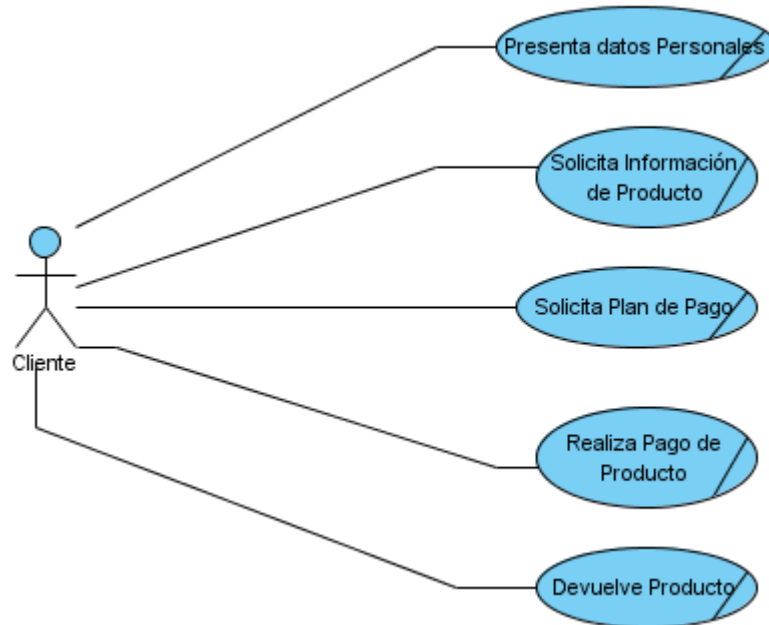


Figura 4: Modelo Casos de Uso del Negocio-Clientes

Modelo Casos de Uso del Negocio: Vendedor(a)

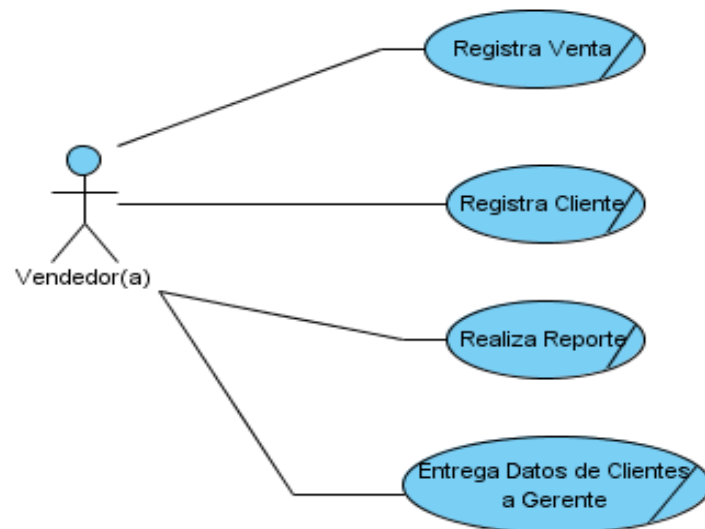


Figura 5: Modelo Casos de Uso del Negocio-Vendedor(a)

Modelo Casos de Uso del Negocio: Cajero(a)

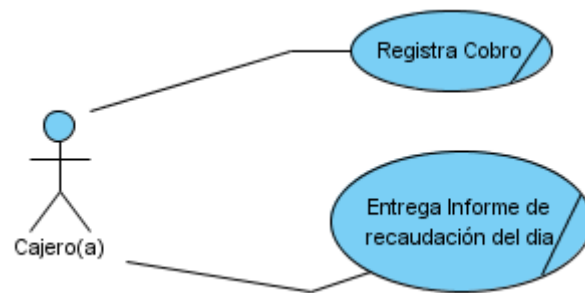


Figura 6: Modelo Casos de Uso del Negocio-Cajero(a)

Modelo Casos de Uso del Negocio: Gerente-General

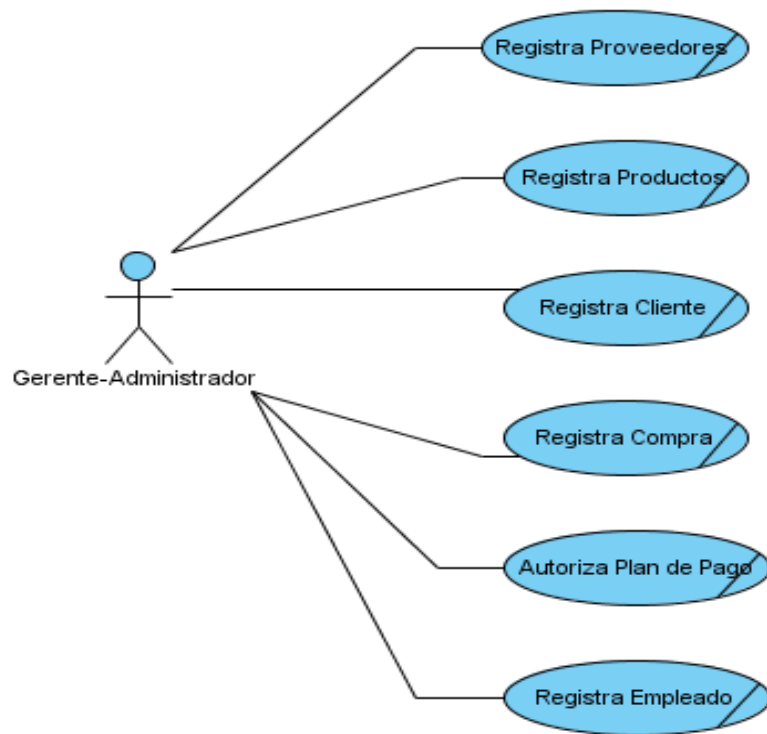


Figura 7: Modelo Casos de Uso del Negocio-Gerente Administrador

2.1.2.2.3.3 MODELO DE PROCESOS DEL NEGOCIO

Modelo de Proceso del Negocio: Consultar Información de Producto

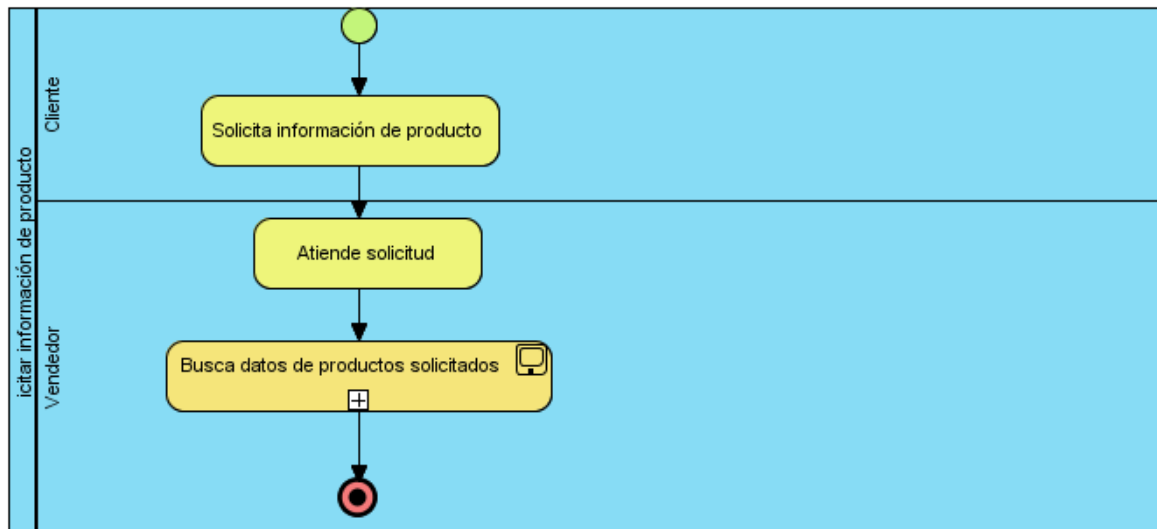


Figura 8: Modelo de Proceso del Negocio-Consultar Producto

Modelo de Proceso del Negocio: Solicitar Registro

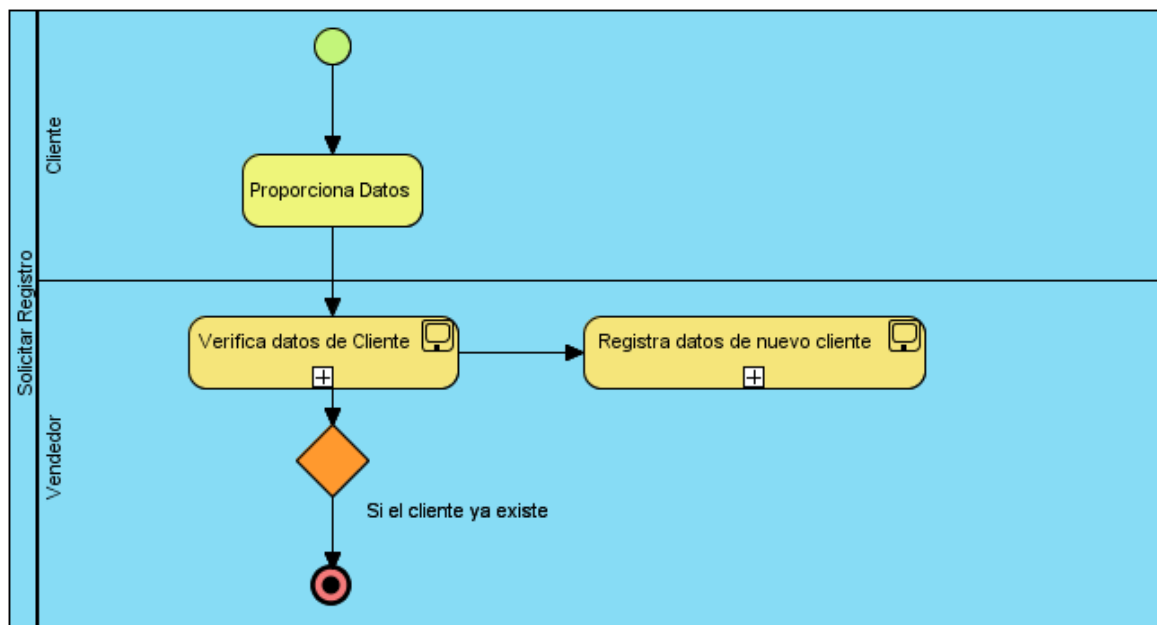


Figura 9: Modelo de Proceso del Negocio-Solicitar Registro

Modelo de Proceso del Negocio: Gerente Solicita Reporte

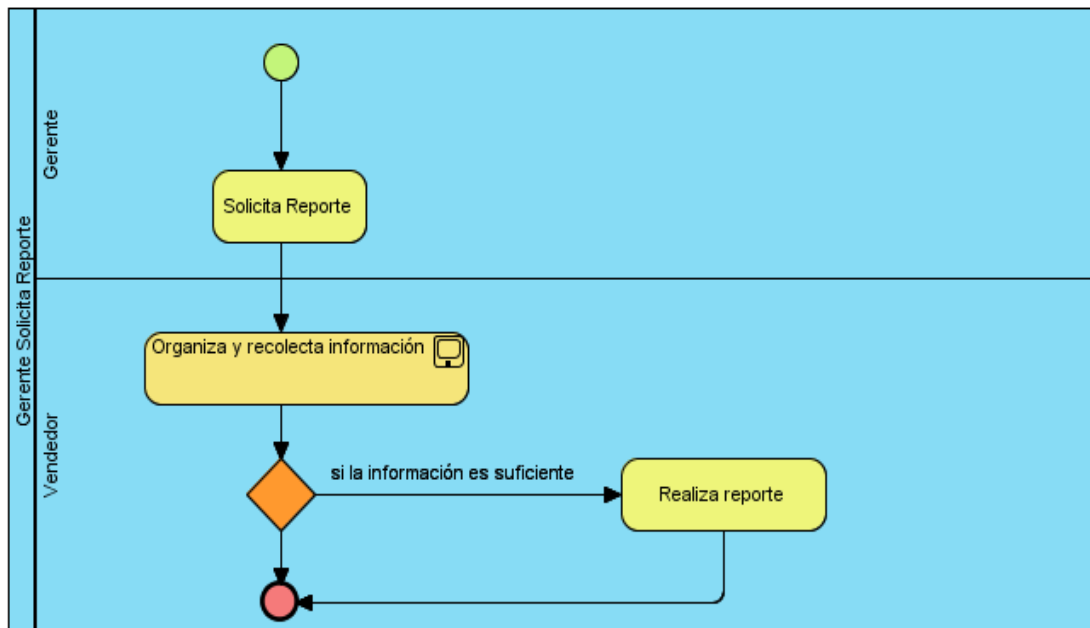


Figura 10: Modelo de Proceso del Negocio-Gerente Solicita Reporte

Modelo de Proceso del Negocio: Registrar Venta

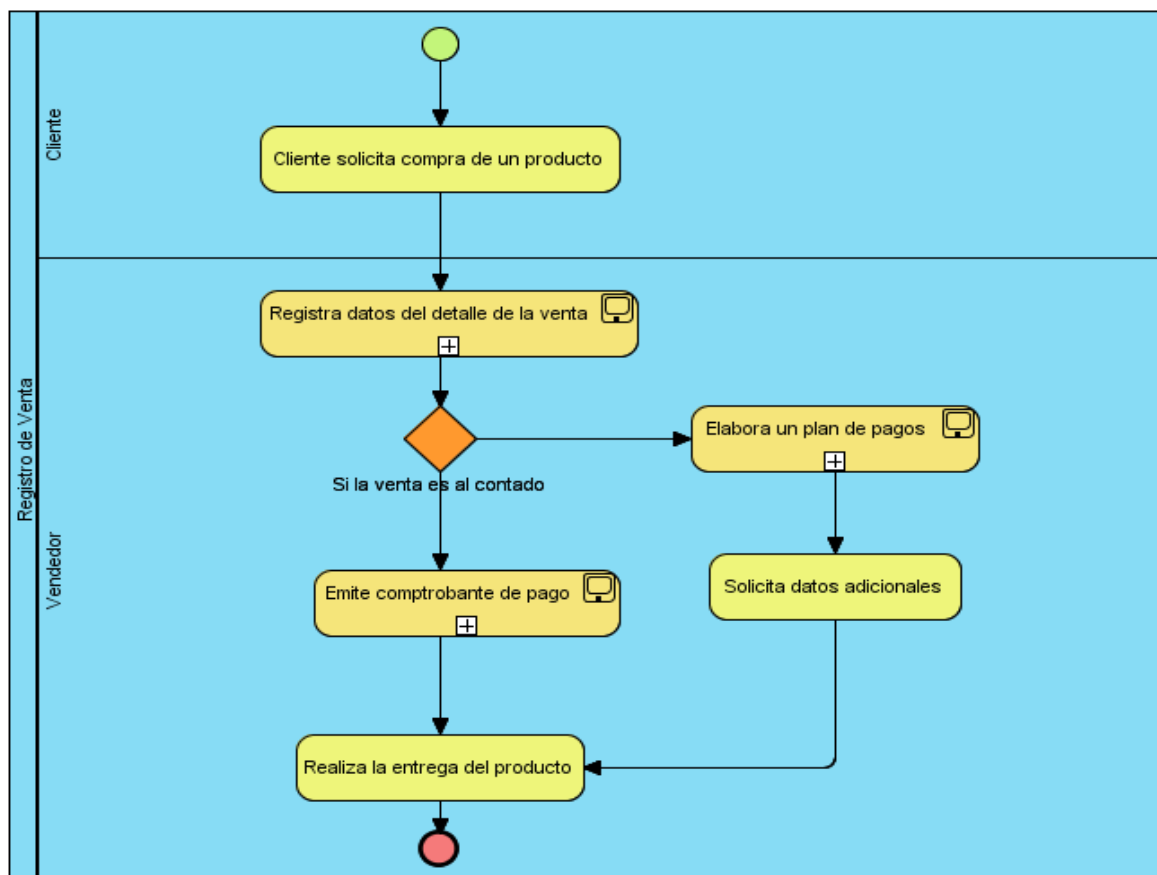


Figura 11: Modelo de Proceso del Negocio-Registrar Venta

Modelo de Proceso del Negocio: Almacenar Producto

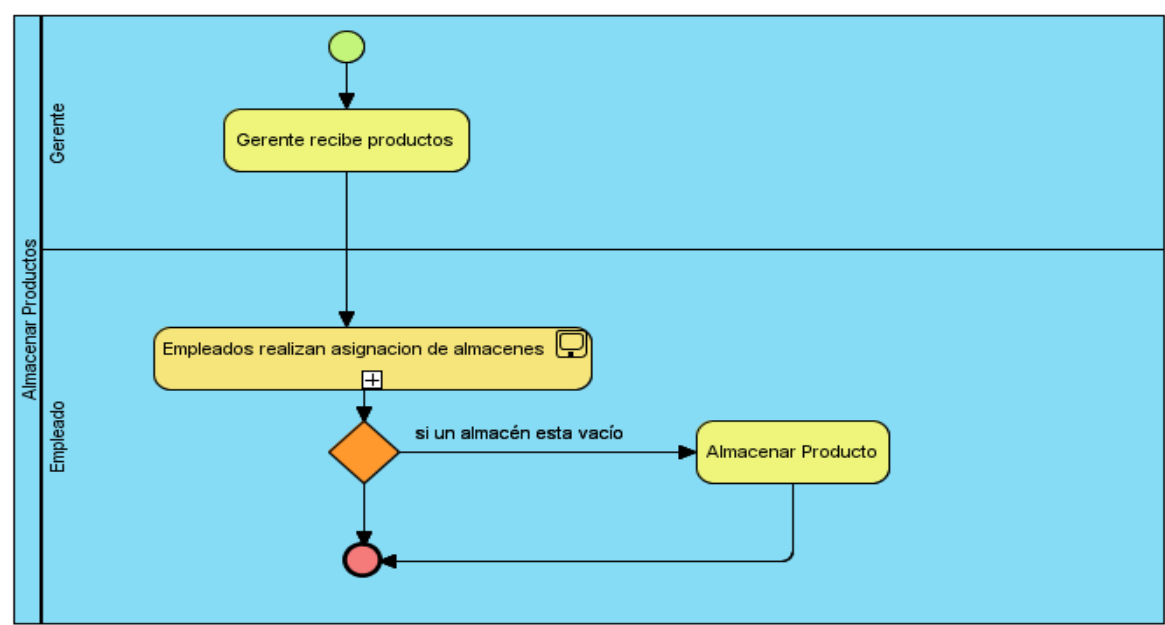


Figura 12: Modelo de Proceso del Negocio-Almacenar Producto

Modelo de Proceso del Negocio: Registra Pago de Cuota

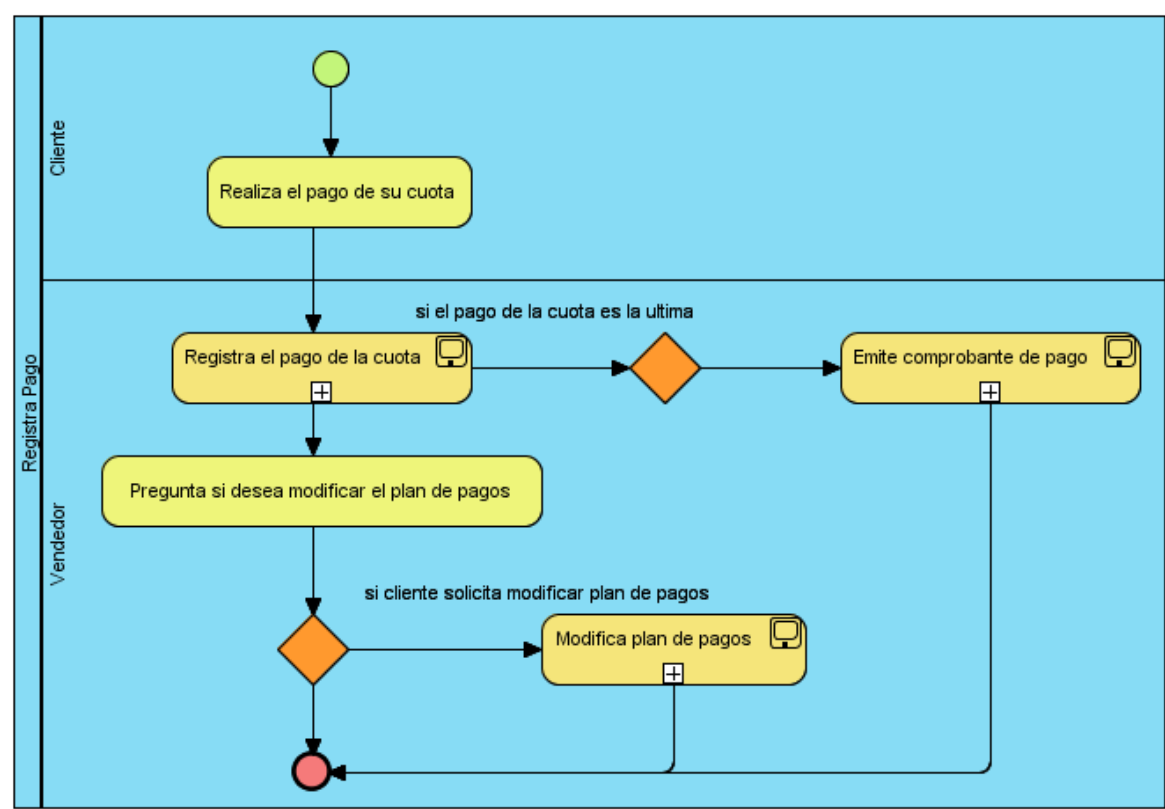


Figura 13: Modelo de Proceso del Negocio- Registra Pago de Cuota

Modelo de Proceso del Negocio: Realizar Compra

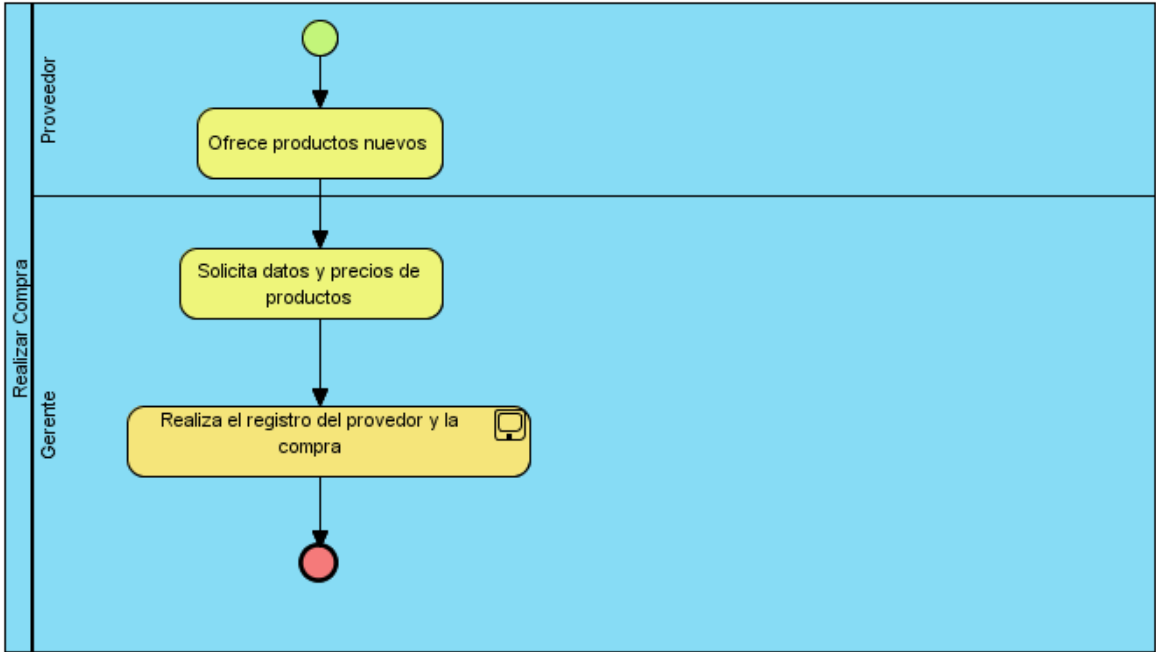


Figura 14: Modelo de Proceso del Negocio-Realizar Compra

Modelo de Proceso del Negocio: Contabilizar ventas mensuales

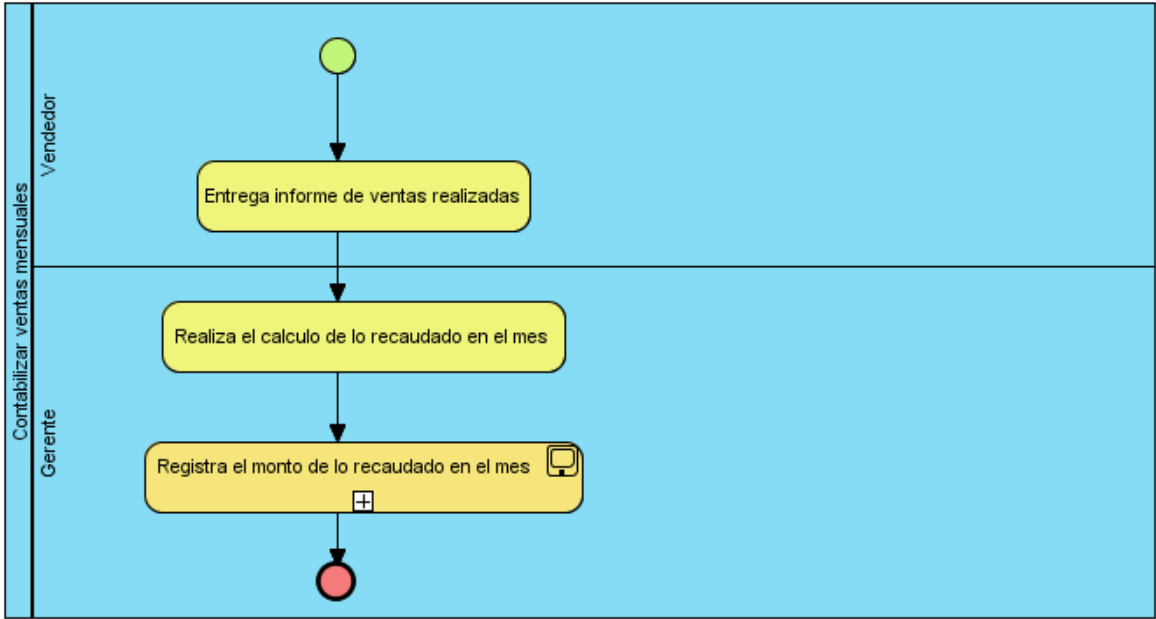


Figura 15: Modelo de Proceso del Negocio-contabilizar ventas mensuales

Modelo de Proceso del Negocio: Solicitar informe de compras de clientes

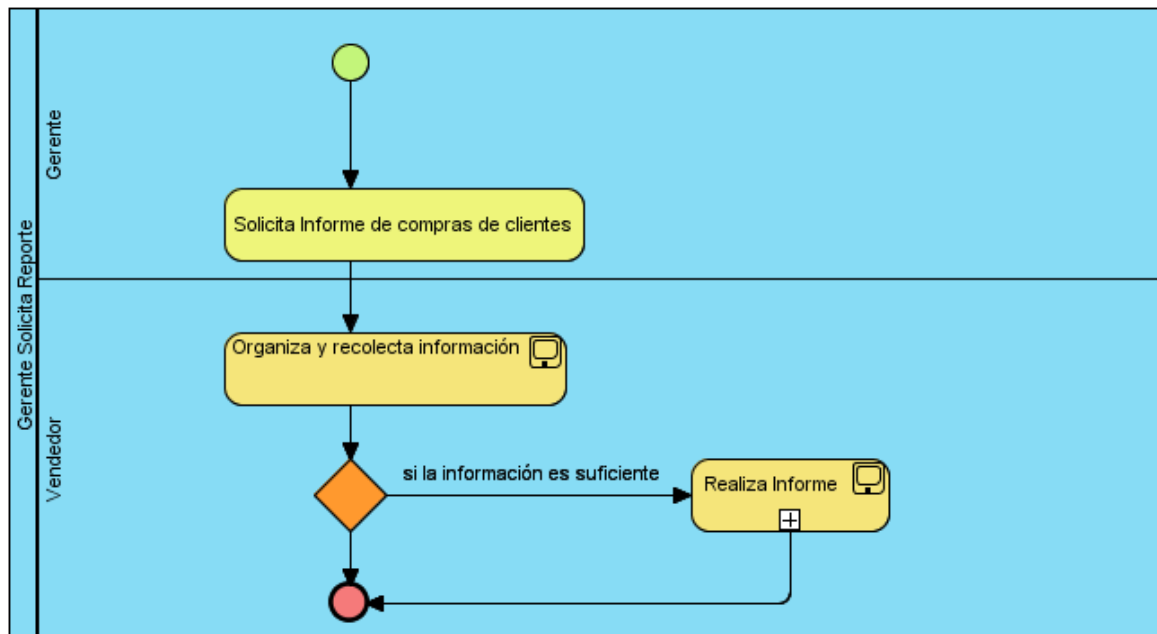


Figura 16: Modelo de Proceso del Negocio-Solicitar informe de compras de clientes

2.1.2.2.3.4 GLOSARIO

2.1.2.2.3.4.1 INTRODUCCIÓN

Este documento recoge términos manejados a lo largo de todo el desarrollo del proyecto, Administración de la Información de Ventas del “**COMERCIAL FIGUEROA**”, trata de un diccionario informal de datos y definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

2.1.2.2.3.4.1.1 PROPÓSITO

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

2.1.2.2.3.4.1.2 ALCANCE

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

2.1.2.2.3.4.1.3 REFERENCIAS

El presente glosario hace referencia a los siguientes documentos:

- Documento Plan de Desarrollo Software.
- Documento Visión del Proyecto.
- Documentos de Especificación de Casos de Uso.
- Documentos del Modelo Análisis y Diseño.

2.1.2.2.3.4.1.4 ORGANIZACIÓN DEL GLOSARIO

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

2.1.2.2.3.4.2 DEFINICIONES

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

- **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 Proceso 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.
- **Autenticación.** Verificación de la identidad de una persona o de un proceso para acceder a un recurso o poder realizar determinada actividad. También se aplica a la verificación de identidad de origen de un mensaje.
- **Backup** (copia de respaldo, copia de seguridad). Copia de ficheros o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de datos. Esta sencilla acción evita numerosos, y a veces irremediables, problemas si se realiza de forma habitual y periódica.
- **Cajero (a).** Persona encargada de realizar los cobros por cada venta realizada, o por cada plan de pago cancelado.
- **Cliente (Red).** Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio. Una estación de trabajo que solicita el contenido de un fichero a un servidor de ficheros es un cliente de este servidor.
- **Cliente.** Persona que realiza una compra directa en el COMERCIAL o que tiene un

plan de pagos, para cubrir el costo del producto adquirido.

- **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.
- **Compra.** Transacción monetaria a cambio de productos de parte de los proveedores del COMERCIAL.
- **Computadora (computador, ordenador)** Máquina electrónica capaz de procesar información siguiendo instrucciones almacenadas en programas. Antes que electrónicas estas máquinas fueron mecánicas o electromecánicas.
- **Código fuente.** Conjunto de instrucciones que componen un programa informático. Estos programas se escriben en determinados lenguajes (java. Java script, etc.)
- **Java.** Lenguaje de programación desarrollado por la empresa Sun para la elaboración o desarrollo de aplicaciones de Software.
- **Java Script** Lenguaje de programación para WWW desarrollado por Netscape. pertenece a la familia Java pero se diferencia de este último es que los programas están incorporados en el fichero HTML.
- **Navegador.** Aplicación para visualizar todo tipo de información y navegar por el espacio Internet. En su forma más básica son aplicaciones hipertexto que facilitan la navegación por los servidores de información Internet; cuentan con funcionalidades plenamente multimedia y permiten indistintamente la navegación por servidores WWW, FTP.
- **Producto.** Proceso que se encuentre a la venta en los predios del COMERCIAL, entre ellos podemos nombrar, electrodomésticos, muebles, juguetería, etc.
- **Proveedor.** Empresa o persona que proporciona de productos varios o uno en específico al COMERCIAL.
- **Permisos.** Atribuciones que se dé a un Usuario del Sistema para la interacción en el mismo.
- **Sistema Operativo.** Un sistema operativo es un programa especial que se carga en un ordenador tras ser encendido y cuya función es Administrar los demás programas, o aplicaciones, que se ejecutarán en dicho ordenador, como, por ejemplo, un procesador de texto o una hoja de cálculo, o la impresión de un texto en una impresora o una conexión a Internet.
- **Usuario:** Persona con una cuenta y con ciertos permisos asignados en el sistema.

- **Vendedor.** Persona encargada de brindar atención e información a los clientes, acerca de los productos y su manejo, existentes en los predios del COMERCIAL.
- **Venta.** Transacción económica a cambio de un producto de parte del COMERCIAL hacia sus clientes.

2.1.2.2.3.5 MODELO DE CASOS DE USO

Modelo de Caso de Uso: Caso de Uso General

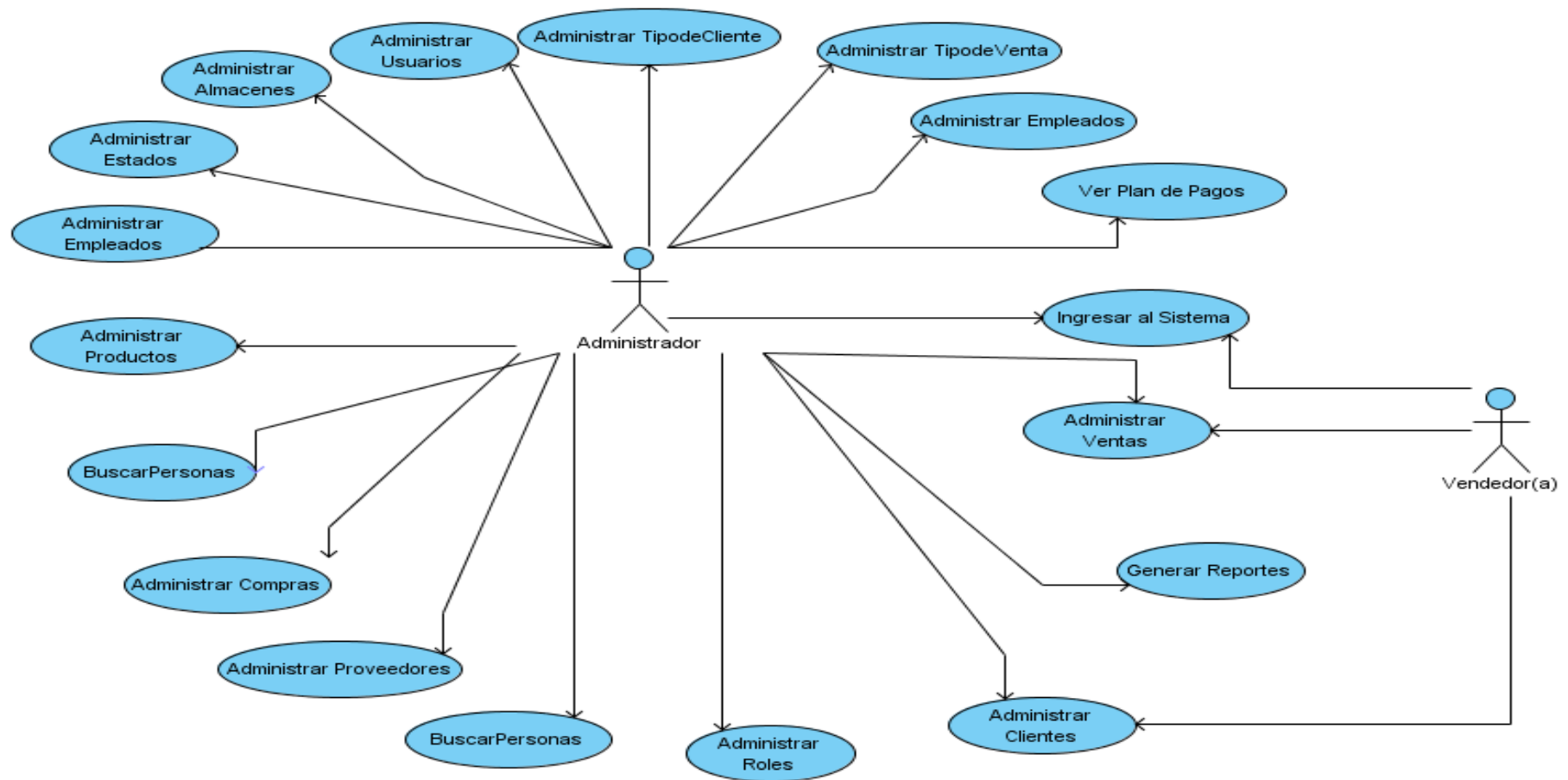


Figura 17: Modelo del Caso de Uso-Caso de Uso General

Modelo de Caso de Uso: Caso de Uso Ingresar al Sistema

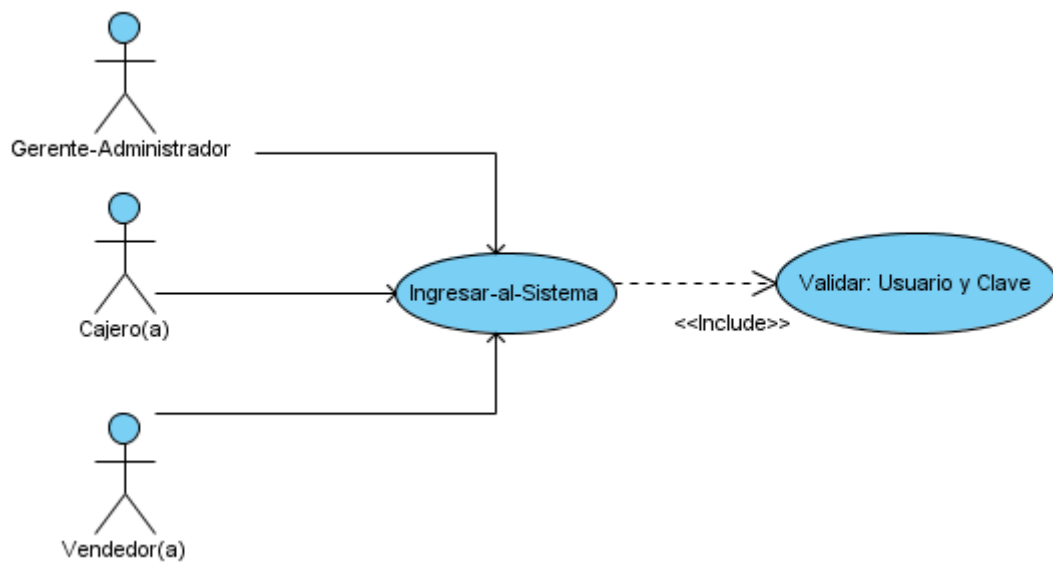


Figura 18: Modelo de Caso de Uso-Caso de Uso Ingresar al Sistema

Modelo de Caso de Uso: Caso de Uso Administrar Usuario

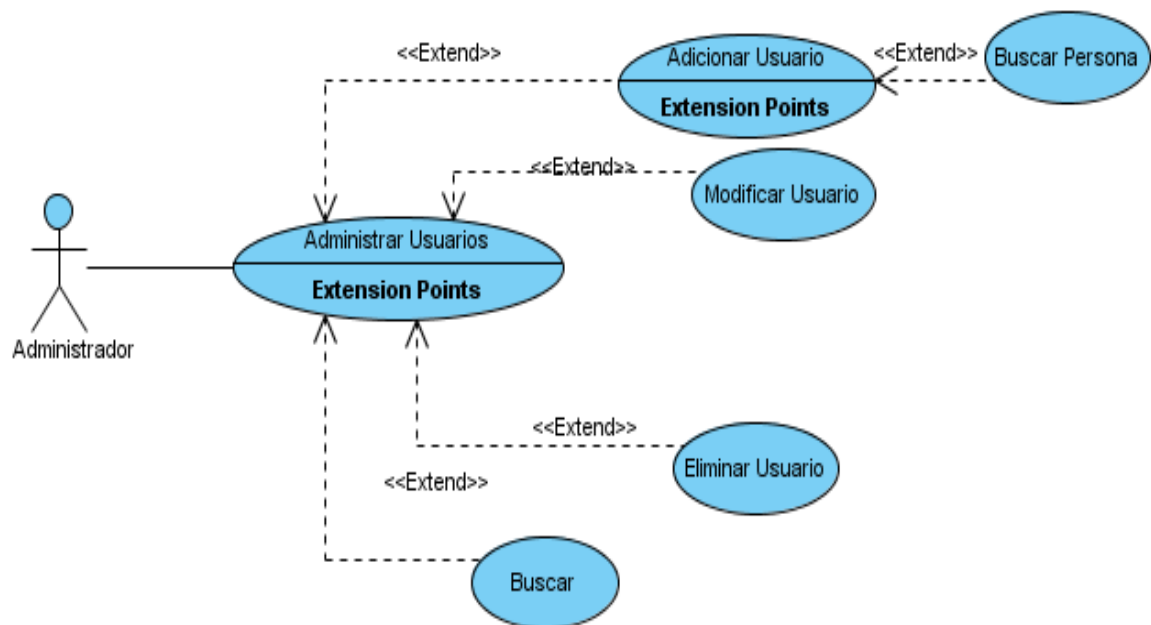


Figura 19: Modelo de Caso de Uso- Administrar Usuario

Modelo de Caso de Uso: Caso de Uso Administrar Almacén

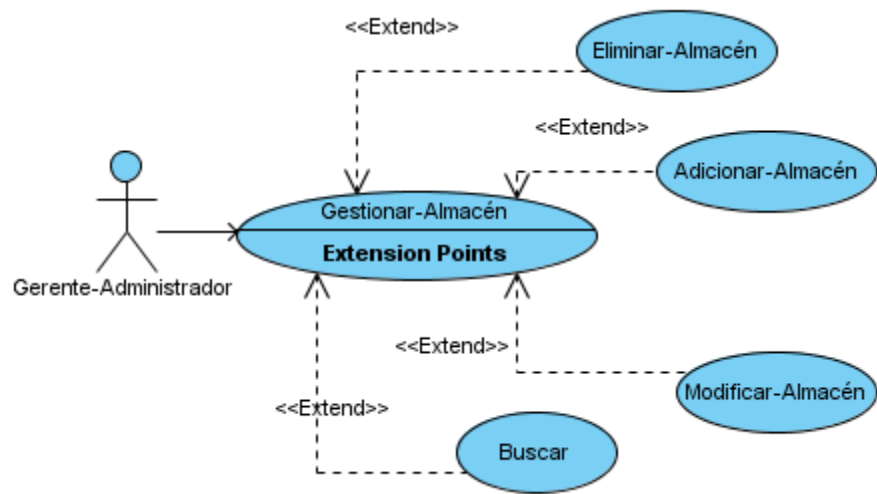


Figura 20: Modelo de Caso de Uso- Administrar Almacén

Modelo de Caso de Uso: Caso de Uso Administrar Cargo

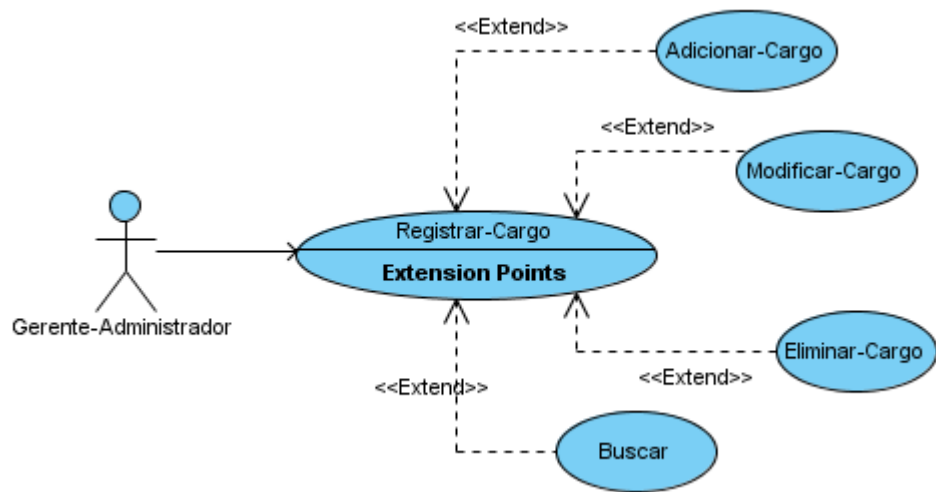


Figura 21: Modelo de Caso de Uso- Administrar Cargo

Modelo de Caso de Uso: Caso de Uso Administrar Compra

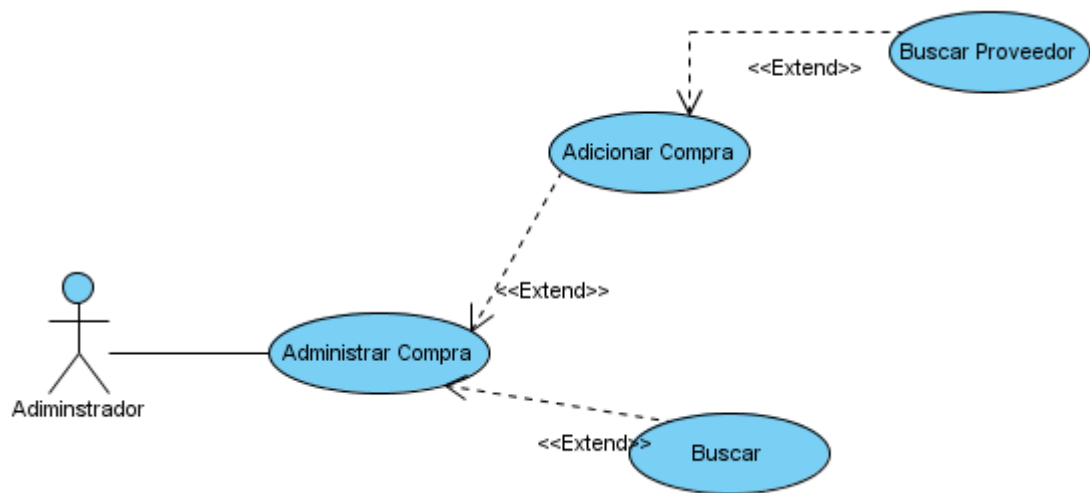


Figura 22: Modelo de Caso de Uso- Administrar Compra

Modelo de Caso de Uso: Caso de Uso Administrar Empleado

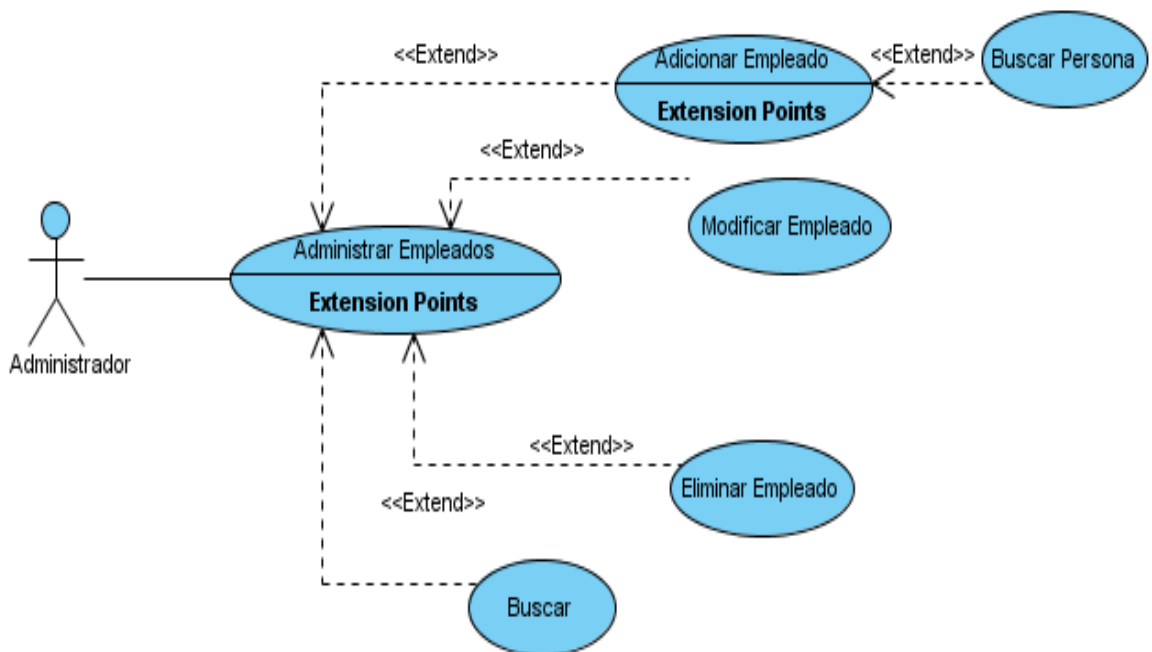


Figura 23: Modelo de Caso de Uso- Administrar Empleado

Modelo de Caso de Uso: Caso de Uso Buscar Personas

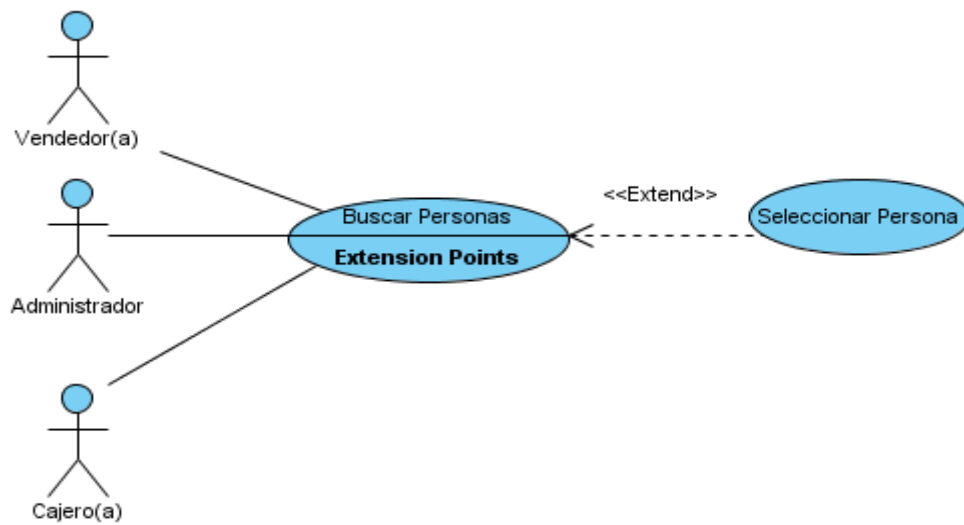


Figura 24: Modelo de Caso de Uso-Buscar Personas

Modelo de Caso de Uso: Caso de Uso Administrar Productos

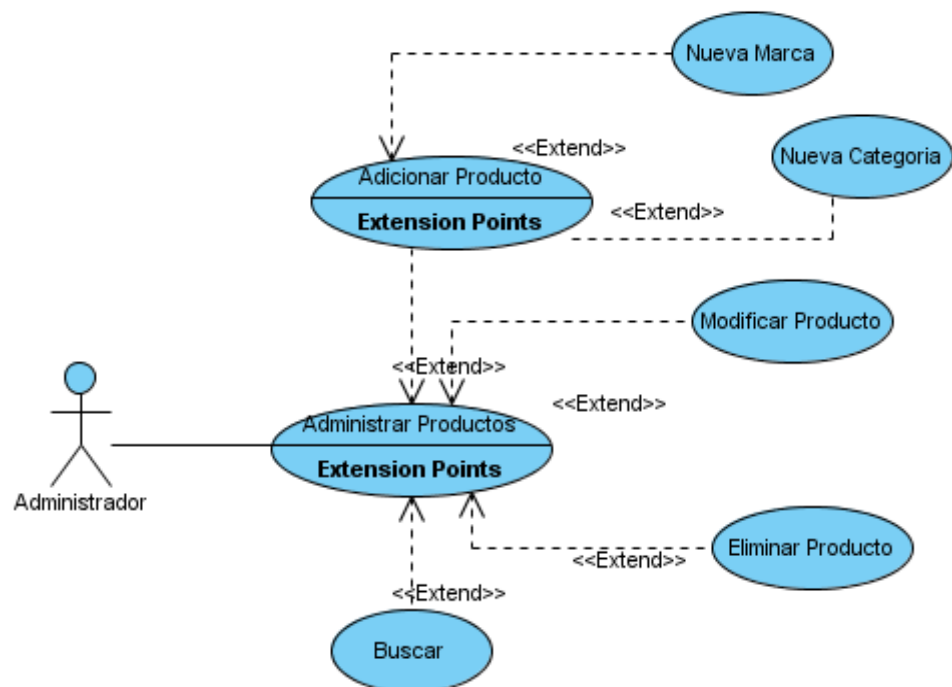


Figura 25: Modelo de Caso de Uso- Administrar Productos

Modelo de Caso de Uso: Caso de Uso Administrar Proveedores

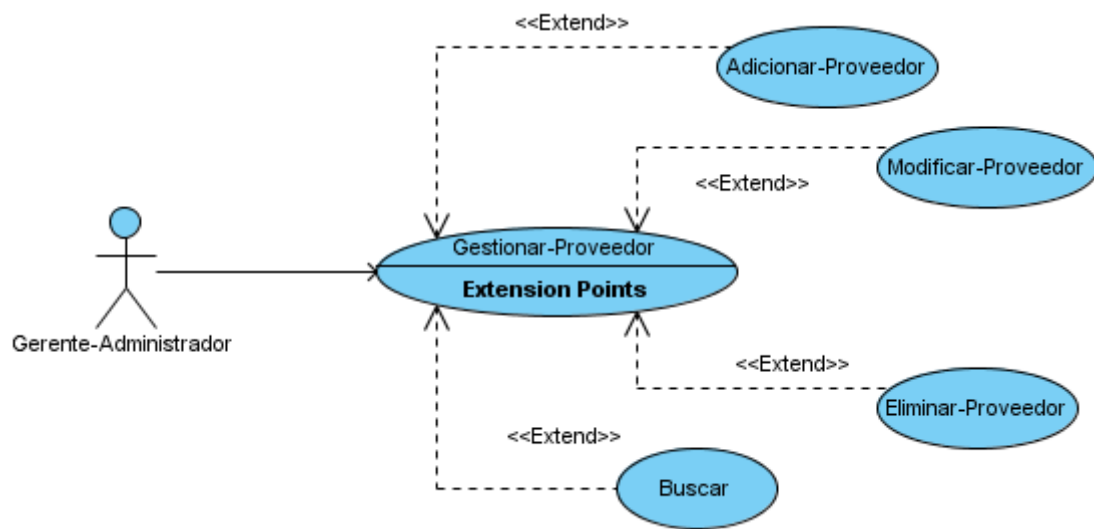


Figura 26: Modelo de Caso de Uso- Administrar Proveedores

Modelo de Caso de Uso: Caso de Uso Administrar Roles

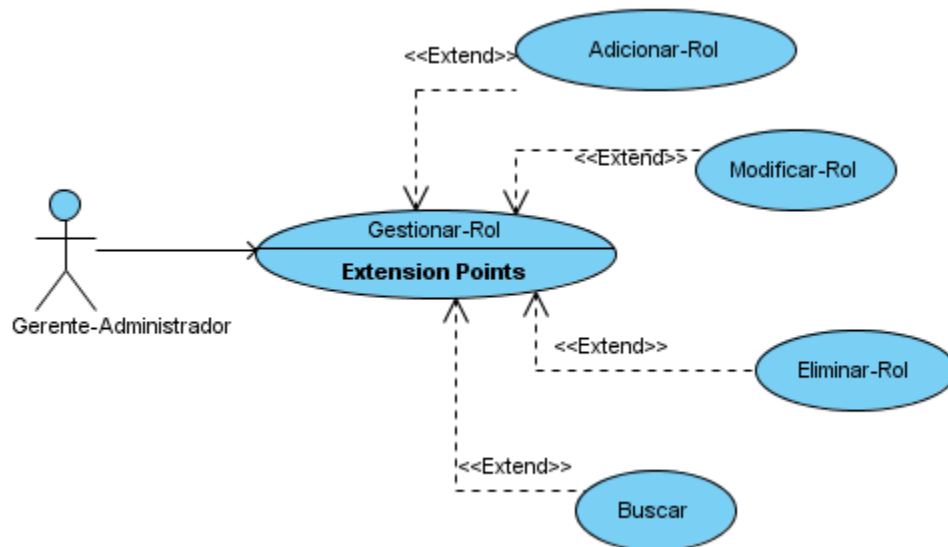


Figura 27: Modelo de Caso de Uso- Administrar Roles

Modelo de Caso de Uso: Caso de Uso Administrar Clientes

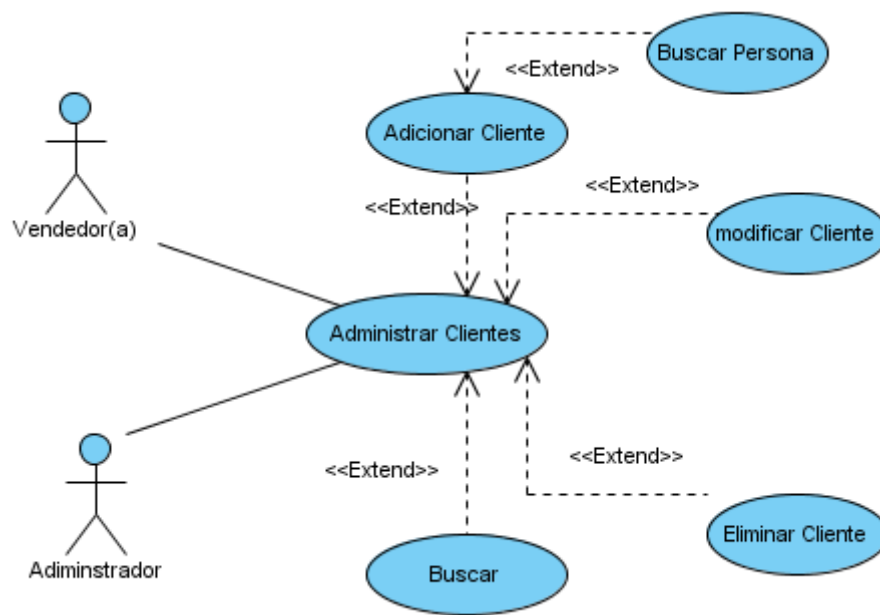


Figura 28: Modelo de Caso de Uso- Administrar Clientes

Modelo de Caso de Uso: Caso de Uso Administrar Estados

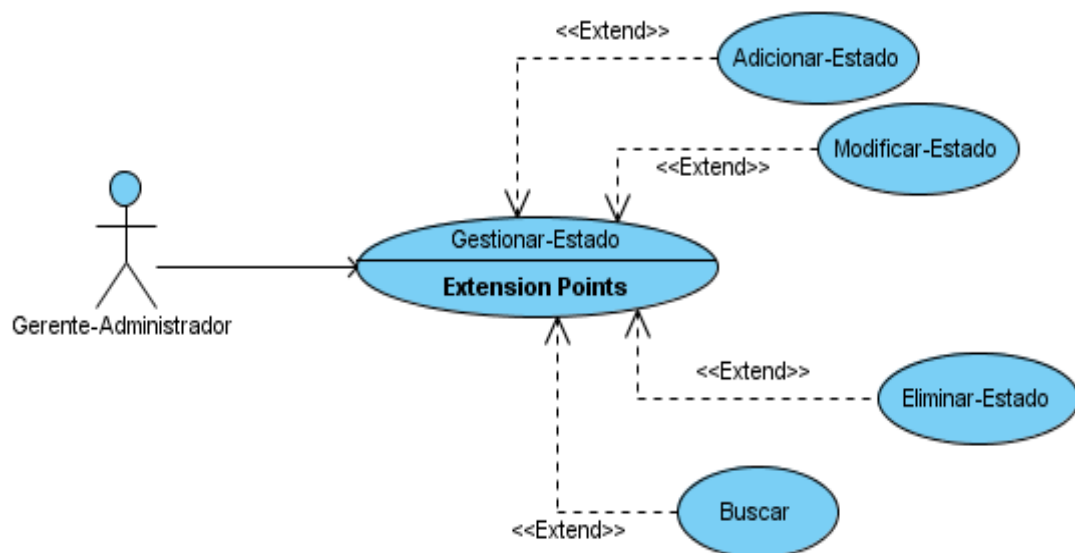


Figura 29: Modelo de Caso de Uso- Administrar Estados

Modelo de Caso de Uso: Caso de Uso Almacenar Producto



Figura 30: Modelo de Caso de Uso- Almacenar Productos

Modelo de Caso de Uso: Caso de Uso Administrar Ventas

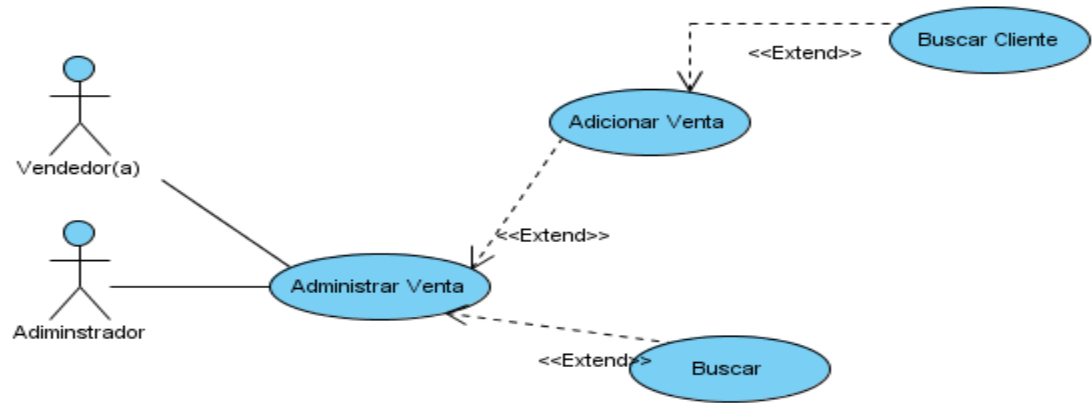


Figura 31: Modelo de Caso de Uso-Caso de Uso Administrar Ventas

Modelo de Caso de Uso: Caso de Uso Registrar Dosificación



Figura 32: Modelo de Caso de Uso-Caso de Uso Registrar Dosificación

Modelo de Caso de Uso: Caso de Uso Administrar Tipo-Clientes

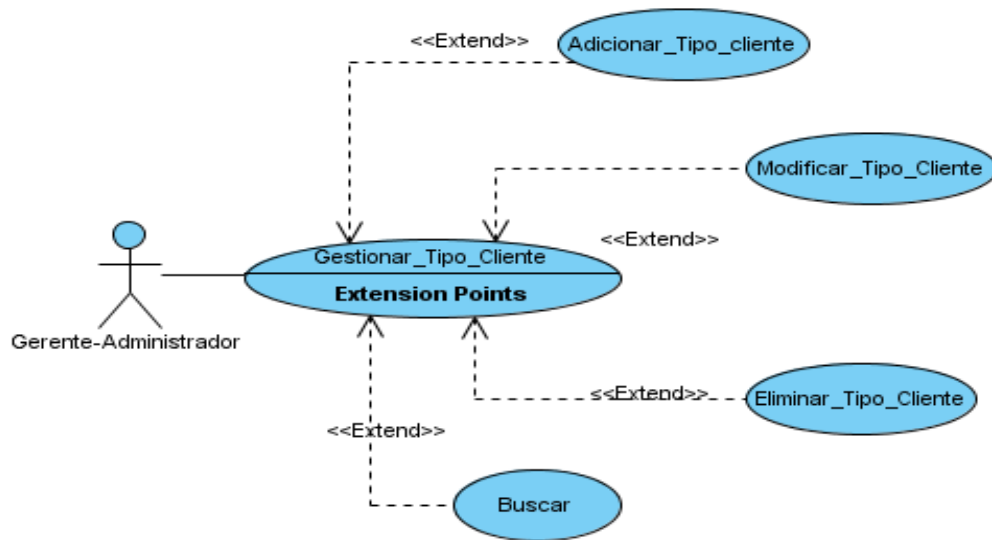


Figura 33: Modelo de Caso de Uso- Administrar Tipo-clientes

Modelo de Caso de Uso: Caso de Uso Ver Plan de Pagos

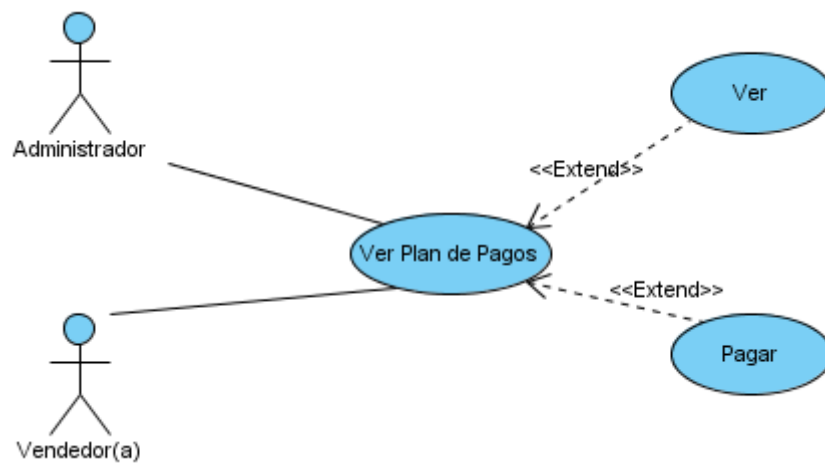


Figura 34: Modelo de Caso de Uso-Ver Plan de Pagos

Modelo de Caso de Uso: Caso de Uso Administrar Tipo de Ventas

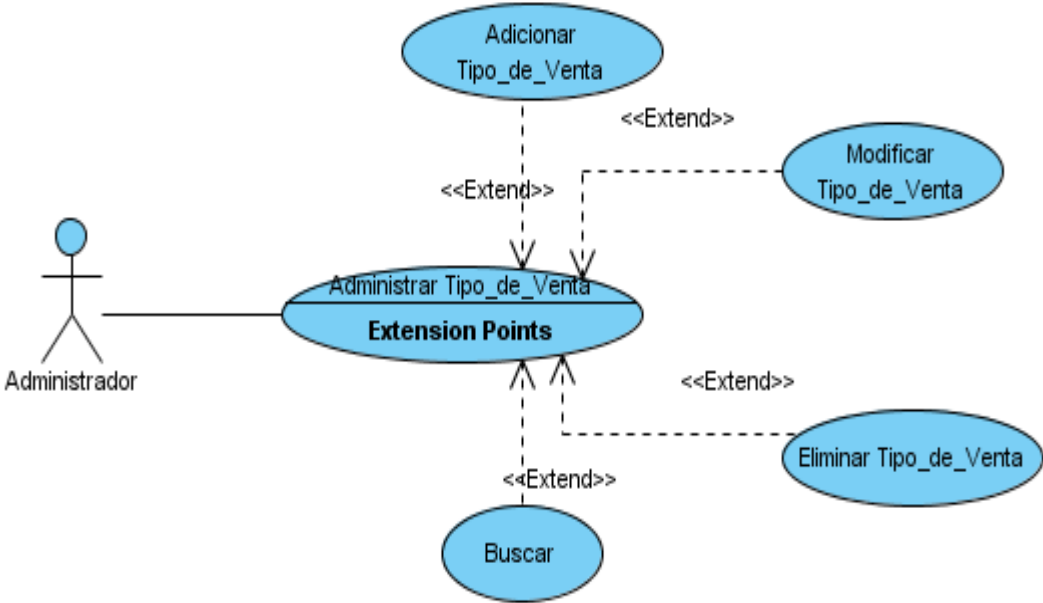


Figura 35: Modelo de Caso de Uso-Administrar Tipo de Venta

Modelo de Caso de Uso: Caso de Uso Generar Reporte

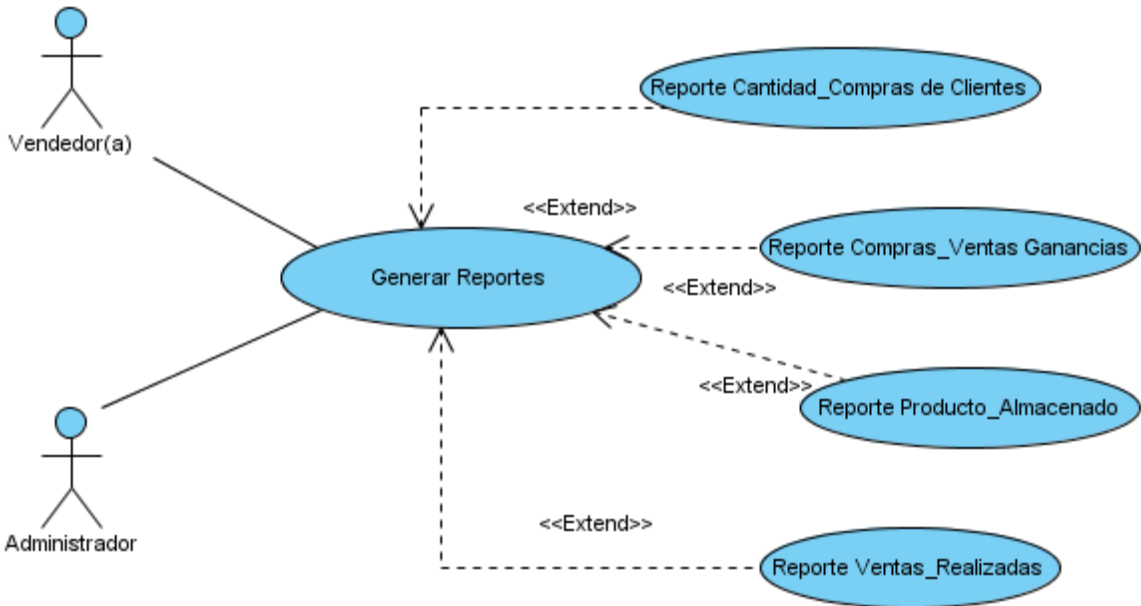


Figura 36: Modelo de Caso de Uso-Generar Reporte

2.1.2.2.3.6 VISIÓN

2.1.2.2.3.6.1 PROPÓSITO

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.

Para determinar la perspectiva o visión del cliente, respecto al producto que ofrecemos en este documento, se realizó entrevistas con los clientes, los cuales nos expusieron sus expectativas que tiene en relación al software de aplicación que se pretende desarrollar. Se rescataron las siguientes apreciaciones:

- Seguridad en el resguardo de los datos que se manejan.
- Confiabilidad en el manejo de información de clientes.
- Registro eficiente de empleados, usuarios, el sistema debería permitir introducir los datos necesarios con la menor cantidad de pasos, pero además, cumplir con todas las expectativas que son necesarias para una institución seria.
- Registro fácil y correcto de compras, ventas, registro de productos en almacenes, ventas al crédito, etc. El sistema debería permitir introducir los datos necesarios con la menor cantidad de pasos.
- Fácil manejo del sistema en todas sus aplicaciones, que permita un manejo eficiente y eficaz de su información.
- Obtención de los reportes necesarios en cualquier momento y de forma fácil.
- Al sistema, solo tendrá acceso, personal autorizado.
- Acceso sin problemas a la aplicación.
- Que se puedan obtener reportes confiables, conforme a lo que en realidad se haya registrado en el Sistema.
- Las pantallas deben ser modernas, elegantes y sobre todo bien diseñadas.
- El sistema debe brindar información confiable.
- Clasificación de la información.
- Buena organización en la presentación de la información en las pantallas.

Los detalles de cómo el sistema cubre los requerimientos, se pueden observar en la especificación de los casos de uso y otros documentos adicionales que se hayan generado en el desarrollo del sistema.

2.1.2.2.3.6.2 ALCANCE

El documento **Visión** se ocupa, como ya se ha visto en el Plan de desarrollo, de la administración y control de la información de ventas del “COMERCIAL FOGUEROA”. Dicho sistema será desarrollado por la universitaria, Griselda Gallardo A.

El sistema permitirá a los usuarios del comercial, controlar todo lo relativo al, registro de compras, ventas, productos, generar reportes, estado de empleados, etc.

2.1.2.2.3.6.3 REFERENCIAS

- Glosario.
- Plan de desarrollo de software.
- RUP (Rational Unified Process).
- Diagrama de proceso del Negocio
- Diagrama de casos de uso.
- Diagrama de Actividades.
- Diagrama de Secuencia.
- Diagrama de Clases.
- Diagrama de componentes
- Diagrama de colaboración

2.1.2.2.3.6.4 POSICIONAMIENTO

2.1.2.2.3.6.4.1 OPORTUNIDAD DEL NEGOCIO

Este sistema permitirá automatizar toda la información del “COMERCIAL FOGUEROA” referida a compras, ventas, productos y clientes, lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas sencillas, atractivas y amigables.

2.1.2.2.3.6.4.2 SENTENCIA QUE DEFINE EL PROBLEMA

El problema de	Administración de la información de Ventas, clientes, productos, reportes de almacenes, etc.
afecta a	Al “COMERCIAL FIGUEROA”
El impacto asociado es	Almacenar toda la información referente al registro de compras, ventas, productos, almacenes, y que esta información esté al instante, accesible y actualizada.
Una solución adecuada sería	Automatización de la información (segura, confiable, etc.) referente a los registros nuevos de Ventas.

2.1.2.2.3.6.4.3 RESUMEN DE STAKEHOLDERS

Nombre	Descripción	Responsabilidades
“COMERCIAL FIGUEROA”	Clientes en potencia.	El stakeholder realiza: Representa a todos los usuarios posibles del sistema mediante su personal autorizado. Seguimiento del desarrollo del proyecto, brinda y aprueba los requisitos y funcionalidad del sistema.
Griselda Gallardo A.	Directora del proyecto	Responsable e interesado principal en el desarrollo del proyecto. Diseña las posibles interfaces finales a ser implementadas en el sistema. Programa el software de acuerdo a los requisitos del sistema. Ejecuta y realiza reiteradas pruebas al sistema en su posible versión final.
Lic.	Profesional	El stakeholder realiza: Realiza un seguimiento y control del

		desarrollo del proyecto. Analiza los distintos documentos presentados acerca del producto.

2.1.2.2.3.6.4.4 RESUMEN DE USUARIOS

Nombre	Descripción
Administrador	Responsable del control de usuarios, permisos, empleados, clientes, realización backups.
Vendedor(a)	Encargado de la atención directa a los clientes, registro de ventas y registro de productos en predios del comercial.

2.1.2.2.3.6.4.5 ENTORNO DE USUARIOS

Los usuarios entrarán al sistema identificándose sobre un ordenador con un sistema operativo Windows XP o superior, y tras este paso, podrán acceder a las funcionalidades del sistema, con sus claves asignadas según el rol que desempeñan en la institución. El usuario solo tiene que saber usar un navegador web (por ejemplo: google chrome).

2.1.2.2.3.6.4.6 DESCRIPCIÓN GLOBAL DEL PRODUCTO

2.1.2.2.3.6.4.6.1 PERSPECTIVA DEL PRODUCTO

El producto a desarrollar es un sistema para mejorar la administración de información de Ventas del “COMERCIAL FIGUEROA”. El sistema podría ser implementado o usado en otro comercial con similares características a este.

2.1.2.2.3.7 ESPECIFICACIONES DE CASOS DE USO

Para los casos de uso y que se entienda su funcionalidad, se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: Nombre, Descripción, precondiciones, post-condiciones, flujo normal, flujo alternativo.

Nombre:	Ingresar al sistema
Fecha de creación:	08/05/2014
Descripción: Permite ingresar al sistema	
Actores: Administrador, Vendedor(a) y Cajero(a)	
Precondiciones:	
1. Flujo Normal: 2. El actor introduce su nombre y usuario al formulario del sistema. 3. Presiona la opción “ACCEDER” 4. El sistema valida sus datos. 5. El actor espera su validación.	
Flujo Alternativo: Si los datos introducidos en el formulario no son válidos el sistema le devuelve un mensaje,” introduzca bien los datos”.	
Post condiciones: Ninguno	

Tabla 1: Especificación de C.U. – Ingresar al Sistema

Nombre:	Administrar Usuarios
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Usuarios	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Usuarios”	

2. El sistema le muestra lista de Usuarios y opciones.
Flujo Alternativo: Ninguno
Post condiciones: ninguno

Tabla 2: Especificación de C.U. – Administrar Usuario

Nombre:	Adicionar Usuarios
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Usuario	
Actores: Administrador	
Precondiciones: seleccionar Administrar Usuario	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Usuario en la en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Usuarios	
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje: ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 3: Especificación de C.U. – Adicionar Usuario

Nombre:	Modificar Usuario
Fecha de creación:	08/05/2014
Descripción: Permite modificar Usuario	
Actores: Administrador	
Precondiciones: seleccionar Administrar Usuarios	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos de Rol. 4. El actor modifica datos del Usuario en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Usuario	
Flujo Alternativo: - Si no selecciona una fila mensaje: “seleccione un Usuario” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 4: Especificación de C.U. – Modificar Usuario

Nombre:	Eliminar Usuario
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Usuario	
Actores: Administrador	
Precondiciones: seleccionar Administrar Usuario	

Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 1. El actor selecciona la opción eliminar. 2. El sistema muestra mensaje: “¿Está seguro de eliminar?” 3. El actor selecciona “Aceptar” 4. El sistema elimina al id de la fila. 5. Mensaje “ se eliminó satisfactoriamente” 6. Retorna a pantalla Administrar Usuarios.
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione Usuario” - Si presiona cancelar retorna a pantalla Administrar Usuarios.
Post condiciones: Ninguno

Tabla 5: Especificación de C.U. – Eliminar Usuario

Nombre:	Administrar Roles
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Roles	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Roles” 2. El sistema le muestra lista de Roles y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: ninguno	

Tabla 6: Especificación de C.U. – Administrar Rol

Nombre:	Adicionar Roles
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Rol	
Actores: Administrador	
Precondiciones: seleccionar Administrar Roles	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Rol en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje:“ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Roles.	
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 7: Especificación de C.U. – Adicionar Rol

Nombre:	Modificar Roles
Fecha de creación:	08/05/2014
Descripción: Permite modificar Rol	
Actores: Administrador	
Precondiciones: seleccionar Administrar Roles	

Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Rol. 4. El actor modifica datos del Rol en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Roles.
Flujo Alternativo: - Si no selecciona una fila mensaje “seleccione un Rol” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 8: Especificación de C.U. – Modificar Rol

Nombre:	Eliminar roles
Fecha de creación:	08/05/2014
Descripción: Permite eliminar roles	
Actores: Administrador	
Precondiciones: seleccionar Administrar Roles	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Roles.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un rol”	

- Si presiona cancelar retorna a pantalla Administrar Roles.
Post condiciones: Ninguno

Tabla 9: Especificación de C.U. – Eliminar Rol

Nombre:	Administrar Empleado
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Empleado	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Empleado” 2. El sistema le muestra lista de Empleado y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 10: Especificación de C.U. – Administrar Empleado

Nombre:	Adicionar Empleado
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Empleado	
Actores: Administrador	

Precondiciones: seleccionar Administrar Empleado
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos de los Empleado en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Empleado.
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 11: Especificación de C.U. – Adicionar Empleado

Nombre:	Modificar Empleado
Fecha de creación:	08/05/2014
Descripción: Permite modificar Empleado	
Actores: Administrador	
Precondiciones: seleccionar Administrar Empleado	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Empleado. 4. El actor modifica datos del Empleado en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Empleado.	

Flujo Alternativo:	
- Si no selecciona una fila mensaje “seleccione un empleado” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”	
Post condiciones:	
Ninguno	

Tabla 12: Especificación de C.U. – Modificar Empleado

Nombre:	Eliminar Empleado
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Empleado	
Actores: Administrador	
Precondiciones: seleccionar Administrar Empleado	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Empleado.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Empleado” - Si presiona cancelar retorna a pantalla Administrar Empleados.	
Post condiciones:	

Tabla 13: Especificación de C.U. – Eliminar Empleado

Nombre:	Administrar Cargo
Fecha de creación:	08/05/2014

Descripción: Permite Administrar Cargo
Actores: Administrador
Precondiciones: Estar logueado en el sistema como Administrador
Flujo Normal: 1. El actor selecciona la opción “Administrar Cargo” 2. El sistema le muestra lista de Cargo y opciones.
Flujo Alternativo: Ninguno
Post condiciones: Ninguno

Tabla 14: Especificación de C.U. – Administrar Cargo

Nombre:	Adicionar Cargo
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Cargo	
Actores: Administrador	
Precondiciones: seleccionar Administrar Cargo	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos de los Cargo en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Cargo.	
Flujo Alternativo:	

- Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje "introduzca bien los datos"
Post condiciones: Ninguno

Tabla 15: Especificación de C.U. – Adicionar Cargo

Nombre:	Modificar Cargo
Fecha de creación:	08/05/2014
Descripción: Permite modificar Cargo	
Actores: Administrador	
Precondiciones: seleccionar Administrar Cargo	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Cargo. 4. El actor modifica datos del Cargo en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: " se modificó satisfactoriamente" 9. Retorna a pantalla Administrar Cargo.	
Flujo Alternativo: - Si no selecciona una fila mensaje "seleccione un Cargo" - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: "introduzca bien los datos"	
Post condiciones: Ninguno	

Tabla 16: Especificación de C.U. – Modificar Cargo

Nombre:	Eliminar Cargo

Fecha de creación:	08/05/2014
Descripción: Permite eliminar Cargo	
Actores: Administrador	
Precondiciones: seleccionar Administrar Cargo	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Cargo.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Cargo” - Si presiona cancelar retorna a pantalla Administrar Cargos.	
Post condiciones:	

Tabla 17: Especificación de C.U. – Eliminar Cargo

Nombre:	Administrar Clientes
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Clientes	
Actores: Administrador, Cajero(a), Vendedor(a).	
Precondiciones: Estar logueado en el sistema	
Flujo Normal: 1. El actor selecciona la opción “Administrar Clientes” 2. El sistema le muestra lista de Estados y opciones.	

Flujo Alternativo:
Ninguno
Post condiciones:
Ninguno

Tabla 18: Especificación de C.U. – Administrar Clientes

Nombre:	Adicionar Cliente
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Cliente	
Actores: Administrador, Vendedor(a).	
Precondiciones: seleccionar Administrar Cliente	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Cliente en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Cliente.	
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 19: Especificación de C.U. – Adicionar Clientes

Nombre:	Modificar Cliente

Fecha de creación:	08/05/2014
Descripción: Permite modificar Cliente	
Actores: Administrador, Vendedor(a).	
Precondiciones: seleccionar Administrar Cliente	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Cliente. 4. El actor modifica datos del Cliente en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Clientes.	
Flujo Alternativo: - Si no selecciona una fila mensaje “seleccione un Cliente” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 20: Especificación de C.U. – Modificar Clientes

Nombre:	Eliminar Cliente
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Cliente	
Actores: Administrador, Vendedor(a).	
Precondiciones: seleccionar Administrar Cliente	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar.	

3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Clientes.
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Cliente” - Si presiona cancelar retorna a pantalla Administrar Cliente.
Post condiciones:

Tabla 21: Especificación de C.U. – Eliminar Cliente

Nombre:	Administrar Tipo_Cliente
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Tipo_Cliente	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Tipo_Cliente” 2. El sistema le muestra lista de Tipo_Cliente y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 22: Especificación de C.U. – Administrar Tipo_Cliente

Nombre:	Adicionar Tipo_Cliente

Fecha de creación:	08/05/2014
Descripción: Permite adicionar Tipos de Clientes	
Actores: Administrador	
Precondiciones: seleccionar Administrar Tipo de Cliente	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Tipo de Cliente en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Tipo_Cliente.	
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 23: Especificación de C.U. – Adicionar Tipo_Cliente

Nombre:	Modificar Tipo_Cliente
Fecha de creación:	08/05/2014
Descripción: Permite modificar Tipos de Clientes	
Actores: Administrador	
Precondiciones: seleccionar Administrar Tipo_Cliente	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar.	

3. El sistema le muestra formulario con datos del Tipo_Cliente. 4. El actor modifica datos del Tipo_Cliente en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Tipo_Cliente.
Flujo Alternativo: - Si no selecciona una fila mensaje “seleccione un tipo de cliente” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: “introduzca bien los datos”.
Post condiciones: Ninguno

Tabla 24: Especificación de C.U. – Modificar Tipo_Cliente

Nombre:	Eliminar Tipo_Cliente
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Tipo_Cliente	
Actores: Administrador	
Precondiciones: seleccionar Administrar Tipo_Cliente	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Tipo_Cliente.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Tipo_Cliente” - Si presiona cancelar retorna a pantalla Administrar Tipo_Cliente.	

Post condiciones:

Tabla 25: Especificación de C.U. – Eliminar Tipo_Cliente

Nombre:	Administrar Ventas
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Ventas	
Actores: Administrador, Cajero(a), Vendedor(a).	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Ventas” 2. El sistema le muestra lista de Estados y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 26: Especificación de C.U. – Administrar Ventas

Nombre:	Ver Plan de Pagos
Fecha de creación:	08/05/2014
Descripción: Permite ver el plan de pagos	
Actores: Administrador, Vendedor(a).	
Precondiciones: seleccionar ver plan de pagos	
Flujo Normal:	

1. El actor selecciona una venta al crédito. 2. Selecciona “Ver Plan de Pagos”. 3. El sistema muestra el detalle del plan de pago de la venta.
Flujo Alternativo: - Si el actor no selecciona una venta al crédito se muestra el mensaje: “Seleccione una venta”.
Post condiciones: Ninguno

Tabla 27: Especificación de C.U. – Ver Plan de Pagos

Nombre:	Pagar plan de pago
Fecha de creación:	08/05/2014
Descripción: Permite cancelar una cuota del plan de pago	
Actores: Administrador, Vendedor(a).	
Precondiciones: seleccionar Administrar Venta y ver plan de pago	
Flujo Normal: 1. El actor selecciona “pagar”. 2. El pago cambia de debe a pagado.	
Flujo Alternativo: - Si no selecciona una fila muestra el mensaje: “seleccione una Venta” -	
Post condiciones: Ninguno	

Tabla 28: Especificación de C.U. – Modificar Ventas

Nombre:	Administrar Tipo de Venta
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Tipo de Venta	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Tipo de Venta” 2. El sistema le muestra lista de Estados y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 29: Especificación de C.U. – Administrar Tipo de Venta

Nombre:	Adicionar Tipo de Venta
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Tipo de Venta	
Actores: Administrador	
Precondiciones: seleccionar Administrar Tipo de Venta	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Tipo de Venta en el formulario. 4. El actor guarda.	

5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Tipo de Venta.
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 30: Especificación de C.U. – Adicionar Tipo de Venta

Nombre:	Modificar Tipo de Venta
Fecha de creación:	08/05/2014
Descripción: Permite modificar Tipo de Venta	
Actores: Administrador	
Precondiciones: seleccionar Administrar Tipo de Venta	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Tipo de Venta. 4. El actor modifica datos del Tipo de Venta en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Tipo de Venta.	
Flujo Alternativo: - Si no selecciona una fila mensaje “seleccione un Tipo de Venta” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”	
Post condiciones:	

Ninguno

Tabla 31: Especificación de C.U. – Modificar Tipo de Venta

Nombre:	Eliminar Tipo de Venta
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Tipo de Venta	
Actores: Administrador	
Precondiciones: seleccionar Administrar Tipo de Venta	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Tipo de Venta.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Tipo de Venta” - Si presiona cancelar retorna a pantalla Administrar Tipo de Venta.	
Post condiciones:	

Tabla 32: Especificación de C.U. – Eliminar Tipo de Venta

Nombre:	Ver Plan de Pagos
Fecha de creación:	08/05/2014
Descripción: Permite ver el manejo del sistema	
Actores: Administrador, Vendedor(a)	

Precondiciones: Debe ingresar a “Ventas de la pantalla Principal” y seleccionar la opción “ Ver Plan de Pagos”
Flujo Normal: 1. El actor selecciona la opción “Ver Plan de Pagos” 2. El sistema muestra la lista de plan de pagos. 3. El actor selecciona una de las opciones del menú 4. El sistema carga la pantalla con pasos y pantallas acerca del manejo del sistema.
Flujo Alternativo: El actor no selecciona la opción Ver Plan de Pagos.
Post condiciones: Ninguno

Tabla 33: Especificación de C.U. –Ver Plan de Pagos

Nombre:	Administrar Producto
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Producto	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Gerente-Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Producto” 2. El sistema le muestra lista de Producto y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 34: Especificación de C.U. – Administrar Producto

Nombre:	Adicionar Producto
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Producto	
Actores: Administrador	
Precondiciones: seleccionar Administrar Producto	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Producto en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Producto.	
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 35: Especificación de C.U. – Adicionar Producto

Nombre:	Modificar Producto
Fecha de creación:	08/05/2014
Descripción: Permite modificar Producto	
Actores: Administrador	
Precondiciones: seleccionar Administrar Producto	
Flujo Normal:	

1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Producto. 4. El actor modifica datos del Producto en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Producto.
Flujo Alternativo: - Si no selecciona una fila mensaje “seleccione un Producto” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 36: Especificación de C.U. – -Modificar Producto

Nombre:	Eliminar Producto
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Producto	
Actores: Administrador	
Precondiciones: seleccionar Administrar Producto	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Producto.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Producto”	

- Si presiona cancelar retorna a pantalla Administrar Productos.
Post condiciones: Ninguno

Tabla 37: Especificación de C.U. – Eliminar Producto

Nombre:	Administrar Almacén
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Almacén	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Almacén” 2. El sistema le muestra lista de Almacén y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: ninguno	

Tabla 38: Especificación de C.U. – Administrar Almacén

Nombre:	Adicionar Almacén
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Almacén	
Actores: Administrador	

Precondiciones: seleccionar Administrar Almacén
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Almacén en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “se guardó satisfactoriamente”. 8. Retorna a pantalla Administrar Almacén.
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje: ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 39: Especificación de C.U. – Adicionar Almacén

Nombre:	Modificar Almacén
Fecha de creación:	08/05/2014
Descripción: Permite modificar Almacén	
Actores: Administrador	
Precondiciones: seleccionar Administrar Almacén	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos de Almacén. 4. El actor modifica datos del Almacén en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Almacén.	

Flujo Alternativo:	
- Si no selecciona una fila mensaje: “seleccione Almacén” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”	
Post condiciones:	
Ninguno	

Tabla 40: Especificación de C.U. – Modificar Almacén

Nombre:	Eliminar Almacén
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Almacén	
Actores: Administrador	
Precondiciones: seleccionar Administrar Almacén	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Almacén.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione Almacén” - Si presiona cancelar retorna a pantalla Administrar Almacén.	
Post condiciones:	
Ninguno	

Tabla 41: Especificación de C.U. – Eliminar Almacén

Nombre:	Administrar Compras
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Compras	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Compras” 2. El sistema le muestra lista de Compras y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 42: Especificación de C.U. –Administrar Compras

Nombre:	Adicionar Compras
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Compras	
Actores: Administrador	
Precondiciones: seleccionar Administrar Compras	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario.	

3. El actor introduce datos de la Compra en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Compras.
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 43: Especificación de C.U. –Adicionar Compras

Nombre:	Administrar Proveedor
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Proveedor	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Proveedores” 2. El sistema le muestra lista de Proveedor y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: ninguno	

Tabla 44: Especificación de C.U. – Administrar Proveedor

Nombre:	Adicionar Proveedor
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Proveedor	
Actores: Administrador	
Precondiciones: seleccionar Administrar Proveedores	
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos del Proveedor en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “se guardó satisfactoriamente”. 8. Retorna a pantalla Administrar Proveedores.	
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje: ”introduzca bien los datos”	
Post condiciones: Ninguno	

Tabla 45: Especificación de C.U. – Adicionar Proveedor

Nombre:	Modificar Proveedor
Fecha de creación:	08/05/2014
Descripción: Permite modificar Proveedor	
Actores: Administrador	
Precondiciones: seleccionar Administrar Proveedor	
Flujo Normal:	

1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos de Proveedor. 4. El actor modifica datos del Proveedor en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Proveedores.
Flujo Alternativo: - Si no selecciona una fila mensaje: “seleccione un Proveedor” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 46: Especificación de C.U. –Modificar Proveedor

Nombre:	Eliminar Proveedor
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Proveedor	
Actores: Administrador	
Precondiciones: seleccionar Administrar Proveedor	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Proveedores.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Proveedor”	

- Si presiona cancelar retorna a pantalla Administrar Proveedor.
Pos condiciones: Ninguno

Tabla 47: Especificación de C.U. –Eliminar Proveedor

Nombre:	Administrar Estados
Fecha de creación:	08/05/2014
Descripción: Permite Administrar Estados	
Actores: Administrador	
Precondiciones: Estar logueado en el sistema como Administrador	
Flujo Normal: 1. El actor selecciona la opción “Administrar Estados” 2. El sistema le muestra lista de Estados y opciones.	
Flujo Alternativo: Ninguno	
Post condiciones: Ninguno	

Tabla 48: Especificación de C.U. – Administrar Estados

Nombre:	Adicionar Estado
Fecha de creación:	08/05/2014
Descripción: Permite adicionar Estado	
Actores: Administrador	

Precondiciones: seleccionar Administrar Estado
Flujo Normal: 1. El actor selecciona la opción adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos de los Estado en el formulario. 4. El actor guarda. 5. El sistema valida datos del formulario. 6. El sistema guarda datos. 7. Mensaje: “ se guardó satisfactoriamente” 8. Retorna a pantalla Administrar Estado.
Flujo Alternativo: - Si los datos introducidos en el formulario no son válidos el sistema le devuelve mensaje ”introduzca bien los datos”
Post condiciones: Ninguno

Tabla 49: Especificación de C.U. – Adicionar Estados

Nombre:	Modificar Estado
Fecha de creación:	08/05/2014
Descripción: Permite modificar Estado	
Actores: Administrador	
Precondiciones: seleccionar Administrar Estado	
Flujo Normal: 1. Selecciona una fila de la lista. 2. El actor selecciona la opción modificar. 3. El sistema le muestra formulario con datos del Estado. 4. El actor modifica datos del Estado en el formulario. 5. El actor guarda el formulario. 6. El sistema valida datos del formulario. 7. El sistema guarda datos. 8. Mensaje: “ se modificó satisfactoriamente” 9. Retorna a pantalla Administrar Estado.	

Flujo Alternativo:	
- Si no selecciona una fila mensaje “seleccione un Estado” - Si los datos introducidos en el formulario no son válidos el sistema le devuelve Mensaje: ”introduzca bien los datos”	
Post condiciones:	
Ninguno	

Tabla 50: Especificación de C.U. – Modificar Estados

Nombre:	Eliminar Estado
Fecha de creación:	08/05/2014
Descripción: Permite eliminar Estado	
Actores: Administrador	
Precondiciones: seleccionar Administrar Estado	
Flujo Normal: 1. Selecciona la fila a eliminar de la lista. 2. El actor selecciona la opción eliminar. 3. El sistema muestra mensaje: “¿Está seguro de eliminar?” 4. El actor selecciona “Aceptar” 5. El sistema elimina al id de la fila. 6. Mensaje “ se eliminó satisfactoriamente” 7. Retorna a pantalla Administrar Estado.	
Flujo Alternativo: - Si no selecciona una fila, mensaje; “seleccione un Estado” - Si presiona cancelar retorna a pantalla Administrar Estados.	
Post condiciones:	

Tabla 51: Especificación de C.U. – Eliminar Estados

Nombre:	Buscar Persona

Fecha de creación:	08/05/2014
Descripción: Permite buscar datos de personas	
Actores: Administrador, Vendedor(a)	
Precondiciones: Seleccionar la opción Adicionar (Clientes, Empleados y/o Usuarios)	
Flujo Normal: 2. El actor selecciona la opción “Buscar Persona” 5. El sistema muestra la lista de personas. 6. El actor selecciona a una persona. 7. El actor selecciona la opción “Aceptar”	
Flujo Alternativo: El actor cierra la ventana de Buscar Personas.	
Post condiciones: Ninguno	

Tabla 52: Especificación de C.U. – Buscar Persona

Nombre:	Reporte Recaudaciones
Fecha de creación:	08/05/2014
Descripción: Permite ver reporte Recaudaciones	
Actores: Administrador	
Precondiciones: Seleccionar la opción “Recaudaciones”	
Flujo Normal: 1. El actor selecciona la opción “Reporte Compras_Ventas Recaudaciones” 2. El actor selecciona el rango de fechas en la que desea el reporte. 3. El actor selecciona la opción “Aceptar” 4. El sistema visualiza el reporte requerido. 5. El actor imprime el reporte y sus datos.	
Flujo Alternativo: El actor cierra la ventana de Recaudaciones.	

Post condiciones: Ninguno

Tabla 53: Especificación de C.U. –Reporte Recaudaciones

Nombre:	Reporte Producto Almacenado
Fecha de creación:	08/05/2014
Descripción: Permite ver reporte Cliente	
Actores: Administrador	
Precondiciones: Seleccionar la opción “Producto Almacenado”	
Flujo Normal: 1. El actor selecciona la opción “Reporte Cliente” 2. El actor selecciona el rango de fechas en la que desea el reporte. 3. El actor selecciona la opción “Aceptar” 4. El sistema visualiza el reporte requerido. 5. El actor imprime el reporte y sus datos.	
Flujo Alternativo: El actor cierra la ventana de Producto Almacenado.	
Post condiciones: Ninguno	

Tabla 54: Especificación de C.U. – Producto Almacenado

Nombre:	Reporte Cantidad de Compras de Clientes
Fecha de creación:	08/05/2014
Descripción: Permite ver reporte Cantidad de Compras de Clientes	
Actores: Administrador	
Precondiciones: Seleccionar la opción “Cantidad de Compras de Clientes”	
Flujo Normal:	

1. El actor selecciona la opción “Cantidad de Compras de Clientes” 2. El actor selecciona el rango de fechas en la que desea el reporte. 3. El actor selecciona la opción “Aceptar” 4. El sistema visualiza el reporte requerido. 5. El actor imprime el reporte y sus datos.
Flujo Alternativo: El actor cierra la ventana de Cantidad de Compras de Clientes.
Post condiciones: Ninguno

Tabla 55: Especificación de C.U. – Cantidad de Compras de Clientes

Nombre:	Reporte Ventas Realizadas
Fecha de creación:	08/05/2014
Descripción: Permite ver reporte Venta Realizadas	
Actores: Administrador	
Precondiciones: Seleccionar la opción “Reporte Ventas Realizadas”	
Flujo Normal: 1. El actor selecciona la opción “Reporte Ventas Realizadas” 2. El actor selecciona el rango de fechas en la que desea el reporte. 3. El actor selecciona la opción “Aceptar” 4. El sistema visualiza el reporte requerido. 5. El actor imprime el reporte y sus datos.	
Flujo Alternativo: El actor cierra la ventana de Reporte Ventas Realizadas.	
Post condiciones: Ninguno	

Tabla 56: Especificación de C.U. –Reporte Ventas Realizadas

Nombre:	Registrar Dosificación
Fecha de creación:	08/05/2014

Descripción: Permite registrar la nueva dosificación
Actores: Administrador
Precondiciones: Seleccionar la opción “Registrar Dosificación”
Flujo Normal: 2. El actor selecciona la opción “Registrar Dosificación”. 3. Selecciona “nueva dosificación”. 4. El actor introduce los nuevos datos. 5. Si los datos son correctos, muestra mensaje: “Se adiciono correctamente”. 6. El actor presiona aceptar.
Flujo Alternativo: El actor cierra la ventana de Registrar Dosificación.
Post condiciones: Ninguno

Tabla 57: Especificación de C.U. –Reporte Ventas Realizadas

2.1.2.2.3.8 ESPECIFICACIONES ADICIONALES

2.1.2.2.3.8.1 PROPÓSITO

El propósito de este documento es determinar las especificaciones adicionales o requerimientos No Funcionales, que dan soporte a los procesos y requerimientos solicitados para el proyecto “Mejorar la administración de la información de Ventas del COMERCIAL FIGUEROA”.

2.1.2.2.3.8.2 ALCANCE

El presente documento tiene como fin desplegar el detalle de las Especificaciones Suplementarias o Requerimientos No Funcionales, de manera que no solo de claridad sobre el enfoque requerido para el planteamiento de la solución sino que establezca un alcance para el Software.

2.1.2.2.3.8.2.1 ESPECIFICACIÓN DE HARDWARE

Requerimientos mínimos de hardware para el funcionamiento del Sistema.

Para el Servidor:

- HP ProLiant ML350 G4p, Dual Option SCSI – Tower Server.
- Processor: Intel Xeon 3.40 GHz/800 MHz
- Disco Duro: 72.8 GB Pluggable Ultra320 SCSI 15.000 rpm.
- RAM: 512 MB REG PC23200 (1x512).

- UPS Sistema de alimentación ininterrumpida M/Tripplite.
- Monitor.
- Teclado.
- Mouse

Para el Cliente:

Requerimientos mínimos

- Equipo Pentium IV, con procesador de 2.4 GHz
- RAM 512 MB.
- Disco duro 80 GB.
- Monitor.
- Teclado.
- Mouse

2.1.2.2.3.8.2.2 ESPECIFICACIÓN DE SOFTWARE

PARA EL SERVIDOR:

- Sistema Operativo Windows 2003 Server o superior.
- Plataforma java JDK 1.6n.
- Gestor de Base de Datos MySql 5.5 para Windows.
- Servidor Apache tomcat 7 o superior para Windows.
- Firefox 9.0 o superior.
- Adobe Reader 7.0 o superior.

PARA EL CLIENTE:

- Sistema Operativo Windows XP o superior.
- Firefox 9.0 o superior
- Adobe Reader 7.0 o superior

2.1.2.2.3.8.3 USABILIDAD

El sistema será desarrollado en entorno amigable, para ser usado de una forma fácil y como complemento para manejo apropiado y fácil del sistema se plantea la realización de cursos de capacitación y realización de un texto de usuario como guía.

2.1.2.2.3.8.4 CONFIABILIDAD

El sistema deberá ser sometido a pruebas de software una vez concluido el desarrollo del mismo, la prueba a desarrollar y aplicar es la de caja negra (partición equivalente).

Al finalizar el proyecto por lo menos se deberá estimarse que un 80 a 90% de los errores han sido descubiertos.

2.1.2.2.3.8.5 FUNCIONALIDAD

Se hará uso de la norma 830 para la especificación de requisitos de software, la misma que permite ayudar a los clientes describir claramente que es lo que se desea con el nuevo software, lo cual también ayuda al desarrollador a entender que exactamente quiere el cliente para obtener o desarrollar un producto de acuerdo a los requerimientos requeridos.

2.1.2.2.3.8.6 SEGURIDAD

Integridad de Información

Para la confidencialidad y seguridad de la información se definen Permisos a los menús del sistema, teniendo cada usuario diferentes permisos, según el usuario de acceso que tenga.

Encriptación

Si en un futuro el sistema llega a ser implementado vía internet y como existe la probabilidad de que este esté amenazado, por lo tanto la clave y usuario de los actores del sistema, están encriptados mediante el algoritmo **BASE64** y encriptación de la conexión Cliente-Servidor, con del uso de certificados **SSL** sobre el protocolo **HTTPS**.

2.1.2.2.3.8.7 DISPONIBILIDAD

Para asegurar la disponibilidad del sistema se implanta el uso de variables de sesión, permitiendo el acceso a la información autorizada sin restricciones, una vez realizada la autenticidad del usuario.

Se recomienda realizar backups diariamente para evitar la pérdida fortuita de Información, como también el uso necesario del UPS, asegurando la disponibilidad al sistema.

2.1.2.2.3.8.8 MANTENIMIENTO

El sistema que se desarrolle debe contar con toda la documentación que explique y aclare su funcionamiento, como ser: Documentación con diagramas, especialmente el componente del sistema, de modo que cuando se requiera extender la operación del sistema, se pueda implementar fácilmente con un tiempo y un costo razonable.

2.1.2.2.3.8.9 PORTABILIDAD

El sistema será desarrollado en el lenguaje de programación java por lo cual es necesario la máquina virtual de java JDK 1.7x

2.1.2.2.3.8.10 EFICIENCIA

El sistema será desarrollado de modo que se tenga un tiempo de respuesta optimo, por lo cual se escogió la base de datos PostgreSQL que se caracteriza como una de las mejores y más rápidas base de datos relacionales que existe.

2.1.2.2.3.8.11 PROTOTIPO DE INTERFACE DE USUARIO

Prototipo que permite al usuario hacerse una idea más o menos exacta de las interfaces que proveerá el sistema y así, conseguir una idea de funcionamiento respecto al sistema. Este prototipo se realizara 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 del 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 a medida que se vayan desarrollando las iteraciones para conseguir el producto (software) final.

Pantalla Ingresar al Sistema:

ACCEDER



Usuario:

Clave:

Acceder

Figura 37: Pantalla-Ingresar al sistema

Pantalla Principal:



Figura 38: Pantalla-Principal

Pantalla Administrar Usuarios:

Usuario:grys

LISTA DE USUARIOS

Activos

Buscar

ID	USUARIO	CLAVE	ESTADO	OBSERVACION
1	talleriii	grys	Activo	nada
2	otro	123	Activo	ninguna
3	grys	666	Activo	ninguna
4	no34mn	3216548	Activo	asjdhas
6	dfgdf	dfgrdrdfg	Activo	dfgdfg

10

Pagina 1 de 1

Lista desde 1 a 5. De un total de 5

Adicionar

Modificar

Borrar

Figura 39: Pantalla-Administrar Usuarios

Pantalla Adicionar Usuarios:

ADICIONAR USUARIO

Los campos com * son necesarios

Datos Personales

✖ Datos de Persona

✖ Buscar Persona

Datos de Usuario

Nombre de Usuario:

*

Clave de Usuario :

*

Estado :

Cajero

Observaciones:

*

Cargar Imagen

Adicionar

Cancelar

Figura 40: Pantalla-Adicionar Usuarios

Pantalla Modificar Usuarios:

MODIFICAR USUARIO

Los campos con * son necesarios

Datos Personales
 Datos de Persona  Buscar Persona

Datos de Usuario
ID Usuario :
Nombre de Usuario:
Clave de Usuario :
Estado :
Estado :
Observaciones:
  Cargar Imagen

 Modificar  Cancelar

Figura 41: Pantalla-Modificar Usuarios

Pantalla Cambiar Datos de Usuario:

CAMBIAR DATOS USUARIO

Los campos con * son necesarios

Datos de Usuario
ID Usuario :
Nombre de Usuario:
Clave de Usuario :
Repita Clave : *

  Cargar Imagen

 Cambiar  Cancelar

Figura 42: Pantalla-Cambiar Datos Usuario

Pantalla Administrar Roles:

LISTA DE ROLES

Activos

Buscar

ID	NOMBRE ROL	ESTADO
1	Administrador	Activo
2	Vendedor	Activo
3	Cajero	Activo
4	vendedor1	Activo

10

Pagina 1 de 1

Lista desde 1 a 4. De un total de 4

Adicionar

Modificar

Borrar

Figura 43: Pantalla-Administrar Roles

Pantalla Adicionar Rol:

ADICIONAR ROL

Los campos con * son necesarios

Nombre :

*

Adicionar

Cancelar

Figura 44: Pantalla-Adicionar Rol

Pantalla Modificar Rol:

MODIFICAR ROL

Los campos con * son necesarios

ID Rol : 2

Nombre : Vendedor

Estado : Activo

✓ Modificar

✗ Cancelar

Figura 45: Pantalla-Modificar Rol

Pantalla Administrar Empleados:

Usuario:grys

LISTA DE EMPLEADOS

Activos

Buscar

ID	NOMBRE	A. PATERNO	A. MATERNO	FECHA DE INGRESO	CARGO	ESTADO	OBSERVACION
3	denis	silisqui	aramayo	2011-02-02	Administrador	Activo	NINGUNA
4	denis4	silisqui4	aramayo4	2005-01-02	Cajero	Activo	Ninguna
5	Maritza m.	valeriano	valeriano	2012-08-22	Cajero	Activo	cajero bjbjbjbjb

10

⏪ ⏩

Pagina 1 de 1

⏪ ⏩

Lista desde 1 a 3. De un total de 3

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 46: Pantalla-Administrar Empleados

Pantalla Adicionar Empleados:

ADICIONAR EMPLEADO

Los campos con * son necesarios

Datos Personales

✖ Datos de Persona

✖ Buscar Persona

Datos de Empleado

Fecha de Registro:

Tipo de Empleado :

nombreasd

Observaciones:

✓ Adicionar

✖ Cancelar

Figura 47: Pantalla-Adicionar Empleado

Pantalla Administrar Cargos:

LISTA DE CARGOS

Activos

🔍

Buscar

ID	NOMBRE CARGO	CARGA HORARIA	ESTADO
1	Administrador	de 7:00 a 20:00	Activo
2	Cajero	es un vsjero	Activo
4	nombre de cargo 125	medio dia	Activo
6	otro cargo 345	De Martes a Jueves	Activo
9	reemplazo	solo medio día	Activo

10

⏪ ⏩

Pagina 1 de 1

⏪ ⏩

Lista desde 1 a 5. De un total de 5

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 48: Pantalla-Administrar Cargos

Pantalla Adicionar Cargo:

ADICIONAR CARGO

Los campos con * son necesarios

Nombre del Cargo : *

Carga Horaria : *

☒ Adicionar ☒ Cancelar

Figura 49: Pantalla-Adicionar Cargo

Pantalla Modificar Cargo:

MODIFICAR CARGO

Los campos con * son necesarios

ID Cargo :

Nombre del Cargo :

Carga Horaria :

Estado :

☒ Modificar ☒ Cancelar

Figura 50: Pantalla-Modificar Cargo

Pantalla Administrar Clientes:

Usuario:grys

LISTA DE CLIENTES

Activos

Buscar

ID	CI	NOMBRE	A. PATERNO	A. MATERNO	FECHA DE NACIMIENTO	TELEFONO	ESTADO	OBSERVACION
6638081	grys	arayo	aramayo6		1987-07-21	6635216	Activo	ninguna
6638083	ratl	martinez	aramayo8		1987-07-21	6635216	Activo	Ninguna fdg
6638082	soledad	mastinez	aramayo7		1987-07-21	6635216	Activo	kajshfd askjdhas
6639075	Griselda	Gallardo	Aparicio		1987-04-15	6352954	Activo	jajajajajaj por que3

10

⏮ ⏪ ⏩ ⏭

Pagina 1 de 1

Lista desde 1 a 4. De un total de 4

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 51: Pantalla-Administrar Clientes

Pantalla Adicionar Cliente:

ADICIONAR CLIENTE

Los campos con * son necesarios

Datos Personales

 Datos de Persona

 Buscar Persona

CI:

Nombre(s) : *

Apellido Paterno : *

Apellido Materno : *

Fecha de Nacimiento : *

sexo:

Estado Civil :

Telefono : *

Direccion actual : *

Descripcion : *

Datos de Cliente

Fecha de Registro: *

Tipo de Cliente :

Observaciones:

 Adicionar

 Cancelar

Figura 52: Pantalla-Adicionar Cliente

Pantalla Modificar Cliente:

MODIFICAR CLIENTE

Los campos con * son necesarios

Datos Personales

 Datos de Persona

 Buscar Persona

CI: 6639075

Nombre(s): hilda

Apellido Paterno: Perez

Apellido Materno: Acedebo

Fecha de Nacimiento: 1987-04-15

sexo: Femenino

Estado Civil: Soltero(a)

Telefono: 6352954

Direccion actual: sin numero

Descripcion: no tiene

Datos de Cliente

ID Cliente: 5

Fecha de Registro: 2012-08-21

Tipo de Cliente: tipo2

Estado: Activo

Observaciones: no hay

 Modificar

 Cancelar

Figura 53: Pantalla-Modificar Cliente

Pantalla Administrar Tipos de Clientes:

Usuario:grys

LISTA DE TIPOS DE CLIENTE

Activos

Buscar

ID	TIPO CLIENTE	ESTADO
1	tipo1	Activo
2	tipo2	Activo
3	Cliente De Oro	Activo

10

Pagina 1 de 1

Lista desde 1 a 3. De un total de 3

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 54: Pantalla-Administrar Tipos de Clientes

Pantalla Adicionar Tipo de Cliente:

Usuario:grys

ADICIONAR TIPO CLIENTE

Los campos con * son necesarios

Nombre Tipo : *

✓ Adicionar

✗ Cancelar

Figura 55: Pantalla-Adicionar Tipos de Clientes

Pantalla Modificar Tipo de Cliente:

MODIFICAR ROL

Los campos con * son necesarios

ID Rol :

Nombre :

Estado :

☒ Modificar ☒ Cancelar

Figura 56: Pantalla-Modificar Tipos de Clientes

Pantalla Administrar Ventas:

LISTA DE VENTAS

Todos

ID	FECHA	MONTO TOTAL	CLIENTE	USUARIO	Tipo de Venta	ESTADO	OBSERVACION
1	2014-12-07	180	Iner Maria	grys	Contado	Activo	
2	2014-12-07	260	Abraham Josue	grys	Contado	Activo	
3	2014-12-07	260	Veronica Ines	grys	Contado	Activo	
4	2014-12-07	1775	Abraham Josue	grys	Contado	Activo	
5	2014-12-10	355	Abraham Josue	grys	Credito	Activo	tjujk

10 de 1

Lista desde 1 a 5. De un total de 5

☒ Adicionar ☒ Modificar ☒ Borrar ☒ Ver Plan de Pagos

Figura 57: Pantalla-Administrar Ventas

Pantalla Adicionar Ventas:

NUEVA VENTA

Los campos con * son necesarios

Datos de venta

ID Venta :

Venta al:

Seleccione

Cliente

CI

NIT

Apellidos y Nombres

Telefono

Buscar Cliente

Cantidad de productos:

Almacen

Producto

Precio

Cantidad

Costo

Monto Total

(Bs.)

Observaciones

Adicionar

Cancelar

Figura 58: Pantalla-Adicionar Venta

Registrar Dosificación:

ID	FECHA DE EXPIRACION	N. AUT.	FACTURAS EMITIDAS
1	2015-10-10	563456	36

10

⏮ ⏪

Pagina de 1

⏩ ⏭

Lista desde 1 a 1. De un total de 1

Adicionar

Figura 59: Pantalla-Registrar Dosificación

Lista de Reportes:



Figura 60: Pantalla-Lista de Reportes

Pantalla Ver Plan de Pagos:

PLAN DE PAGOS						
ID	NRO. PAGO	FECHA PAGO	MONTO A PAGAR	MULTA A PAGAR	ESTADO	OBSERVACION
13	1		107.5		Debe	
14	2		107.5		Debe	
 Pagar  Volver a Lista de Ventas						

Figura 61: Pantalla-Ver Plan de Pagos

Pantalla Administrar Tipo de Venta:

LISTA DE TIPOS DE VENTA

Activos

 Buscar

ID	TIPO DE VENTA	ESTADO
1	Contado	Activo
3	Credito no reembolsable	Activo
2	Credito	Activo

10

Pagina 1

de 1

Lista desde 1 a 3. De un total de 3

 Adicionar

 Modificar

 Borrar

Figura 62: Pantalla-Administrar Tipo de Venta

Pantalla Adicionar Tipo de Venta:

ADICIONAR TIPO DE VENTA

Los campos con * son necesarios

Nombre Tipo : *

☒ Adicionar ☒ Cancelar

Figura 63: Pantalla-Adicionar Tipo de Venta

Pantalla Modificar Tipo de Venta:

MODIFICAR TIPO DE VENTA

Los campos con * son necesarios

ID Tipo Venta :

Nombre Tipo :

Estado :

☒ Modificar ☒ Cancelar

Figura 64: Pantalla-Modificar Tipo de Venta

Pantalla Administrar Productos:

LISTA DE PRODUCTOS

Activos

Buscar

ID	IMAGEN	NOMBRE	MARCA	CATEGORIA
1		pro1	marca1	cat1
2		pro2	marca1	cat1
3		pro3	marca3	cat2
4		pro4	marca4	cat3
5		producto23	marca1	cat1
6		producto asd	marca4	cat2
7		asdwsdf	marca234	cat1
8		andador	marca3	cat1

10

⏪ ⏩

Pagina 1 de 1

⏪ ⏩

Lista desde 1 a 8. De un total de 8

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 65: Pantalla-Administrar Productos

Pantalla Adicionar Producto:

ADICIONAR PRODUCTO

Los campos con * son necesarios

Nombre del Producto : *

Marca del Producto : *  Marca

Categoria del Producto : *  Categoria

Color del Producto : *

Tamaño del Producto :

Precio Unitario del Producto :

Precio Oferta del Producto :

Multa del Producto (%) :

Descuento del Producto :

Observacion :

 Cargar Imagen

 Adicionar  Cancelar

Figura 66: Pantalla- Adicionar Producto

Pantalla Modificar Producto:

MODIFICAR PRODUCTO

Los campos con * son necesarios

ID Producto : 6

Nombre del Producto : celular

Marca del Producto : marca4

Categoría del Producto : cat2

Color del Producto : negro

Tamaño del Producto : mediano

Precio Unitario del Producto : 12.6

Precio Oferta del Producto : 0.0

Multa del Producto (%): 2.0

Descuento del Producto : 3.0

Estado : Activo

Observacion : jashdasd

+ Marca

+ Categoría



 Cargar Imagen

✓ Modificar

✗ Cancelar

Figura 67: Pantalla- Modificar Producto

Pantalla Administrar Almacenes:

Activos

INGRESE EL NOMBRE DEL ALMACEN A BUSCAR

ID	NOMBRE ALMACEN	DIRECCION	ESTADO	OBSERVACION
2	A1	direccion 1 all	Activo	ninguna
3	A2	Direccion nro 2	Activo	alskjdhajkl asldaksjhdaks asldkajsdlkajs
5	A4	Direccion nro 4	Activo	lkjsldfnsiohsdiknsf34 asdw
12	A7	direccion nro 7	Activo	ningunaningunasd
22	B8	direccion nro 13B	Activo	ningningunaunad34 ghf
26	C1	direccion nro 17C	Activo	ninningsunagunad asdas
27	C2	direccion nro 18C	Activo	ninguningunana asfasfd
29	C4	direccion nro 20C	Activo	ningningunauna
30	C5	direccion nro 21C	Activo	ninguna
31	C6	direccion nro 22C	Activo	ningningunauna

10

⏮ ⏪ ⏩ ⏭

Pagina 1 de 7

Lista desde 1 a 10. De un total de 68

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 68: Pantalla-Administrar Almacenes

Pantalla Adicionar Almacén:

ADICIONAR ALMACEN

Los campos com * son necesarios

Nombre del Almacen :

*

Direccion :

*

Observaciones:

*

✓ Adicionar

✗ Cancelar

Figura 69: Pantalla-Adicionar Almacén

Pantalla Modificar Almacén:

MODIFICAR ALMACEN

Los campos com * son necesarios

ID Almacen : 12

Nombre del Almacen : A7

Direccion : direccion nro 7

Observaciones : ningunaningunasd

Estado : Activo

✓ Modificar

✗ Cancelar

Figura 70: Pantalla-Adicionar Almacén

Pantalla Administrar Compras:

LISTA DE COMPRAS

Activo

Buscar

ID	FECHA	MONTO TOTAL	ID PROVEEDOR	NOMBRE	TELEFONO	ESTADO	OBSERVACION
1	2012-11-05	212.01	prove1		6678432	Activo	asdwe werf sdf sdf
2	2014-10-14	100.01	prove1		6678432	Activo	En oferta
3	2014-11-11	350.01	prove1		6678432	Activo	ytjtt

10

⏪ ⏩

Pagina 1 de 1

⏪ ⏩

Lista desde 1 a 3. De un total de 3

✓ Adicionar

✎ Modificar

🗑 Borrar

Figura 71: Pantalla-Administrar Compras

Pantalla Adicionar compras:

NUEVA COMPRA

Los campos con * son necesarios

Datos de Compra
ID Compra : 0 en Fecha 2014-12-13

Datos de Proveedor  Buscar Proveedor
NIT 123456789 Nombre prove1
E-mail cdiego@entelnet.com Telefono 6678432

Cantidad de productos:
5

Almacen	Producto	Precio	Cantidad	Costo	
A2	Telefono FT988LS	260.0	1	260	X
A2	Sauna SF871	355.0	1	355	X
A3	Procesador CP4239	360.0	1	360	X
A4	Hervidor HA760	186.0	1	186	X
A2	Telefono FT988LS	260.0	1	260	X
Monto Total				1421	(Bs.)

Observaciones
prueba.....

 Adicionar  Cancelar

Figura 72: Pantalla-Adicionar Compras

Pantalla Adicionar Proveedores:

ADICIONAR PROVEEDOR

Los campos con * son necesarios

Nit : *

Nombre : *

Correo Electronico : *

Cuenta Bancaria : 0

Telefono : *

Fax: *

Ciudad : *

Direccion : *

Observaciones : *

✓ Adicionar

✗ Cancelar

Figura 73: Pantalla-Adicionar Proveedores

Pantalla Modificar Proveedores:

MODIFICAR PROVEEDOR

Los campos con * son necesarios

ID Proveedor : 1

Nit : 123456789

Nombre : prove1

Correo Electronico : prove1_12@gmail.com

Cuenta Bancaria : 123456

Telefono : 6678432

Fax: 2346543

Ciudad : Tarija

Direccion : No tiehne

Estado : Activo ▾

Observaciones : ningunaningunayh

✓ Modificar

✗ Cancelar

Figura 74: Pantalla-Modificar Proveedores

Pantalla Administrar Estados:

LISTA DE ESTADOS

Activos

Buscar

ID	NOMBRE DE ESTADO	ESTADO	DESCRIPCION
A	Activo	Activo	activo
I	Inactivo	Activo	inactivo
PE	PrimerEstado15	Activo	es otro estado

10

Pagina 1 de 1

Lista desde 1 a 3. De un total de 3

Adicionar

Modificar

Borrar

Figura 75: Pantalla- Administrar Estados

Pantalla Adicionar Estados:

ADICIONAR ESTADO

Los campos con * son necesarios

ID Estado :

*

Nombre del Estado :

*

Descripcion :

*

Adicionar

Cancelar

Figura 76: Pantalla- Adicionar Estados

Pantalla Modificar Estados:

MODIFICAR ESTADO

Los campos con * son necesarios

ID Estado : TO

Nombre del Estado : Todos

Descripcion : un estado general
para mostrar todos los datos

Estado : Activo

✓ Modificar

✗ Cancelar

Figura 77: Pantalla- Modificar Estados

Pantalla Buscar Persona:

Busqueda de Persona

Buscar Persona

ID PERSONA	CI	NOMBRE	A. PATERNO	A. MATERNO	FECHA DE NACIMIENTO	TELEFONO	SEXO	ESTADO CIVIL	DIRECCION	DESCRIPCION
7	6638081	denis6	silisqui6	aramayo6	1987-07-21	6635216	M	S	no tiene 6	nada 6
8	6638082	denis	silisqui	aramayo	1987-07-21	6635216	M	C	no tiene 7	nada 7
9	6638085	denis8	silisqui8	aramayo8	1987-07-21	6635216	M	V	no tiene 8	nada 8
17	6639075	hilda	akumay	akumano	1987-04-15	6352954	F	S	sin numero	no tuetn

Aceptar

Figura 78: Pantalla-Buscar Persona

Reporte Compras de Clientes (cantidad):



[Nombre de la Institucion] NombreInstitucion

Fecha : 19/09/2014

Hora : 10:03:39 AM

Pagina: 1 de 1

LISTADO DE CLIENTES

Desde: 02/01/2011 Hasta: 02/11/2012

ID	CI	APELLIDOS Y NOMBRES	TIPO	F. REGISTRO	ESTADO	NRO COMPRAS
1	6638081	arayo aramayo6 grys	tipo1	23-jun-11	Activo	1
2	6638083	martinez aramayo8 ratl	tipo1	28-ene-11	Activo	1
4	6638082	mastinez aramayo7 soledad	tipo1	20-ago-12	Activo	0
5	6639075	Gallardo Aparicio Griselda	tipo2	21-ago-12	Activo	0

Figura 79: Reporte de Clientes

Reporte Compras-Ventas (recaudaciones):

ID VENTA: 33	TIPO DE VENTA: Contado	EN FECHA: 14/12/2014			
USUARIO : grys	ESTADO DE LA VENTA: Pagado				
CI: 1032578	CLIENTE: Gallardo Aparicio Hernan	NIT: null			
DETALLE					
ID PRO.	NOMBRE PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL	
MONTO TOTAL: 1800Bs					

ID VENTA: 32	TIPO DE VENTA: Contado	EN FECHA: 14/12/2014			
USUARIO : grys	ESTADO DE LA VENTA: Pagado				
CI: 4124124	CLIENTE: Benitez Montero Ludmila	NIT: null			
DETALLE					
ID PRO.	NOMBRE PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL	
MONTO TOTAL: 265Bs					

ID VENTA: 31	TIPO DE VENTA: Contado	EN FECHA: 14/12/2014			
USUARIO : grys	ESTADO DE LA VENTA: Pagado				
CI: 10663971	CLIENTE: Cayani Rivero Abraham Josue	NIT: null			
DETALLE					
ID PRO.	NOMBRE PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL	
MONTO TOTAL: 355Bs					

ID VENTA: 30	TIPO DE VENTA: Contado	EN FECHA: 14/12/2014
USUARIO : grys	ESTADO DE LA VENTA: Pagado	
CI: 4124124	CLIENTE: Benitez Montero Ludmila	NIT: null
DETALLE		

Figura 80: Reporte Compras-Ventas (recaudaciones)

Reporte Producto Almacenado:

PRODUCTO ALMACENADO							
ALMACEN		A1		1			
ID_PRODUCTO	NOMBRE	MARCA	CATEGORIA	PRECIO	COLOR	TAMAÑO	CANTIDAD
10	Hervidor HA760	Ufesa	Electrodomestico	186.0	blanco	mediano	100
7	Hervidor	Magafesa	Electrodomestico	265.0	blanco	mediano	100
6	Procesador CP4239	Magafesa	Electrodomestico	360.0	Platino	mediano	100
3	Telefono TG3721	Panasonic	Dispositivo	280.0	Negro	mediano	100
ALMACEN		A2		2			
ID_PRODUCTO	NOMBRE	MARCA	CATEGORIA	PRECIO	COLOR	TAMAÑO	CANTIDAD
9	Sandwichera	Ufesa	Electrodomestico	390.0	Platino	mediano	100
1	Telefono FT988LS	Panasonic	Dispositivo	260.0	negro	normal	100
8	Sauna SF871	Ufesa	Electrodomestico	355.0	Platino	mediano	100
ALMACEN		A3		3			
ID_PRODUCTO	NOMBRE	MARCA	CATEGORIA	PRECIO	COLOR	TAMAÑO	CANTIDAD
2	Telefono TS500WN	Panasonic	Dispositivo	200.0	blanco	normal	100
8	Sauna SF871	Ufesa	Electrodomestico	355.0	Platino	mediano	100
6	Procesador CP4239	Magafesa	Electrodomestico	360.0	Platino	mediano	100
9	Sandwichera	Ufesa	Electrodomestico	390.0	Platino	mediano	100

Figura 81: Reporte Producto Almacenado

Reporte Ventas:



COMERCIAL "FIGUEROA"

Fecha : 15/12/2014

Hora : 1:46:15 PM

Pagina: 1 de 9

LISTA DE VENTAS

Desde: 01/01/2014 Hasta: 31/12/2014

ID VENTA: 36	TIPO DE VENTA: Credito	EN FECHA: 14/12/2014		
USUARIO : grys	ESTADO DE LA VENTA: Pagado			
CI: 5012457	CLIENTE: Valdez Mendoza Veronica Ines	NIT: null		
DETALLE				
ID PRO.	NOMBRE PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL
9	Sandwichera	1	390.0	390.0
6	Procesador CP4239	1	360.0	360.0
7	Hervidor	1	265.0	265.0
8	Sauna SF871	1	355.0	355.0
10	Hervidor HA760	1	186.0	186.0
8	Sauna SF871	1	355.0	355.0
8	Sauna SF871	1	355.0	355.0
9	Sandwichera	1	390.0	390.0
MONTO TOTAL: 2656Bs				

Figura 82: Reporte de Ventas

2.1.2.2.3.1 MODELO DE ANÁLISIS Y DISEÑO

2.1.2.2.3.1.1 DIAGRAMA DE ACTIVIDADES

Diagrama de actividad: C.U. Ingresar al Sistema

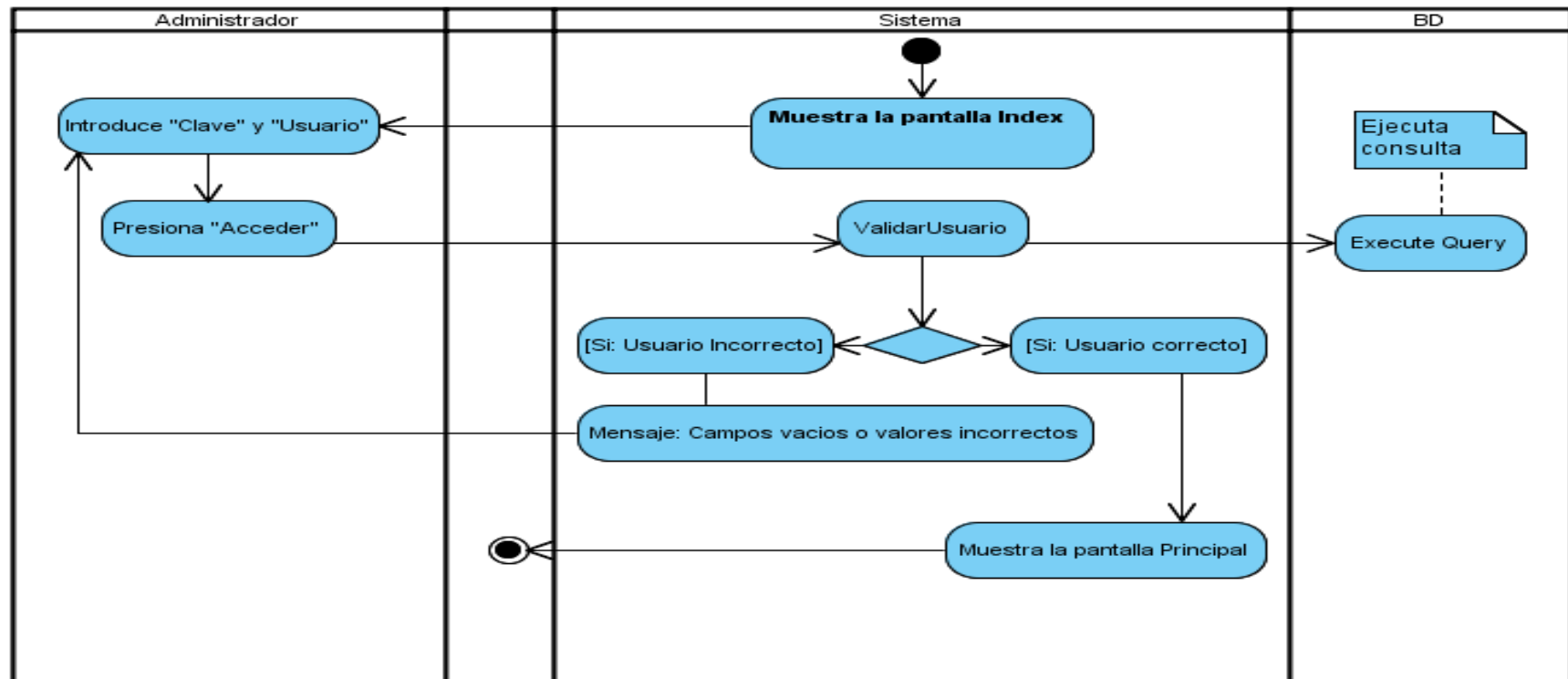


Figura 83: Diagrama de Actividades-Ingresar al Sistema

Diagrama de actividad: C.U. Administrar Usuario

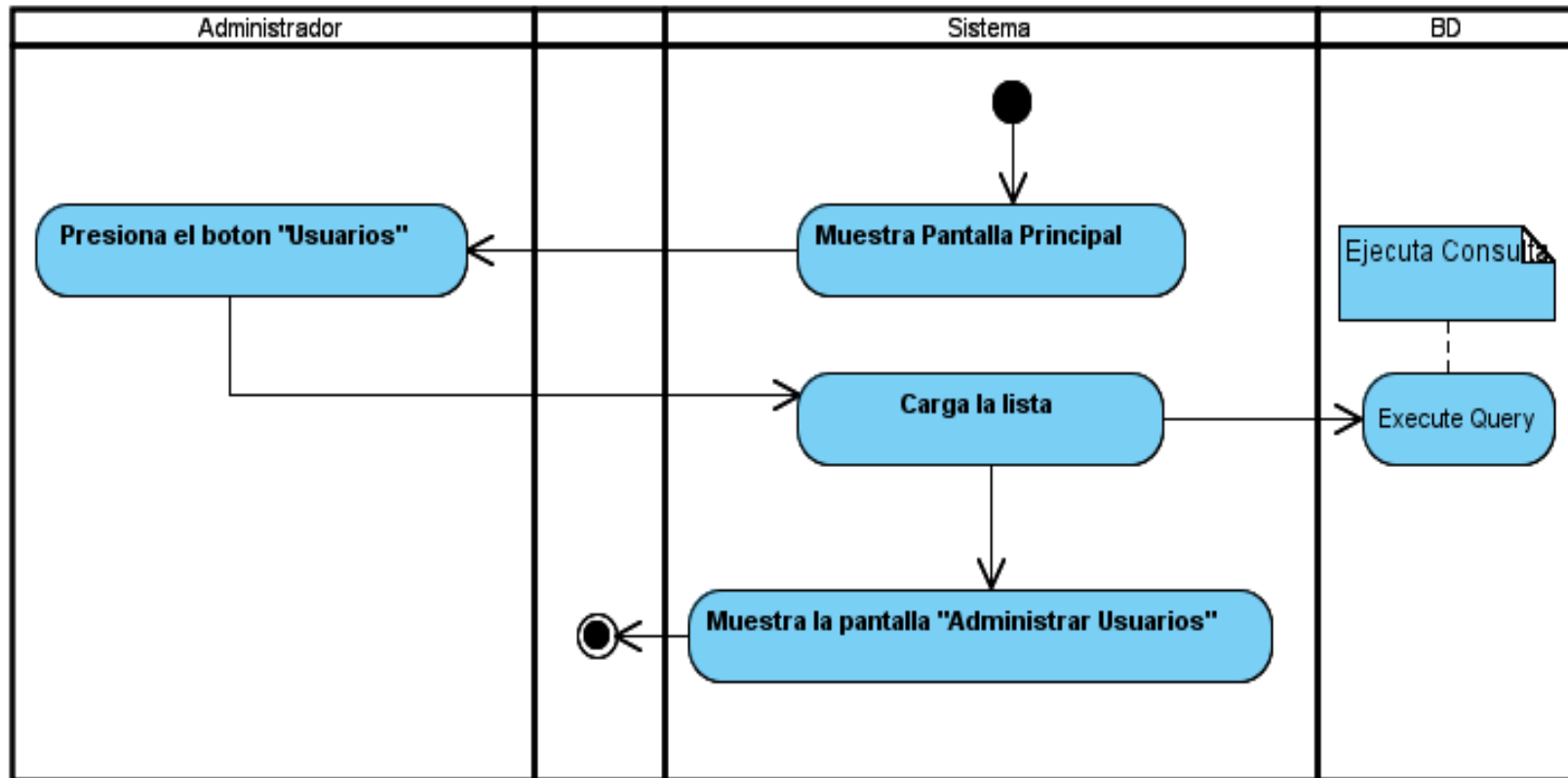


Figura 84: Diagrama de Actividades-Administrar Usuario

Diagrama de actividad: C.U. Adicionar Usuario

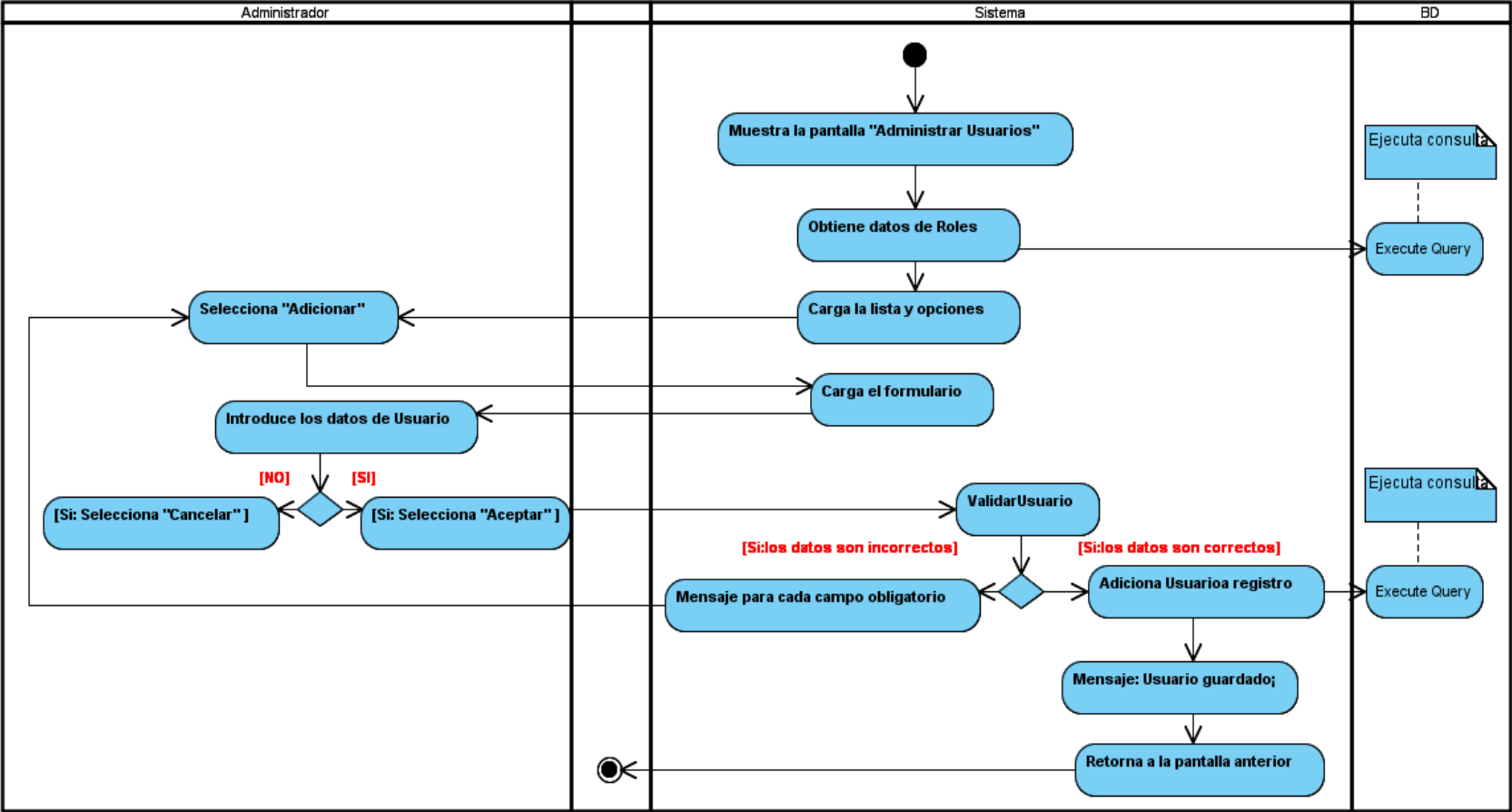


Figura 85: Diagrama de Actividades-Adicionar Usuario

Diagrama de actividad: C.U. Modificar Usuario

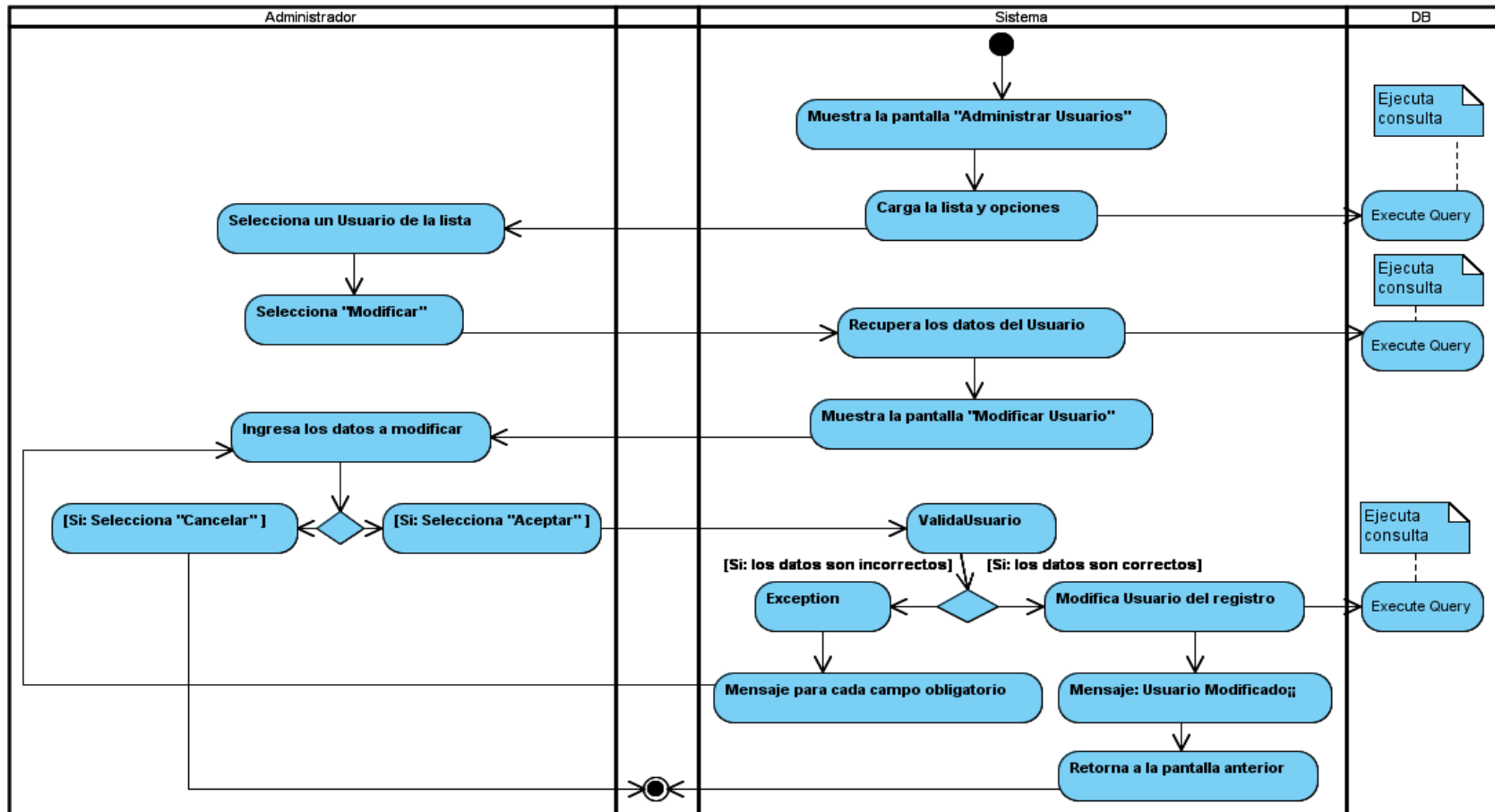


Figura 86: Diagrama de Actividades-Modificar Usuario

Diagrama de actividad: C.U. Eliminar Usuario

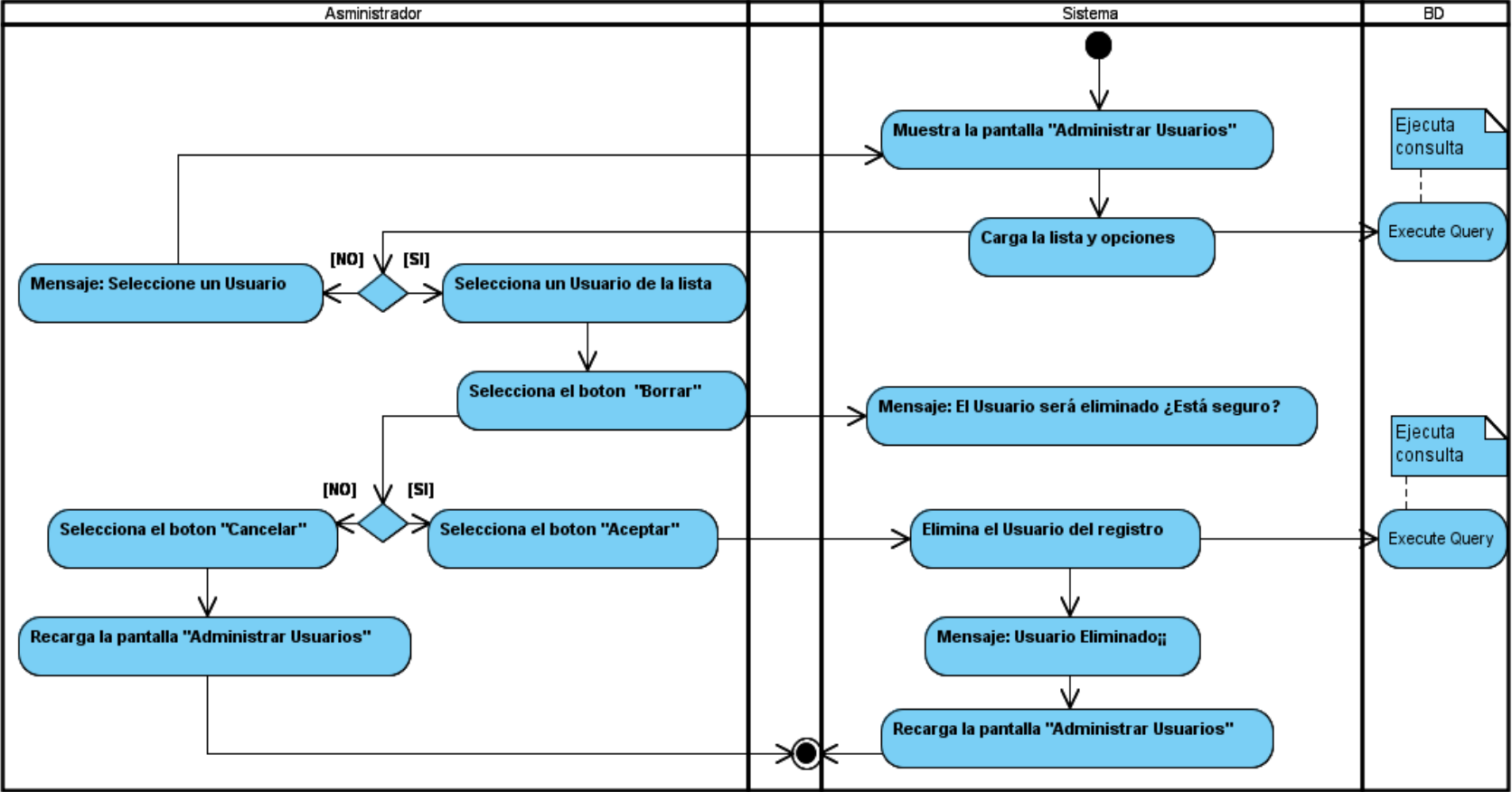


Figura 87: Diagrama de Actividades-Eliminar Usuario

Diagrama de actividad: C.U. Administrar Roles

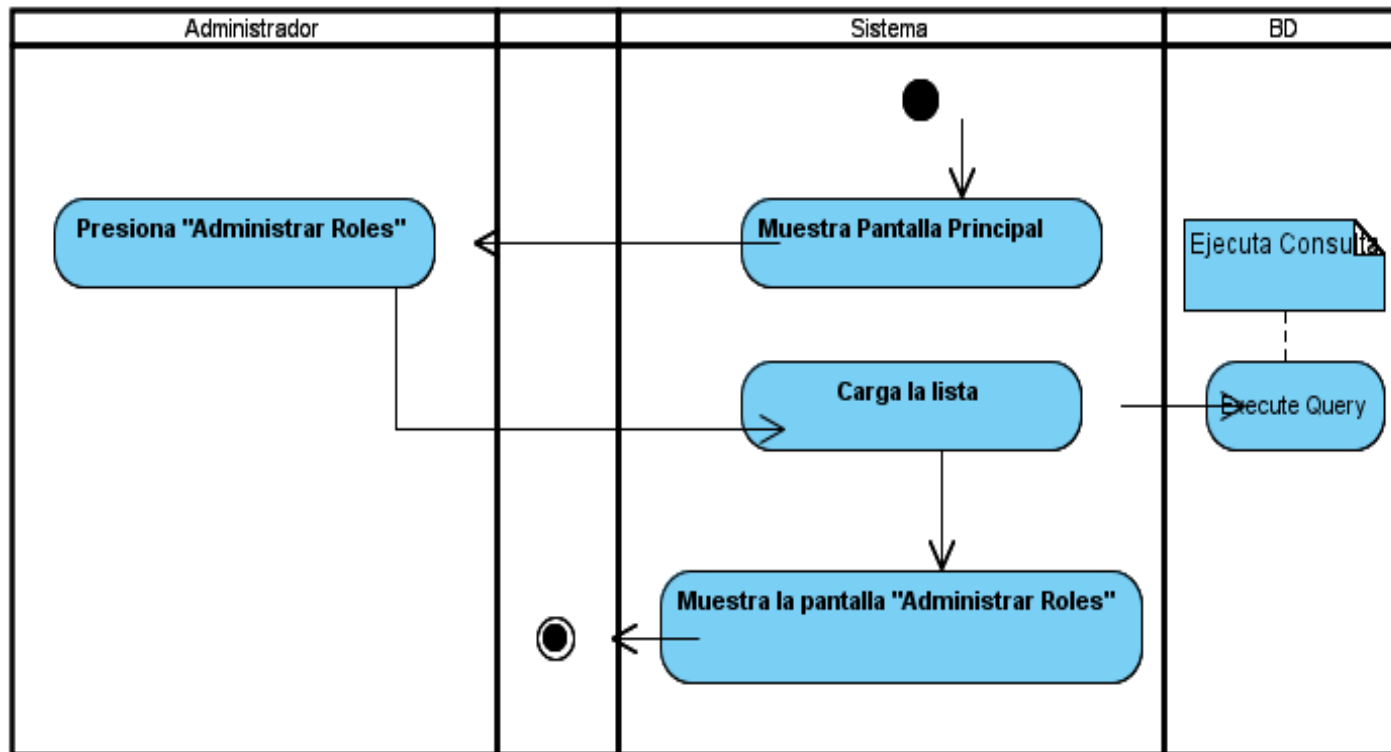


Figura 88: Diagrama de Actividades-Administrar Roles

Diagrama de actividad: C.U. Adicionar Rol

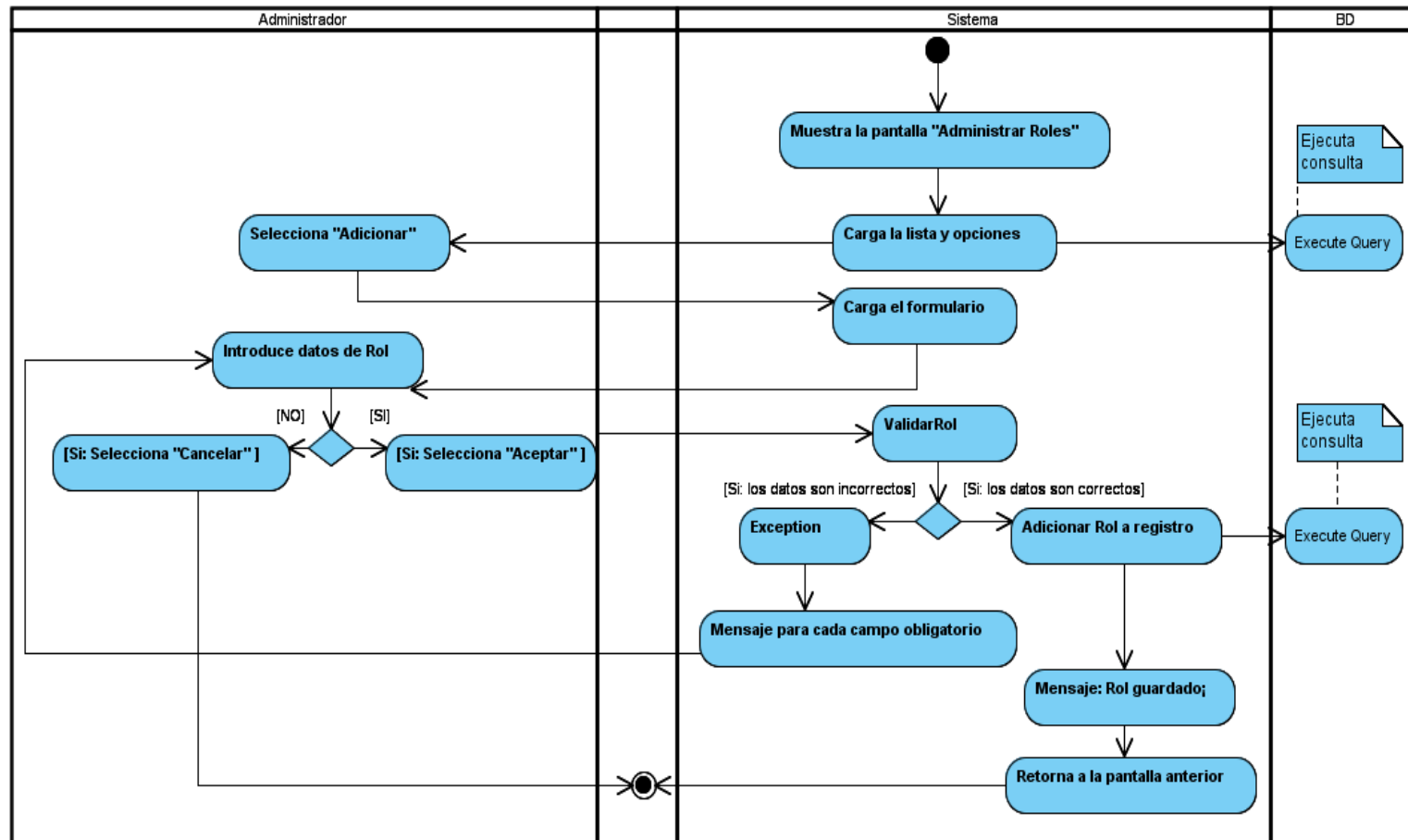


Figura 89: Diagrama de Actividades-Adicionar Rol

Diagrama de actividad: C.U. Modificar Rol

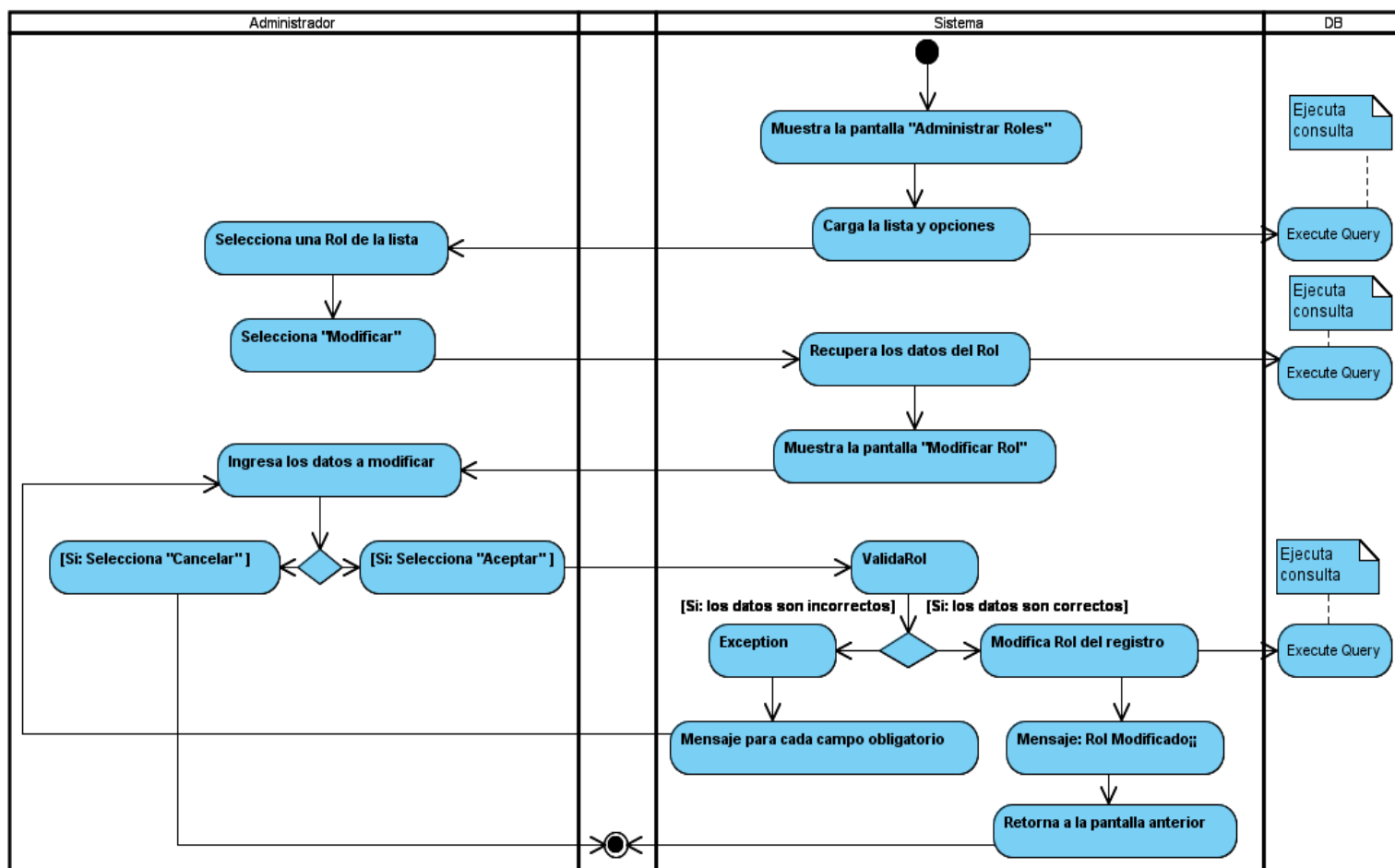


Figura 90: Diagrama de Actividades-Modificar Rol

Diagrama de actividad: C.U. Eliminar Rol

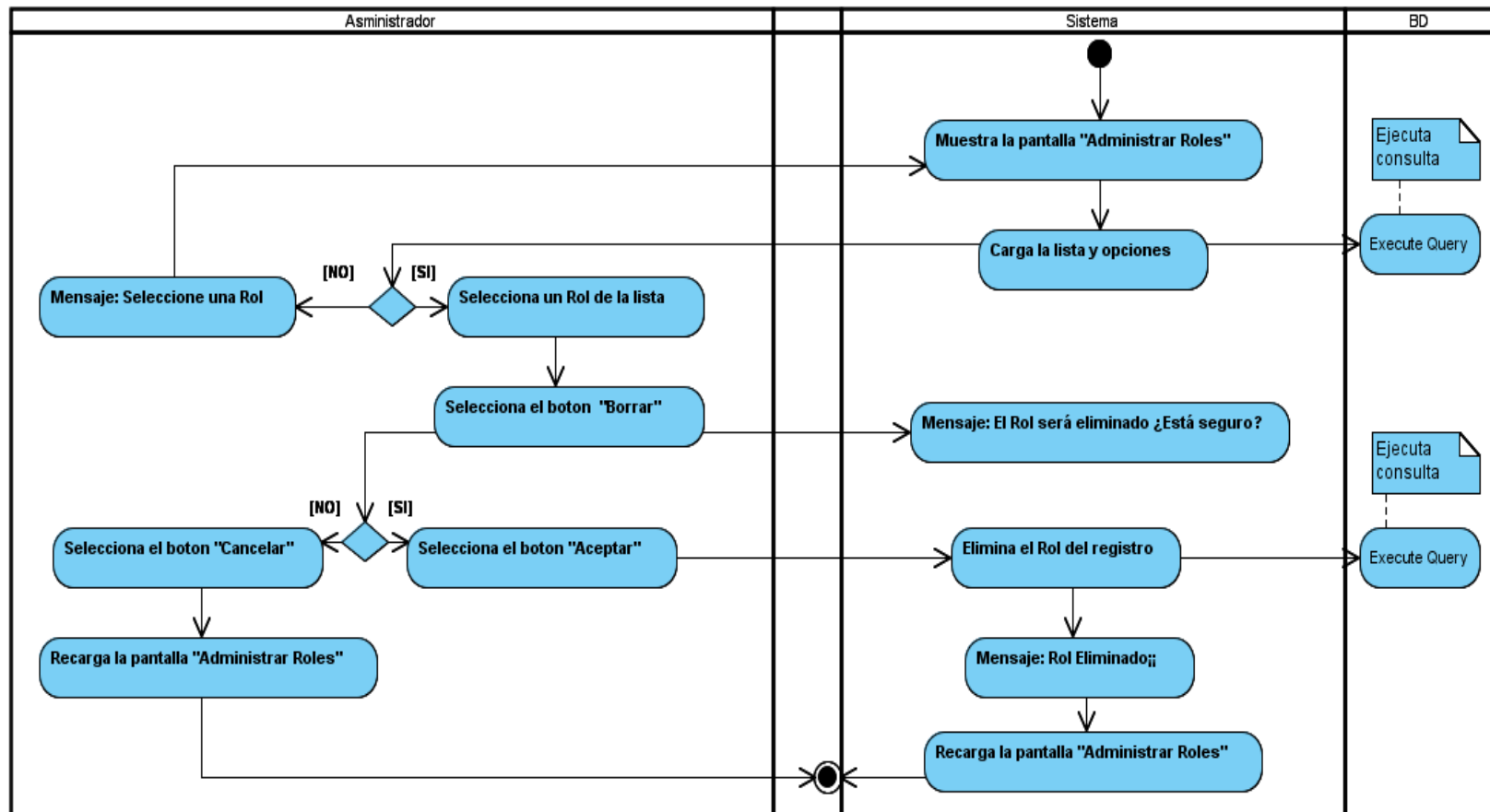


Figura 91: Diagrama de Actividades-Eliminar Rol

Diagrama de actividad: C.U. Administrar Empleados

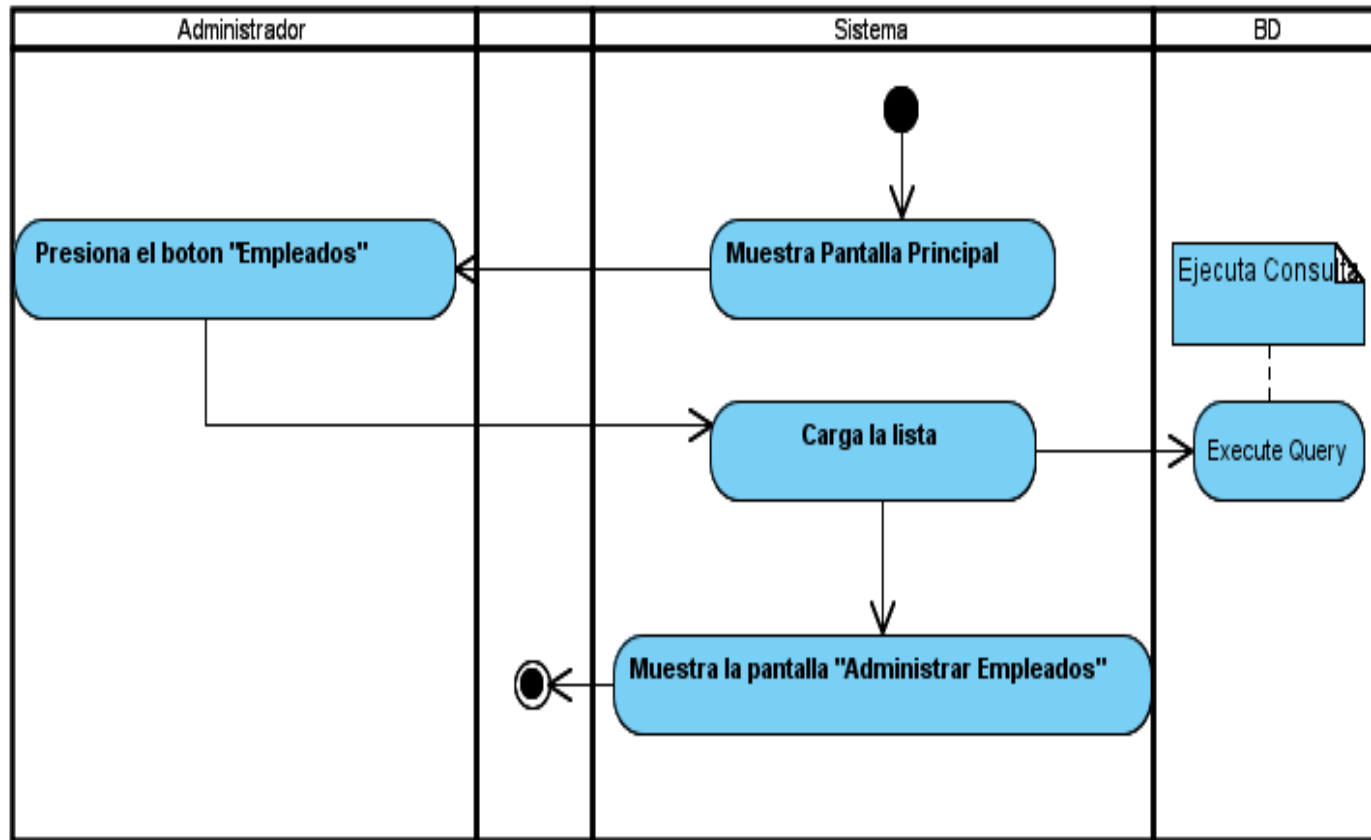


Figura 92: Diagrama de Actividades-Administrar Empleados

Diagrama de actividad: C.U. Adicionar Empleado

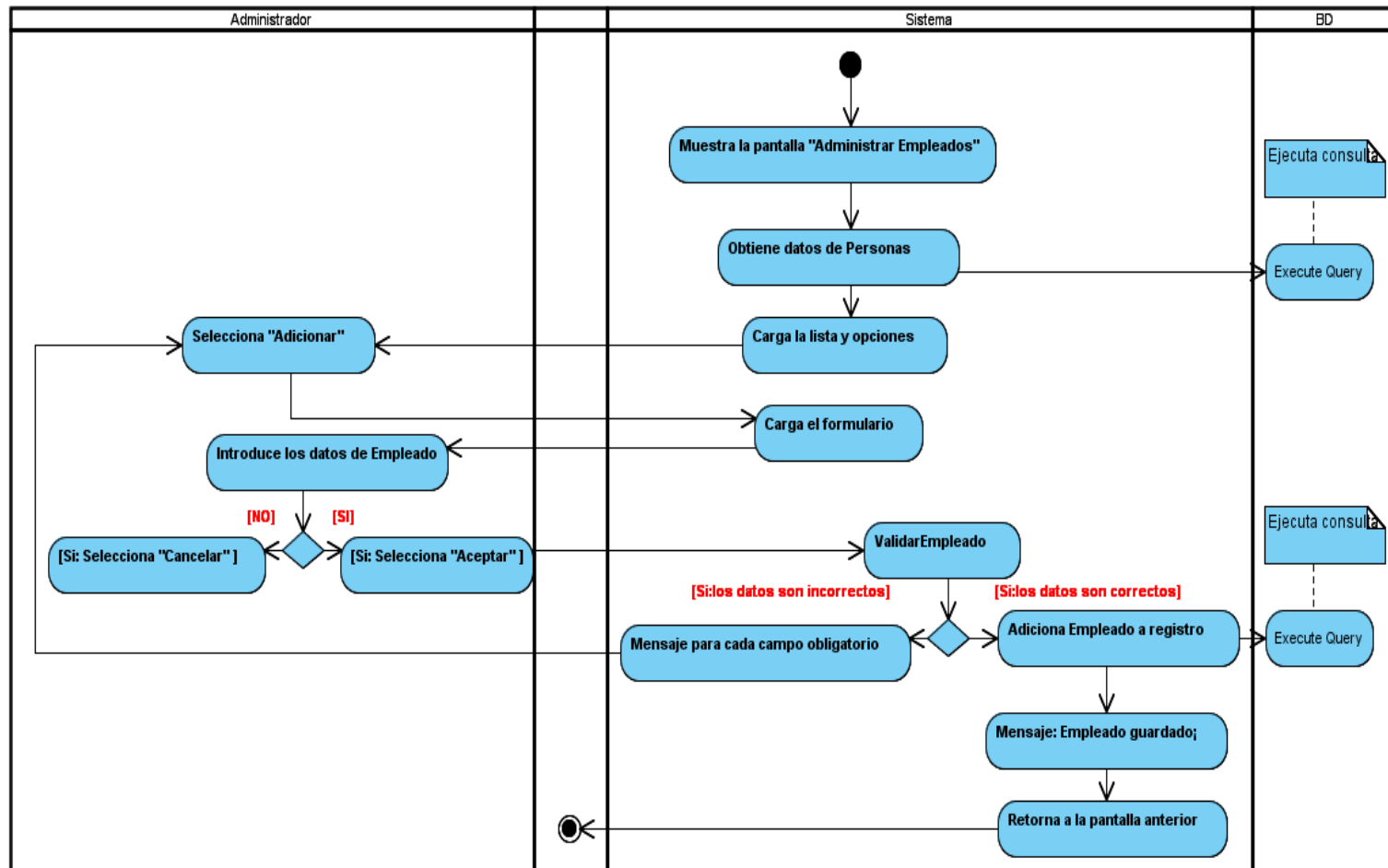


Figura 93: Diagrama de Actividades-Adicionar Empleado

Diagrama de actividad: C.U. Modificar Empleado

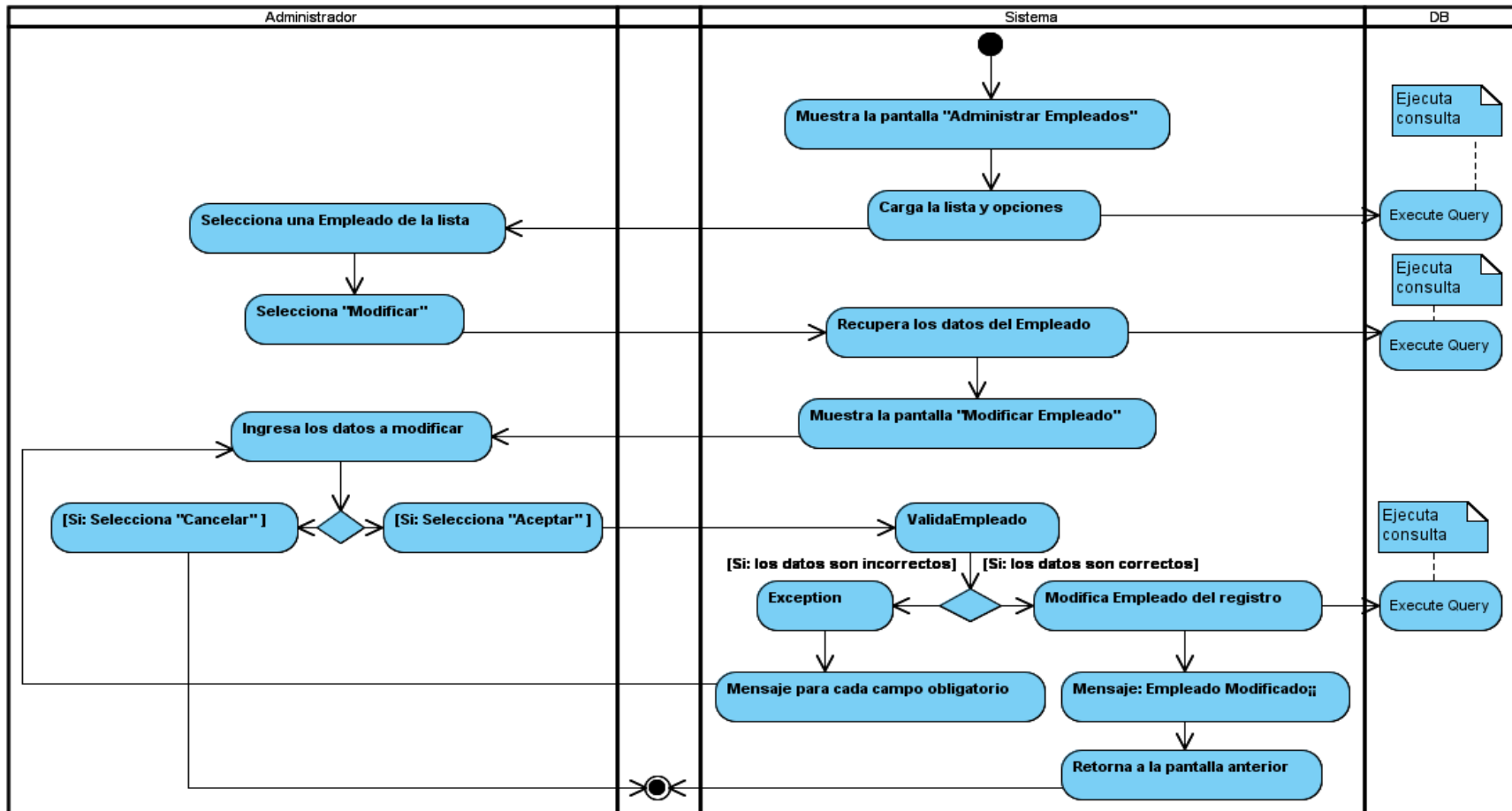


Figura 94: Diagrama de Actividades-Modificar Empleado

Diagrama de actividad: C.U. Eliminar Empleado

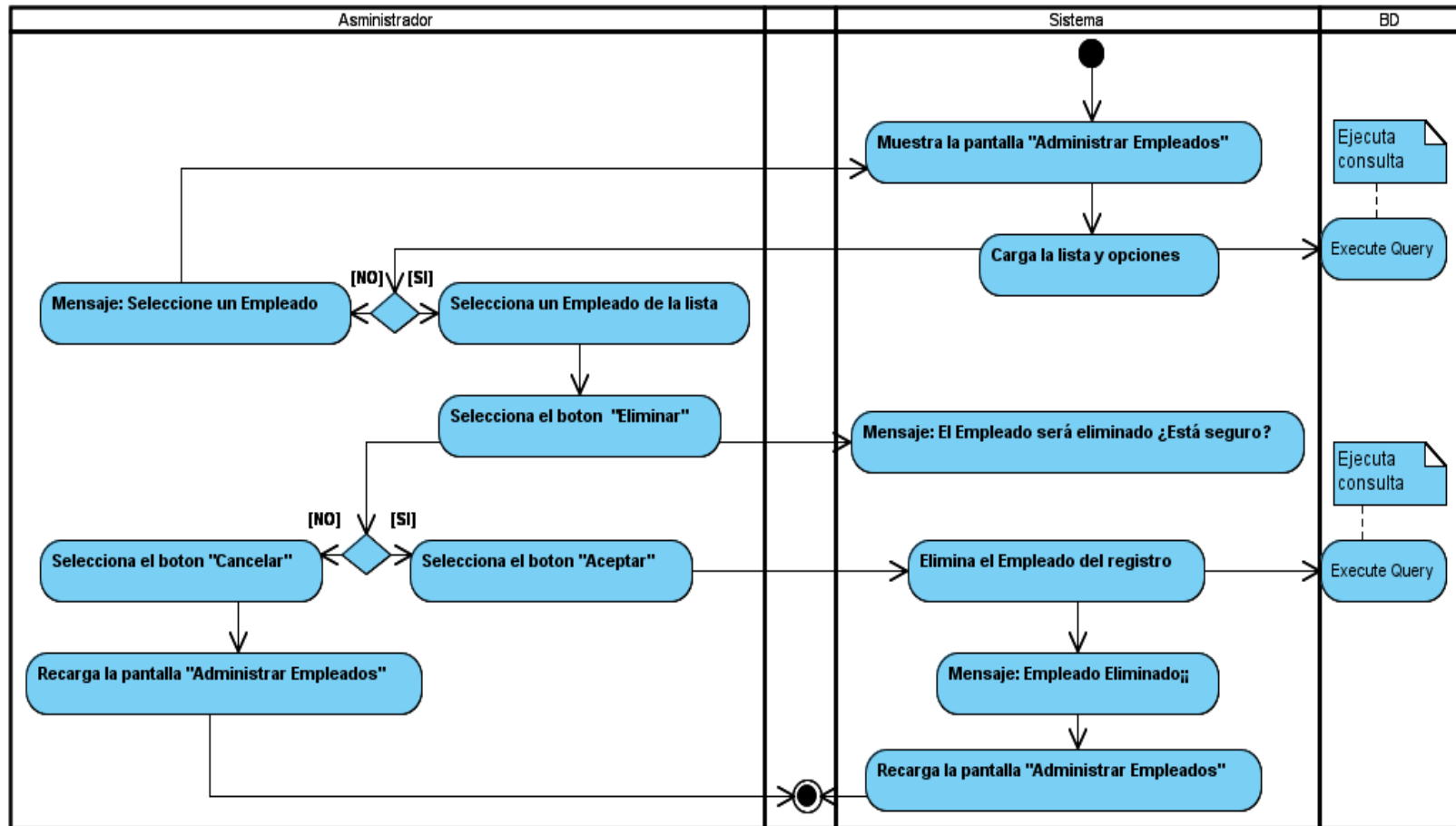


Figura 95: Diagrama de Actividades-Eliminar Empleado

Diagrama de actividades: C.U. Administrar Cargo

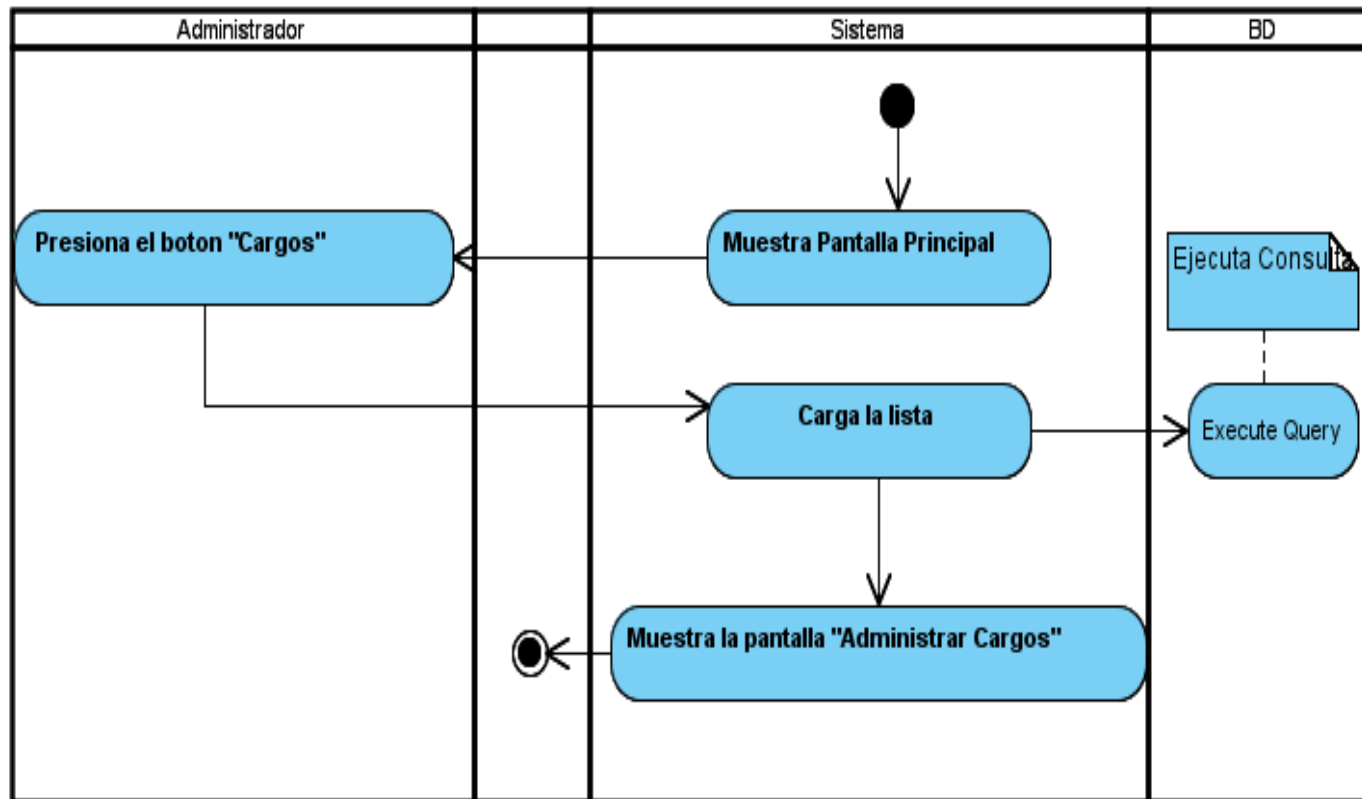


Figura 96: Diagrama de Actividades-Administrar Cargos

Diagrama de actividades: C.U. Adicionar Cargo

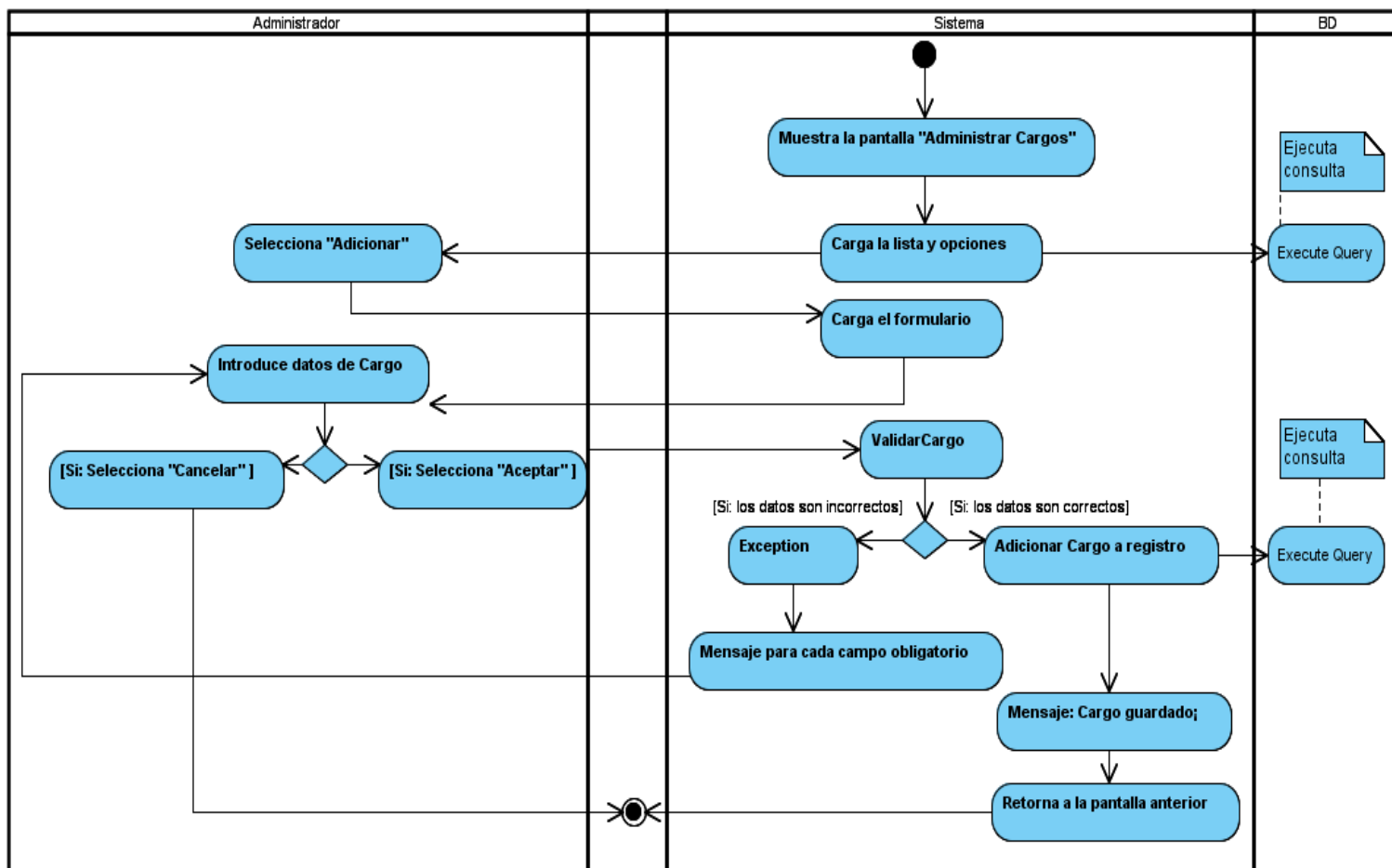


Figura 97: Diagrama de Actividades-Adicionar Cargo

Diagrama de actividades: C.U. Modificar Cargo

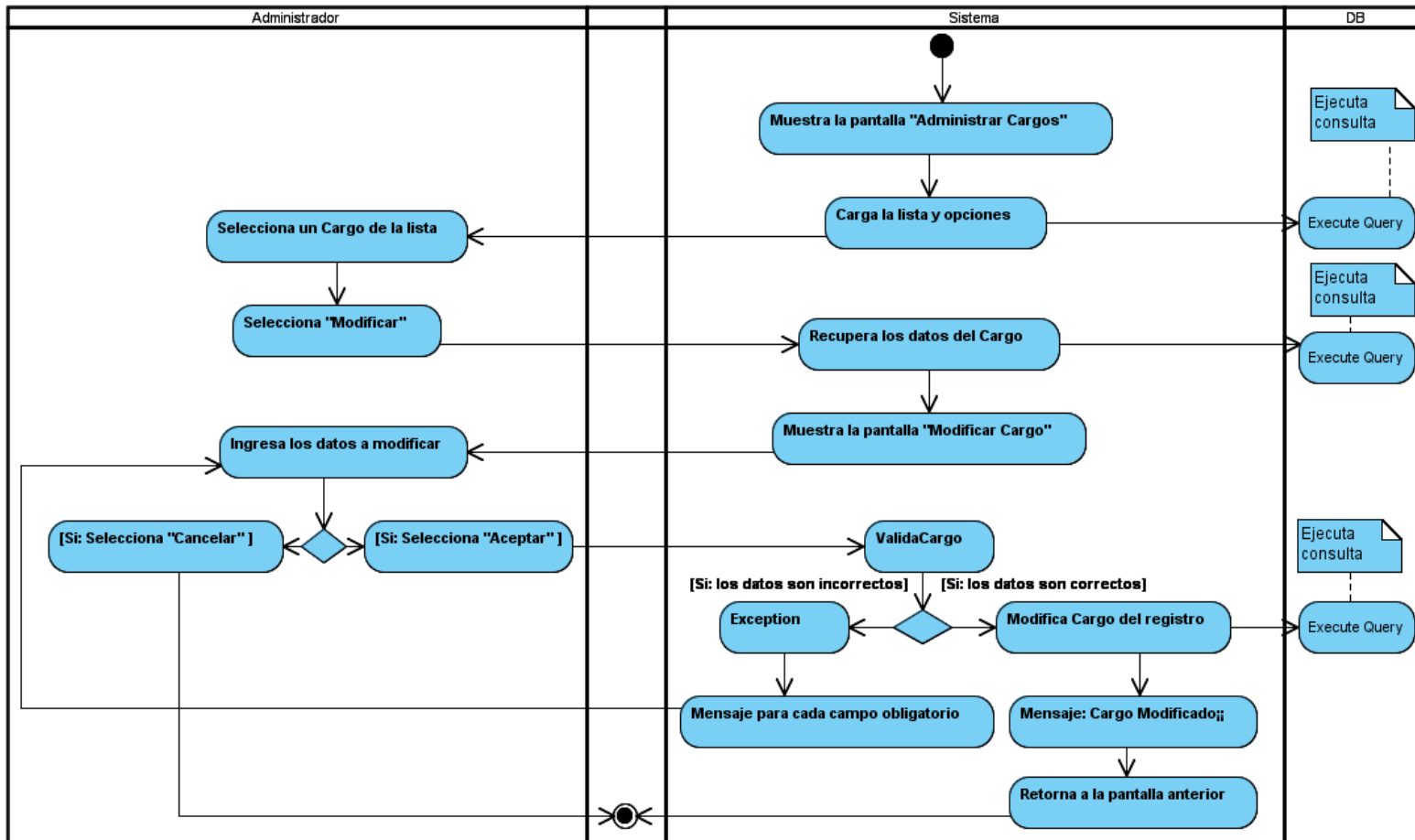


Figura 98: Diagrama de Actividades-Modificar Cargo

Diagrama de actividades: C.U. Eliminar Cargo

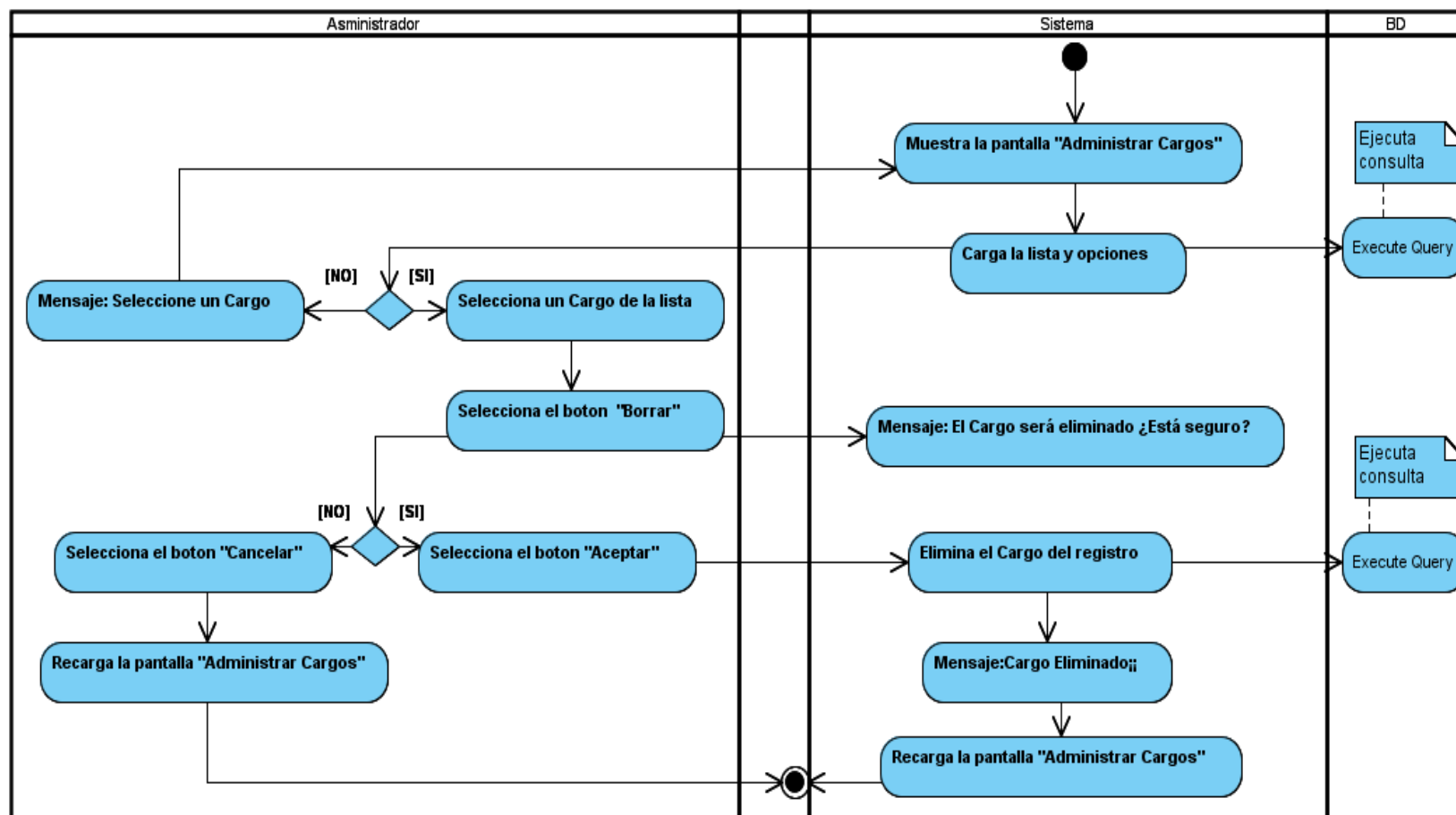


Figura 99: Diagrama de Actividades-Eliminar Cargo

Diagrama de actividad: C.U. Administrar Cliente

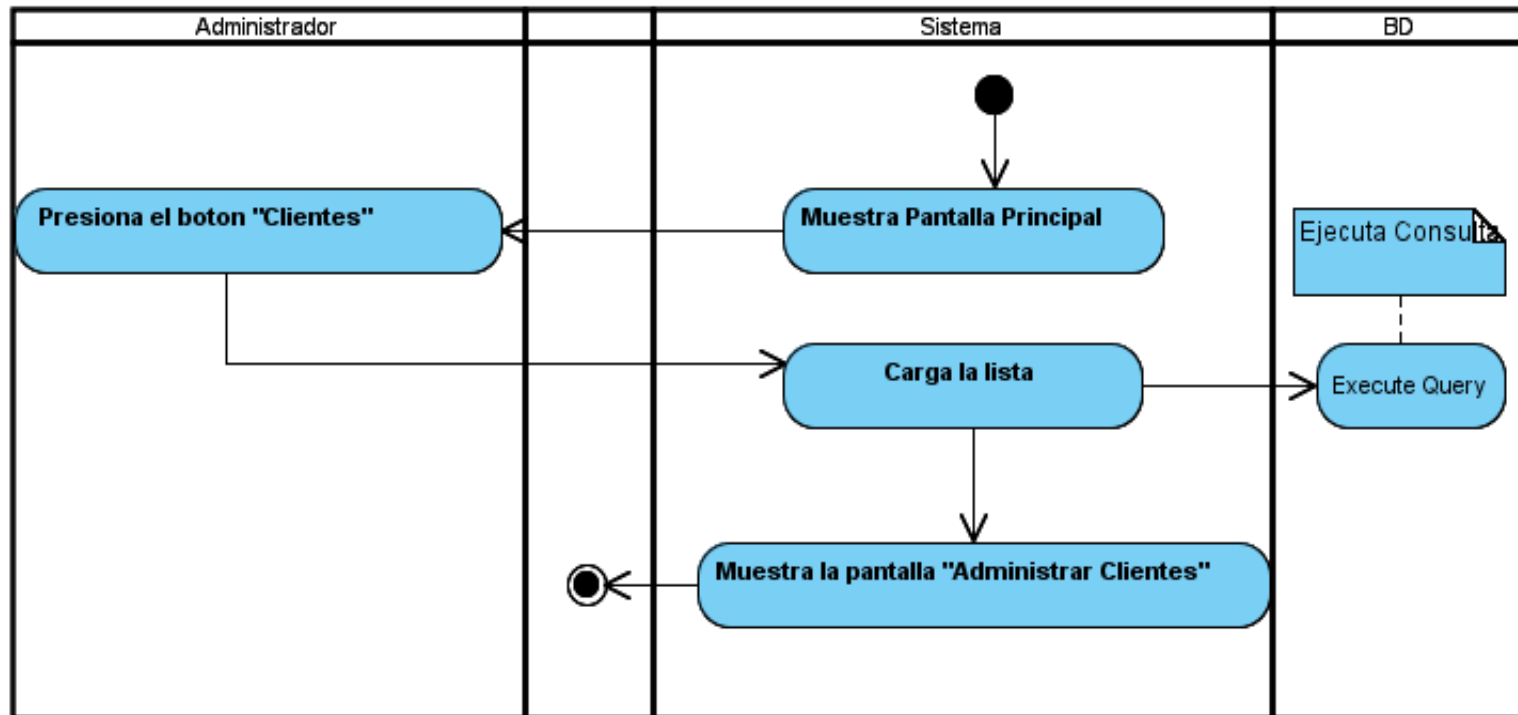


Figura 100: Diagrama de Actividades-Administrar Clientes

Diagrama de actividad: C.U. Adicionar Cliente

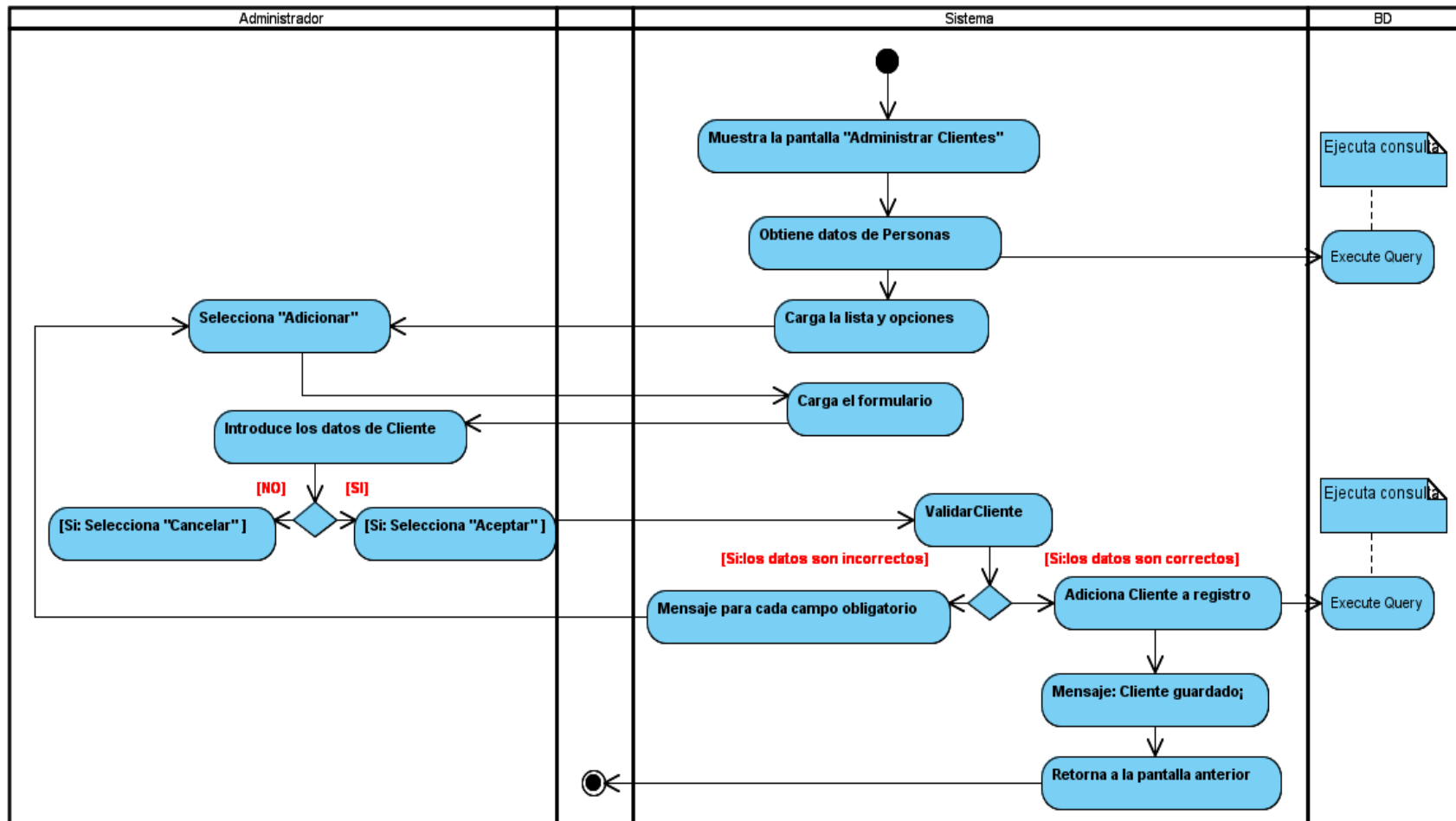


Figura 101: Diagrama de Actividades-Adicionar Cliente

Diagrama de actividad: C.U. Modificar Cliente

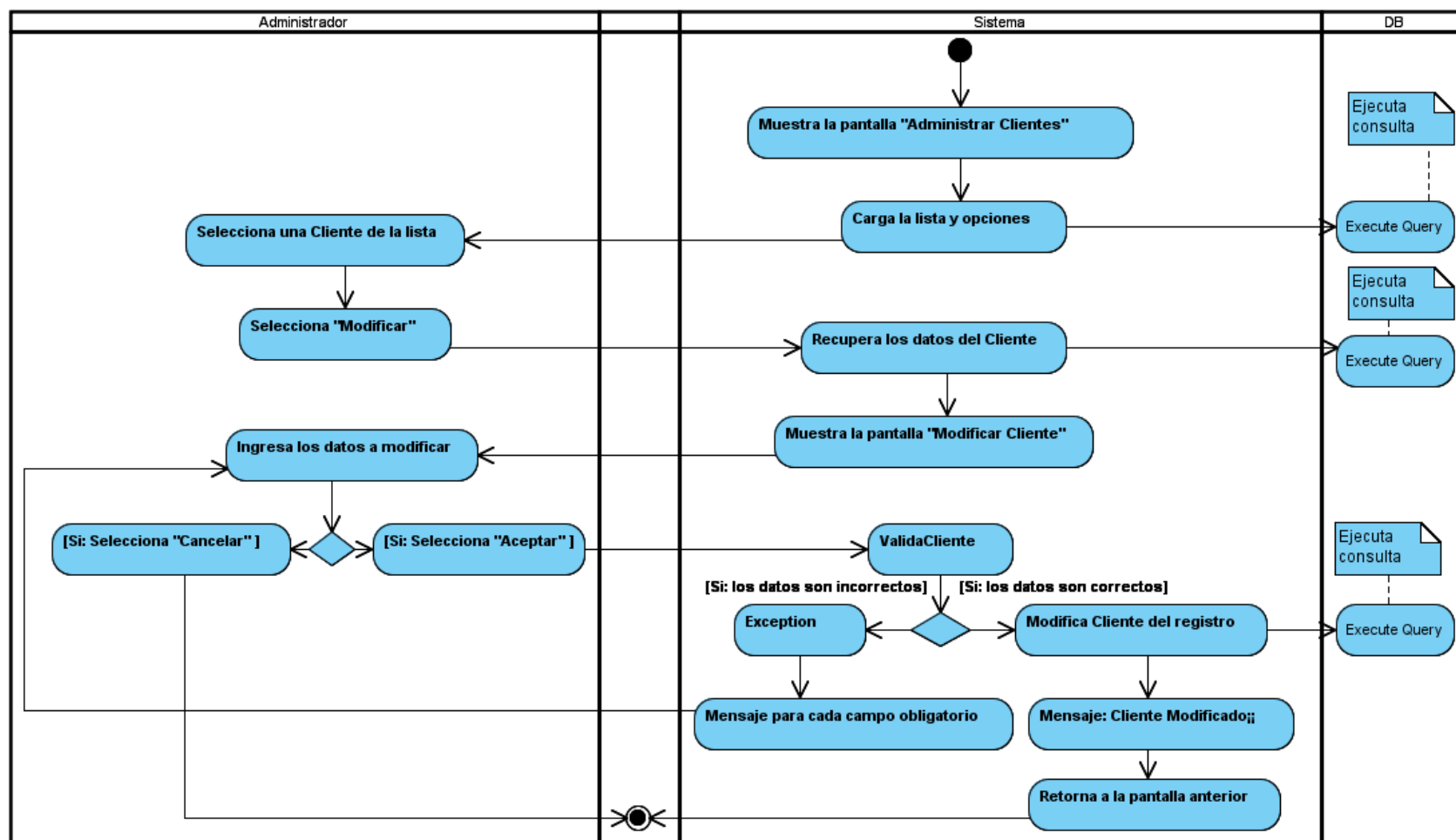


Figura 102: Diagrama de Actividades-Modificar Cliente

Diagrama de actividad: C.U. Eliminar Cliente

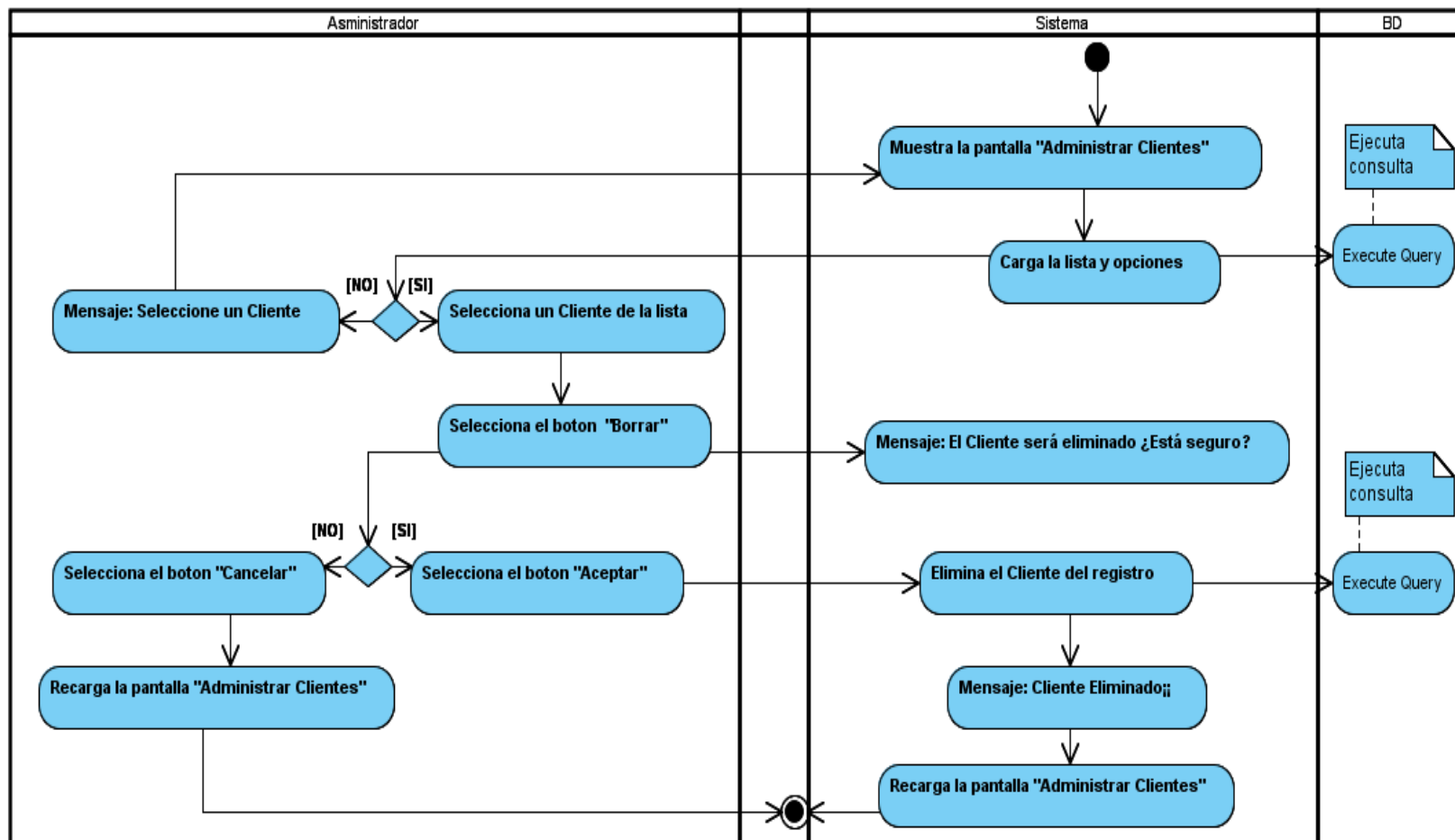


Figura 103: Diagrama de Actividades-Eliminar Cliente

Diagrama de actividad: C.U. Adicionar Tipo de Cliente

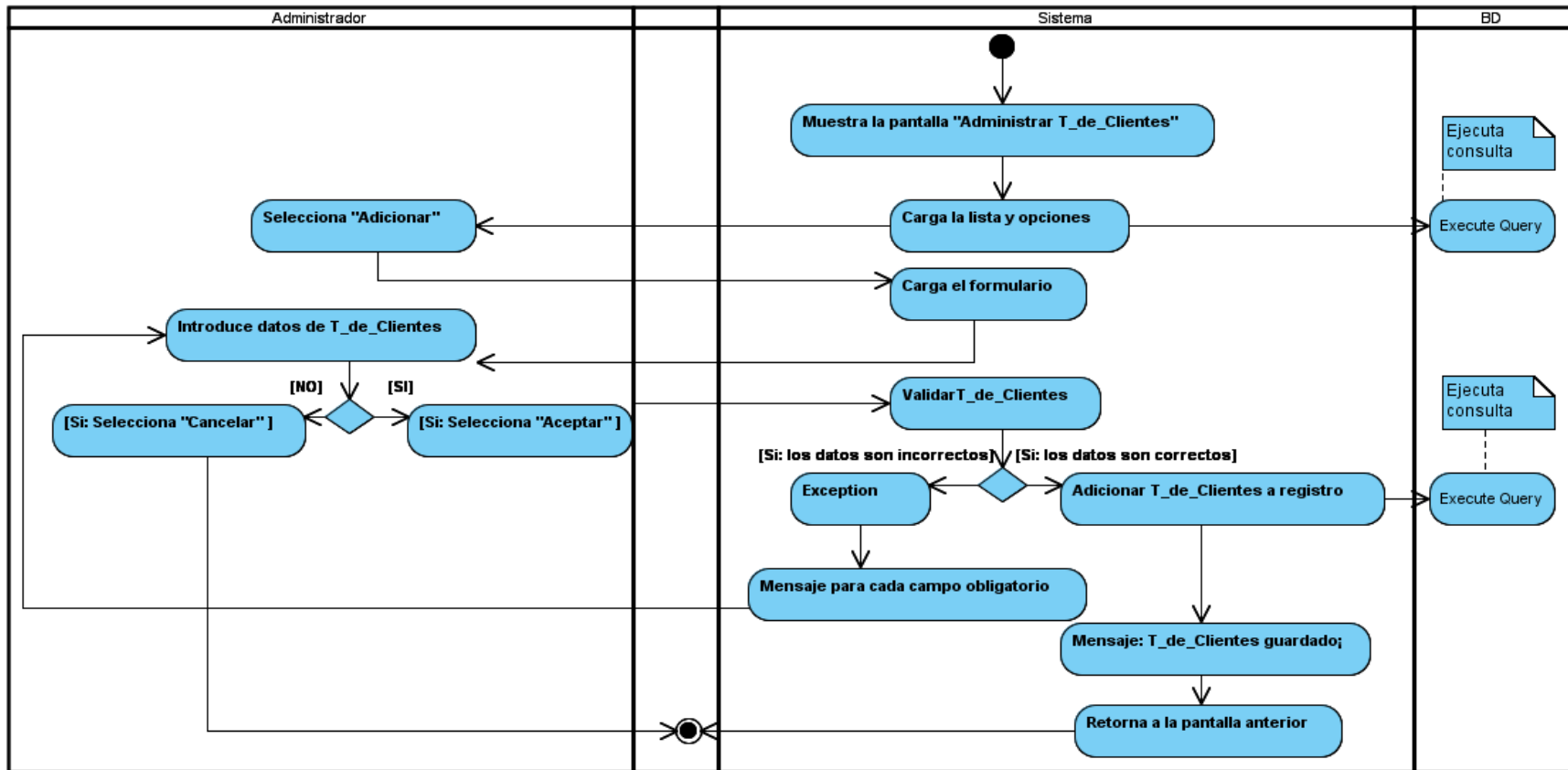


Figura 104: Diagrama de Actividades- Adicionar Tipos de Clientes

Diagrama de actividad: C.U. Modificar Tipo de Cliente

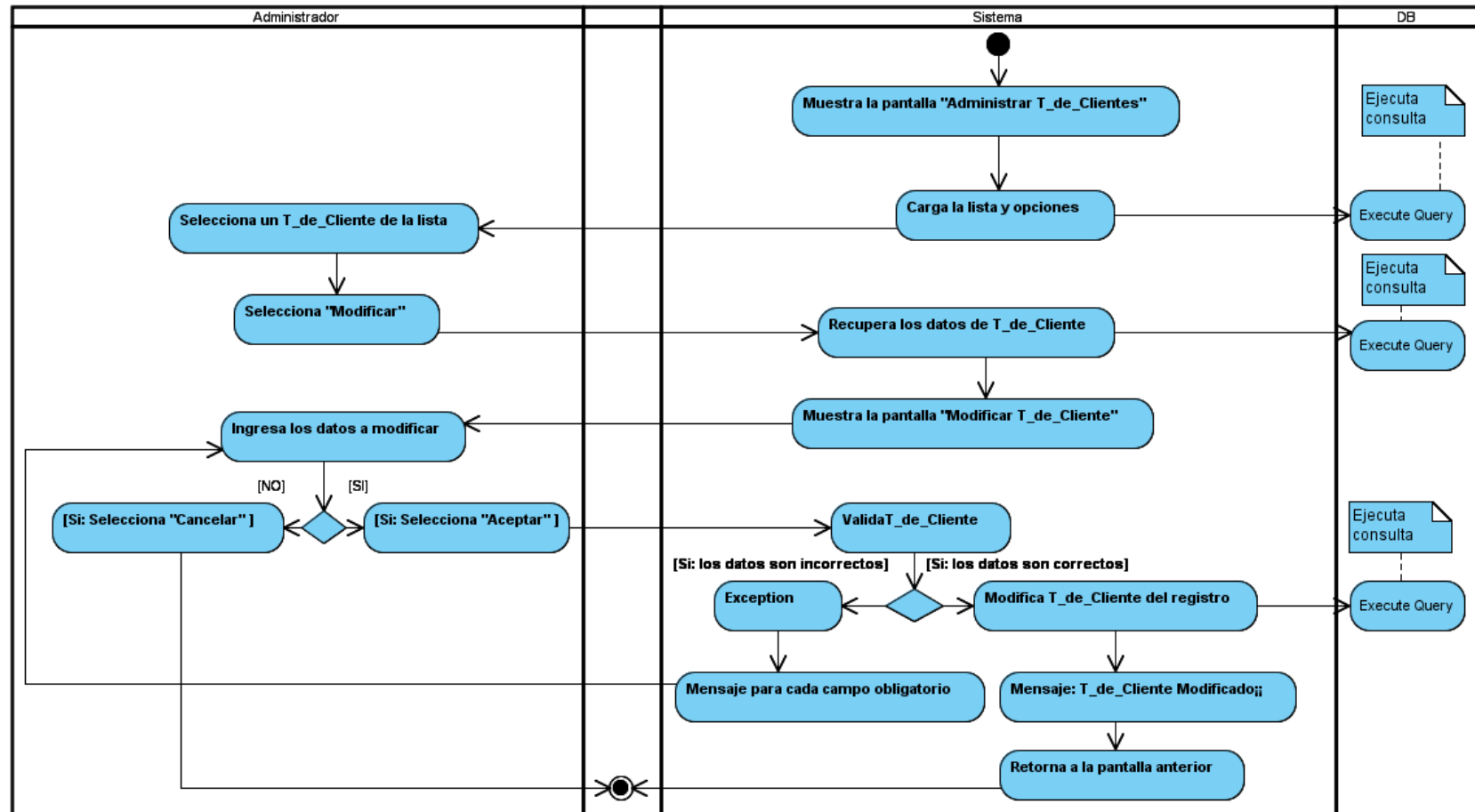


Figura 105: Diagrama de Actividades- Modificar Tipos de Clientes

Diagrama de actividad: C.U. Eliminar Tipo de Cliente

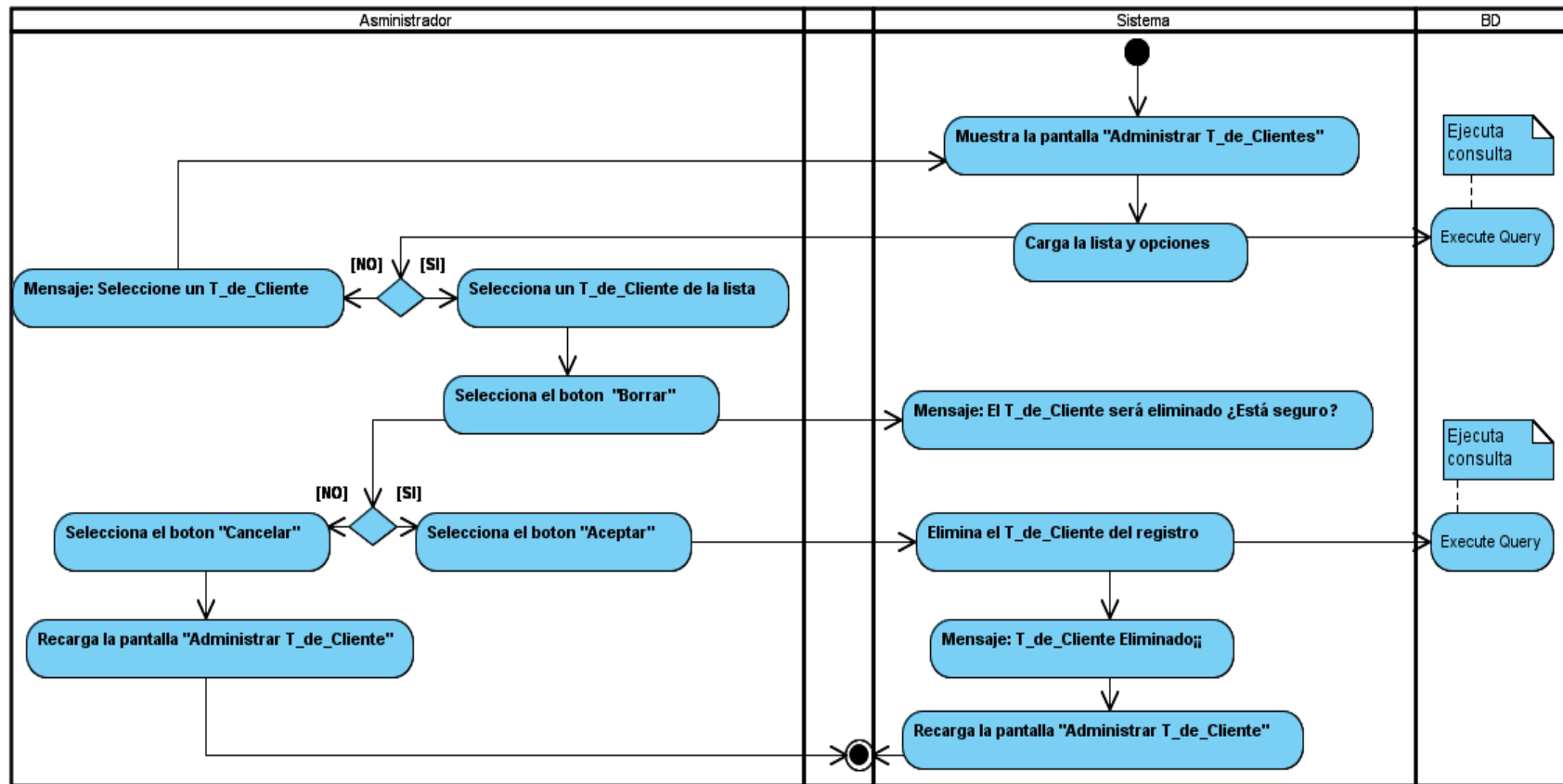


Figura 106: Diagrama de Actividades- Eliminar Tipos de Clientes

Diagrama de actividad: C.U. Administrar Venta

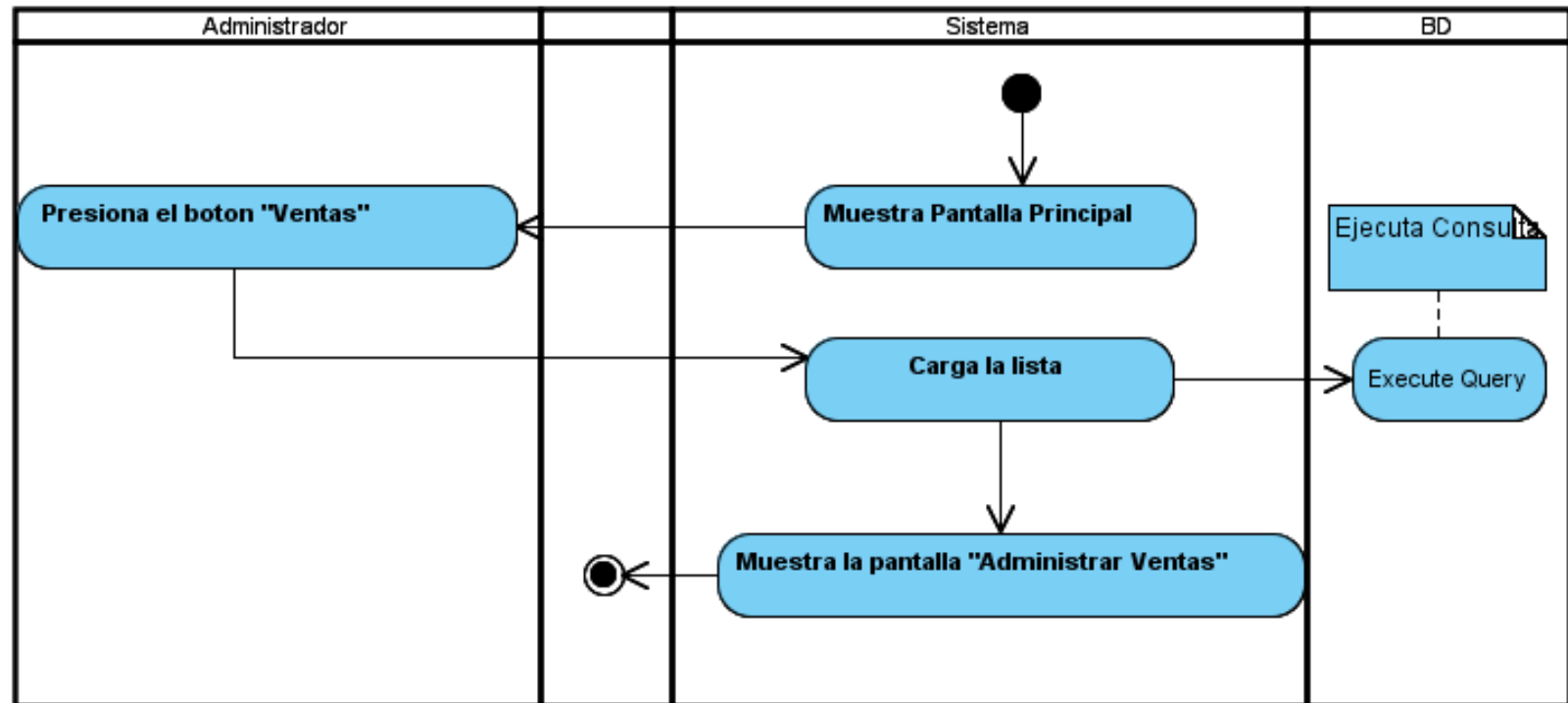


Figura 107: Diagrama de Actividades-Administrar Ventas

Diagrama de actividad: C.U. Adicionar Venta

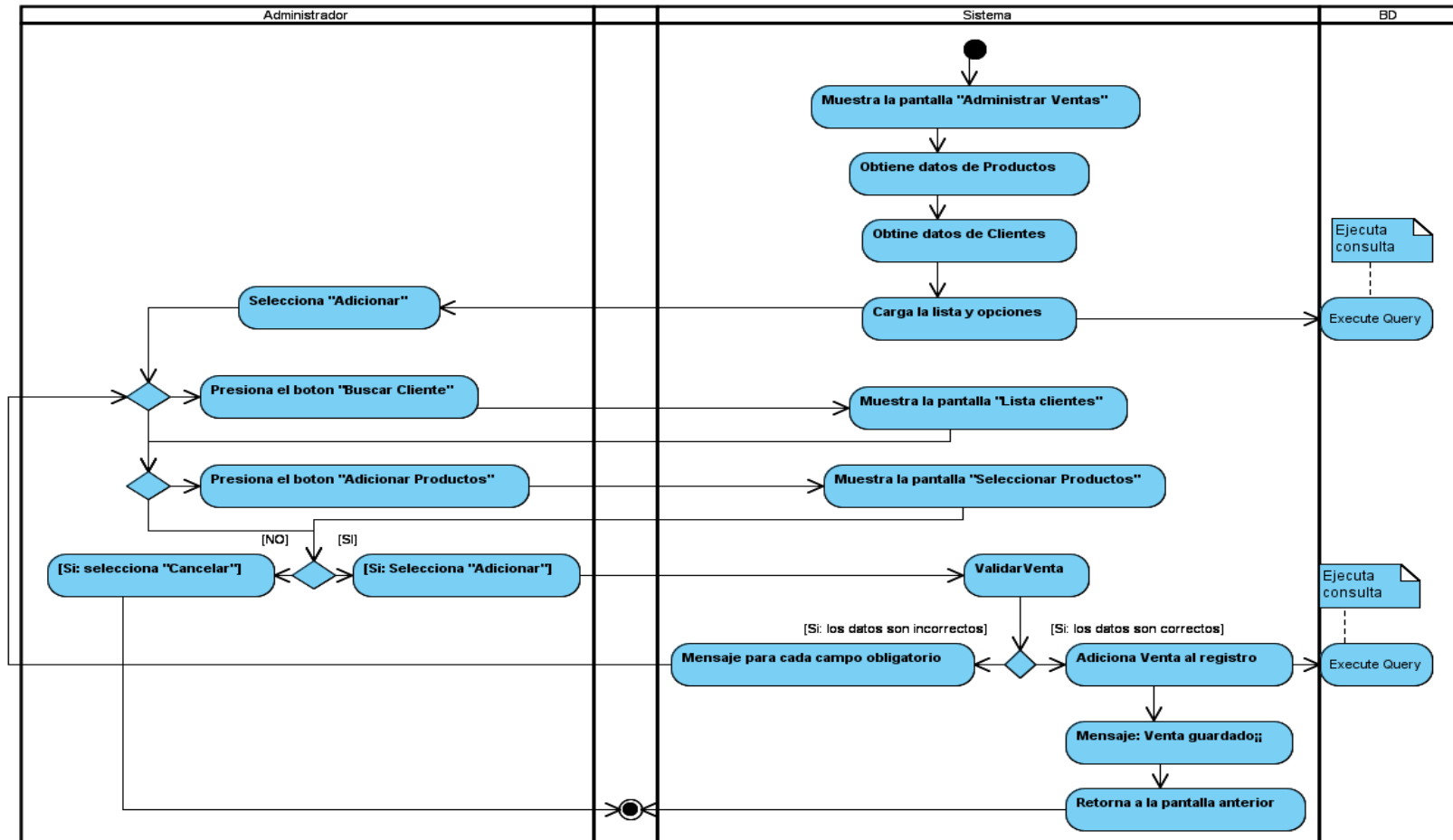


Figura 108: Diagrama de Actividades-Adicionar Venta

Diagrama de actividad: C.U. Administrar Tipo de Ventas

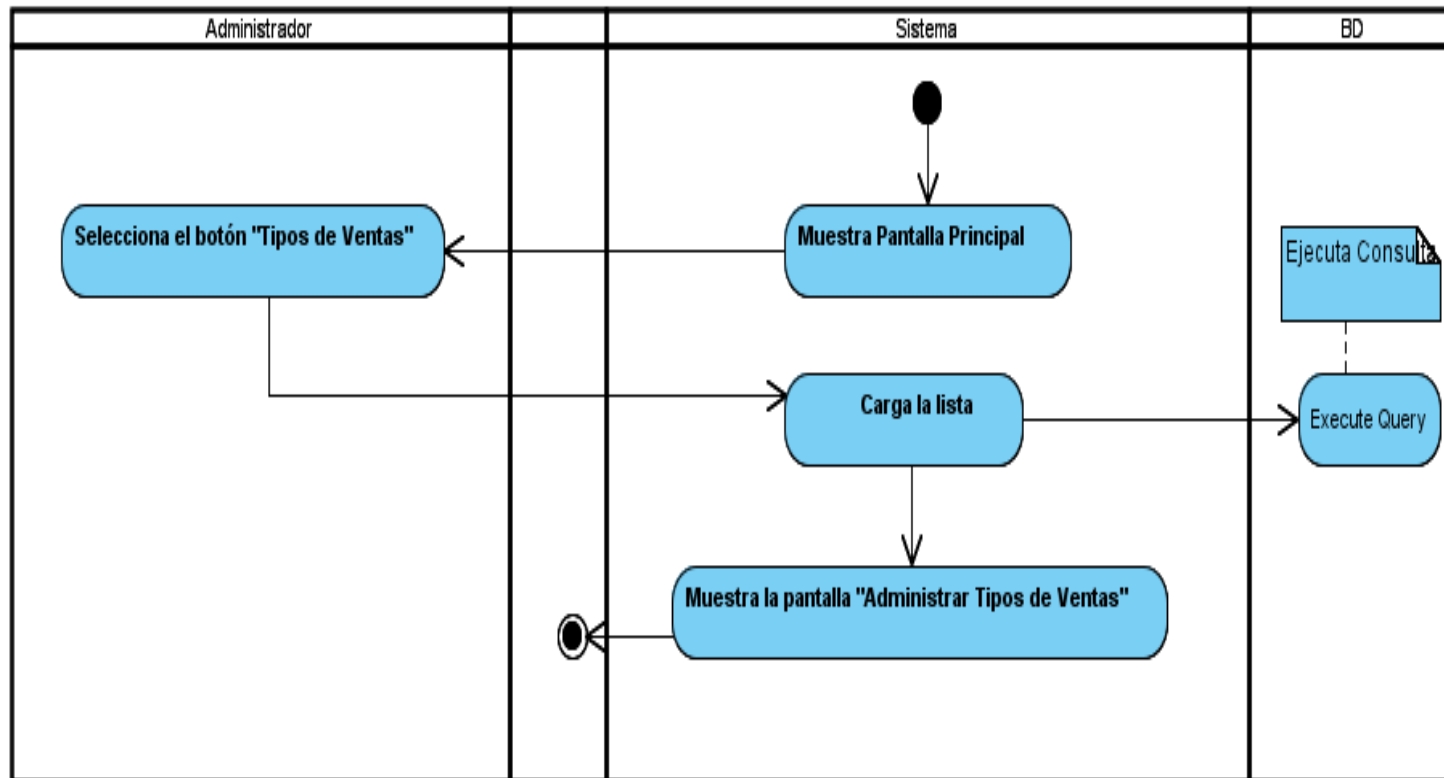


Figura 109: Diagrama de Actividades- Administrar Tipos de Ventas

Diagrama de actividad: C.U. Adicionar Tipo de Venta

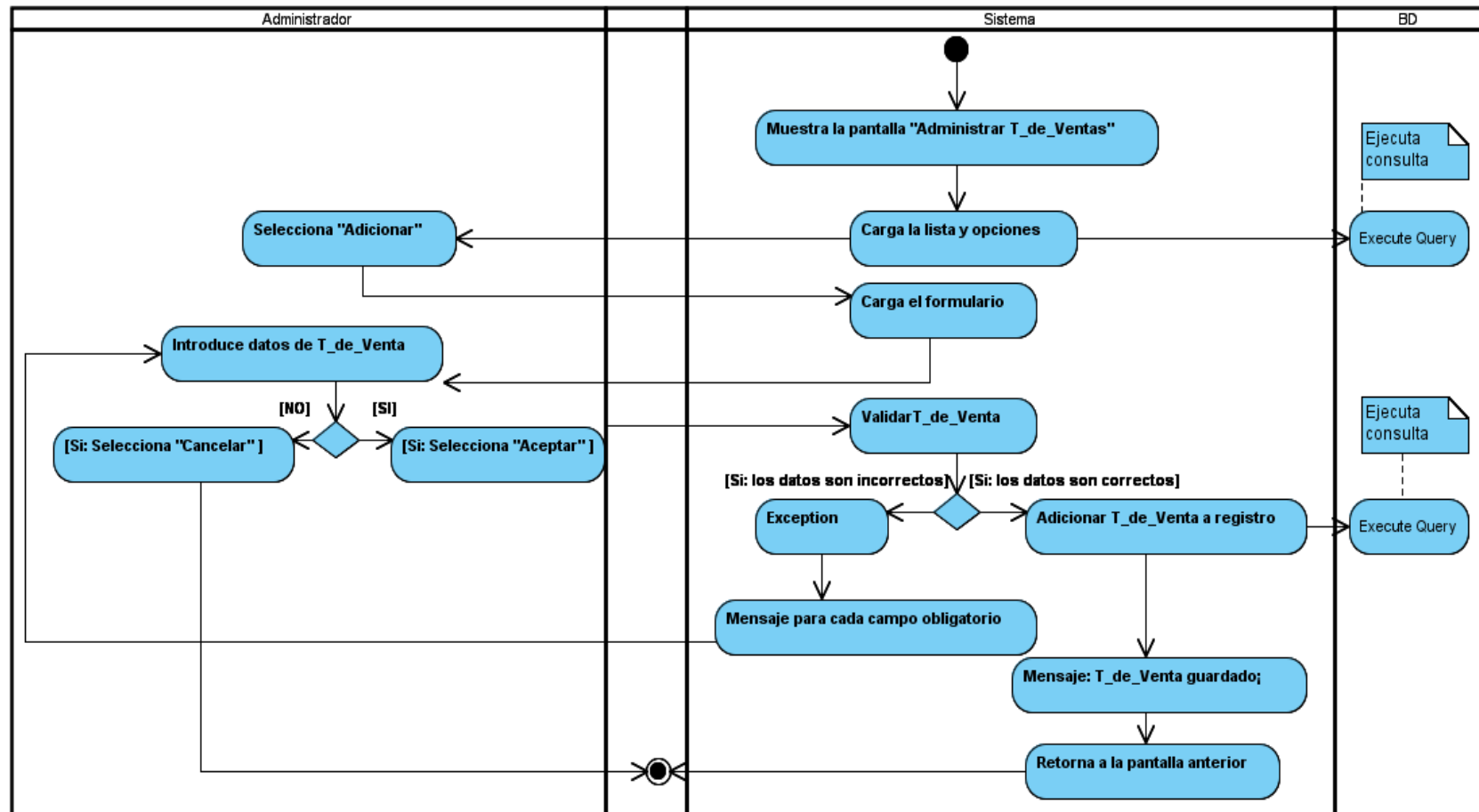


Figura 110: Diagrama de Actividades- Adicionar Tipos de Ventas

Diagrama de actividad: C.U. Modificar Tipo de Venta

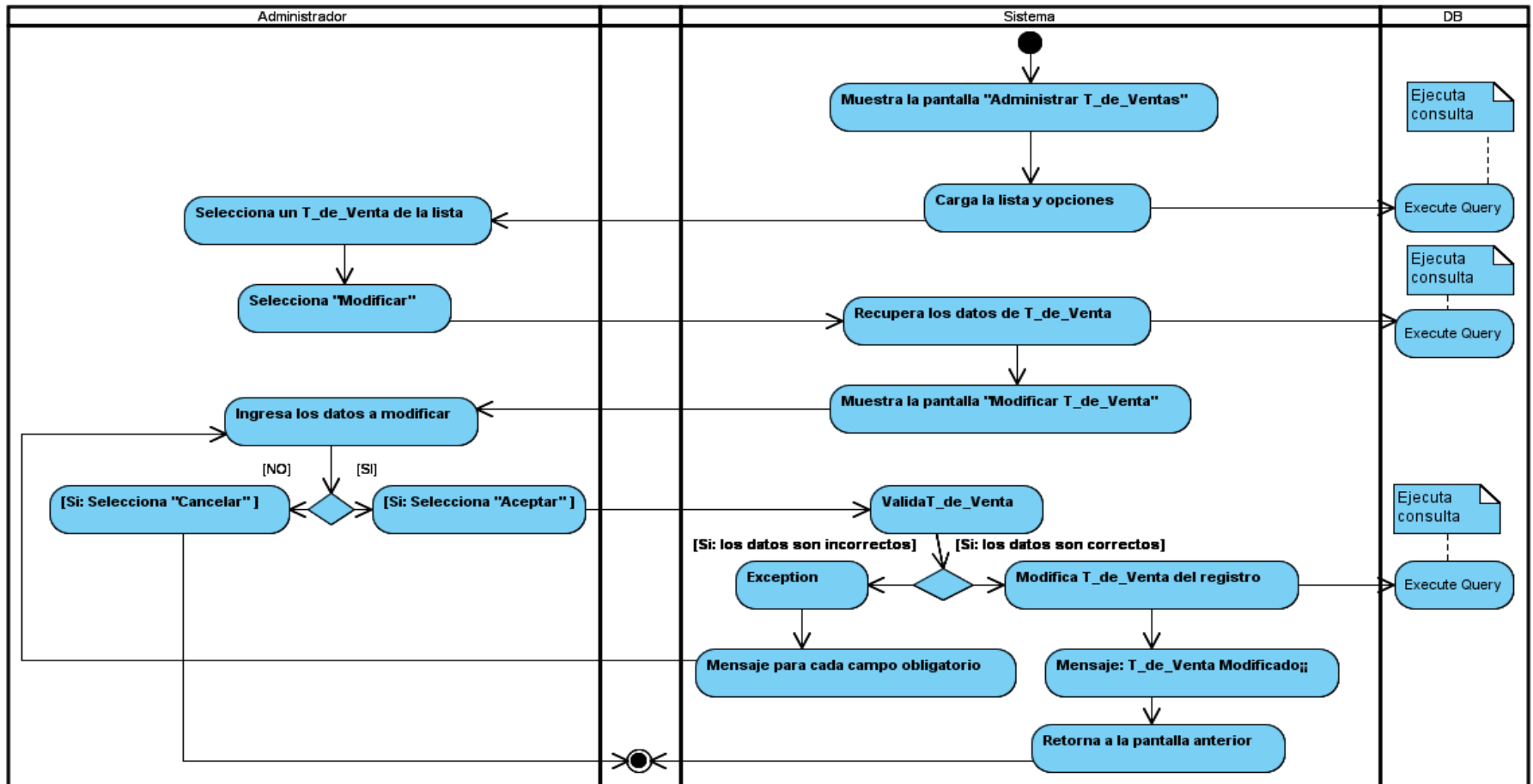


Figura 111: Diagrama de Actividades- Modificar Tipos de Ventas

Diagrama de actividad: C.U. Eliminar Tipo de Venta

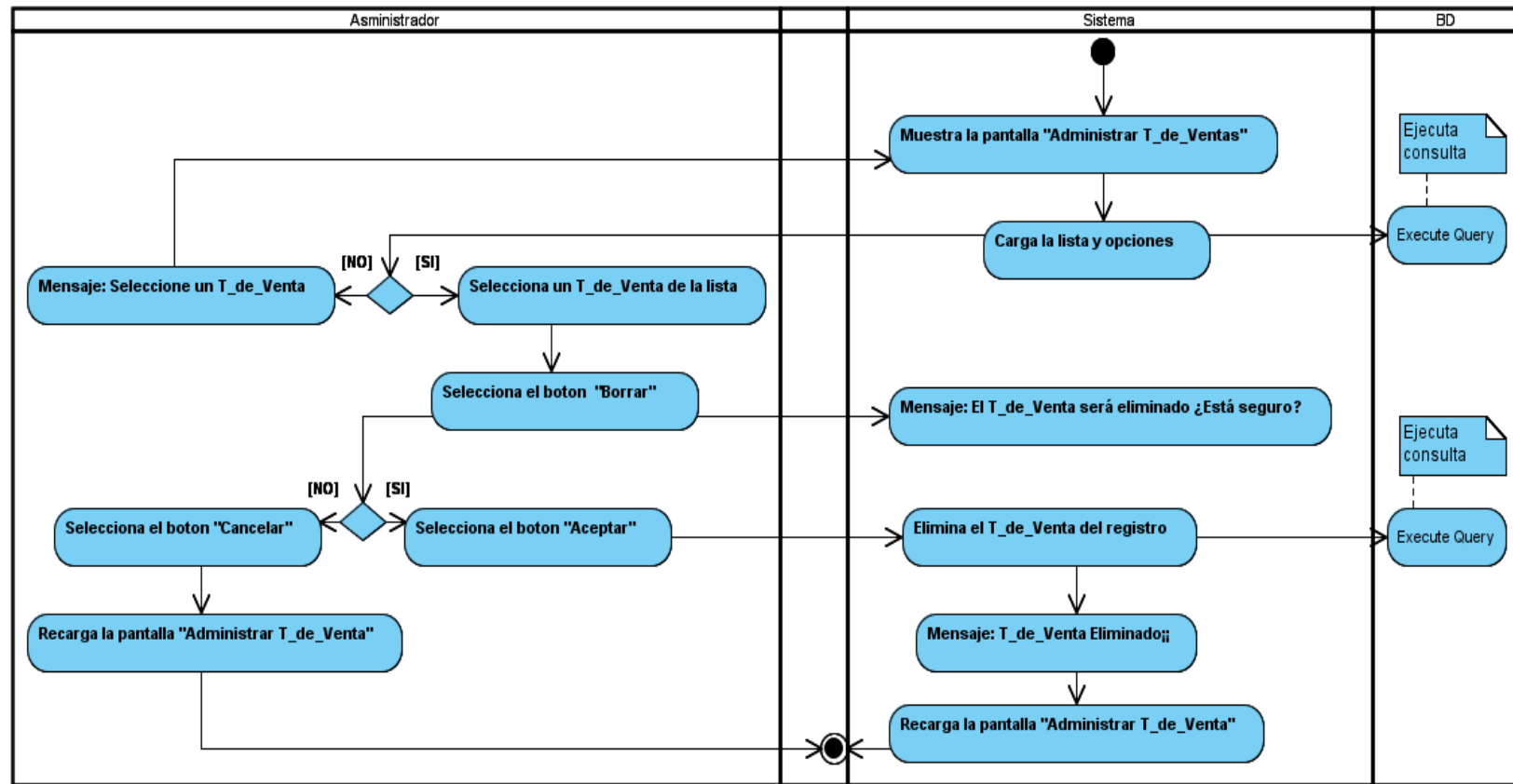


Figura 112: Diagrama de Actividades- Eliminar Tipos de Ventas

Diagrama de Secuencia: C.U. Ver Plan de Pagos

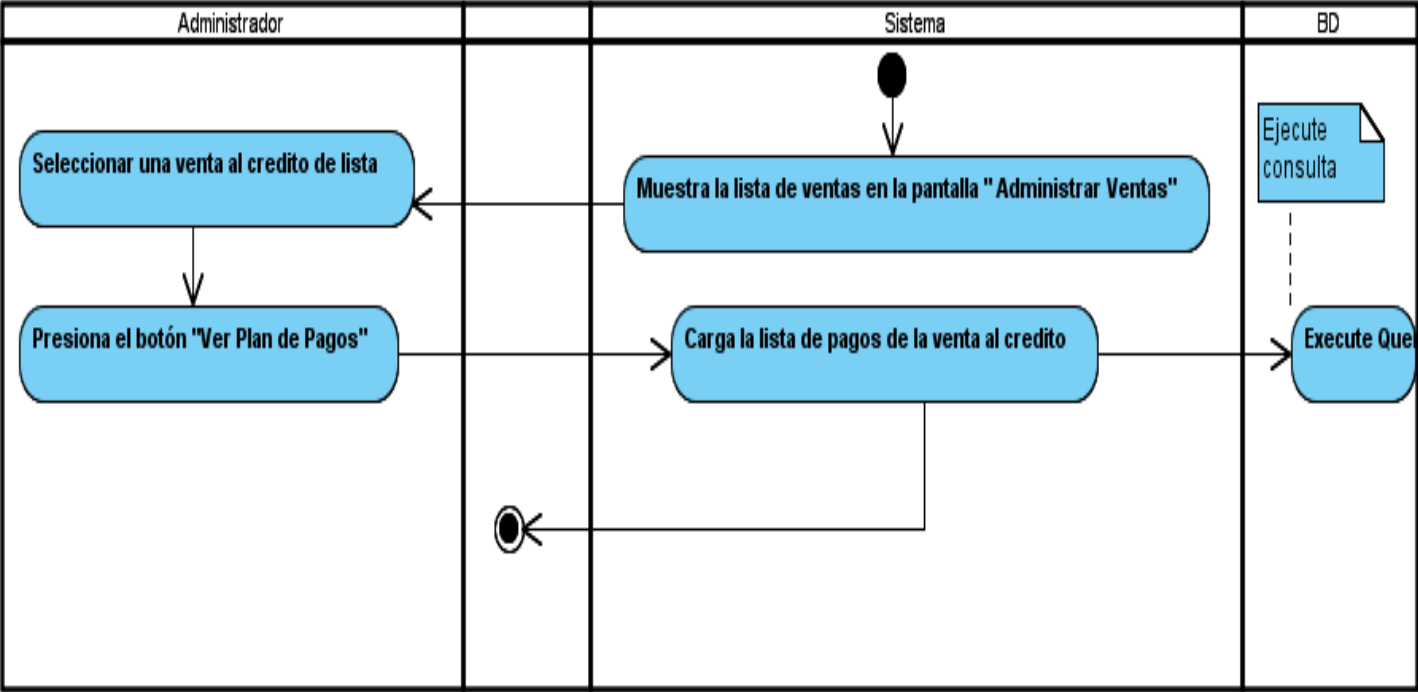


Figura 113: Diagrama de Actividades- Ver Plan de Pagos

Diagrama de actividad: C.U. Pagar Cuota de Plan de Pago

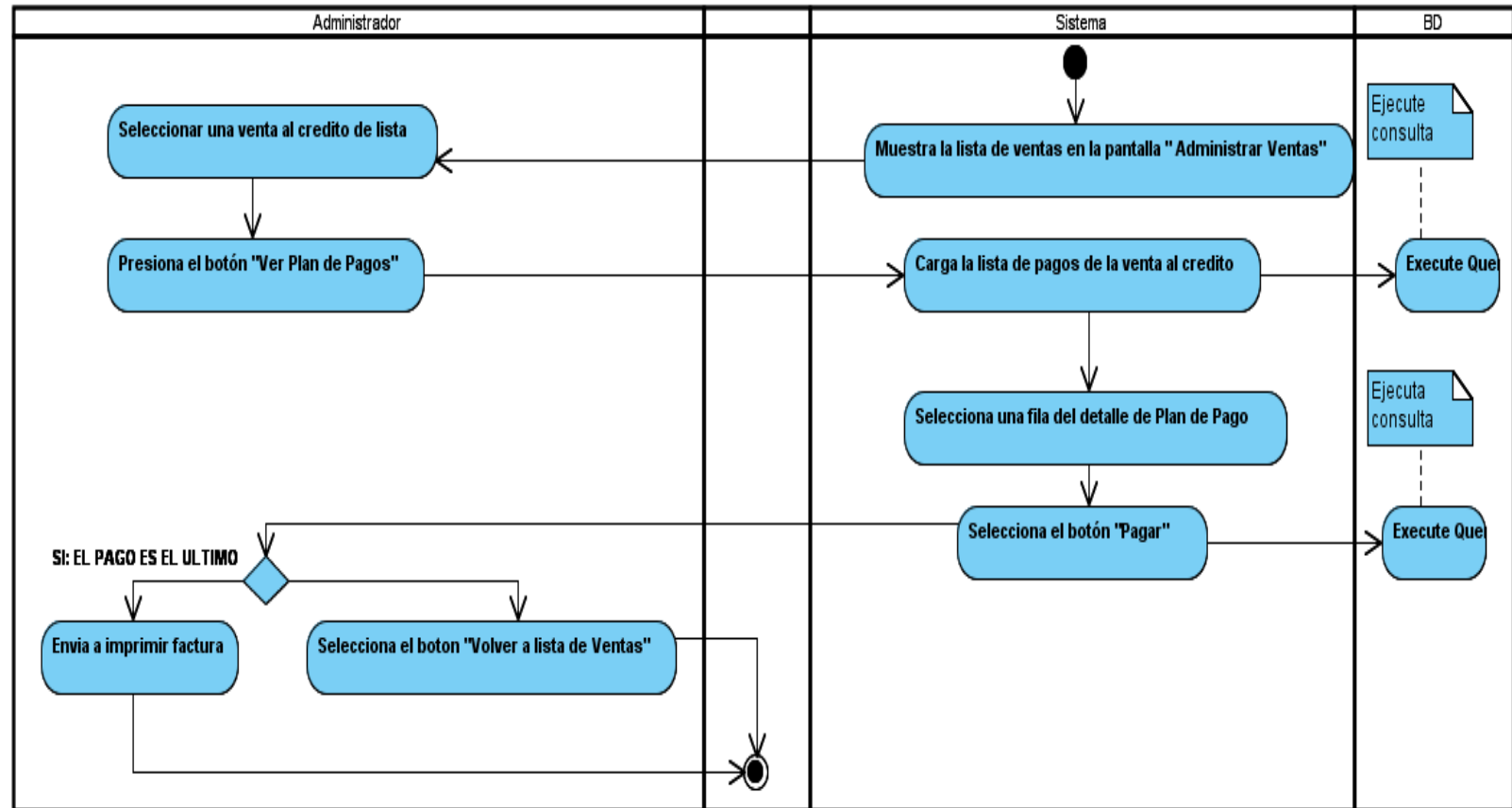


Figura 114: Diagrama de Actividades- Pagar cuota de Plan de Pago

Diagrama de actividad: C.U. Administrar Productos

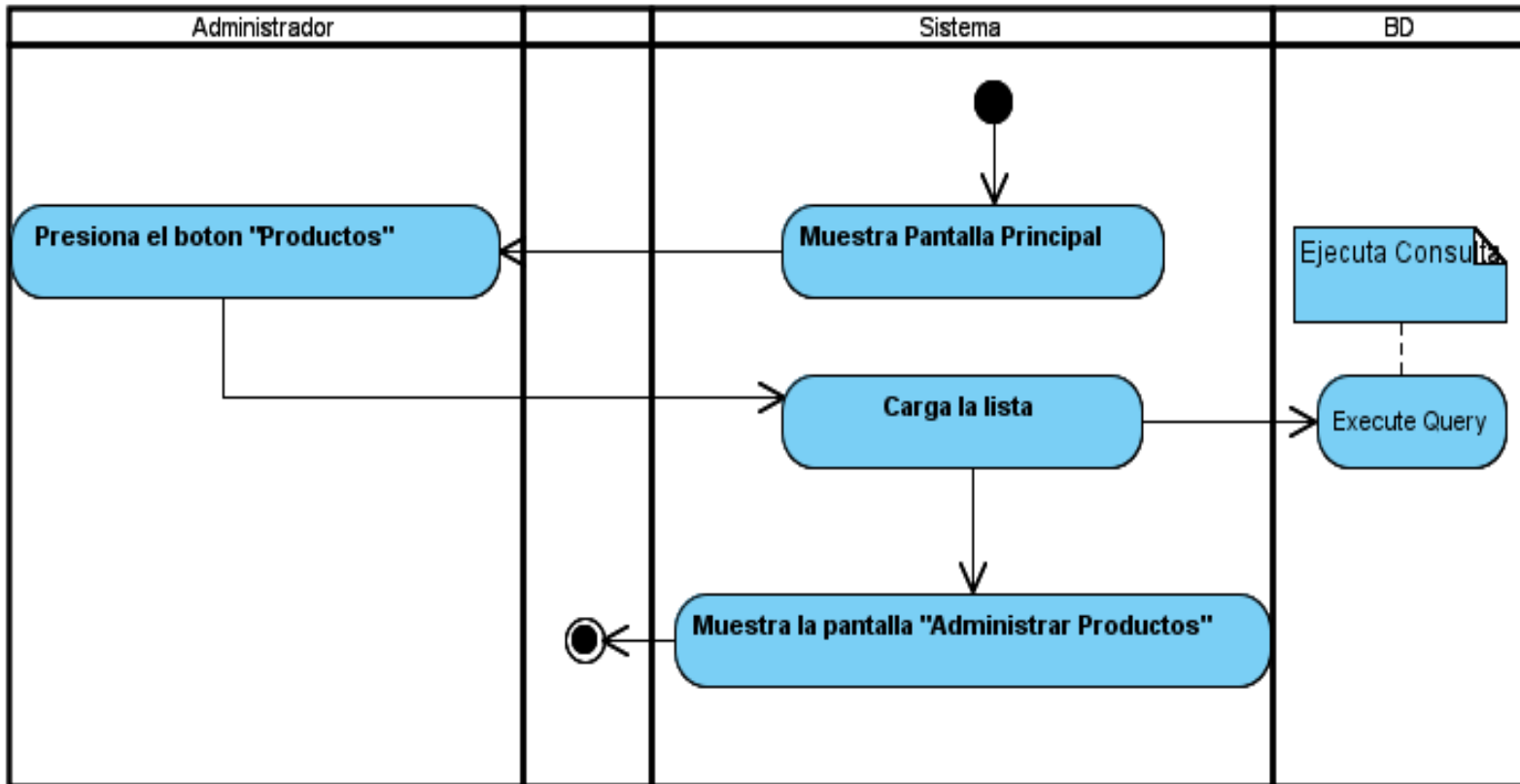


Figura 115: Diagrama de Actividades-Administrar Productos

Diagrama de actividad: C.U. Adicionar Producto

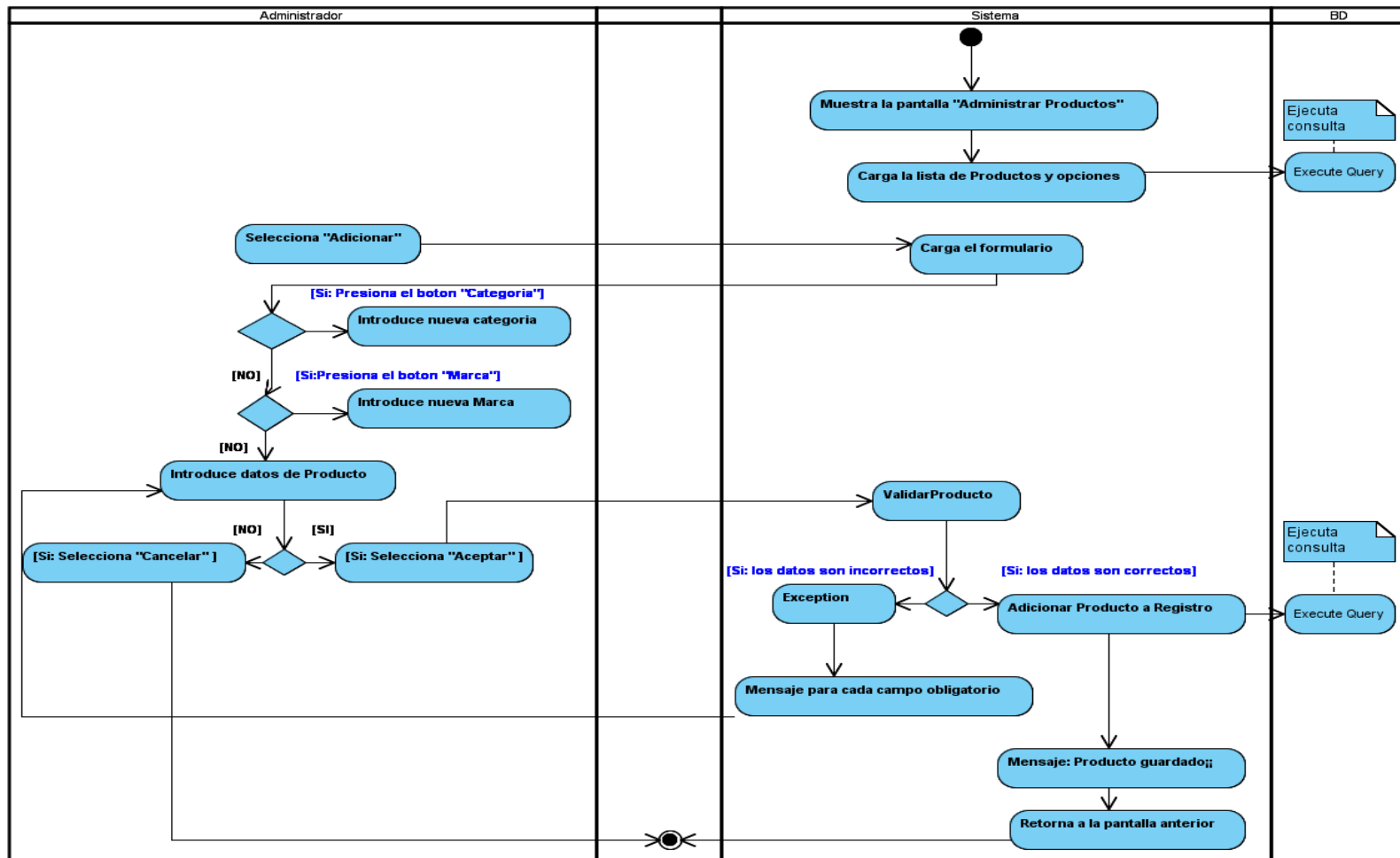


Figura 116: Diagrama de Actividades-Adicionar Producto

Diagrama de actividad: C.U. Modificar Producto

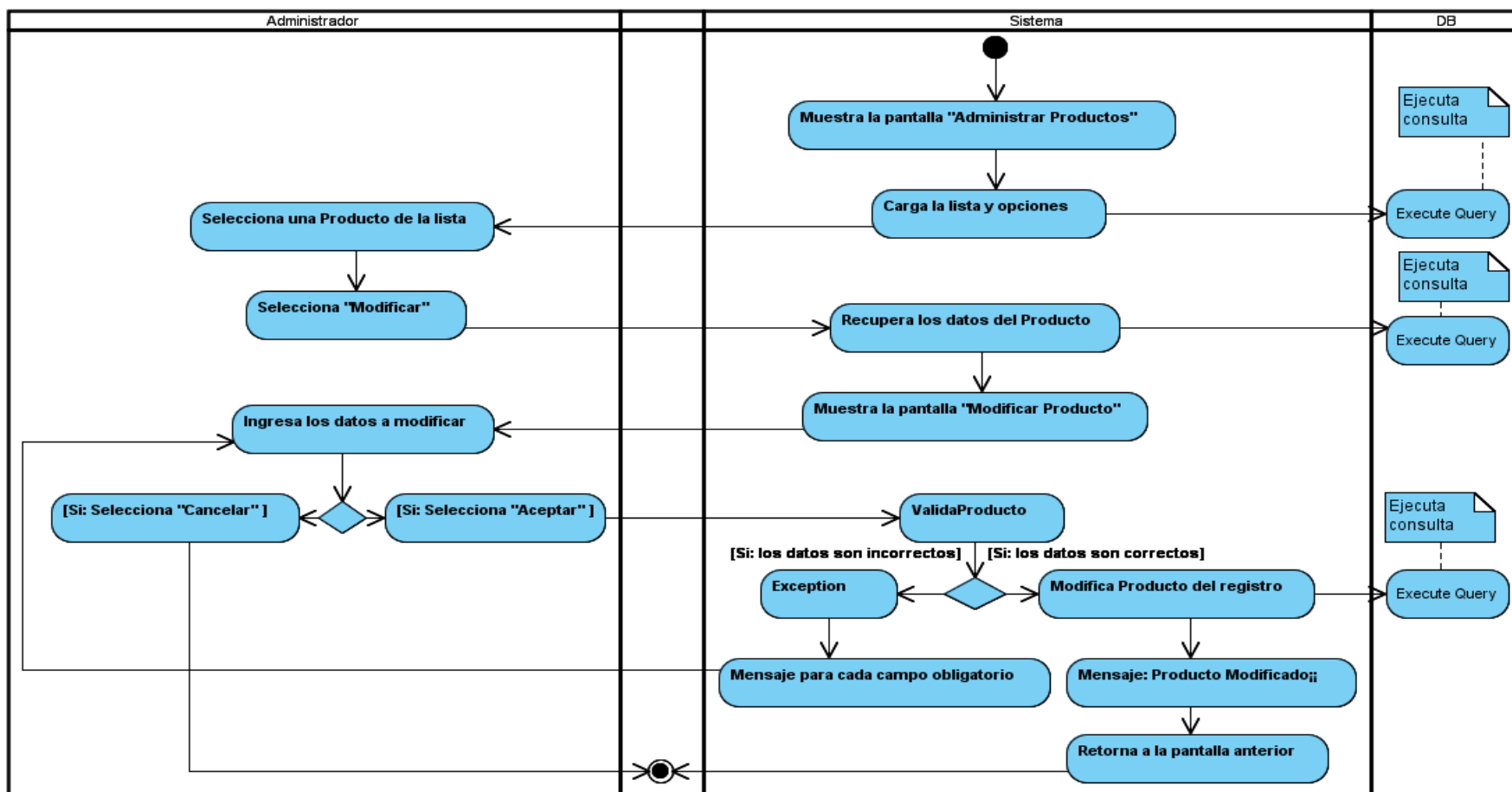


Figura 117: Diagrama de Actividades-Modificar Producto

Diagrama de actividad: C.U. Eliminar Producto

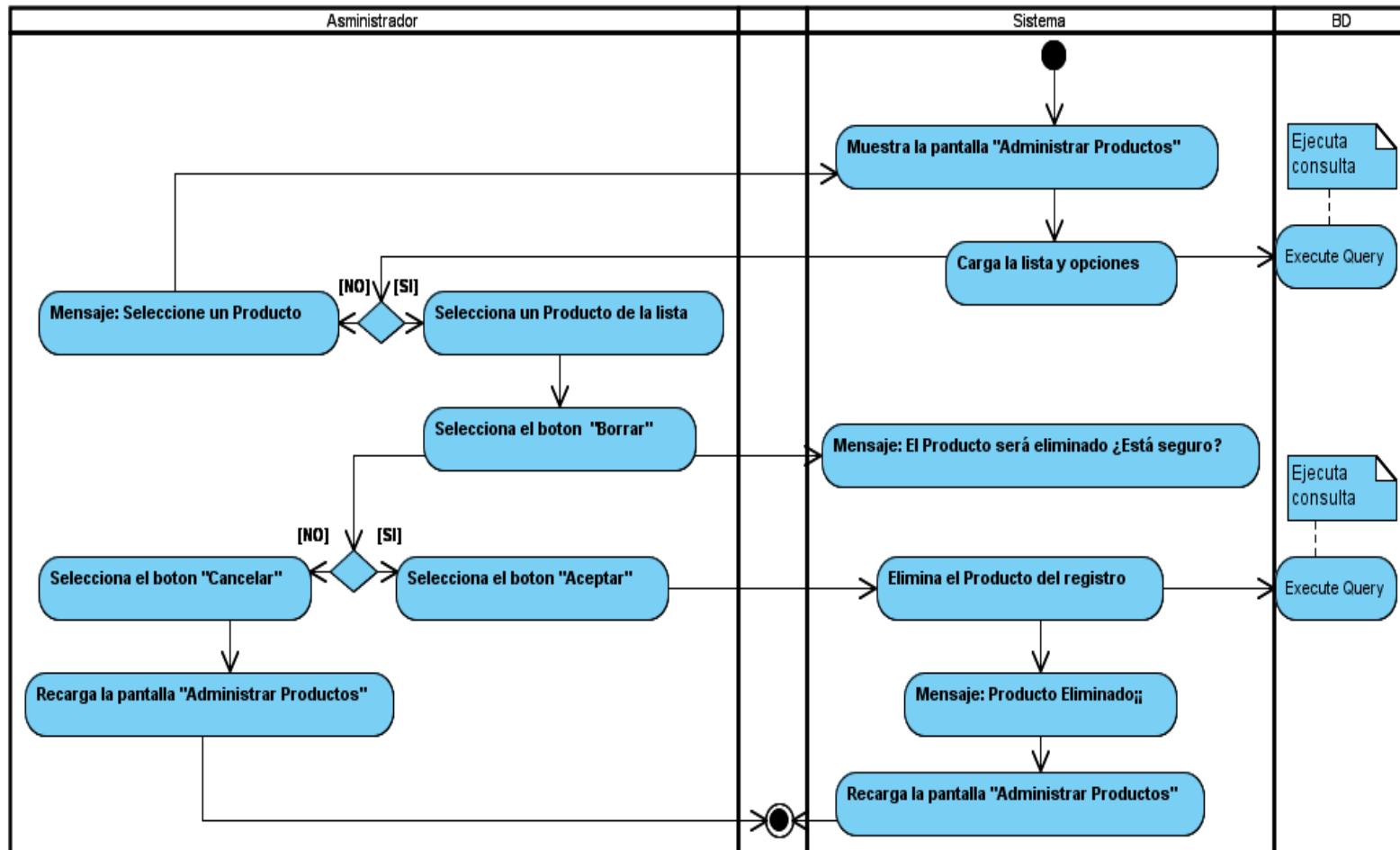


Figura 118: Diagrama de Actividades-Eliminar Producto

Diagrama de actividades: C.U. Administrar Almacén

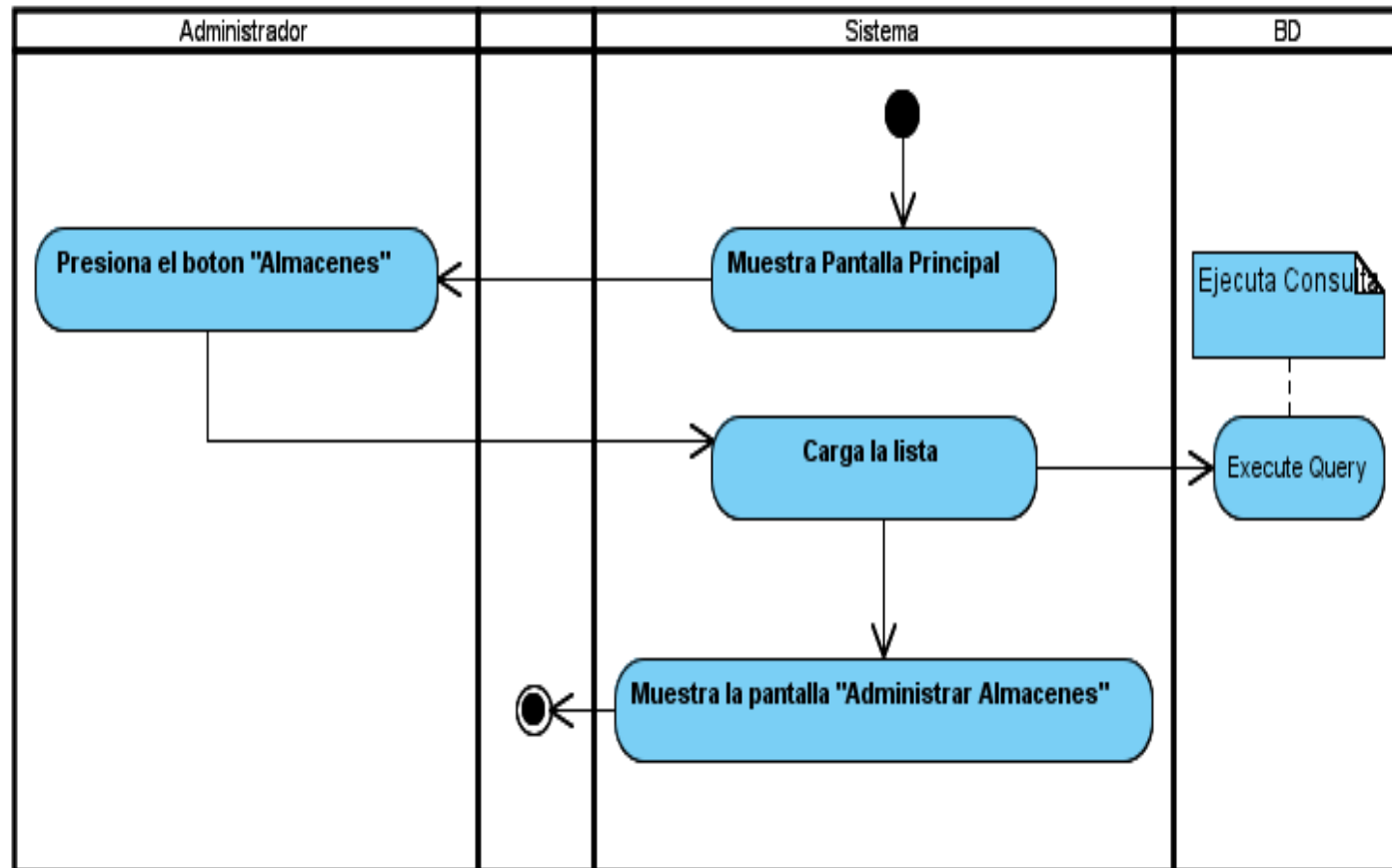


Figura 119: Diagrama de Actividades-Administrar Almacén

Diagrama de actividades: C.U. Adicionar Almacén

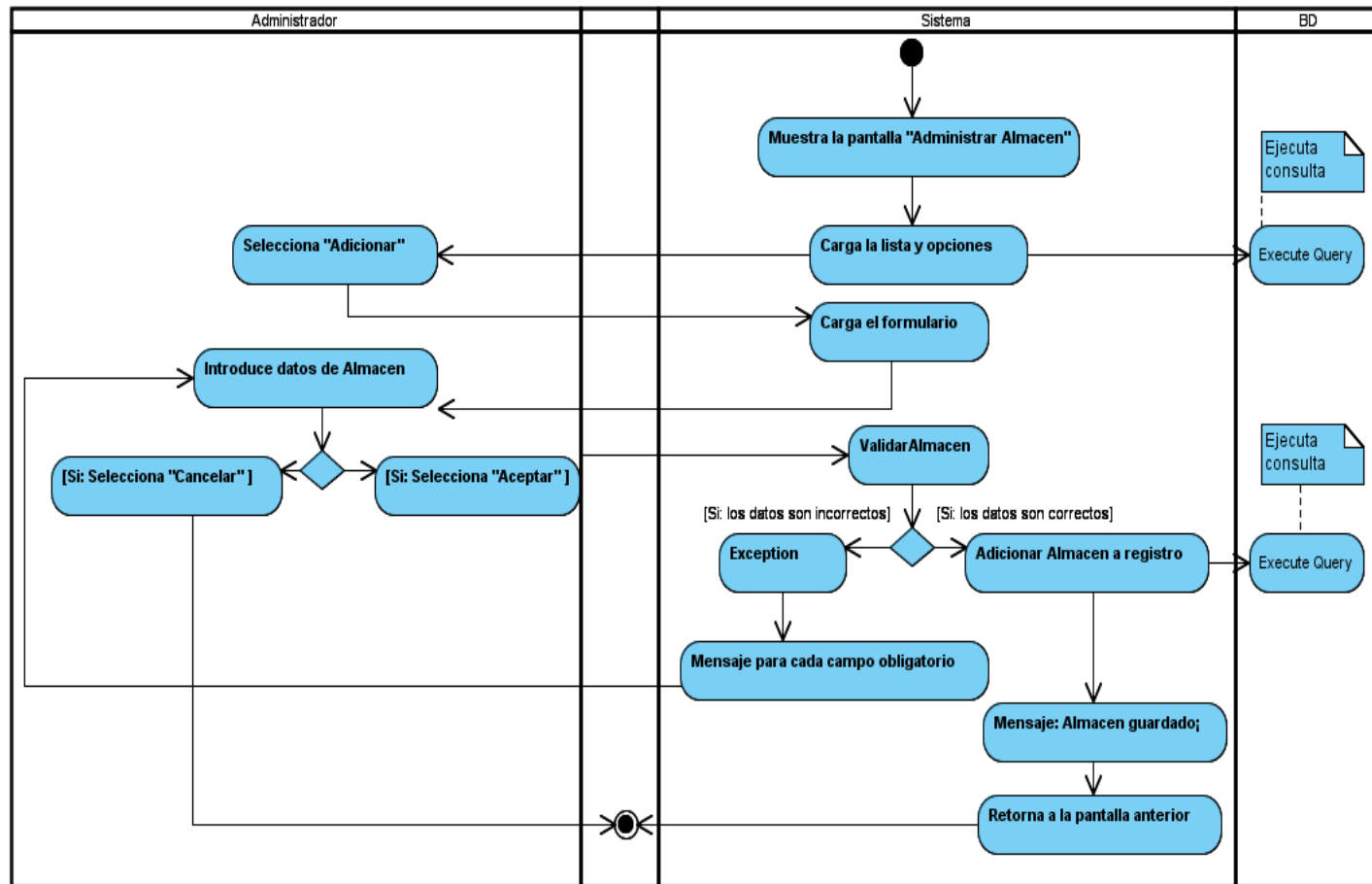


Figura 120: Diagrama de Actividades-Adicionar Almacén

Diagrama de actividades: C.U. Modificar Almacén

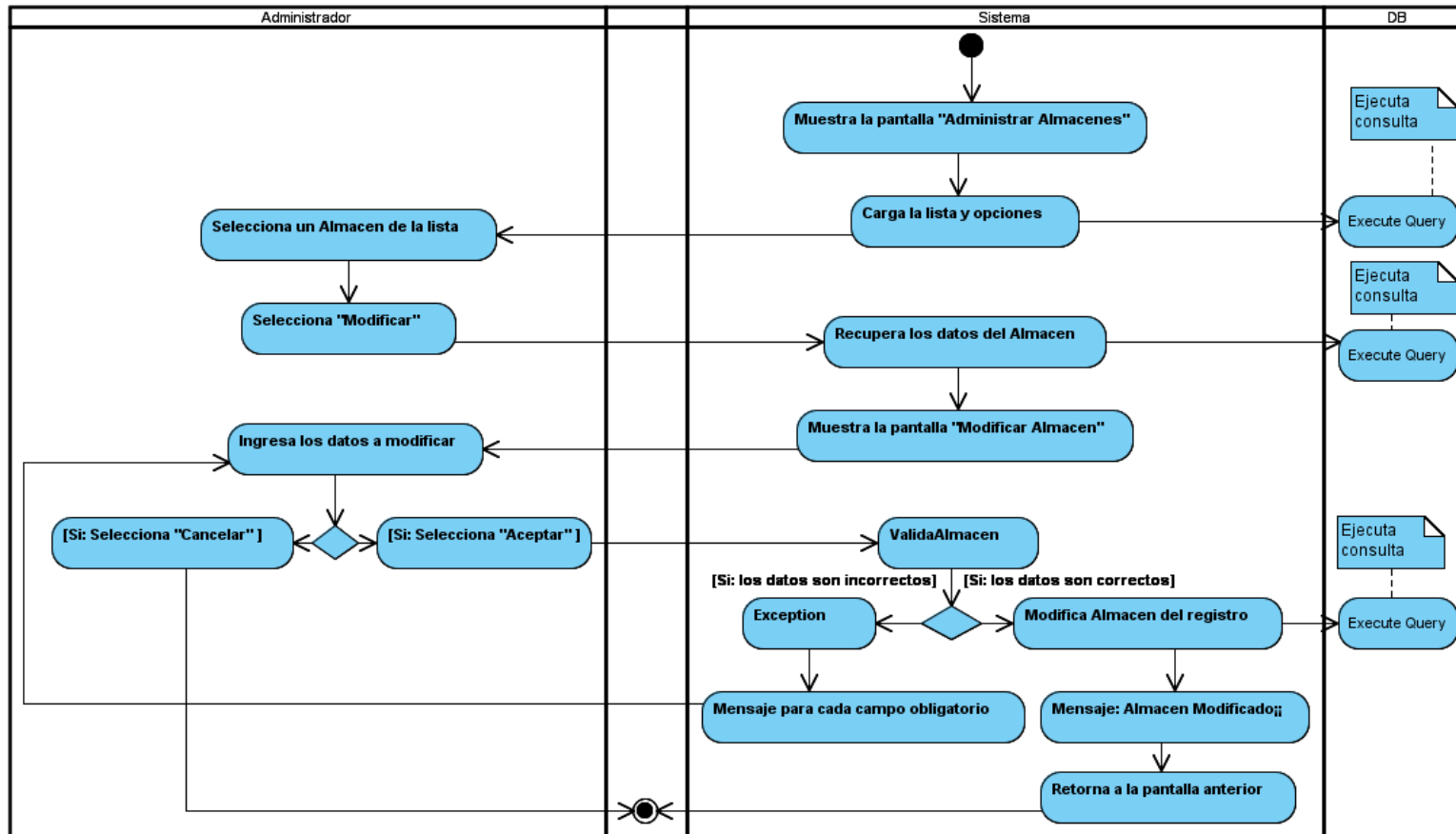


Figura 121: Diagrama de Actividades-Modificar Almacén

Diagrama de actividades: C.U. Eliminar Almacén

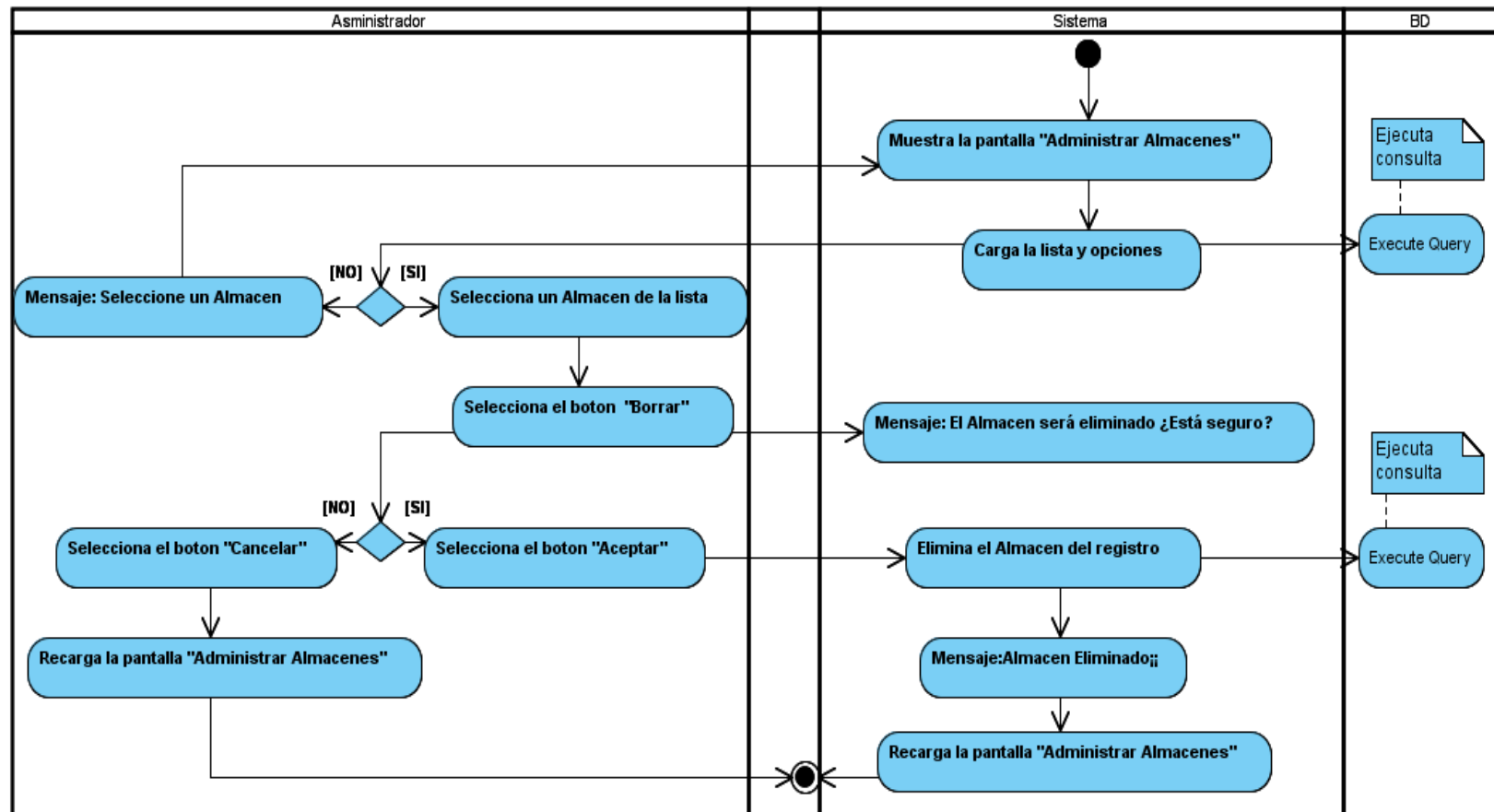


Figura 122: Diagrama de Actividades-Eliminar Almacén

Diagrama de actividades: C.U. Administrar Compra

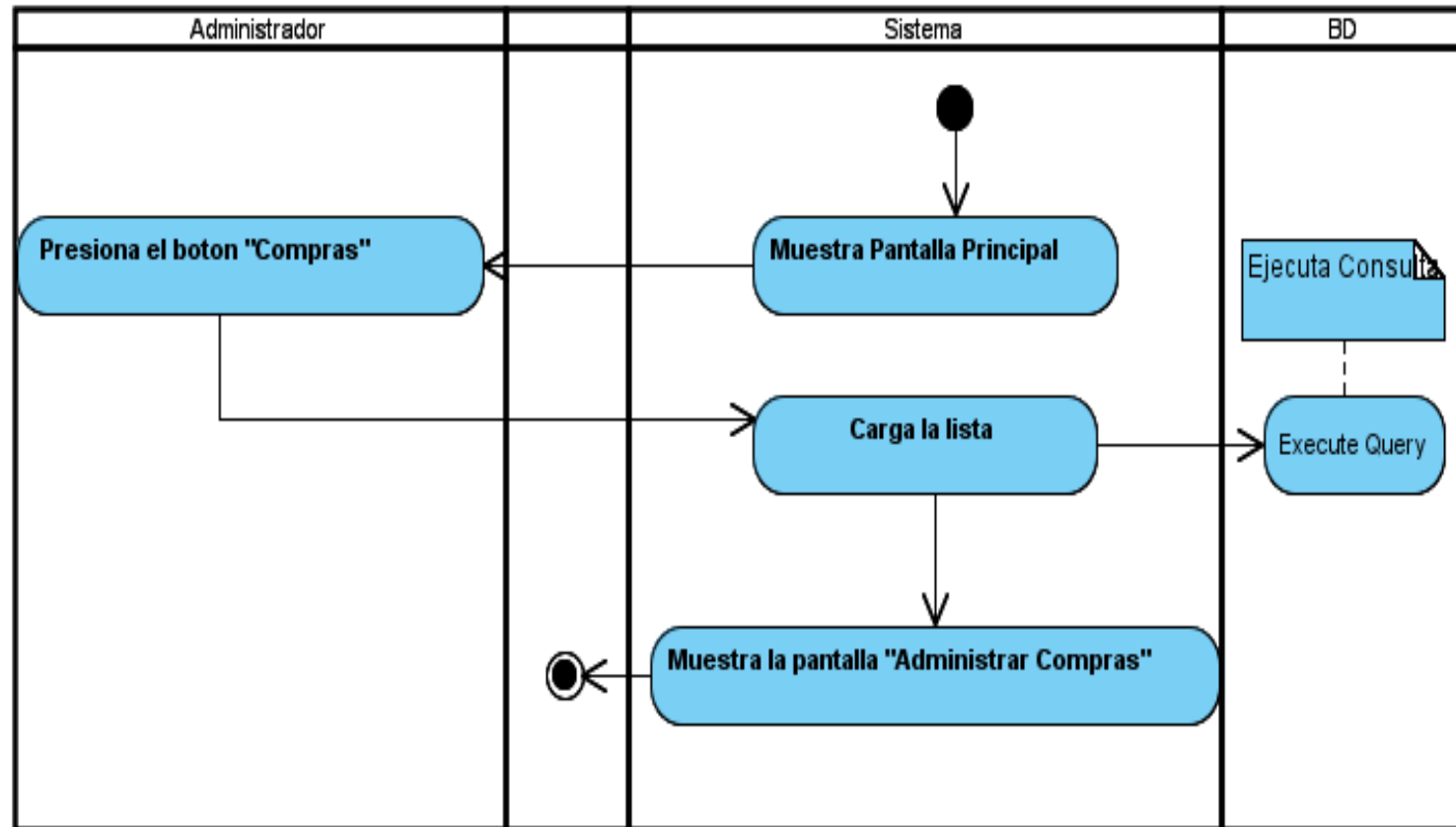


Figura 123: Diagrama de Actividades-Administrar Compra

Diagrama de actividades: C.U. Adicionar Compra

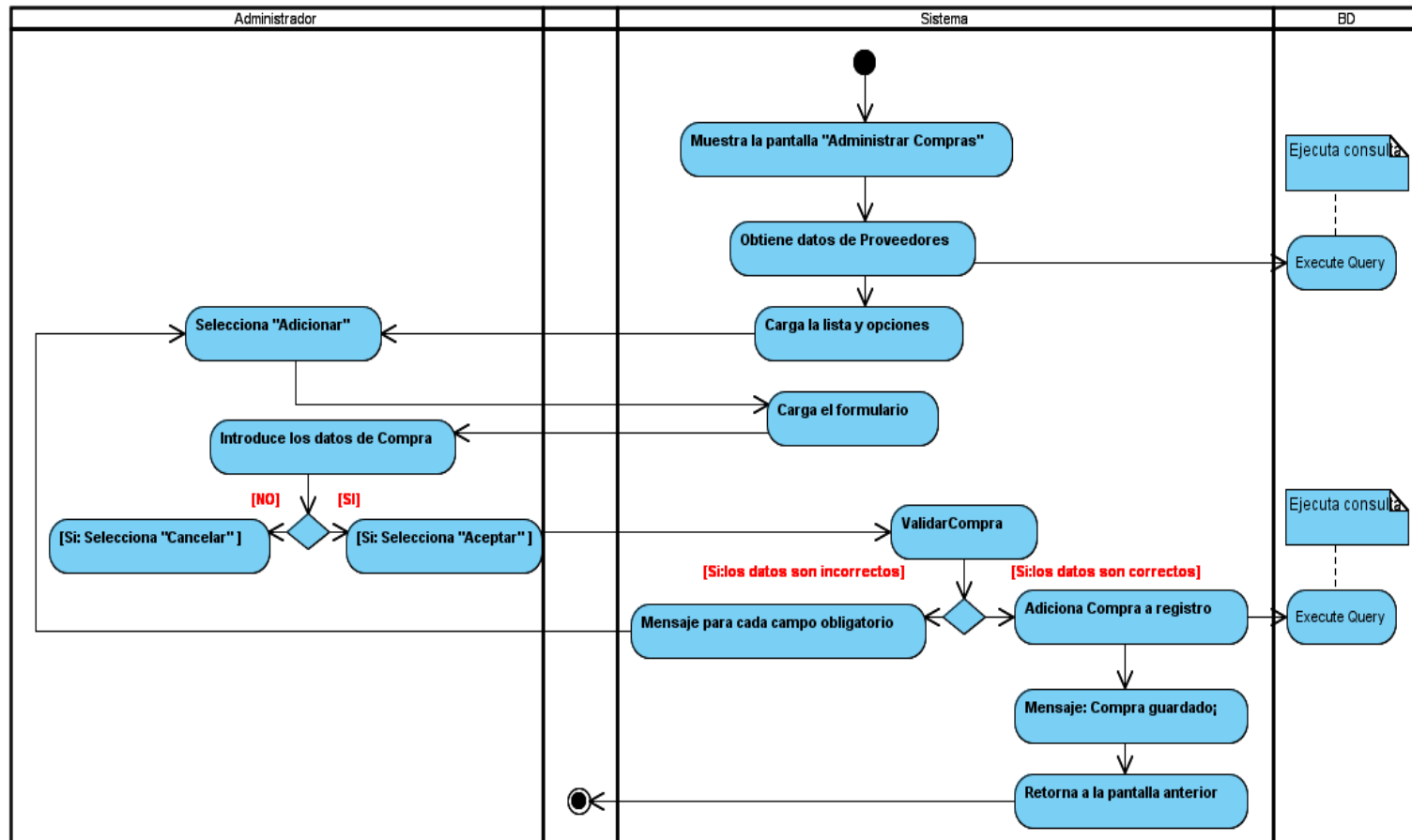


Figura 124: Diagrama de Actividades-Adicionar Compra

Diagrama de actividades: C.U. Modificar Compra

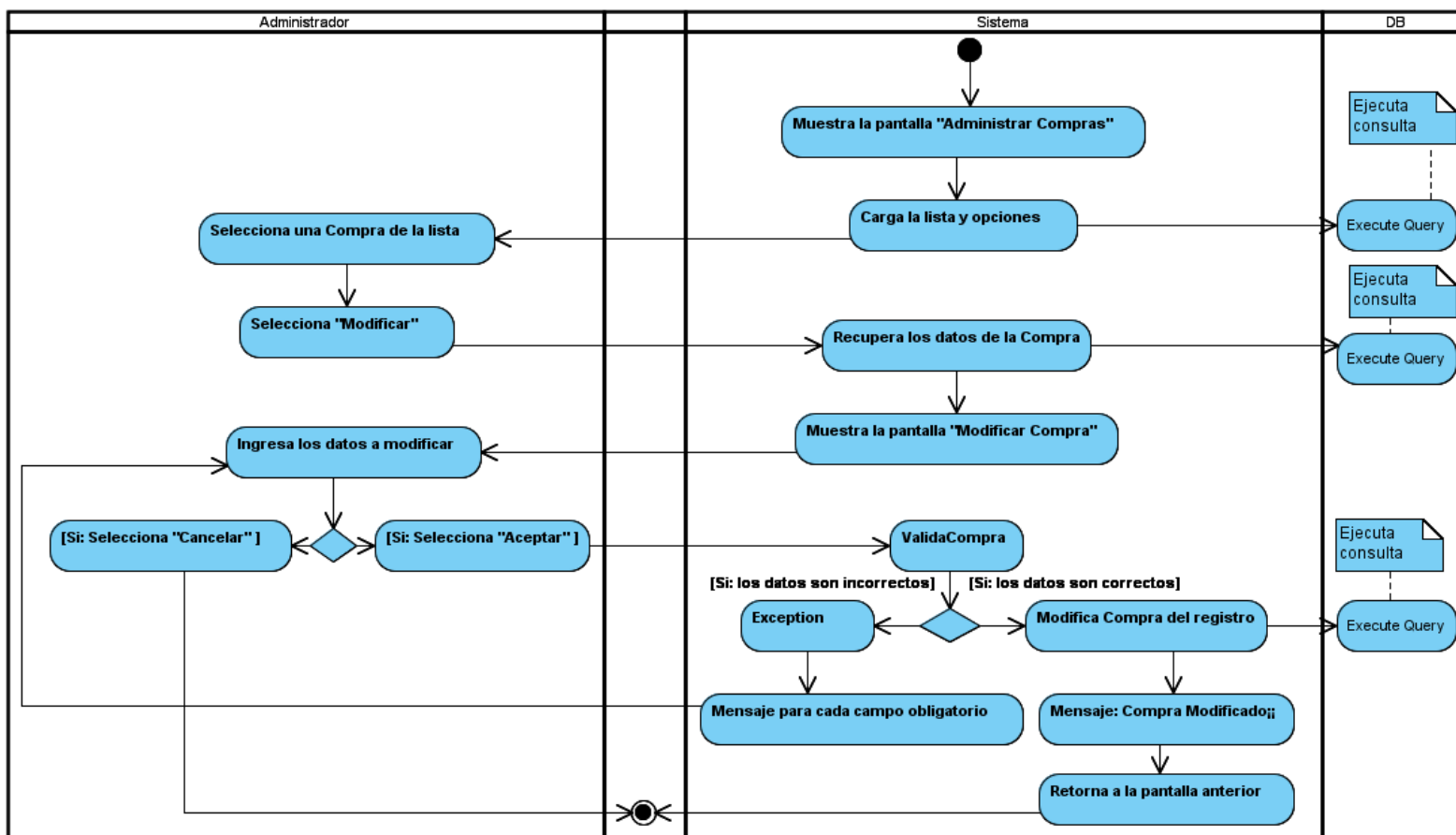


Figura 125: Diagrama de Actividades-Modificar Compra

Diagrama de actividades: C.U. Eliminar Compra

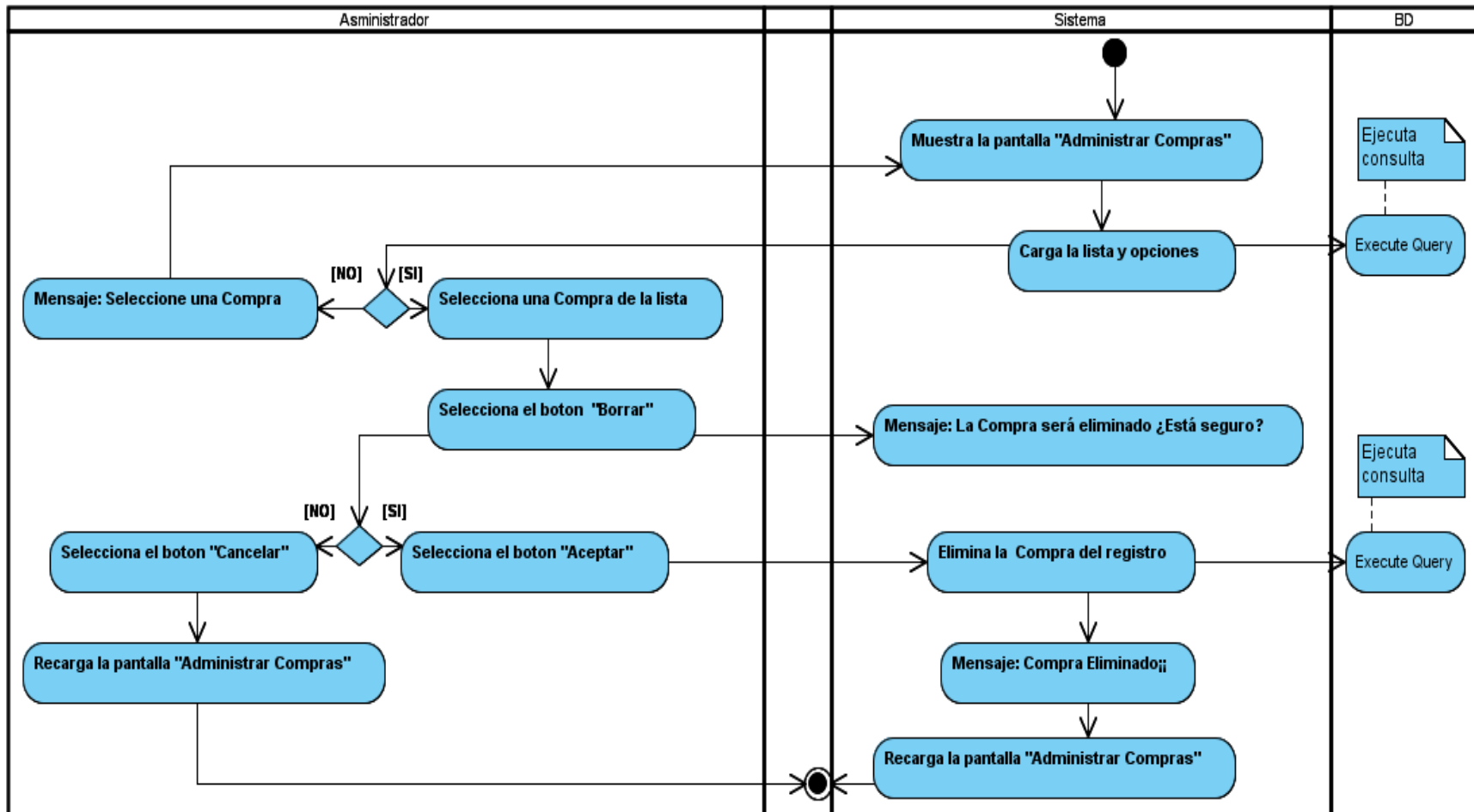


Figura 126: Diagrama de Actividades-Eliminar Compra

Diagrama de actividad: C.U. Administrar Proveedor

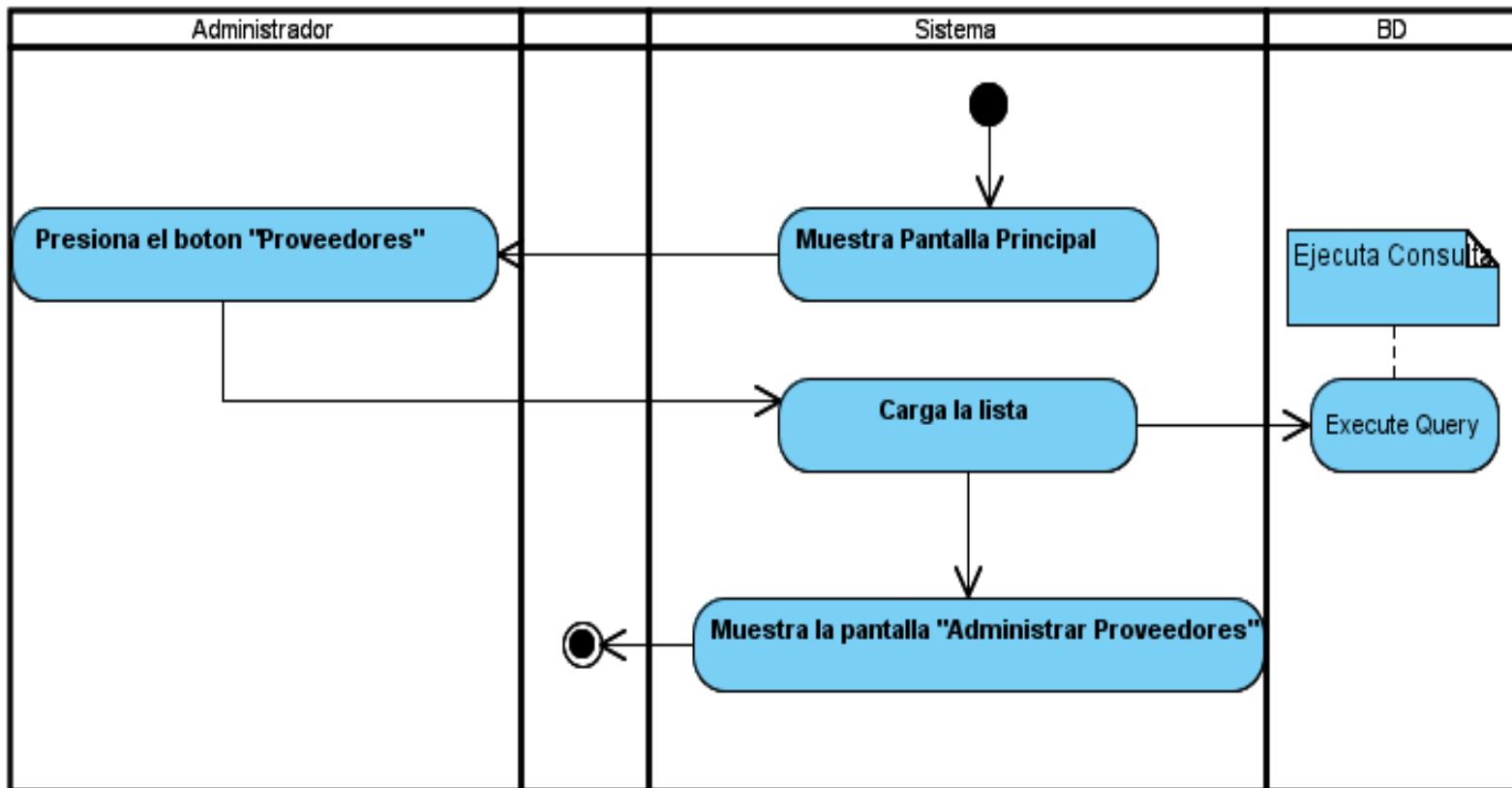


Figura 127: Diagrama de Actividades-Administrar Proveedores

Diagrama de actividad: C.U. Adicionar Proveedor

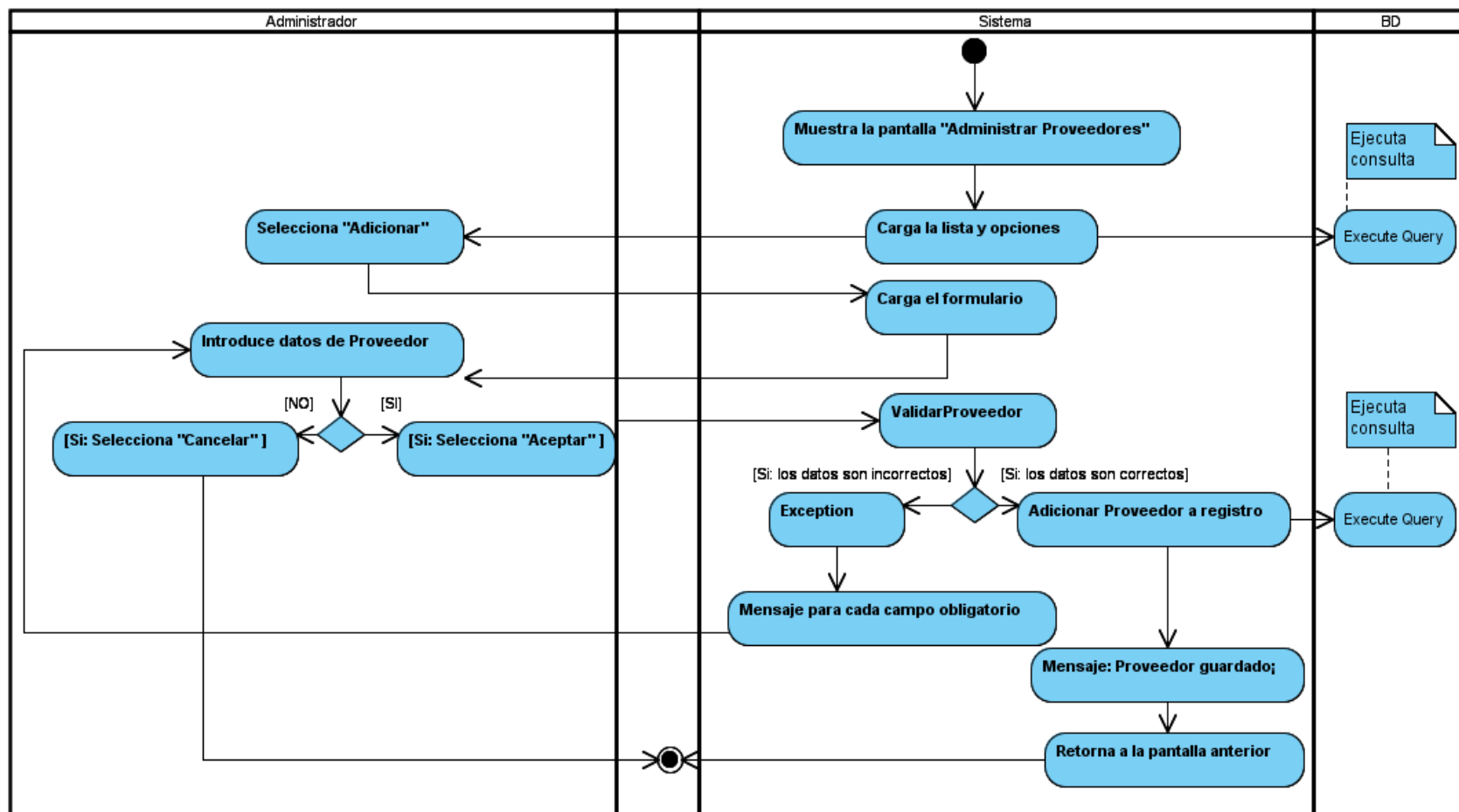


Figura 128: Diagrama de Actividades-Adicionar Proveedor

Diagrama de actividad: C.U. Modificar Proveedor

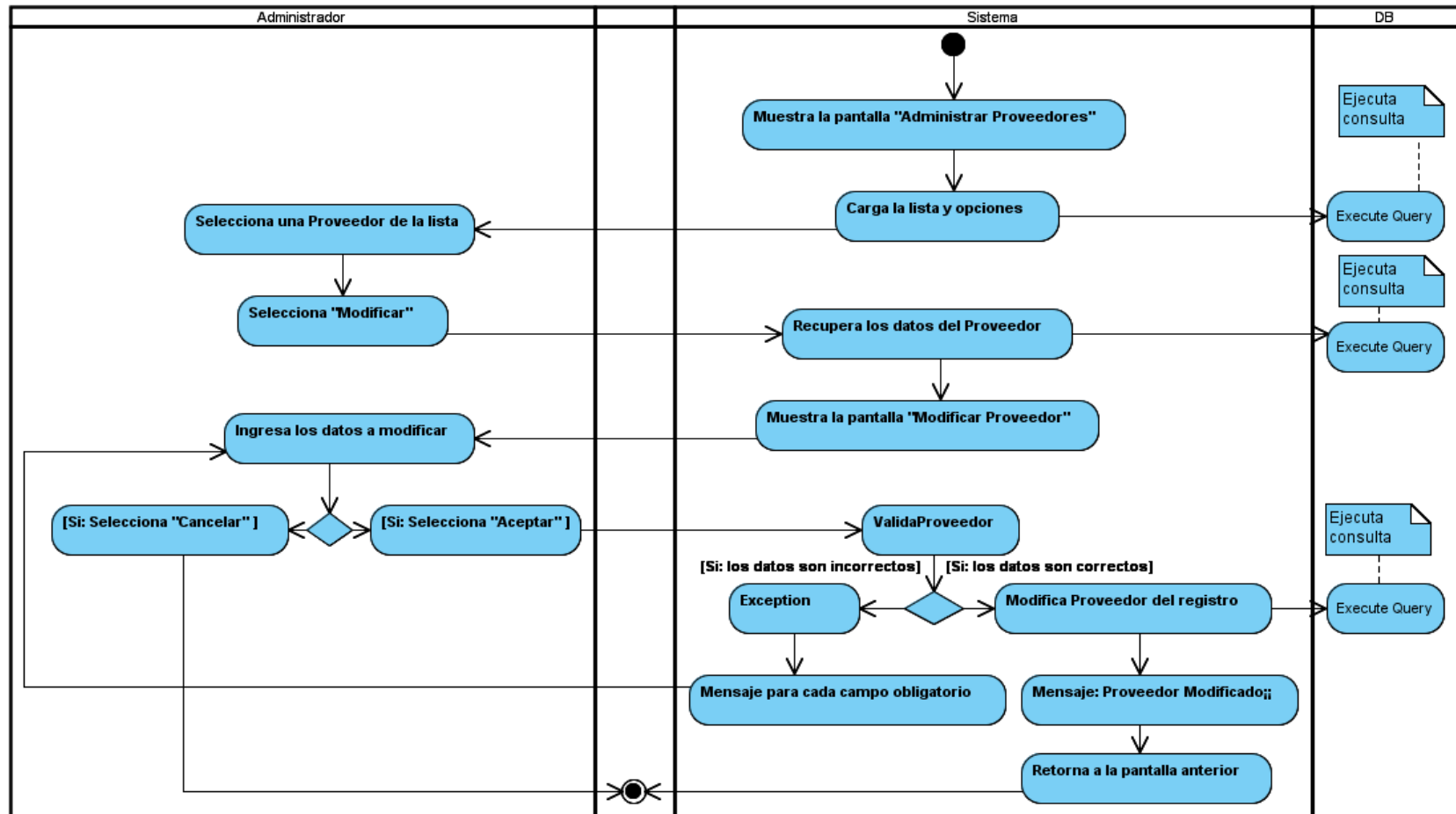


Figura 129: Diagrama de Actividades-Modificar Proveedor

Diagrama de actividad: C.U. Eliminar Proveedor

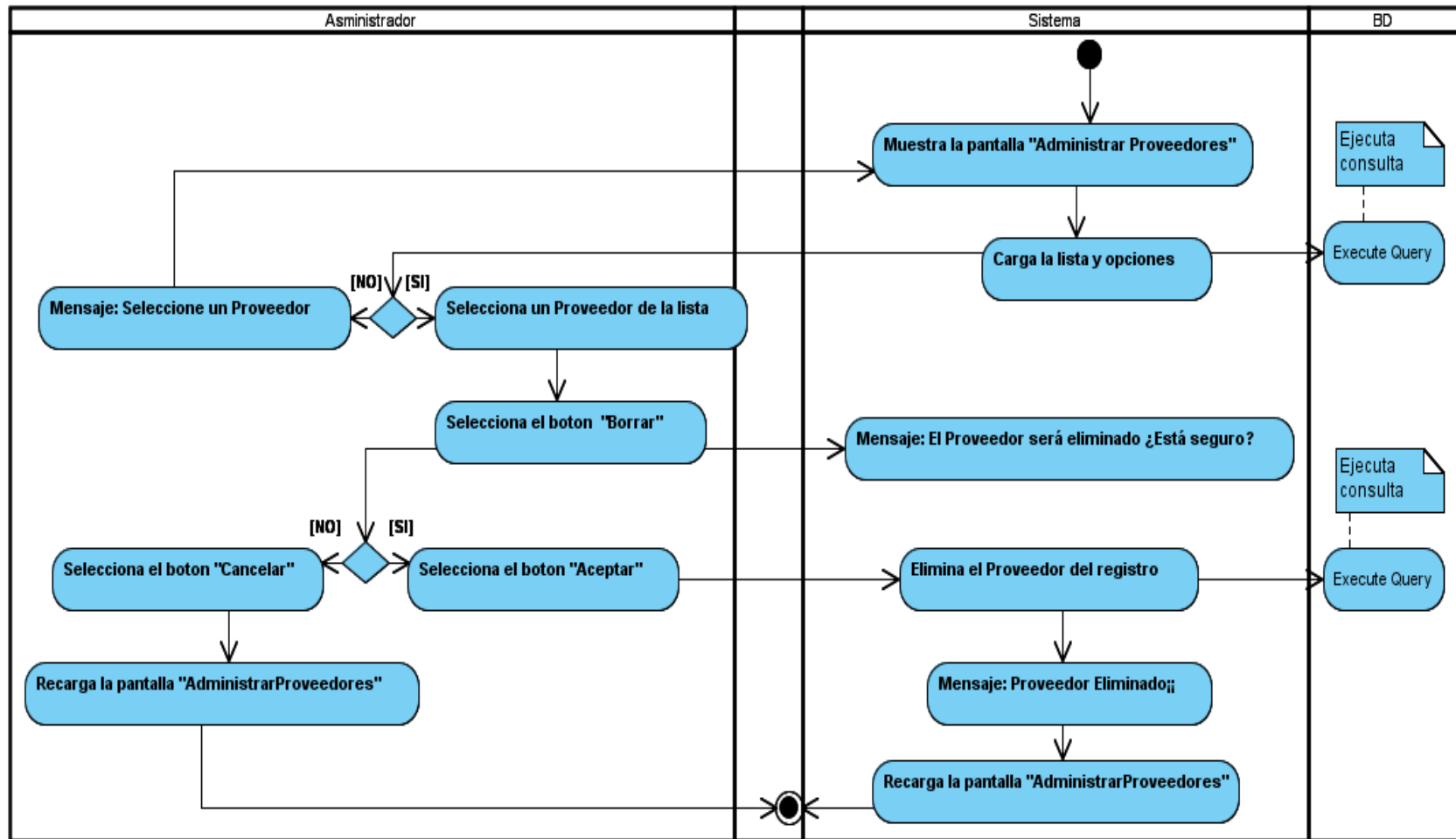


Figura 130: Diagrama de Actividades-Eliminar Proveedor

Diagrama de actividad: C.U. Administrar Estados

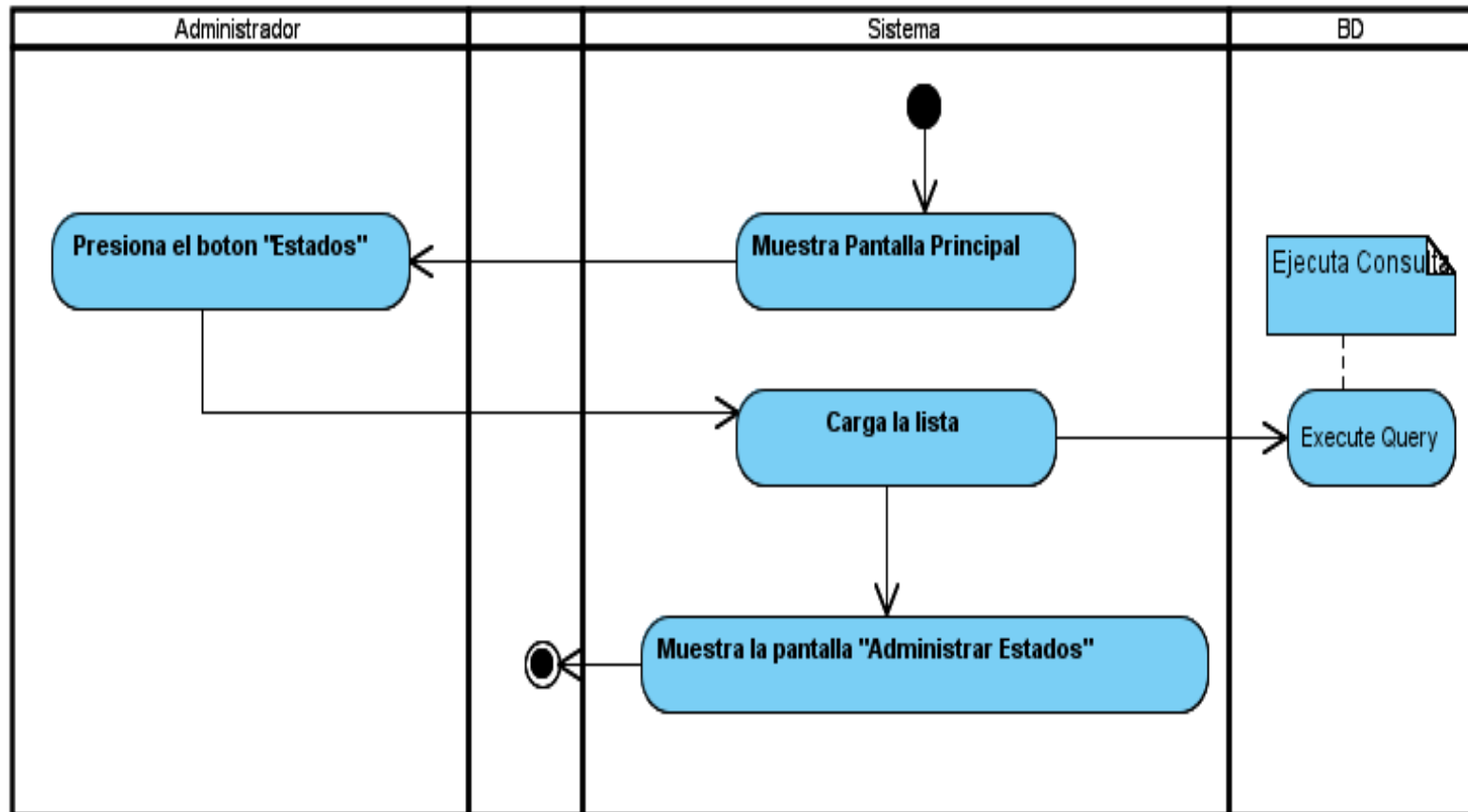


Figura 131: Diagrama de Actividades-Administrar Estados

Diagrama de actividad: C.U. Adicionar Estado

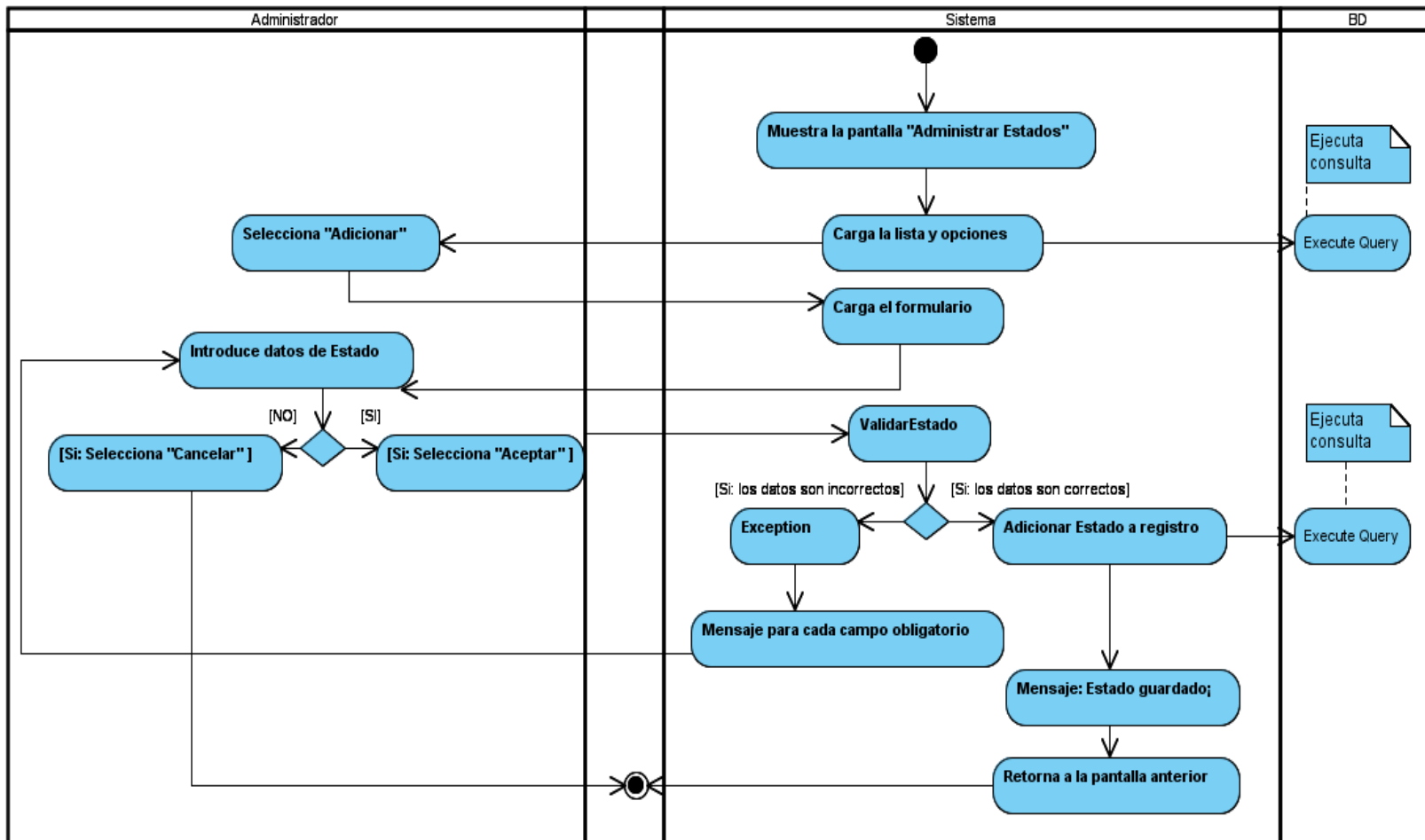


Figura 132: Diagrama de Actividades-Adicionar Estado

Diagrama de actividad: C.U. Modificar Estado

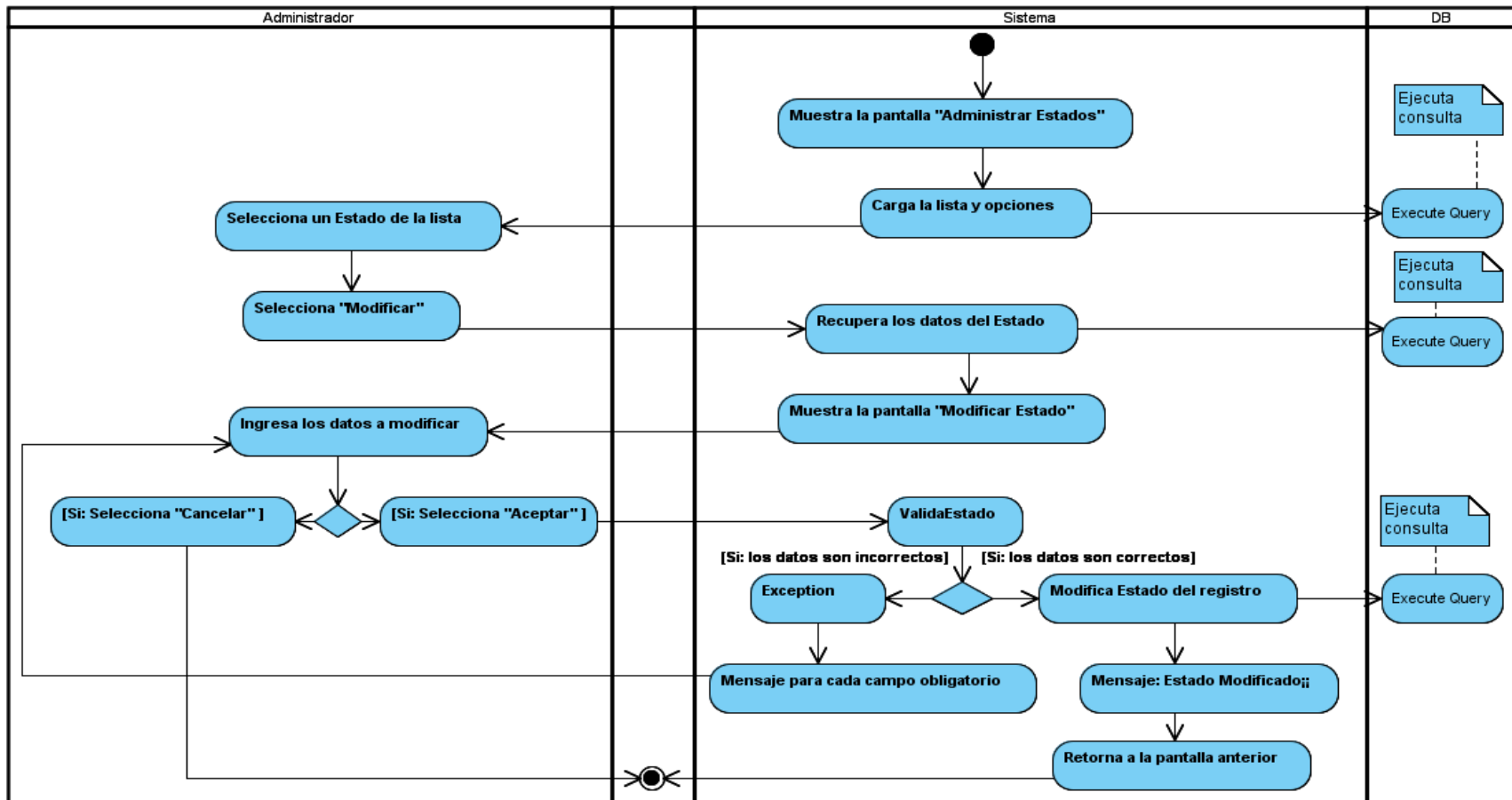


Figura 133: Diagrama de Actividades-Modificar Estado

Diagrama de actividad: C.U. Eliminar Estado

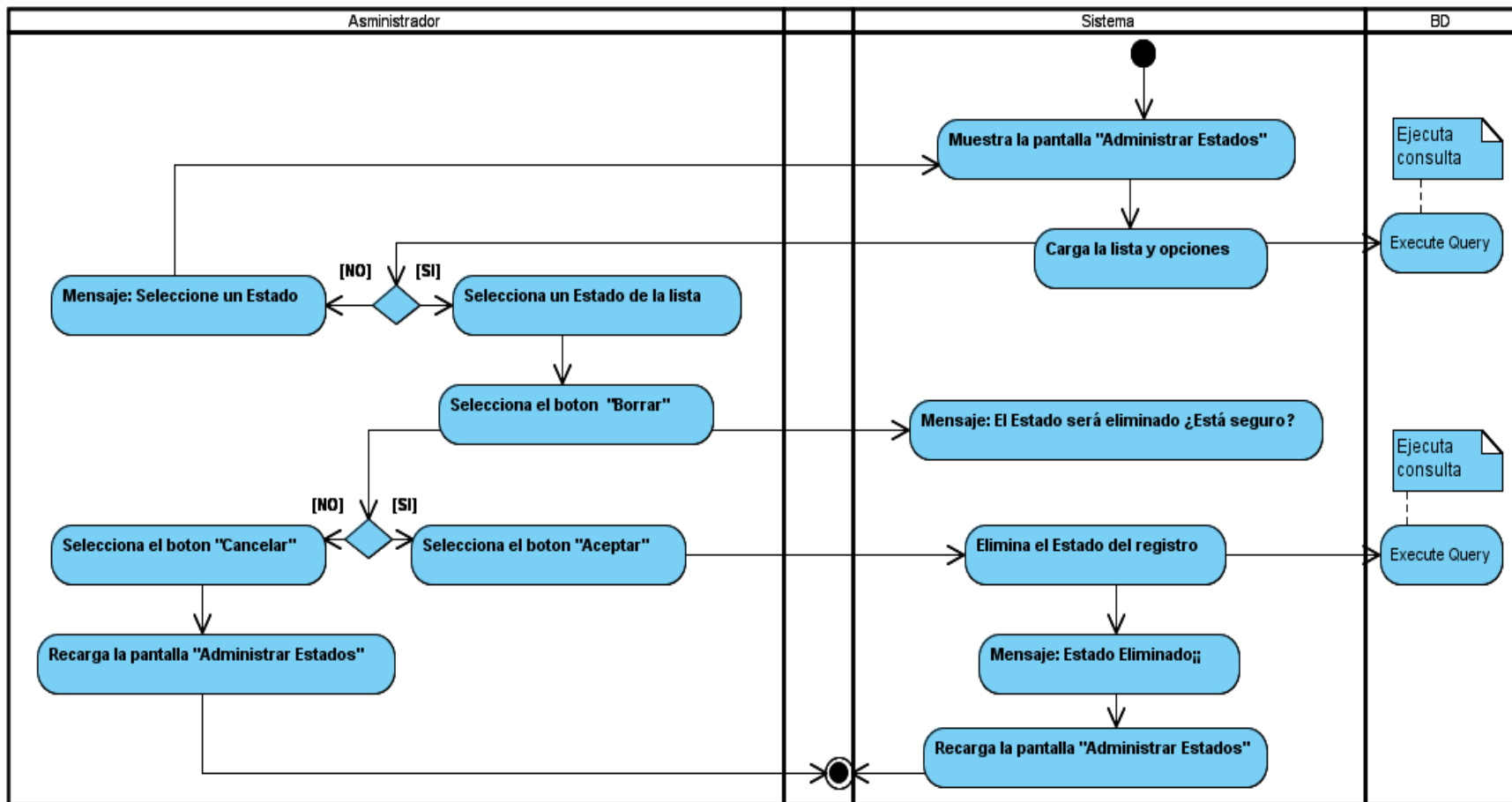


Figura 134: Diagrama de Actividades-Eliminar Estado

Diagrama de actividad: C.U. Buscar Persona

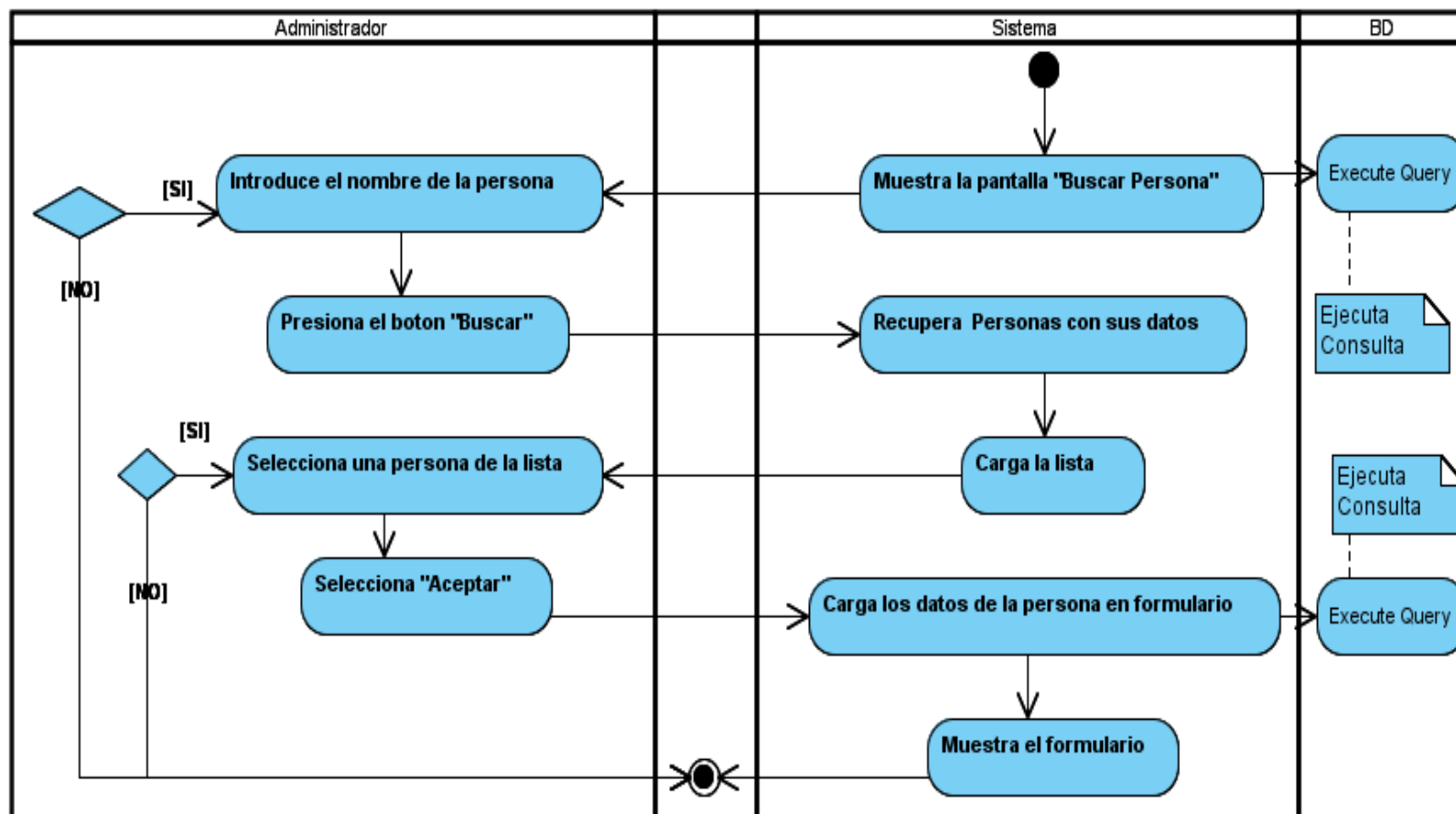


Figura 135: Diagrama de Actividades-Buscar Persona

Diagrama de actividad: C.U. Generar Reporte

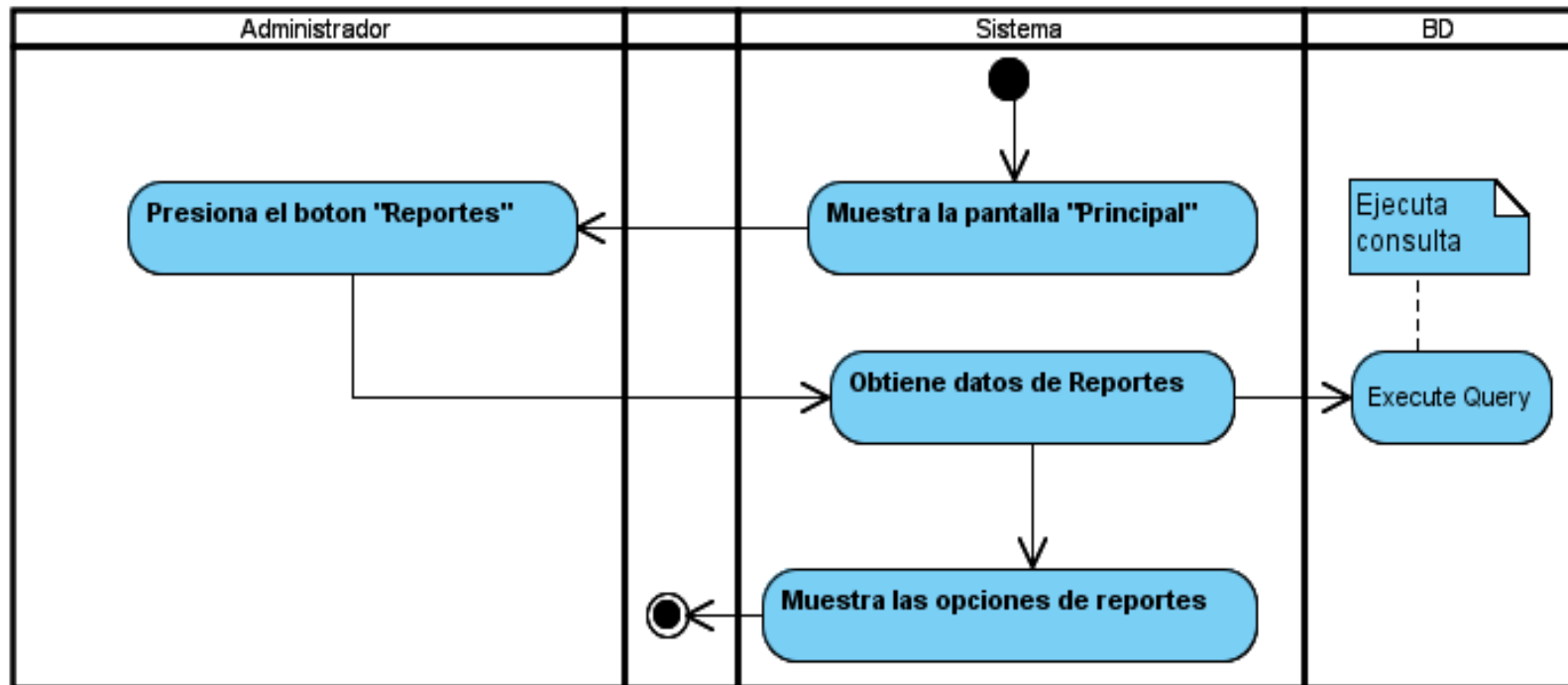


Figura 136: Diagrama de Actividades- Generar Reportes

Diagrama de actividad: C.U. Reporte Ventas

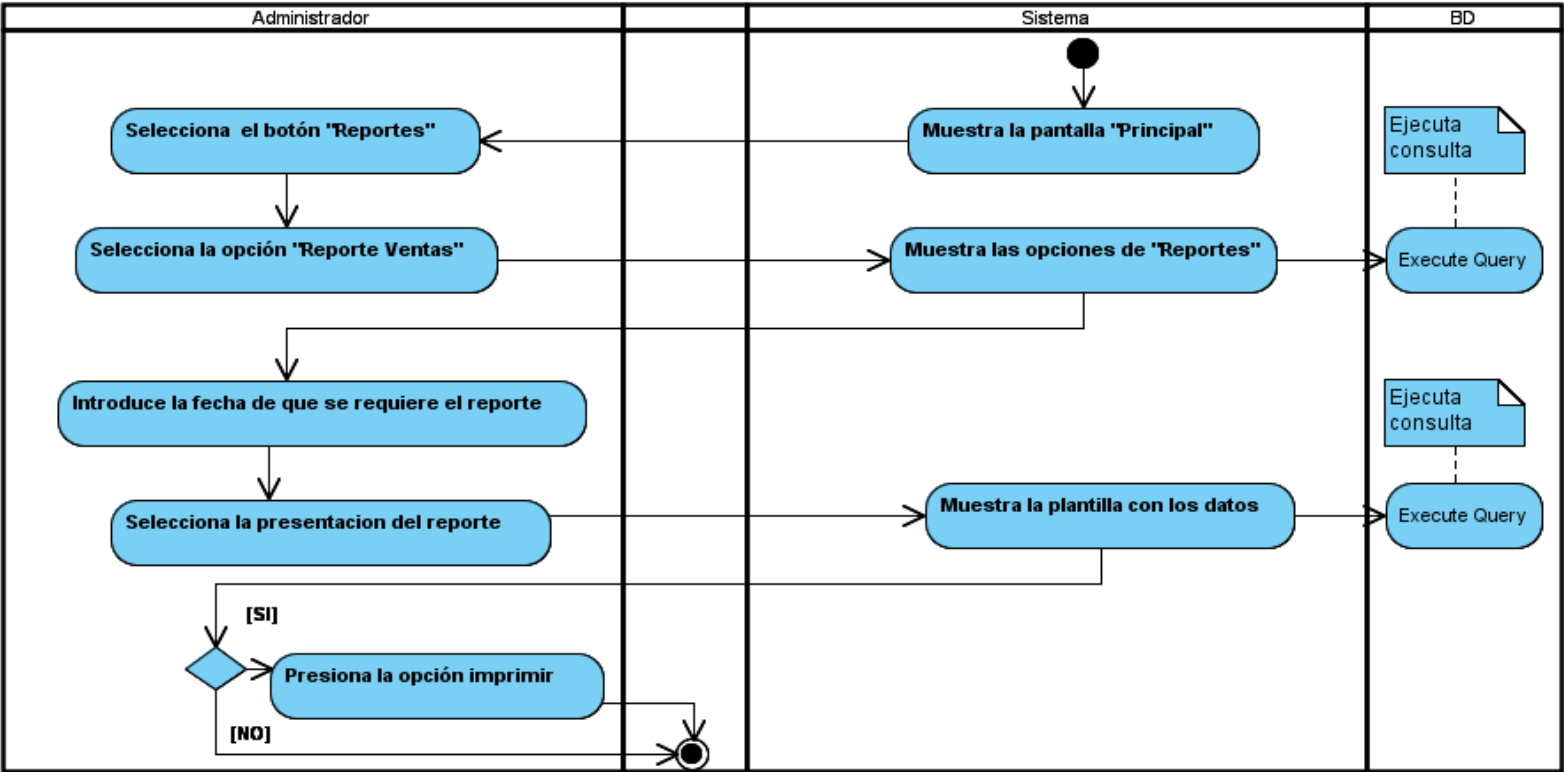


Figura 137: Diagrama de Actividades- Reporte Ventas

Diagrama de actividad: C.U. Reporte Cantidad Compras Clientes

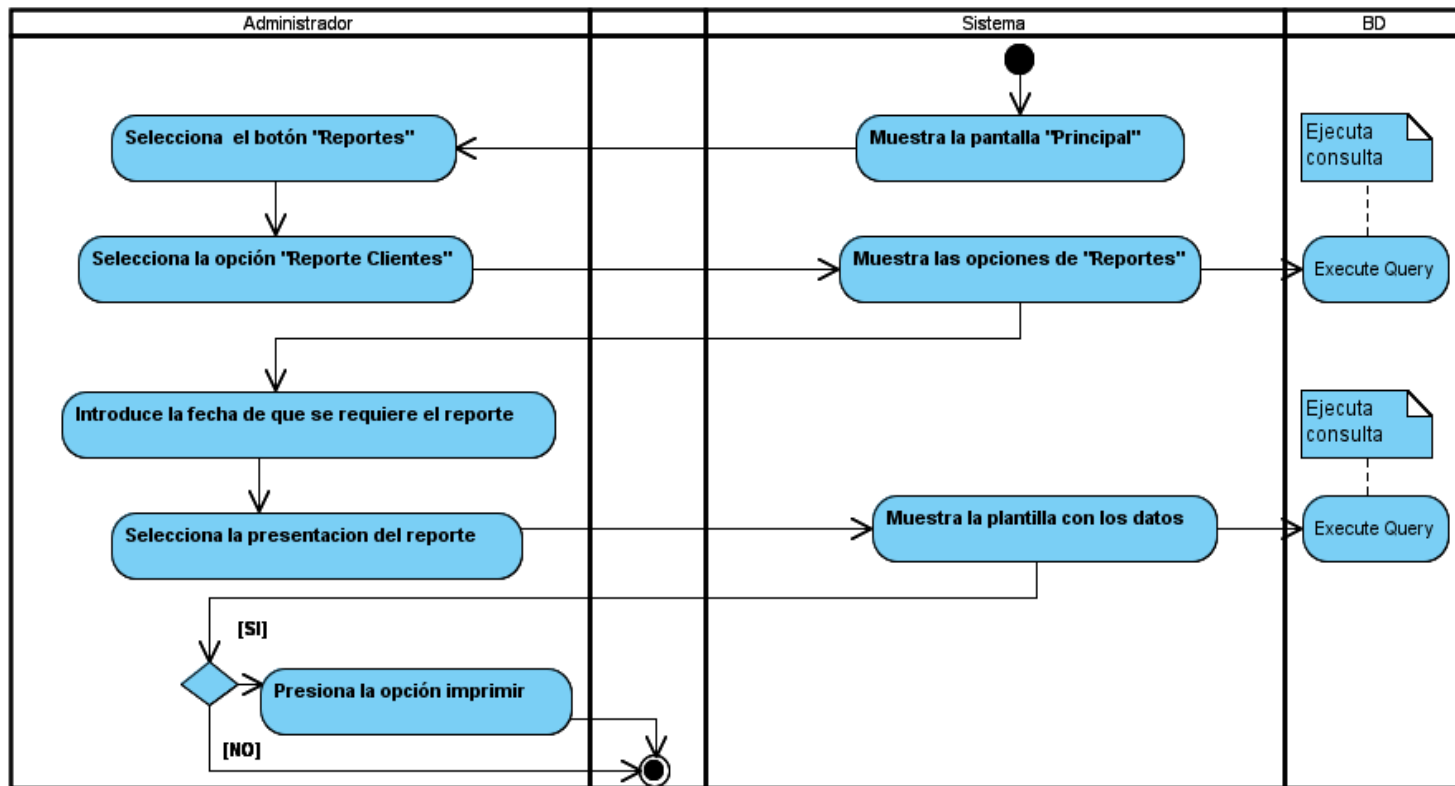


Figura 138: Diagrama de Actividades- Reporte Clientes

Diagrama de actividad: C.U. Reporte Productos Almacenados

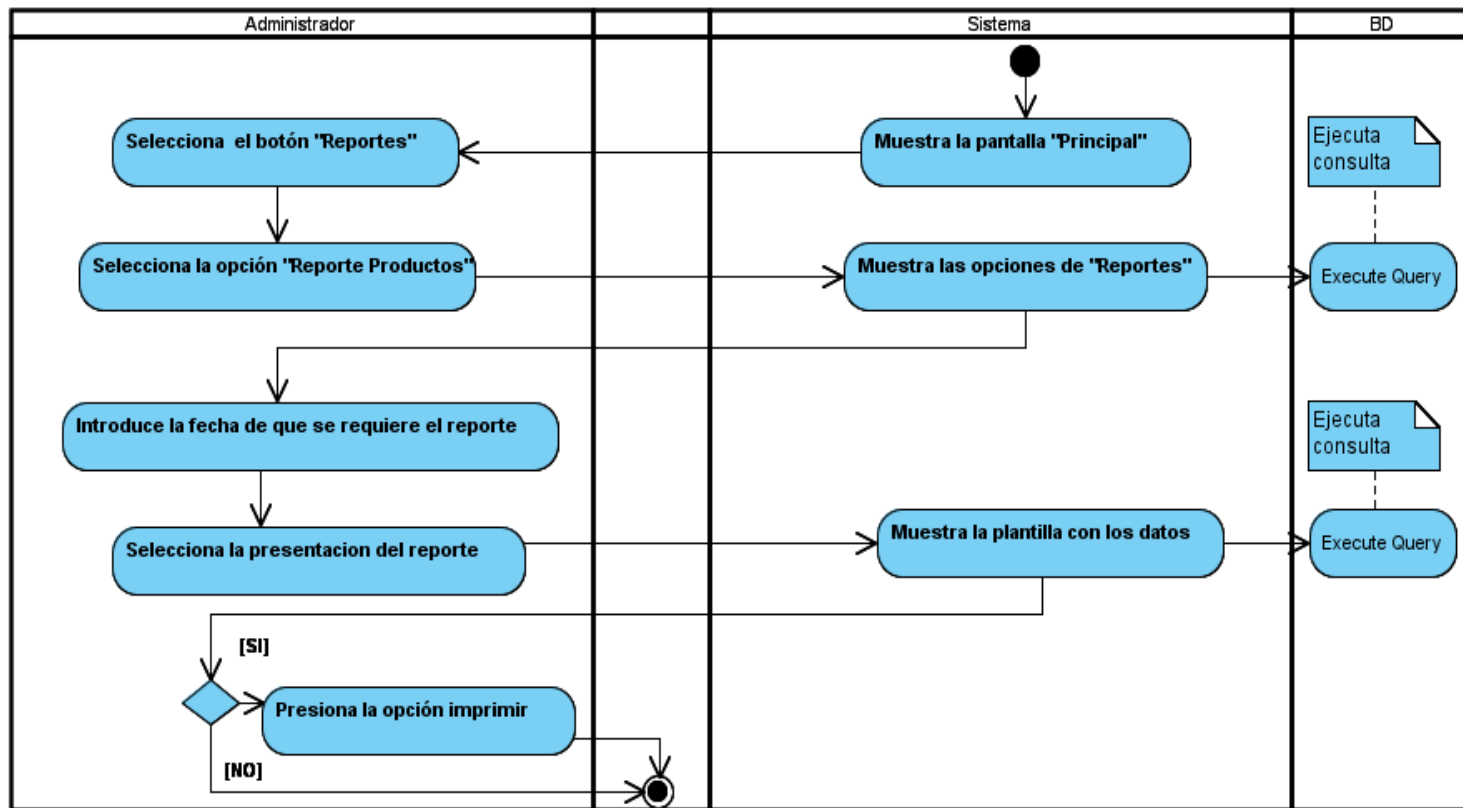


Figura 139: Diagrama de Actividades- Reporte Productos Almacenados

2.1.2.2.3.1.1 DIAGRAMAS DE SECUENCIA

Diagrama de secuencia: C.U. Ingresar al Sistema

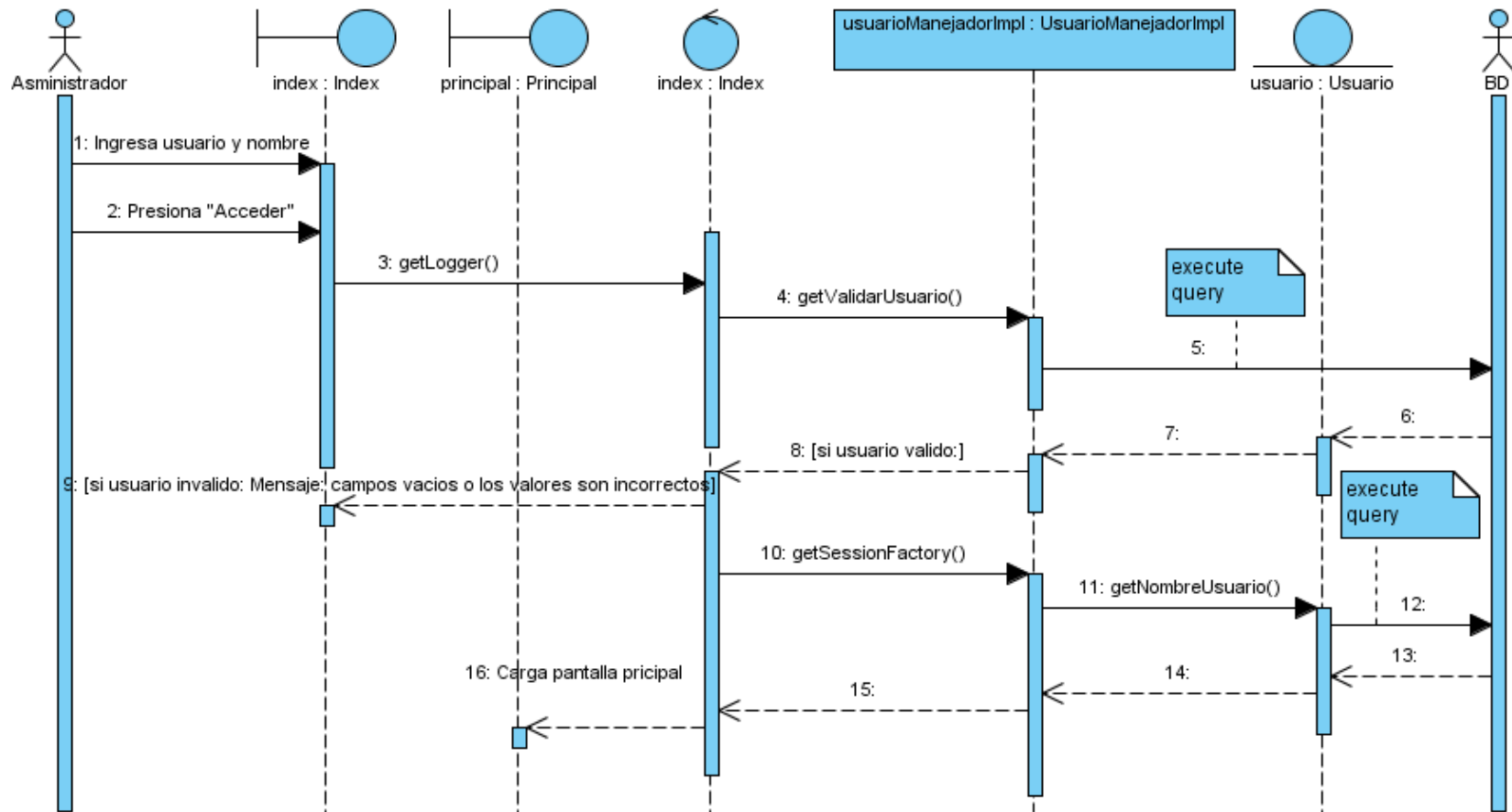


Figura 140: Diagrama de Secuencia- Ingresar al Sistema

Diagrama de secuencia: C.U. Administrar Usuario

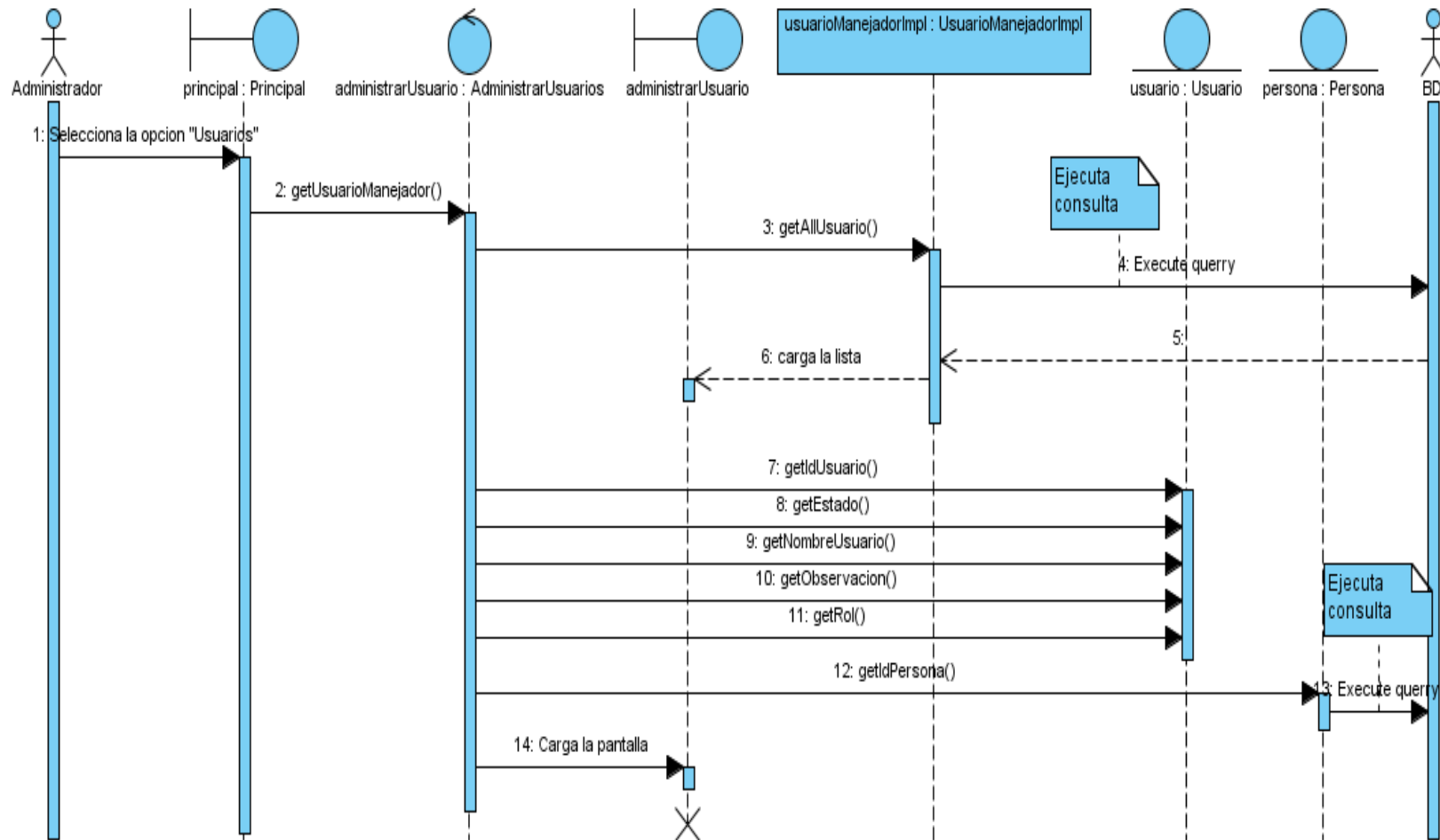


Figura 141: Diagrama de Secuencia-Administrar Usuario

Diagrama de secuencia: C.U. Adicionar Usuario

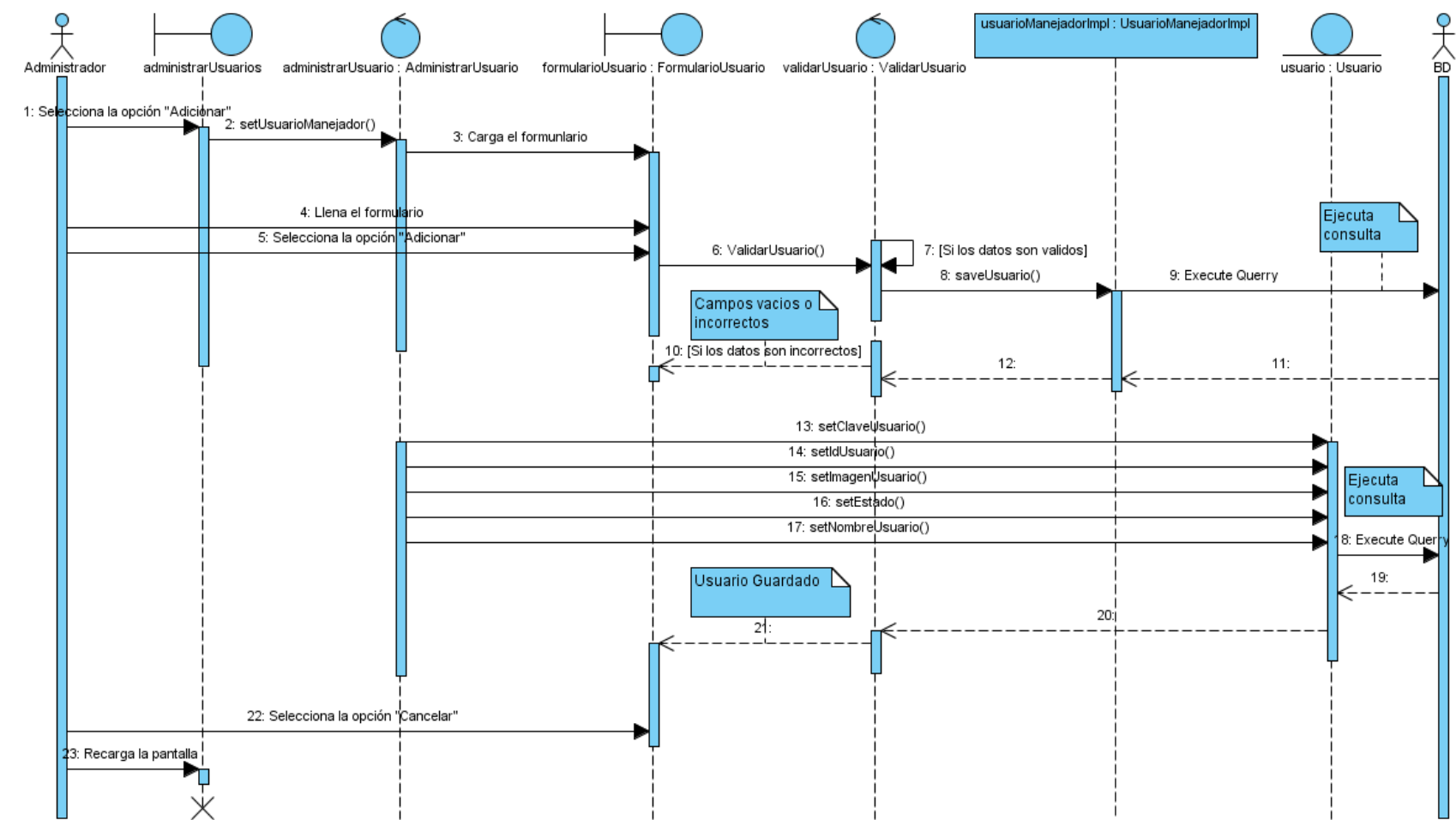


Figura 142: Diagrama de Secuencia-Adicionar Usuario

Diagrama de secuencia: C.U. Modificar Usuario

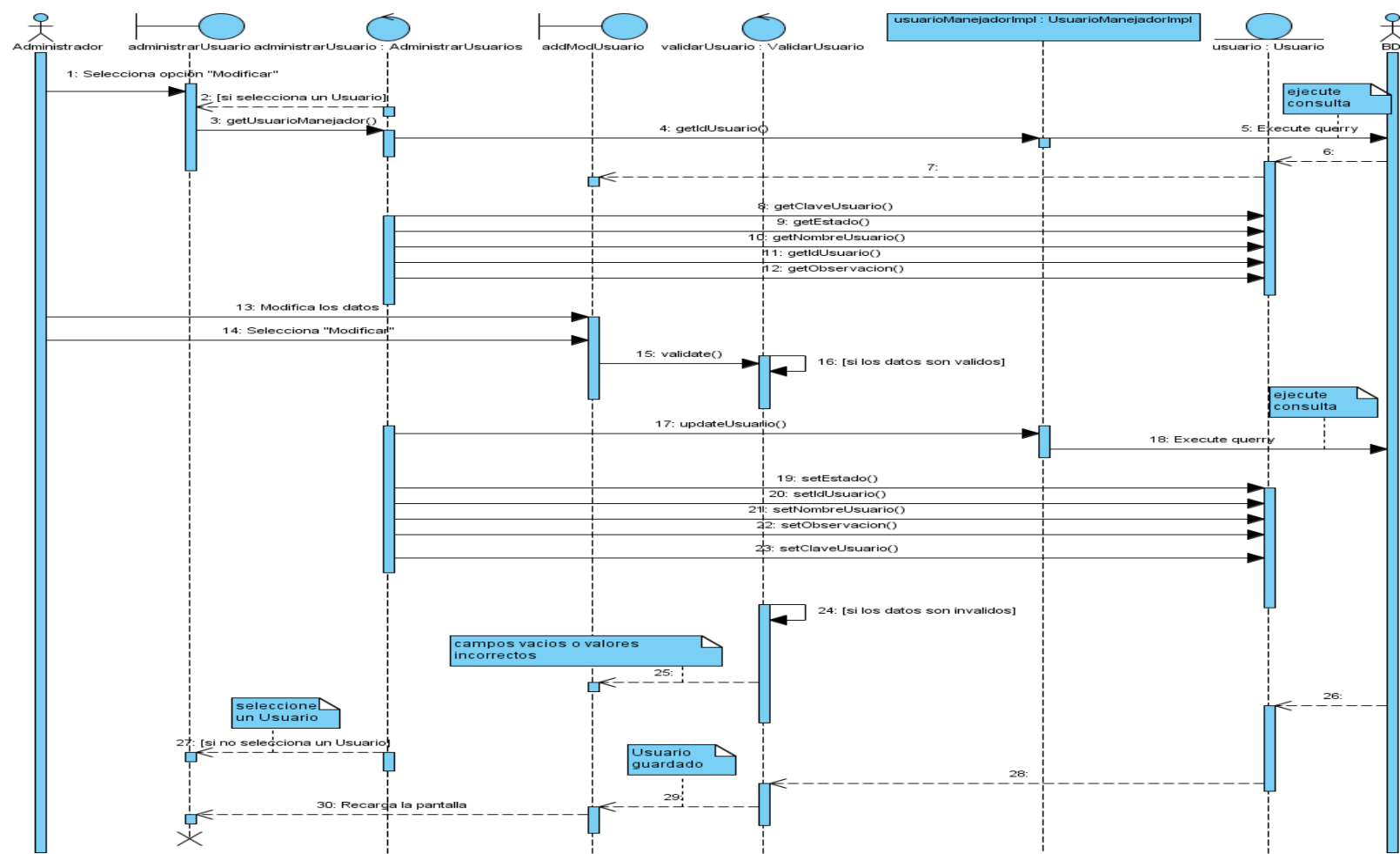


Figura 143: Diagrama de Secuencia-Modificar Usuario

Diagrama de secuencia: C.U. Eliminar Usuario

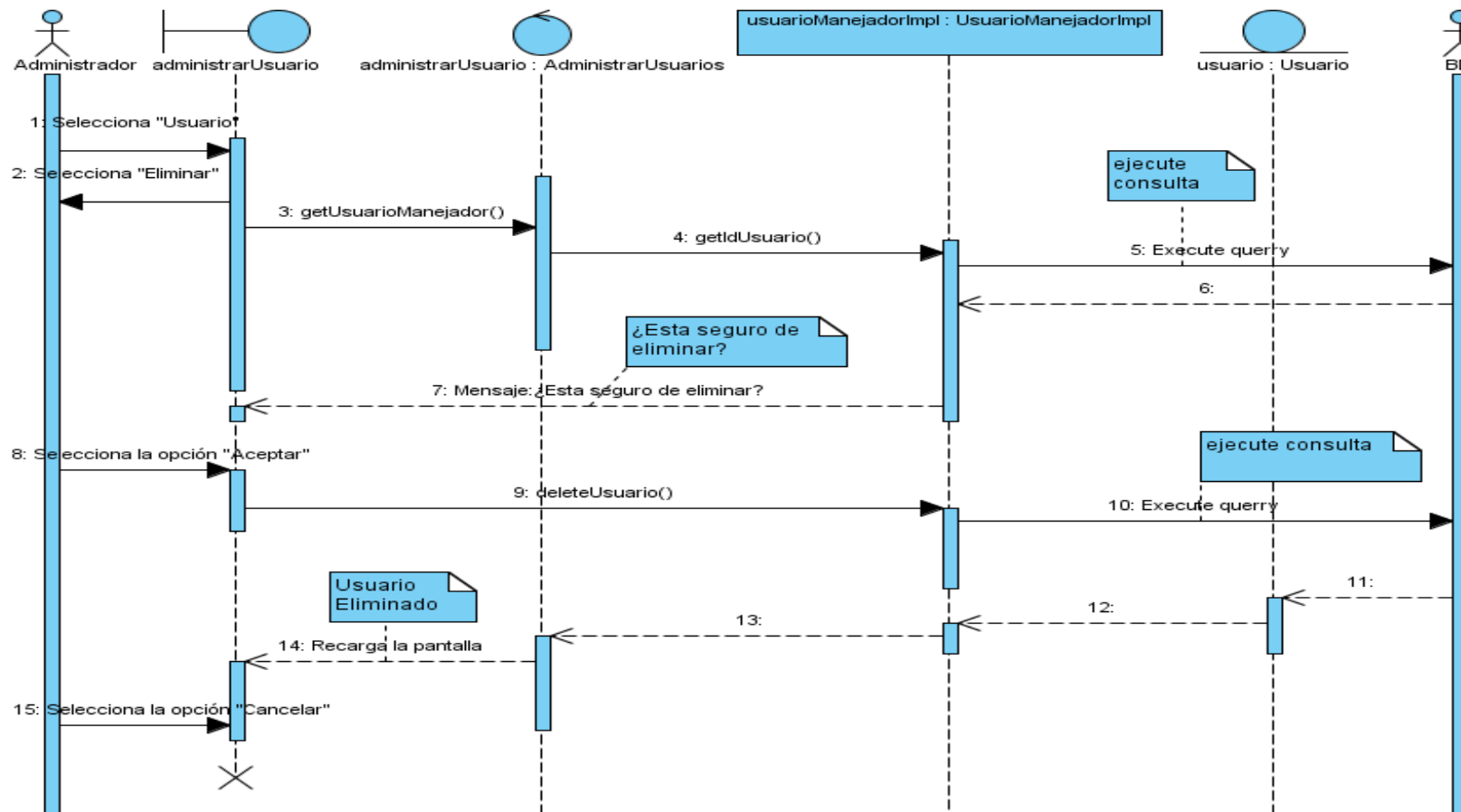


Figura 144: Diagrama de Secuencia-Eliminar Usuario

Diagrama de secuencia: C.U. Administrar Roles

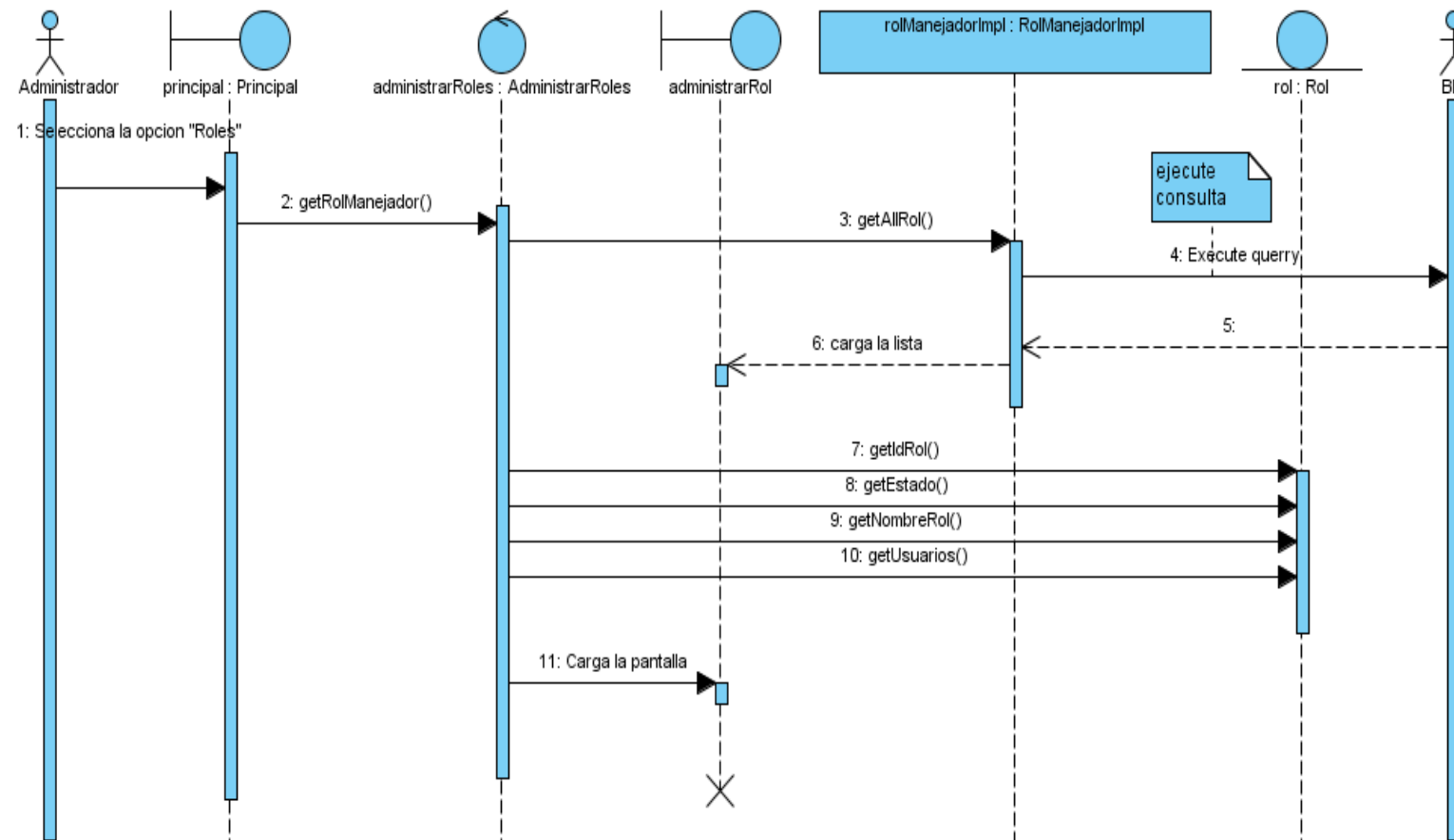


Figura 145: Diagrama de Secuencia-Administrar Roles

Diagrama de secuencia: C.U. Adicionar Rol

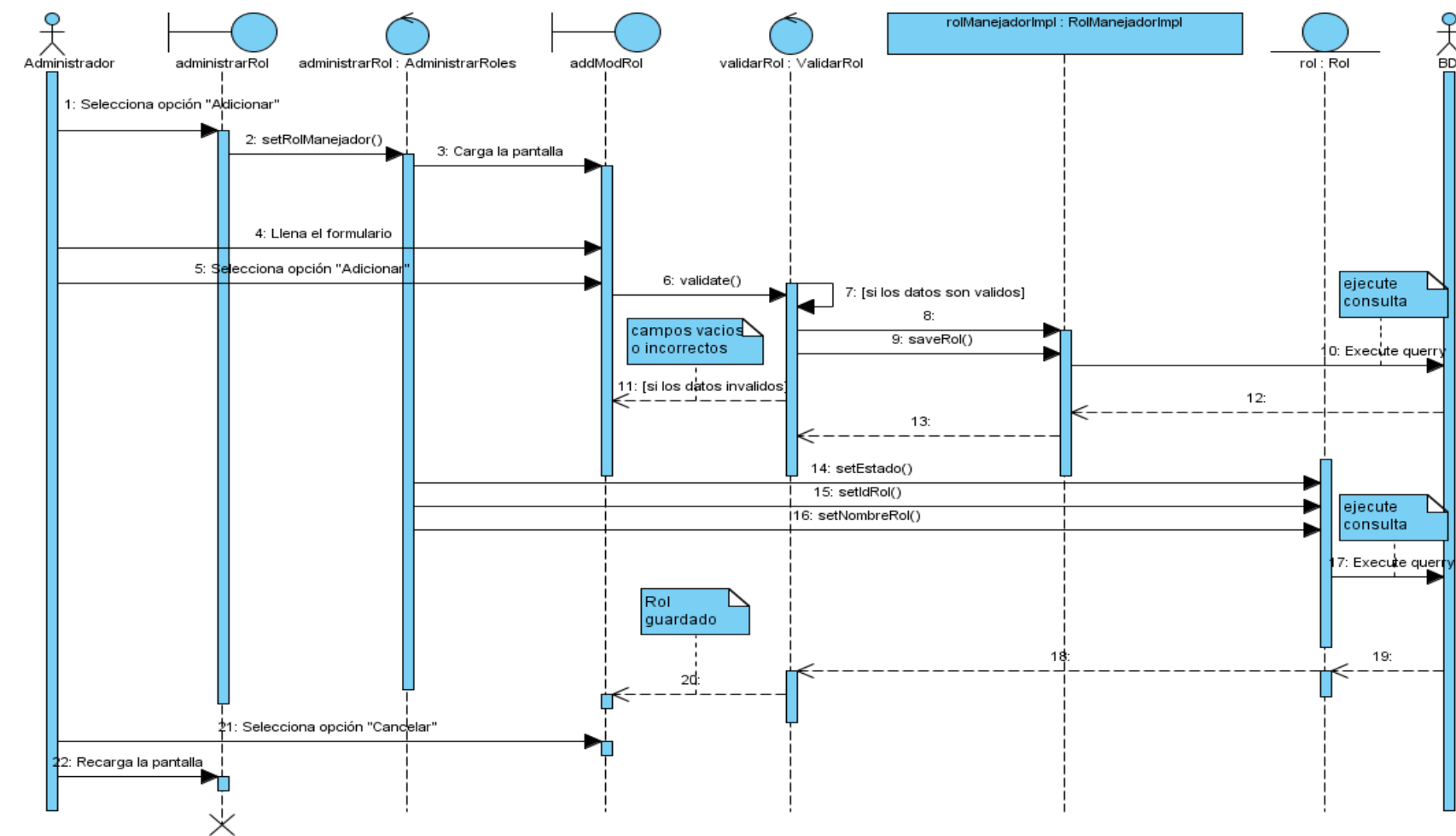


Figura 146: Diagrama de Secuencia-Adicionar Rol

Diagrama de secuencia: C.U. Modificar Rol

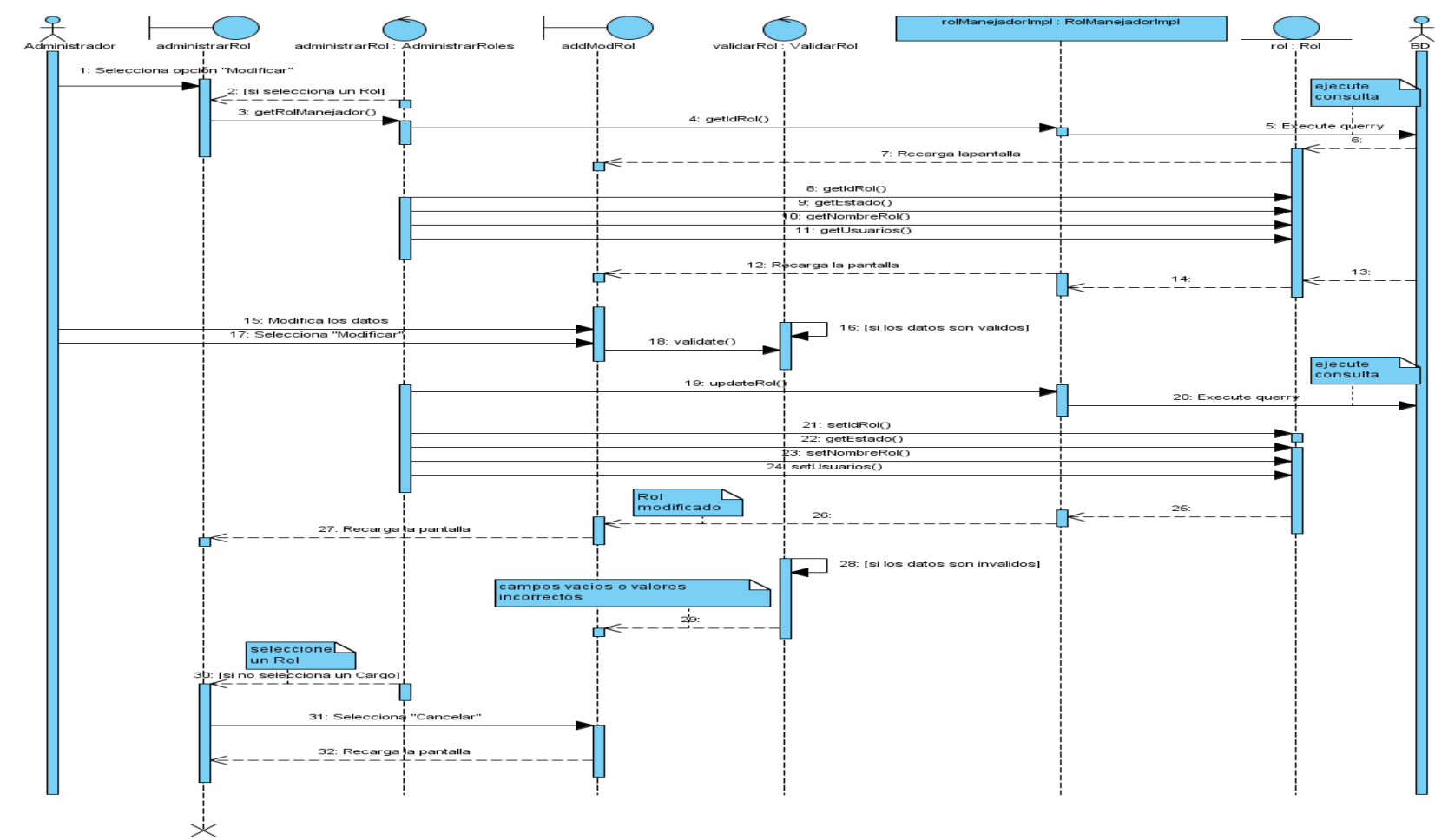


Figura 147: Diagrama de Secuencia-Modificar Rol

Diagrama de secuencia: C.U. Eliminar Rol

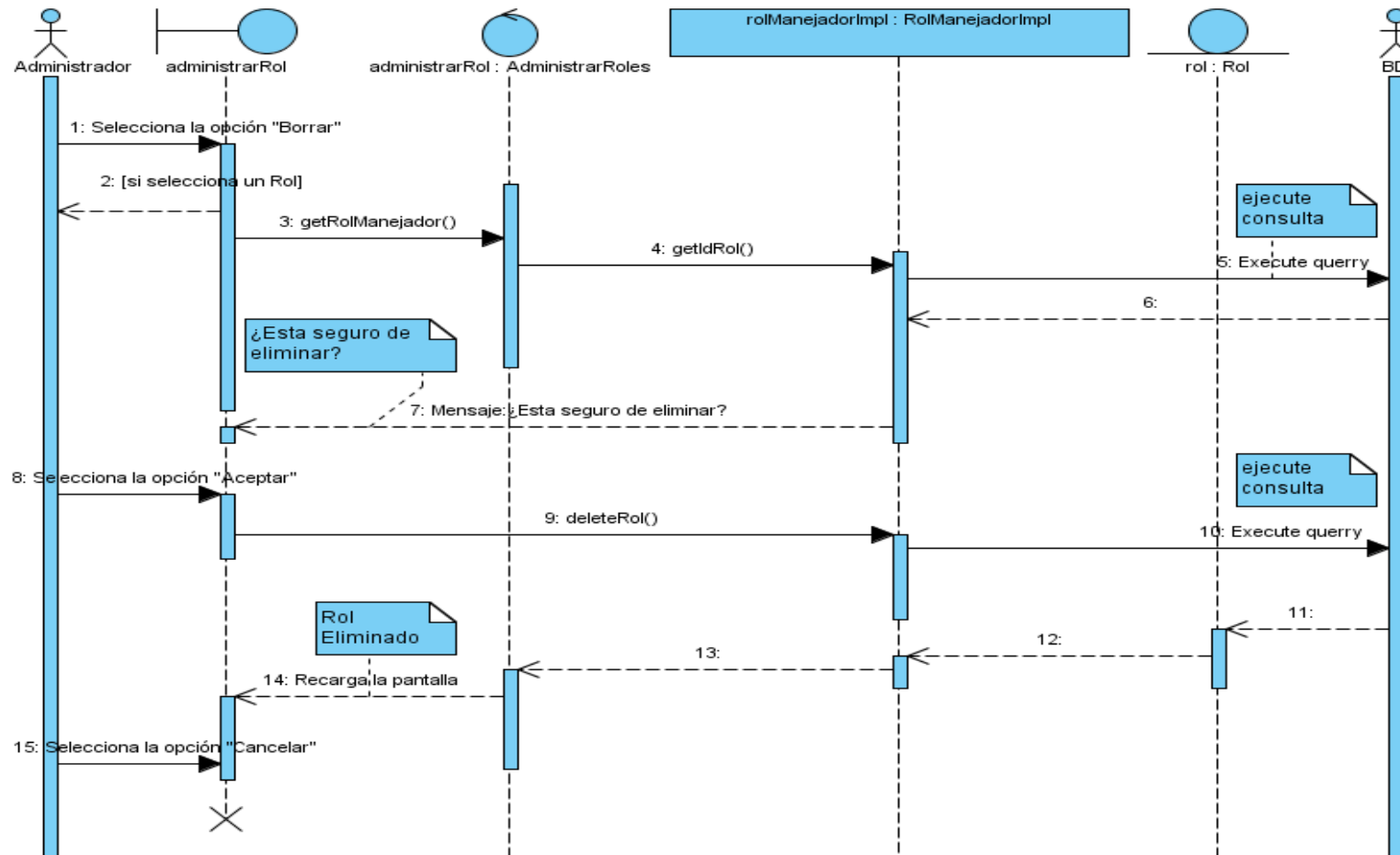


Figura 148: Diagrama de Secuencia-Eliminar Rol

Diagrama de secuencia: C.U. Administrar Roles

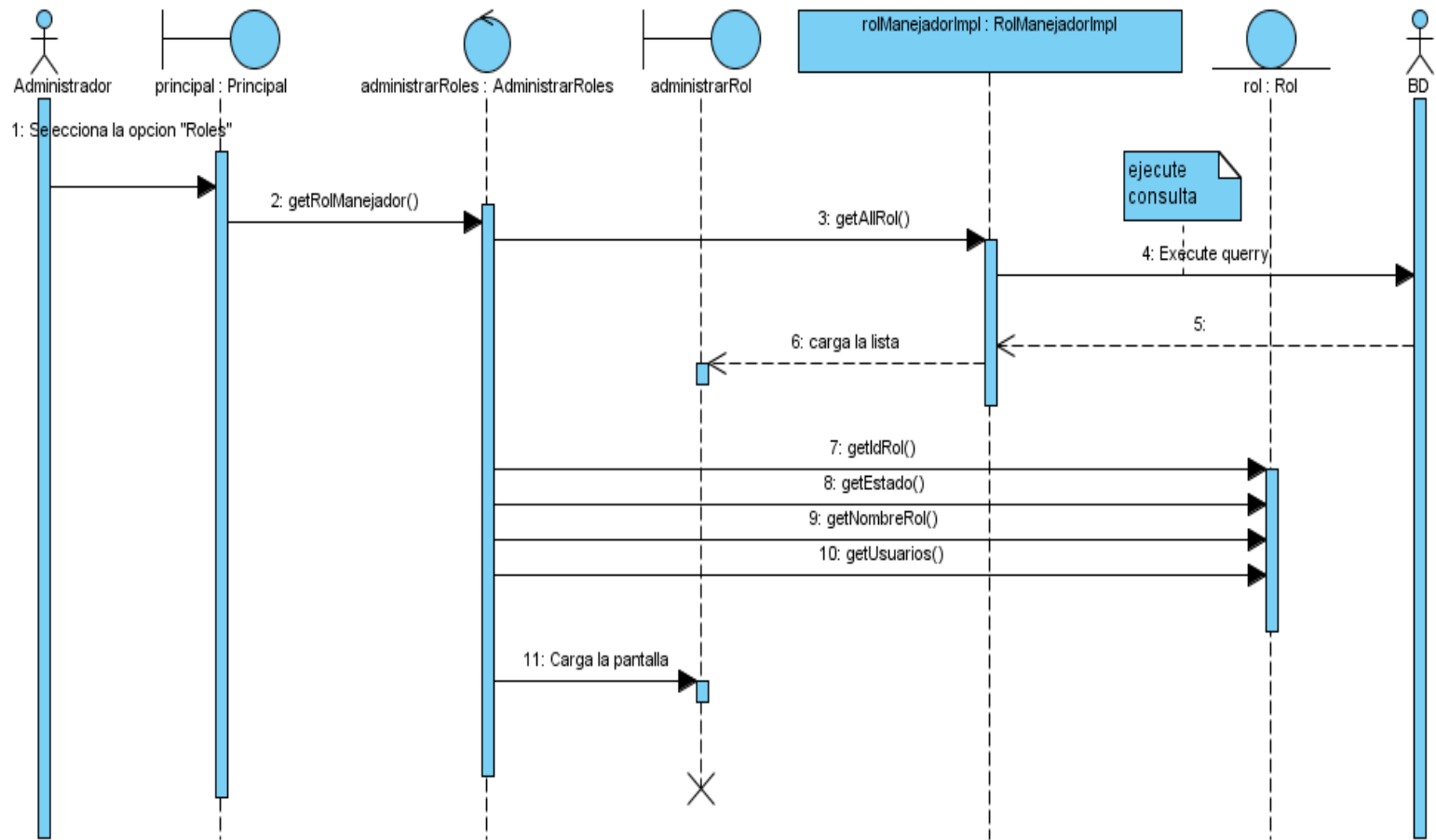


Figura 149: Diagrama de Secuencia-Administrar Roles

Diagrama de secuencia: C.U. Adicionar Rol

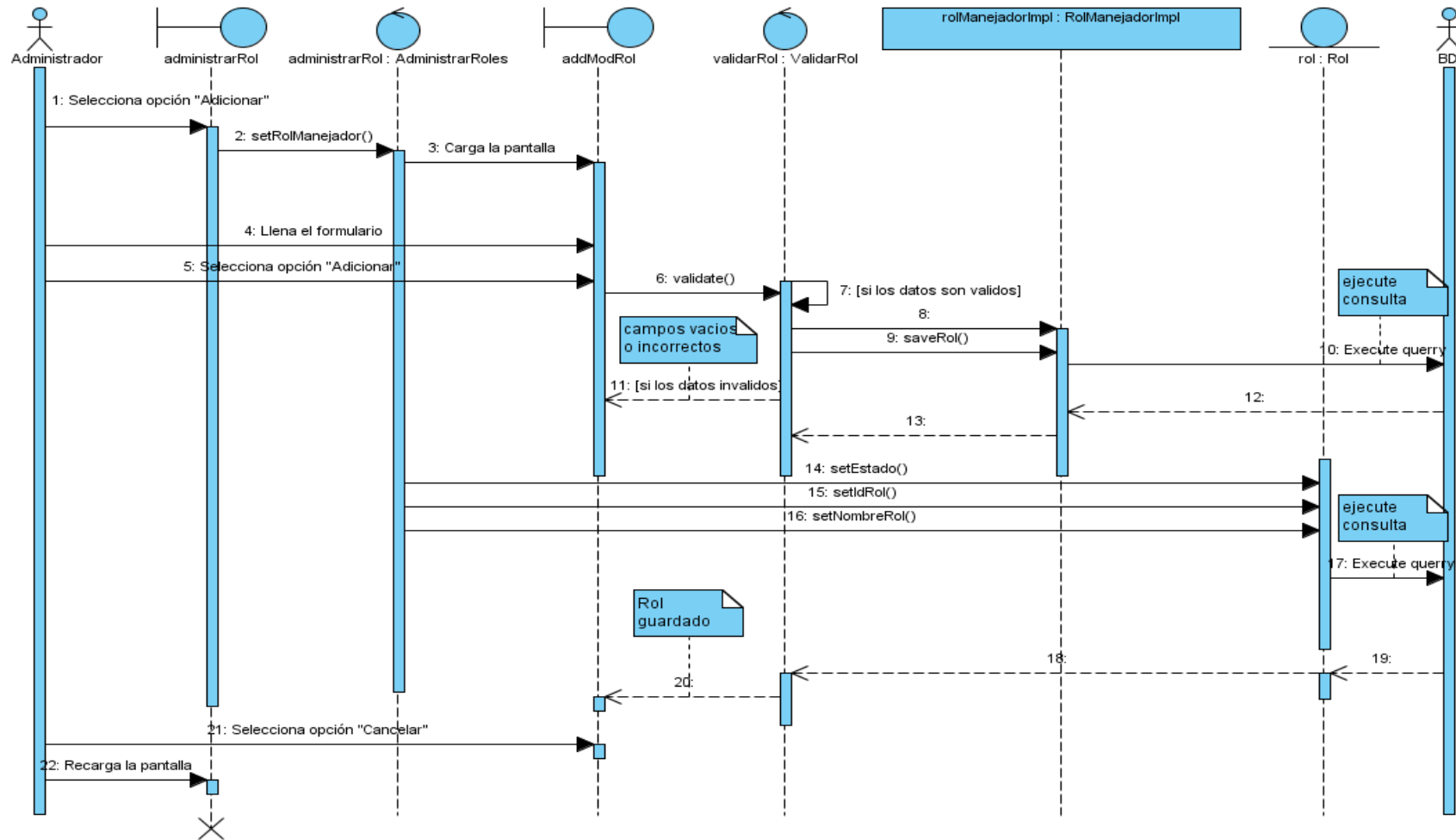


Figura 150: Diagrama de Secuencia-Agregar Rol

Diagrama de secuencia: C.U. Modificar Rol

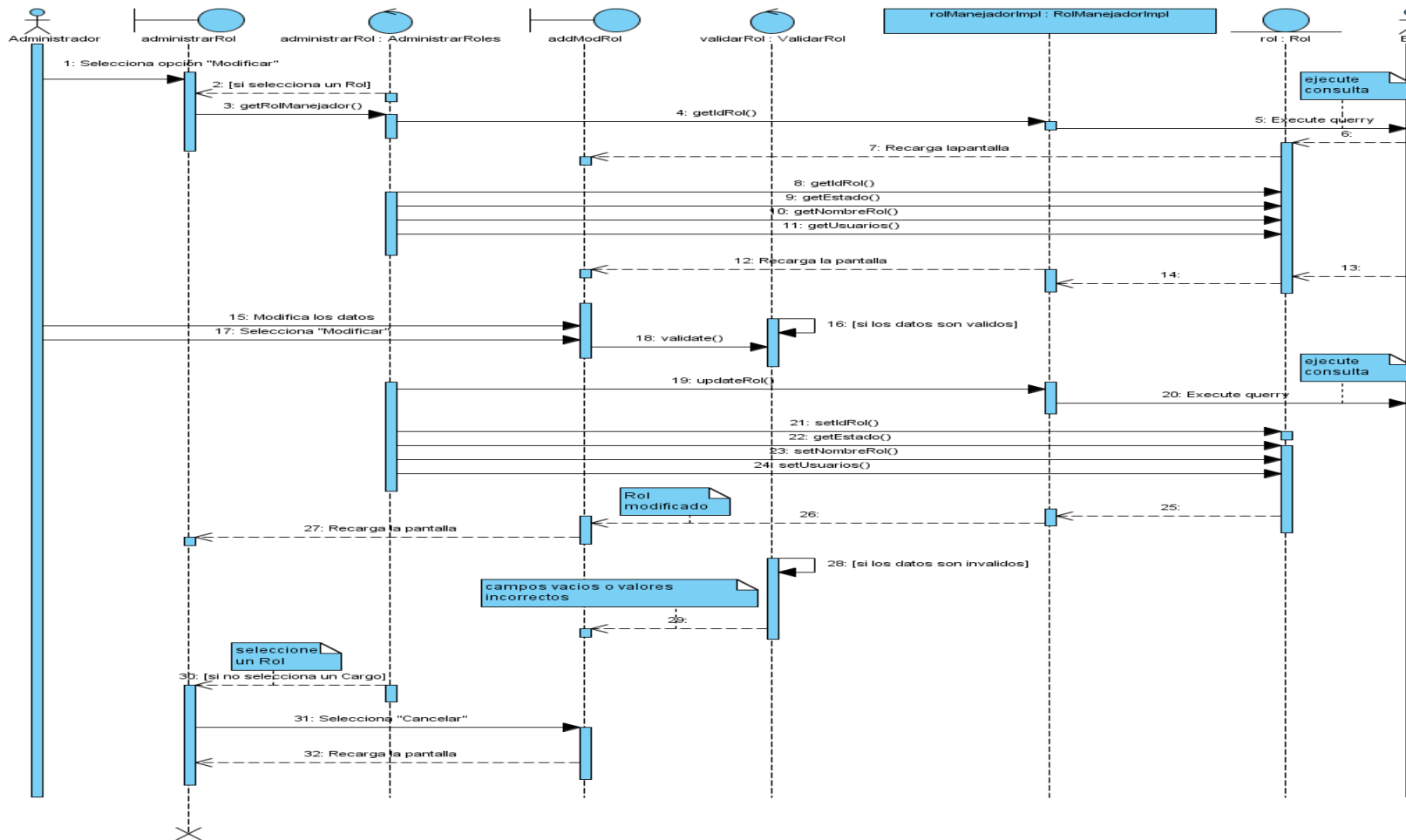


Figura 151: Diagrama de Secuencia-Modificar Rol

Diagrama de secuencia: C.U. Eliminar Rol

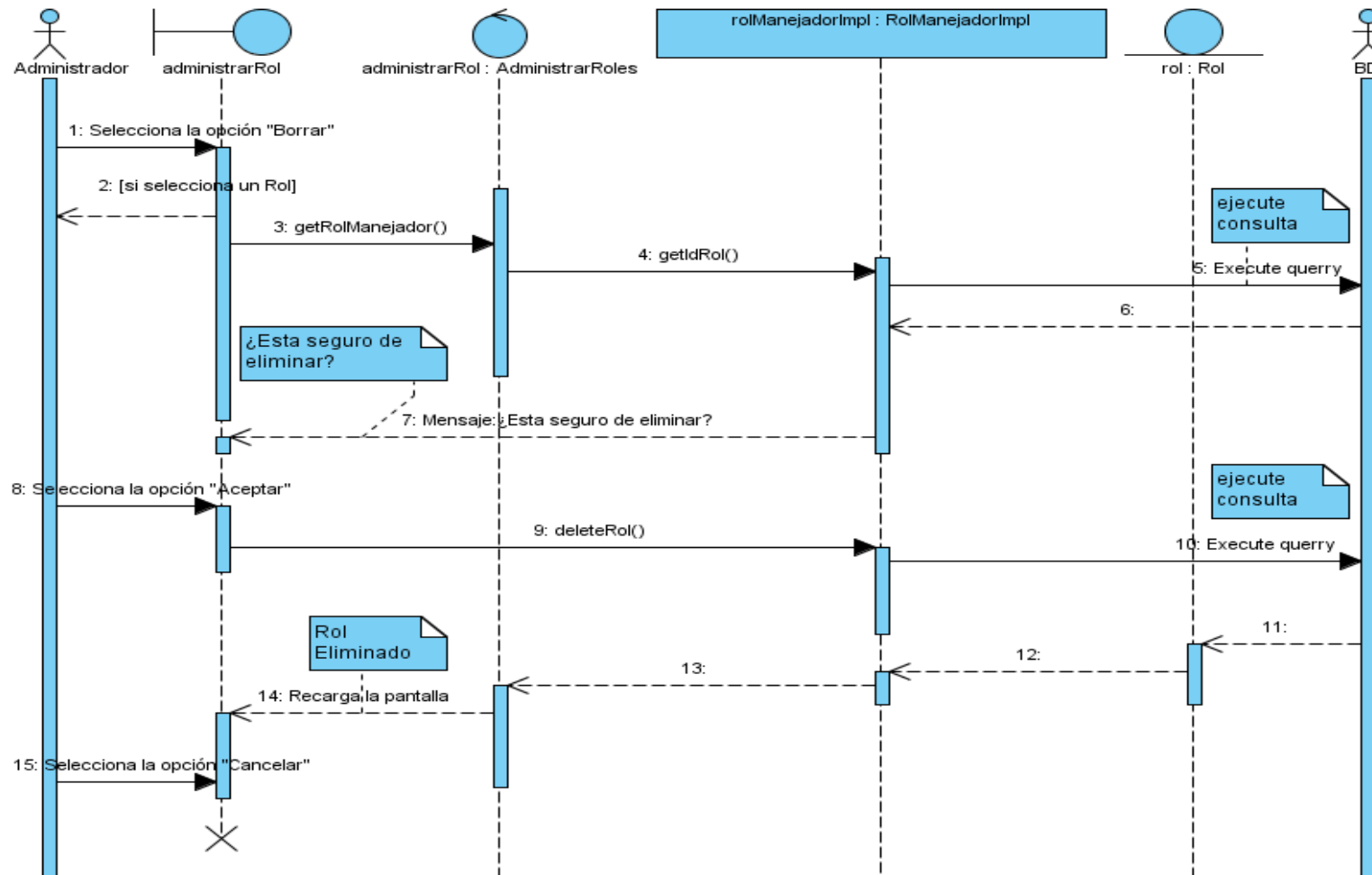


Figura 152: Diagrama de Secuencia-Eliminar Rol

Diagrama de secuencia: C.U. Administrar Empleado

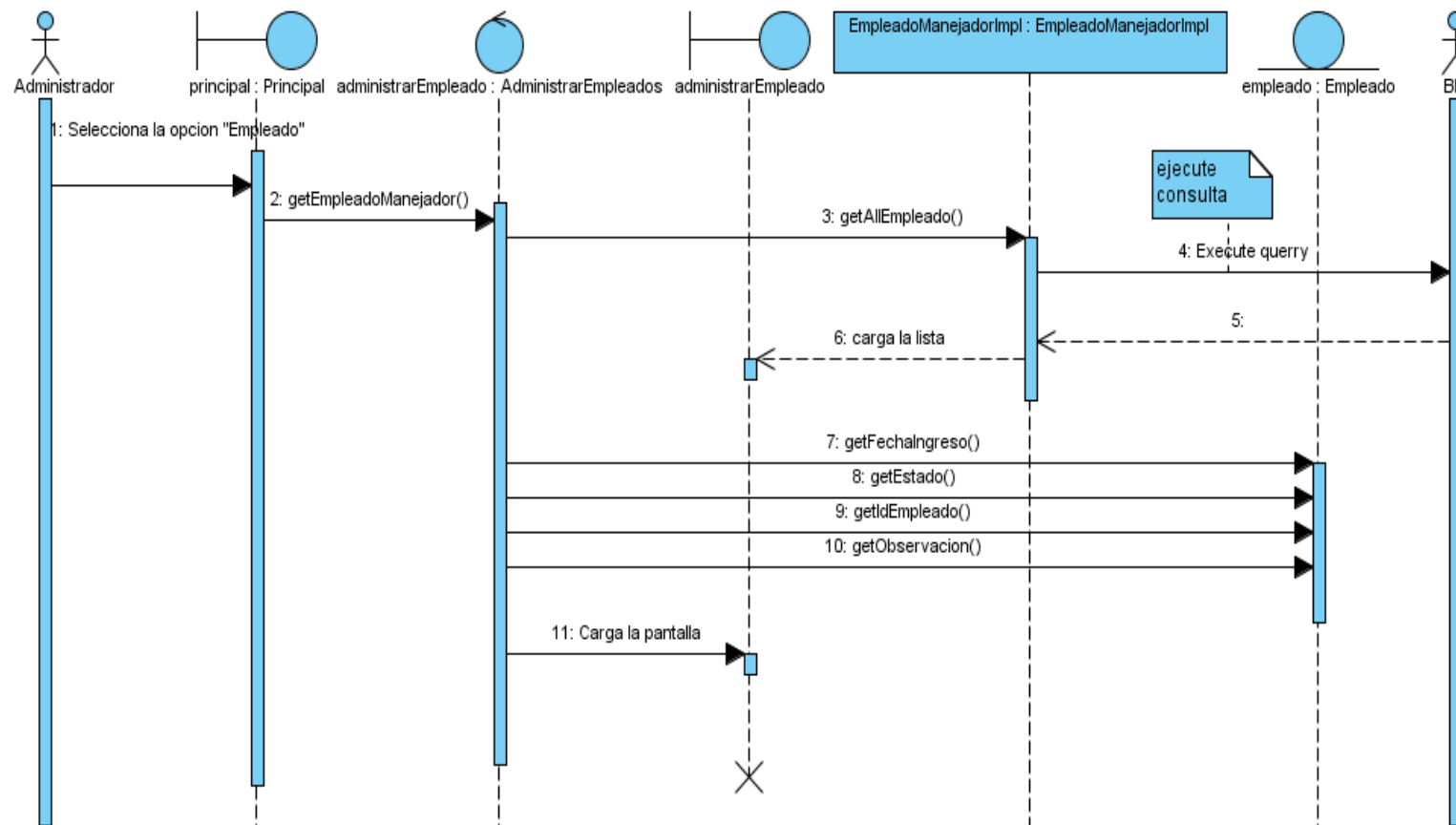


Figura 153: Diagrama de Secuencia-Administrar Empleado

Diagrama de secuencia: C.U. Adicionar Empleado

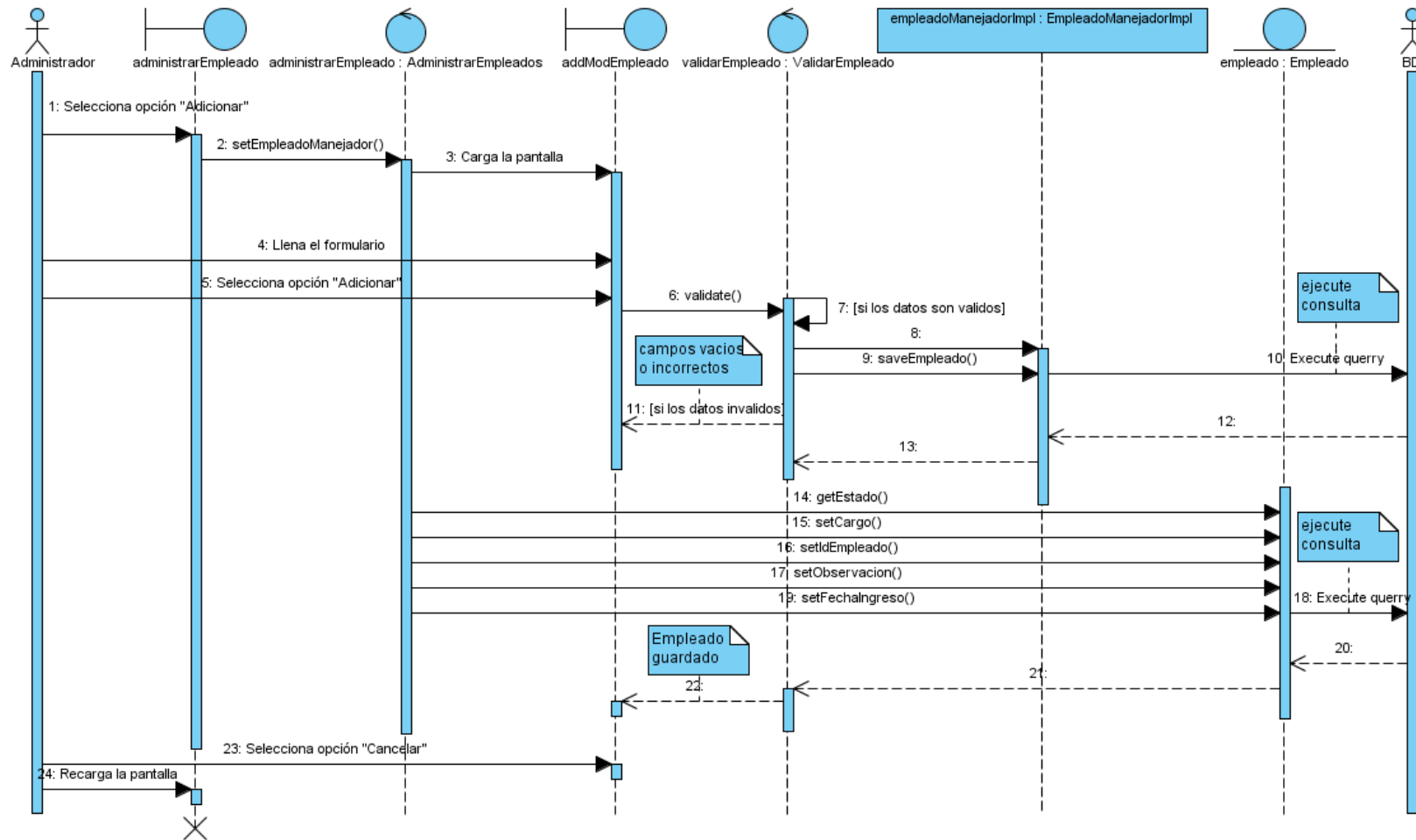


Figura 154: Diagrama de Secuencia-Adicionar Empleado

Diagrama de secuencia: C.U. Modificar Empleado

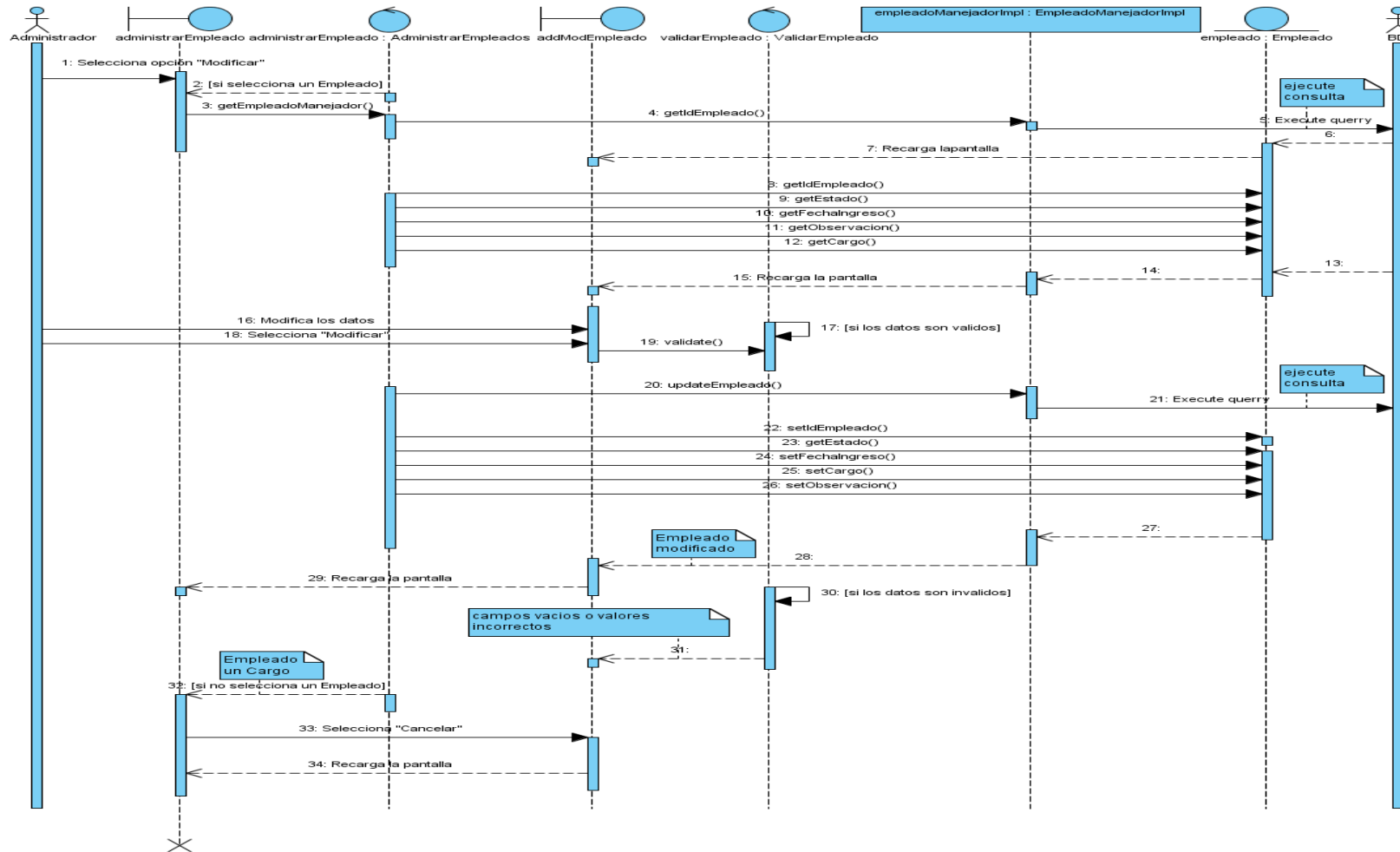


Figura 155: Diagrama de Secuencia-Modificar Empleado

Diagrama de secuencia: C.U. Eliminar Empleado

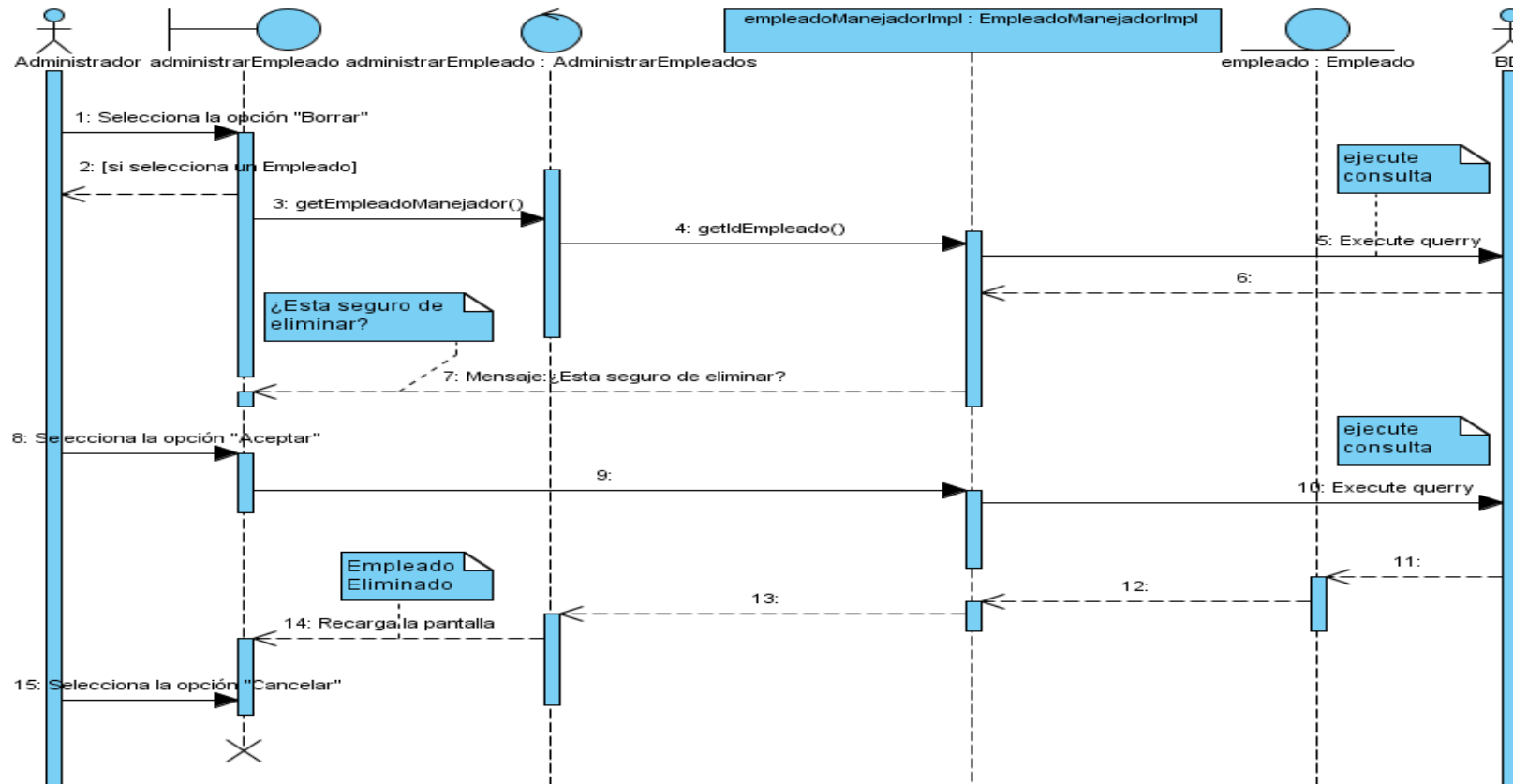


Figura 156: Diagrama de Secuencia-Eliminar Empleado

Diagrama de secuencia: C.U. Administrar Cargo

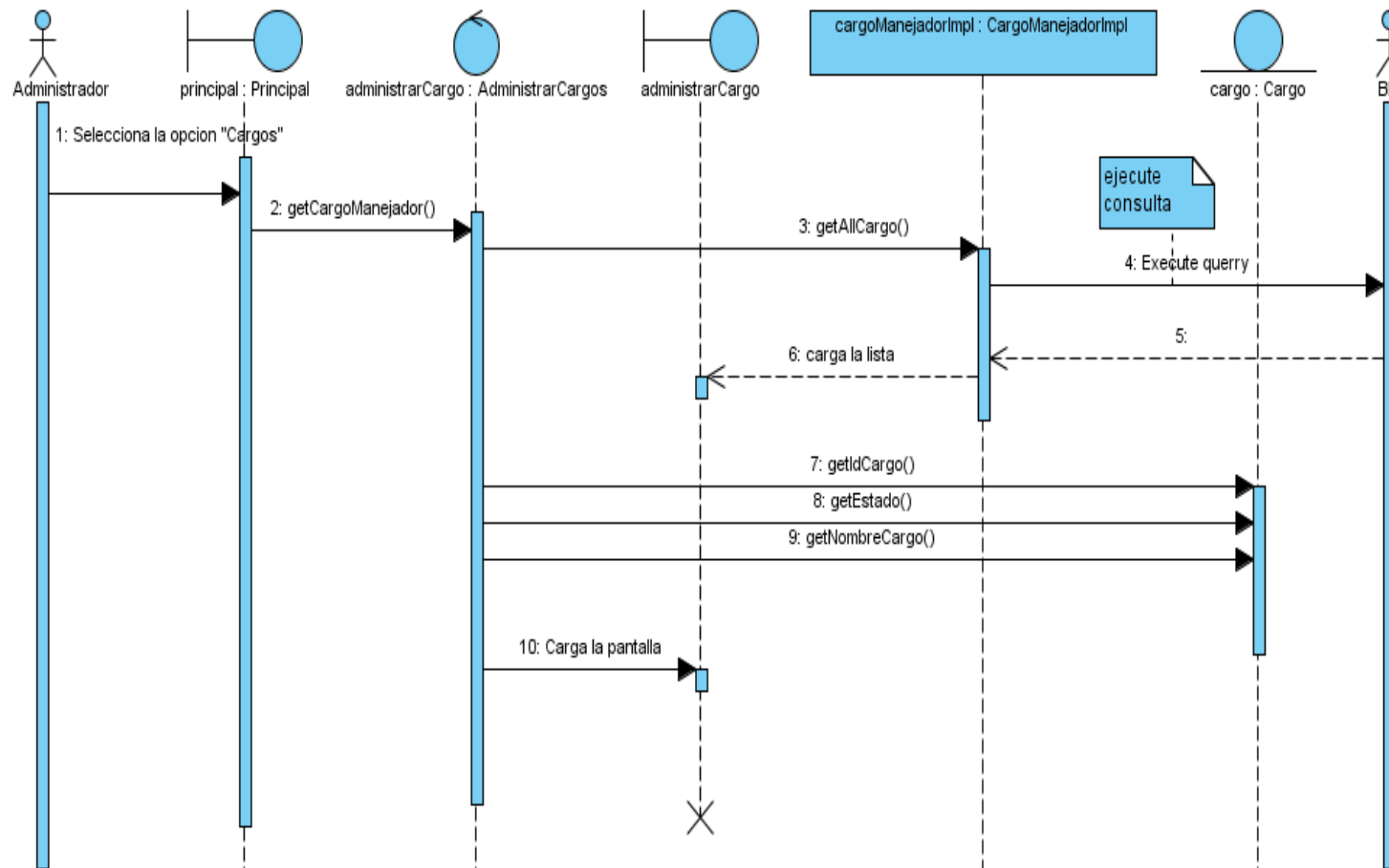


Figura 157: Diagrama de Secuencia-Administrar Cargo

Diagrama de secuencia: C.U. Adicionar Cargo

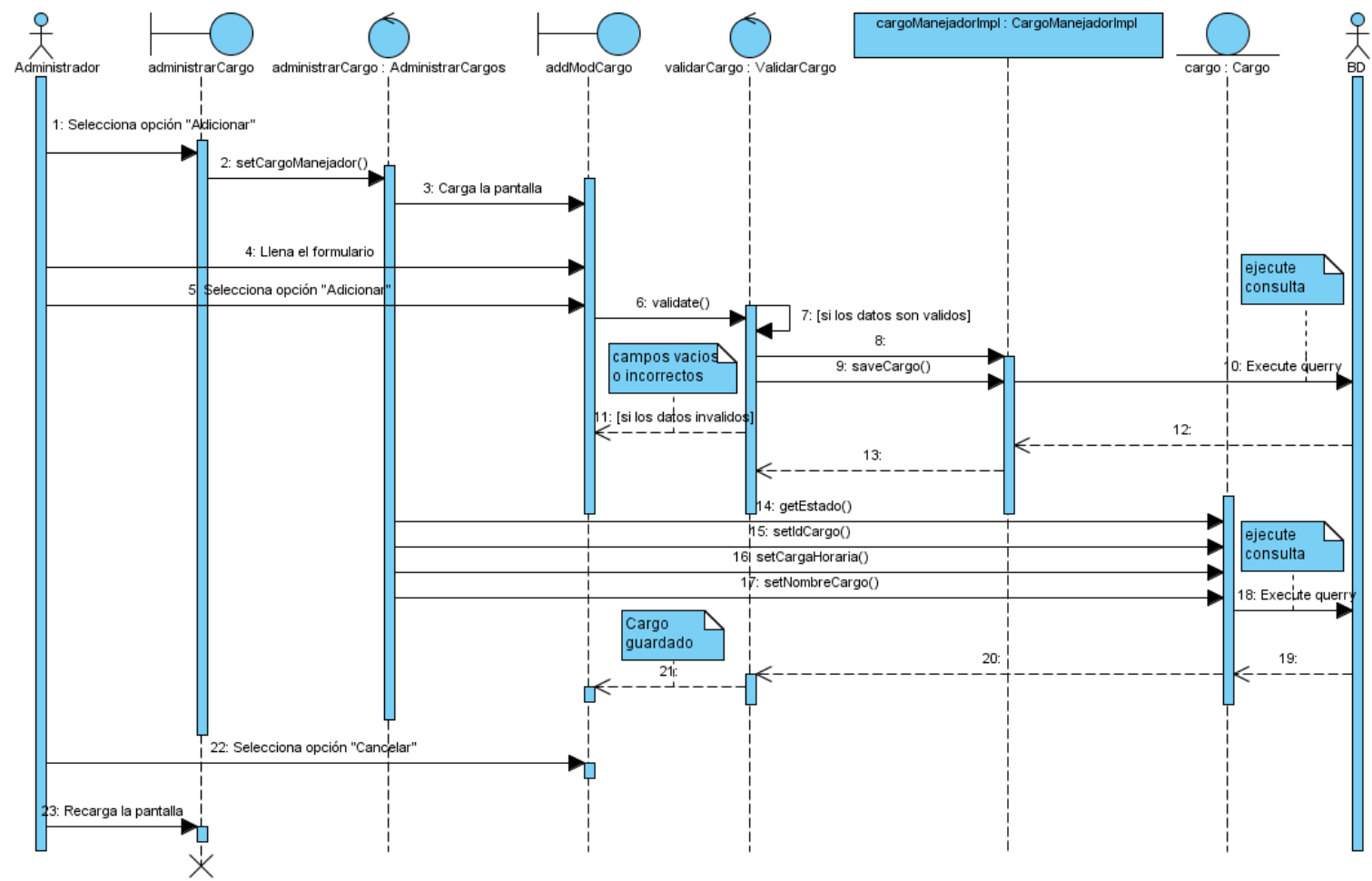


Figura 158: Diagrama de Secuencia-Adicionar Cargo

Diagrama de secuencia: C.U. Modificar Cargo

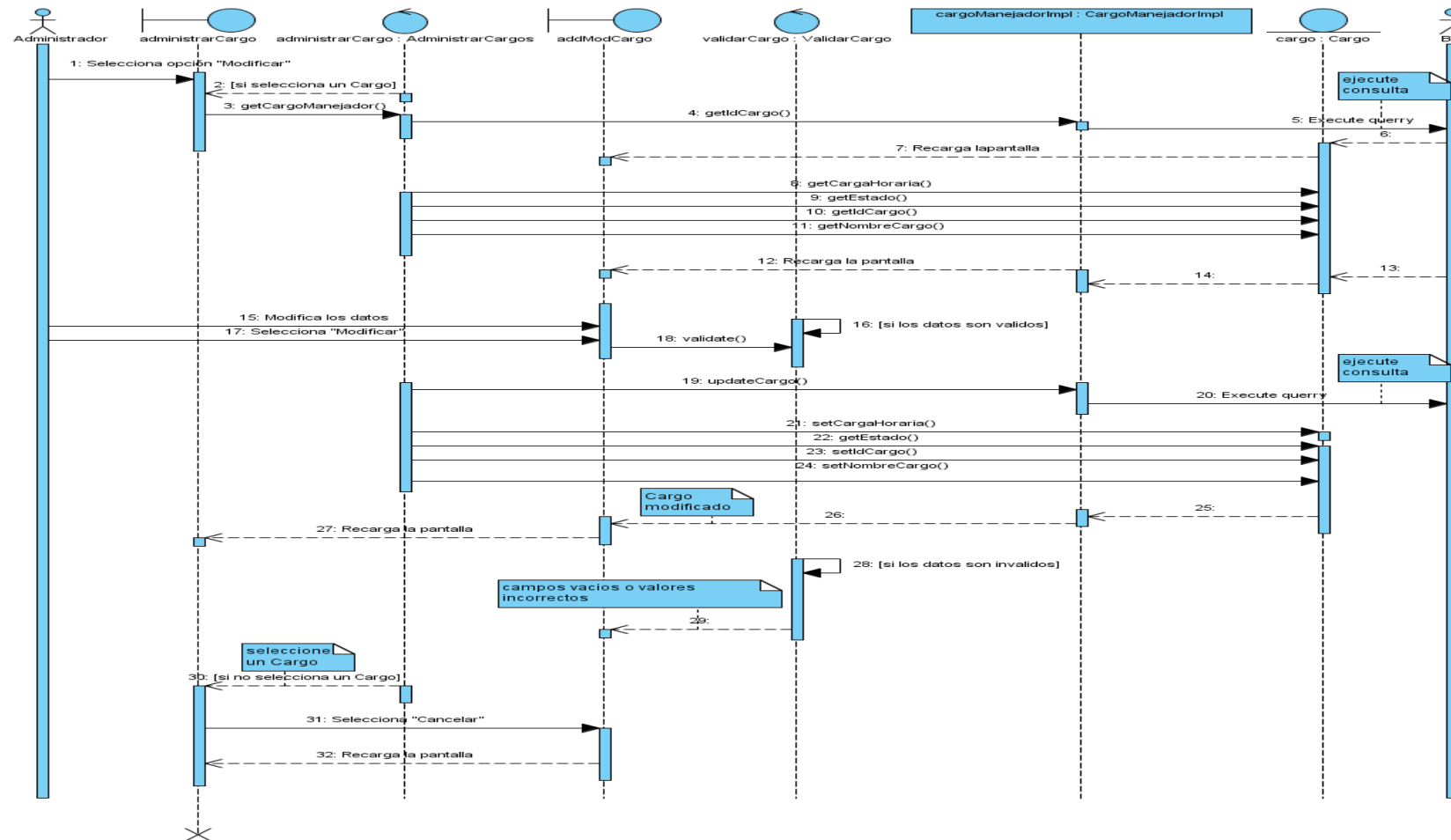


Figura 159: Diagrama de Secuencia-Modificar Cargo

Diagrama de secuencia: C.U. Eliminar Cargo

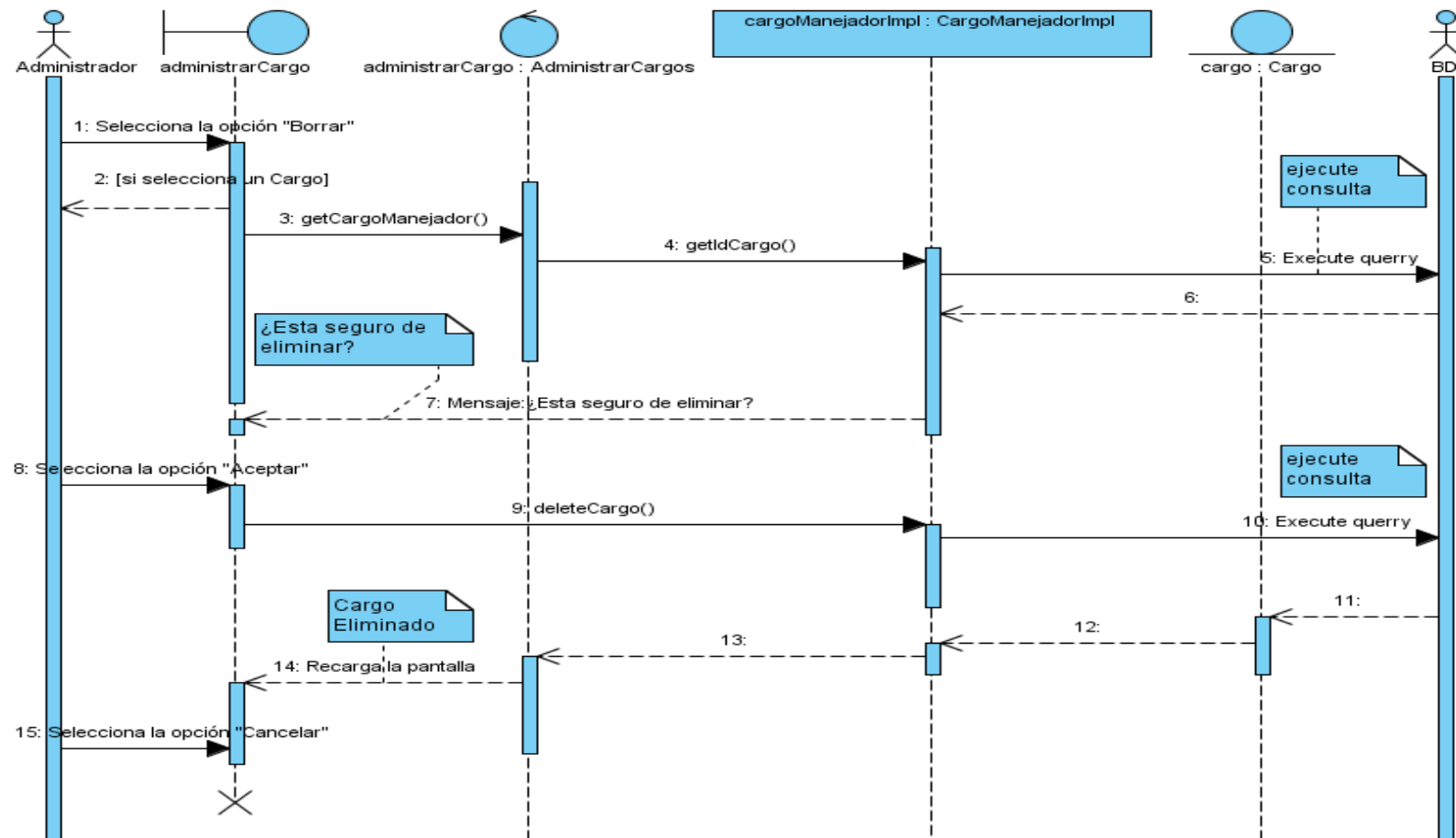


Figura 160: Diagrama de Secuencia-Eliminar Cargo

Diagrama de secuencia: C.U. Administrar Clientes

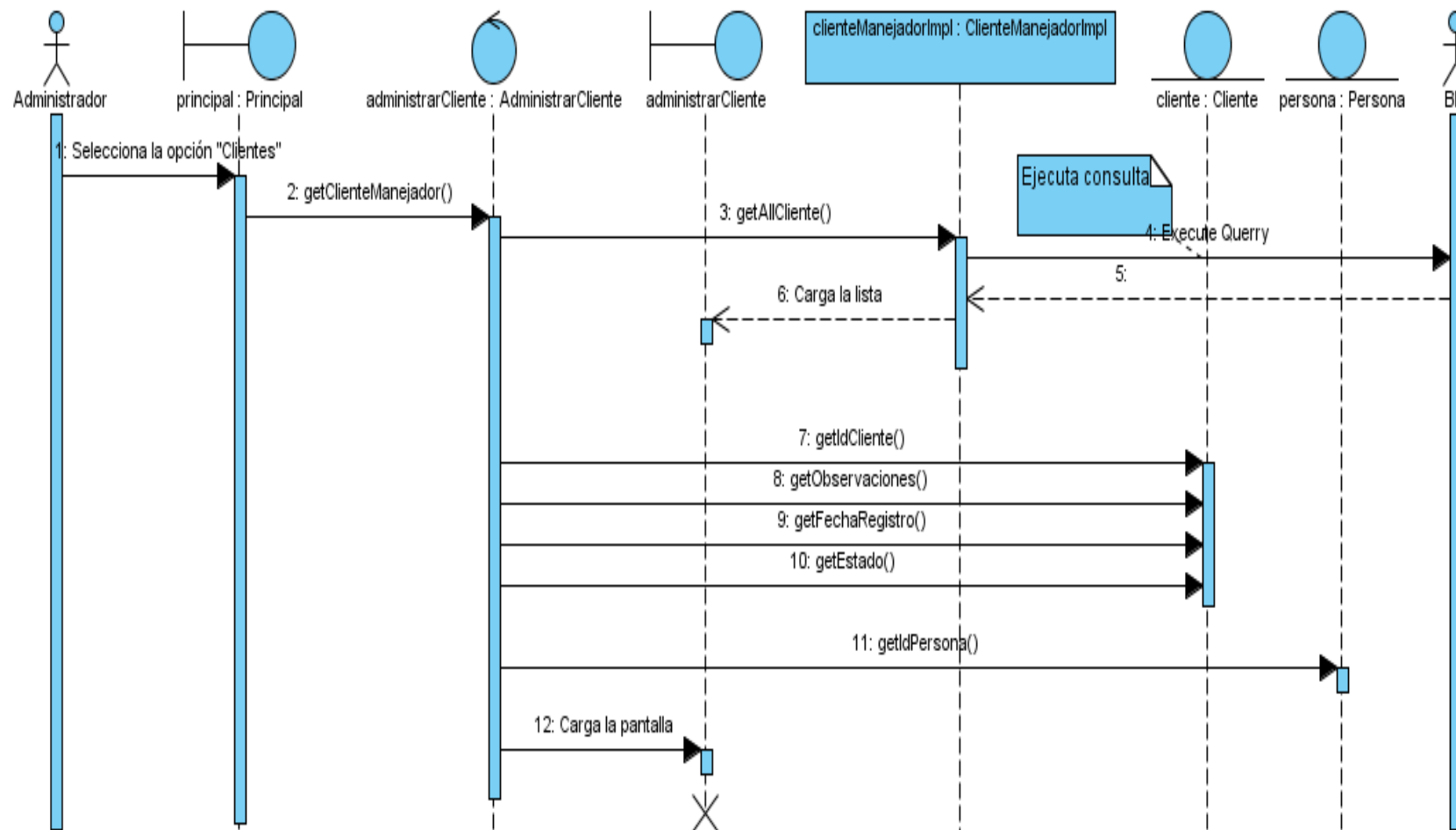


Figura 161: Diagrama de Secuencia-Administrar Clientes

Diagrama de secuencia: C.U. Adicionar Cliente

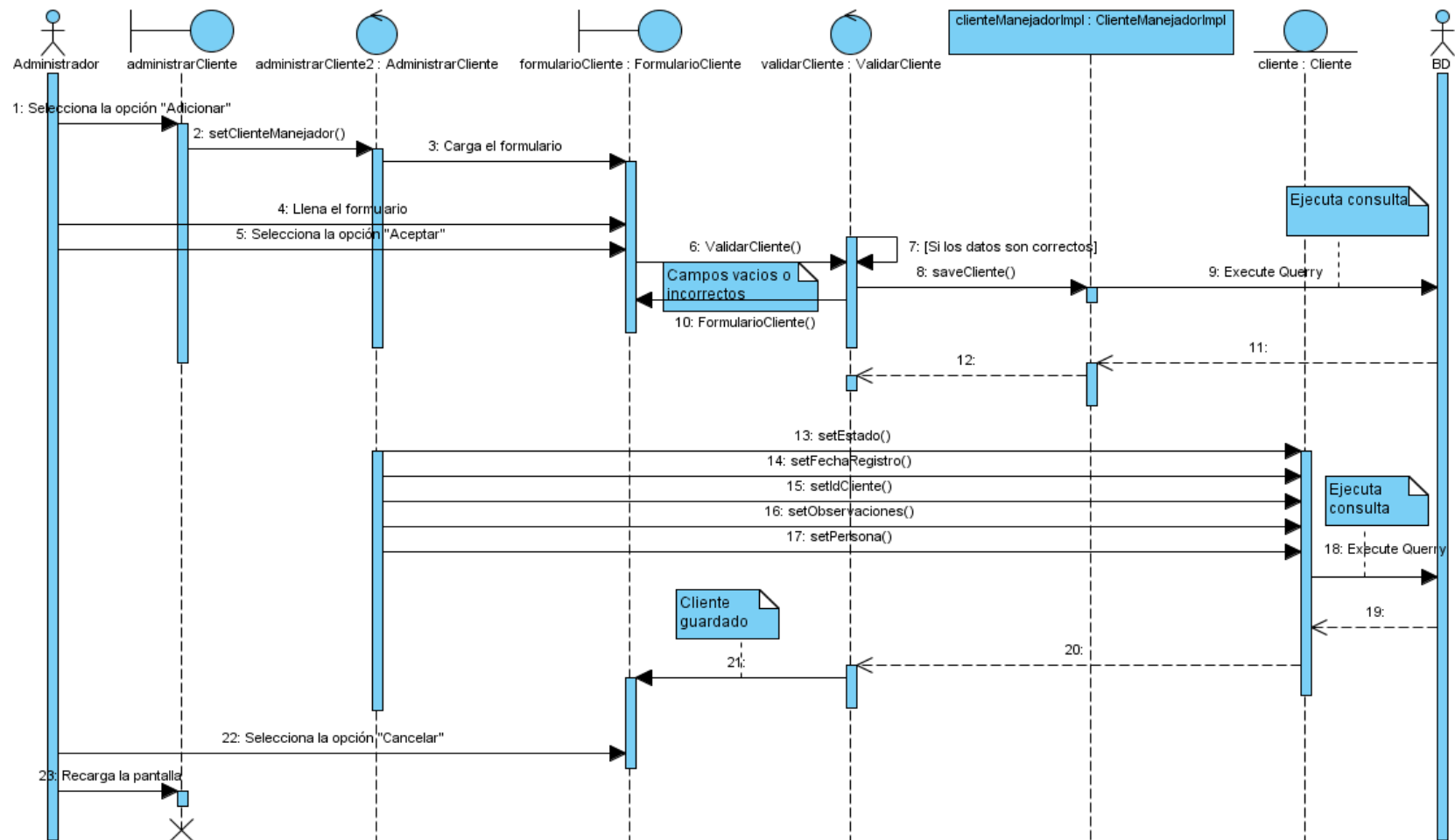


Figura 162: Diagrama de Secuencia-Adicionar Cliente

Diagrama de secuencia: C.U. Modificar Cliente

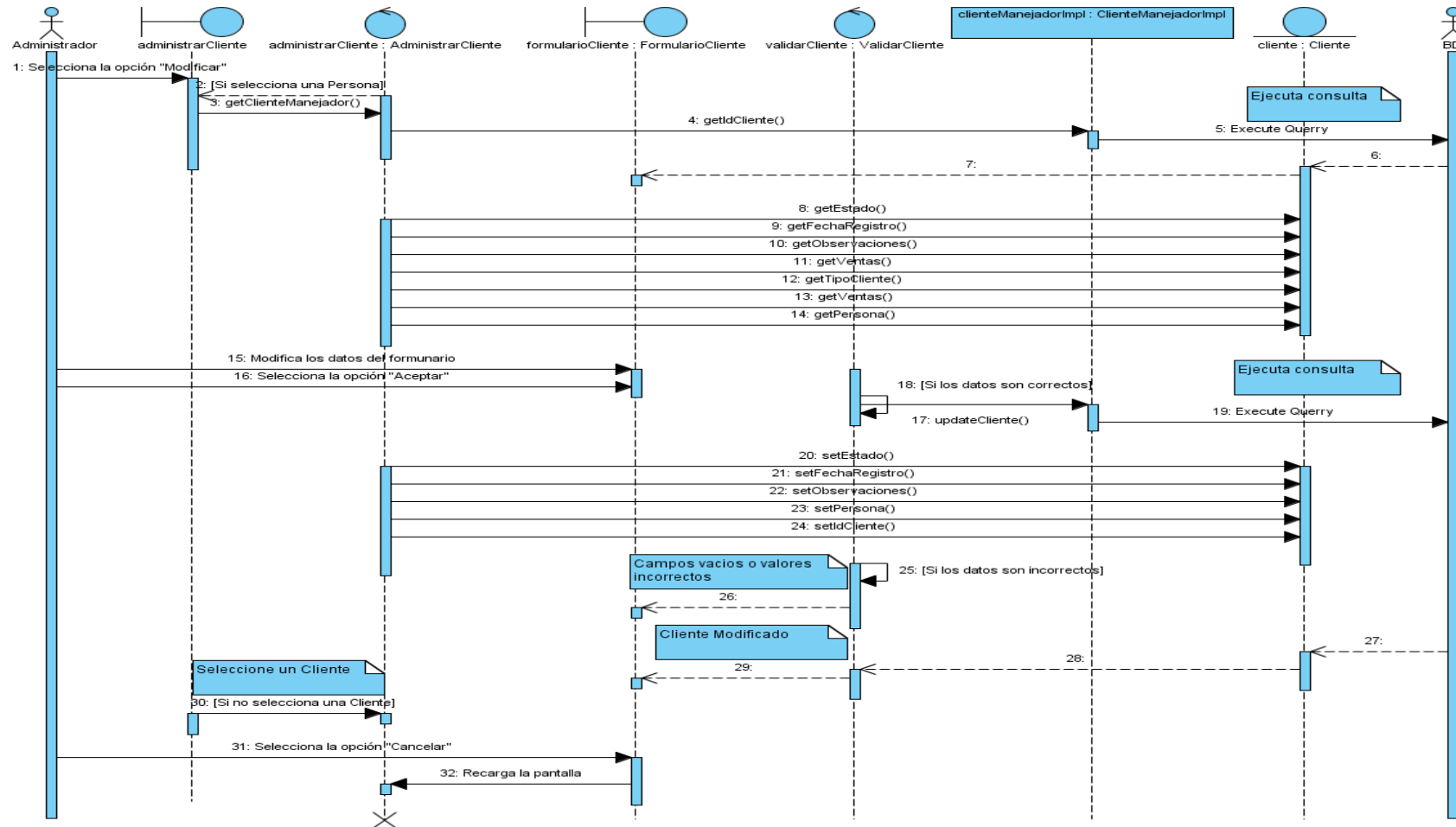


Figura 163: Diagrama de Secuencia-Modificar Cliente

Diagrama de secuencia: C.U. Eliminar Cliente

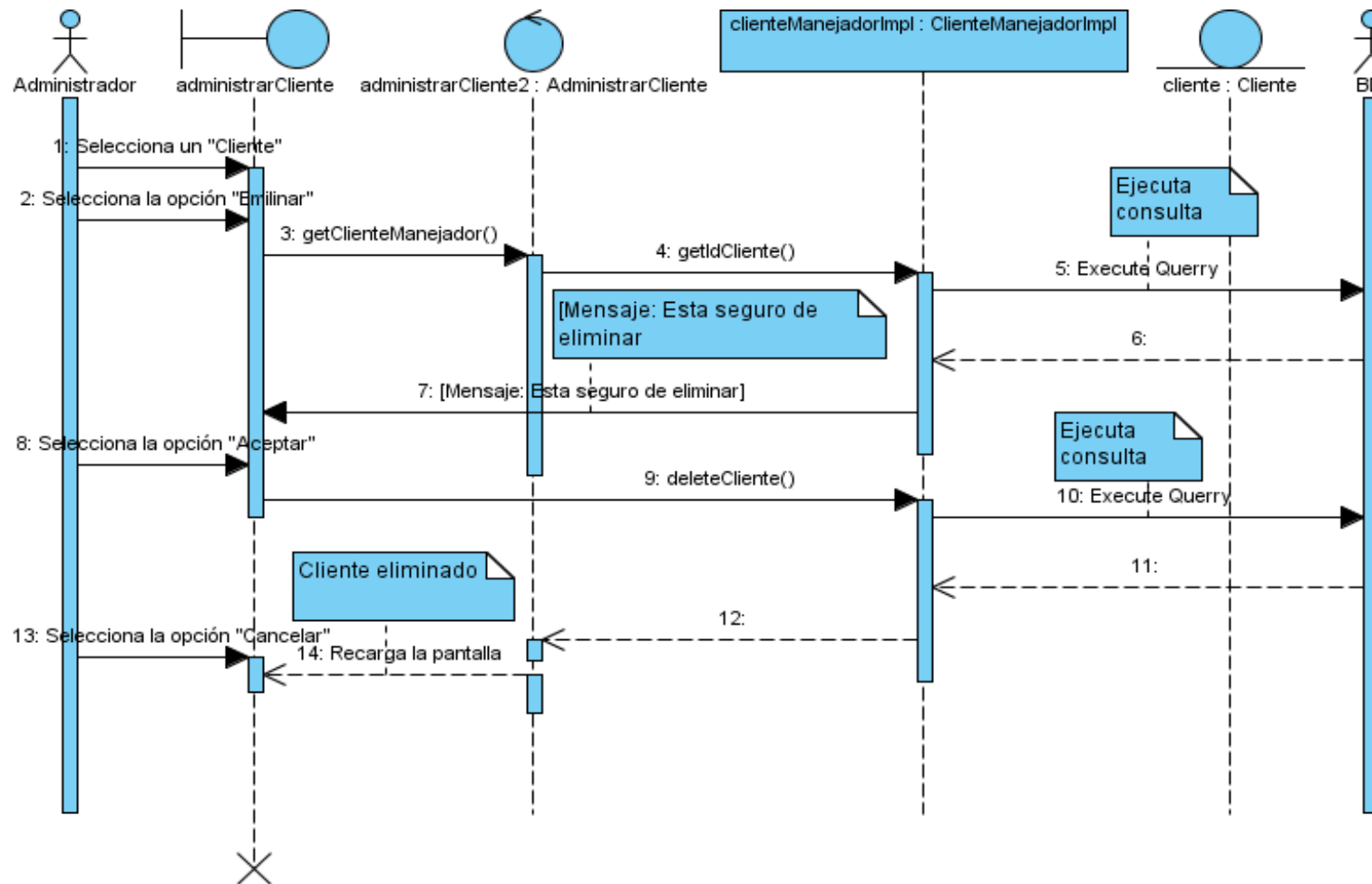


Figura 164: Diagrama de Secuencia-Eliminar Cliente

Diagrama de secuencia: C.U. Administrar Tipo de Clientes

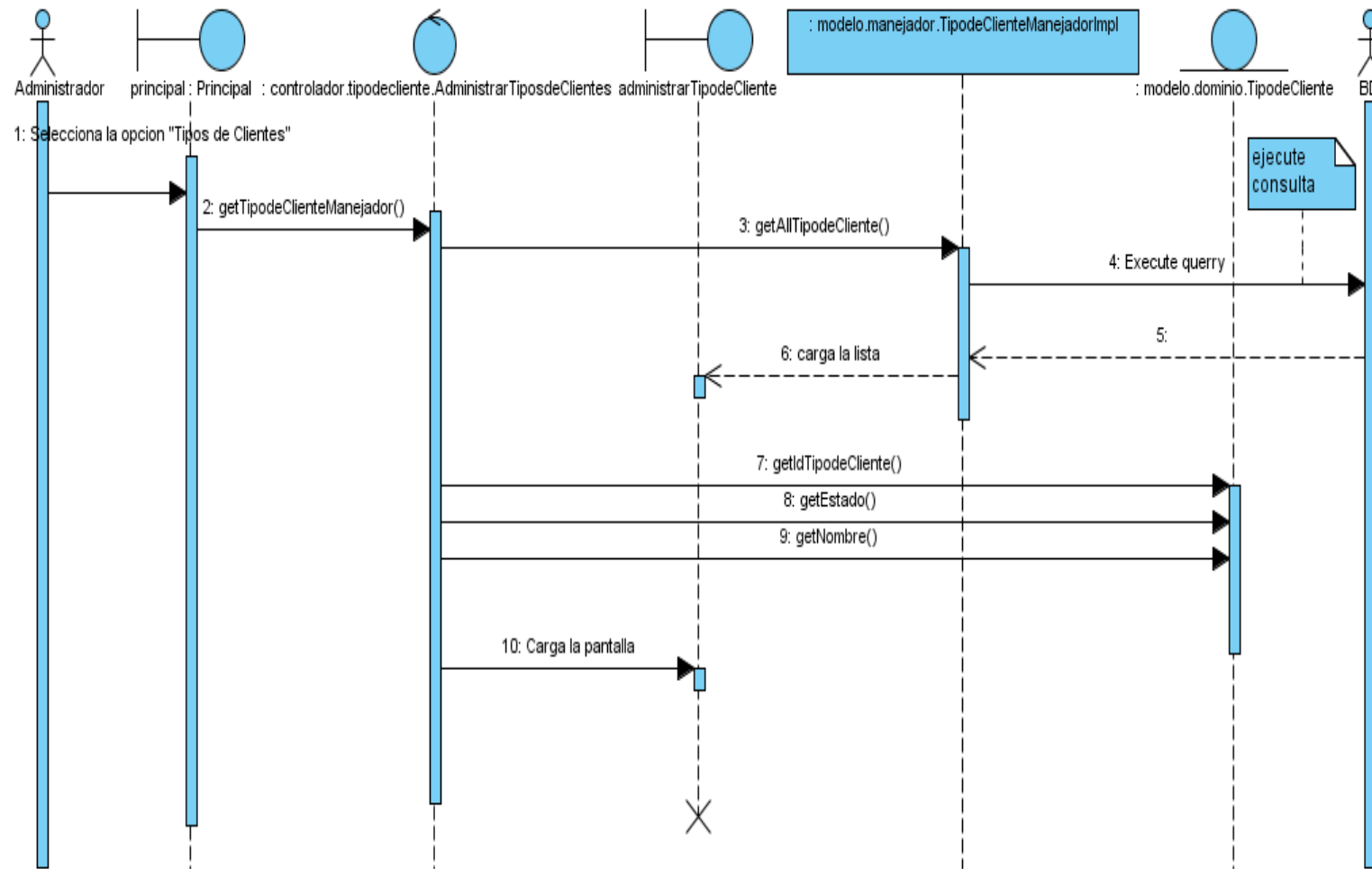


Figura 165: Diagrama de Secuencia-Administrar Tipo de Cliente

Diagrama de secuencia: C.U. Adicionar Tipo de Cliente

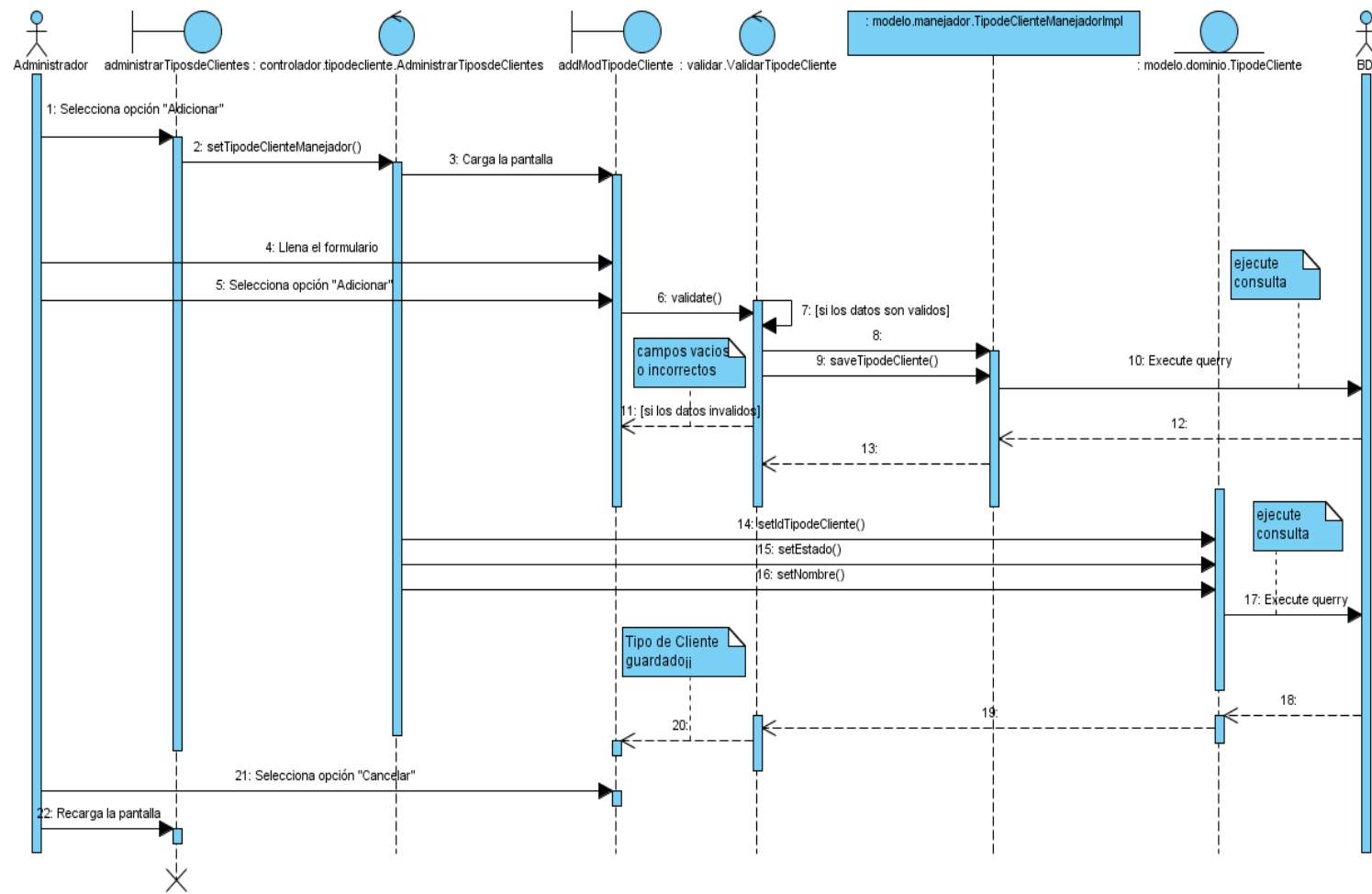


Figura 166: Diagrama de Secuencia-Adicionar Tipo de Cliente

Diagrama de secuencia: C.U. Modificar Tipo de Cliente

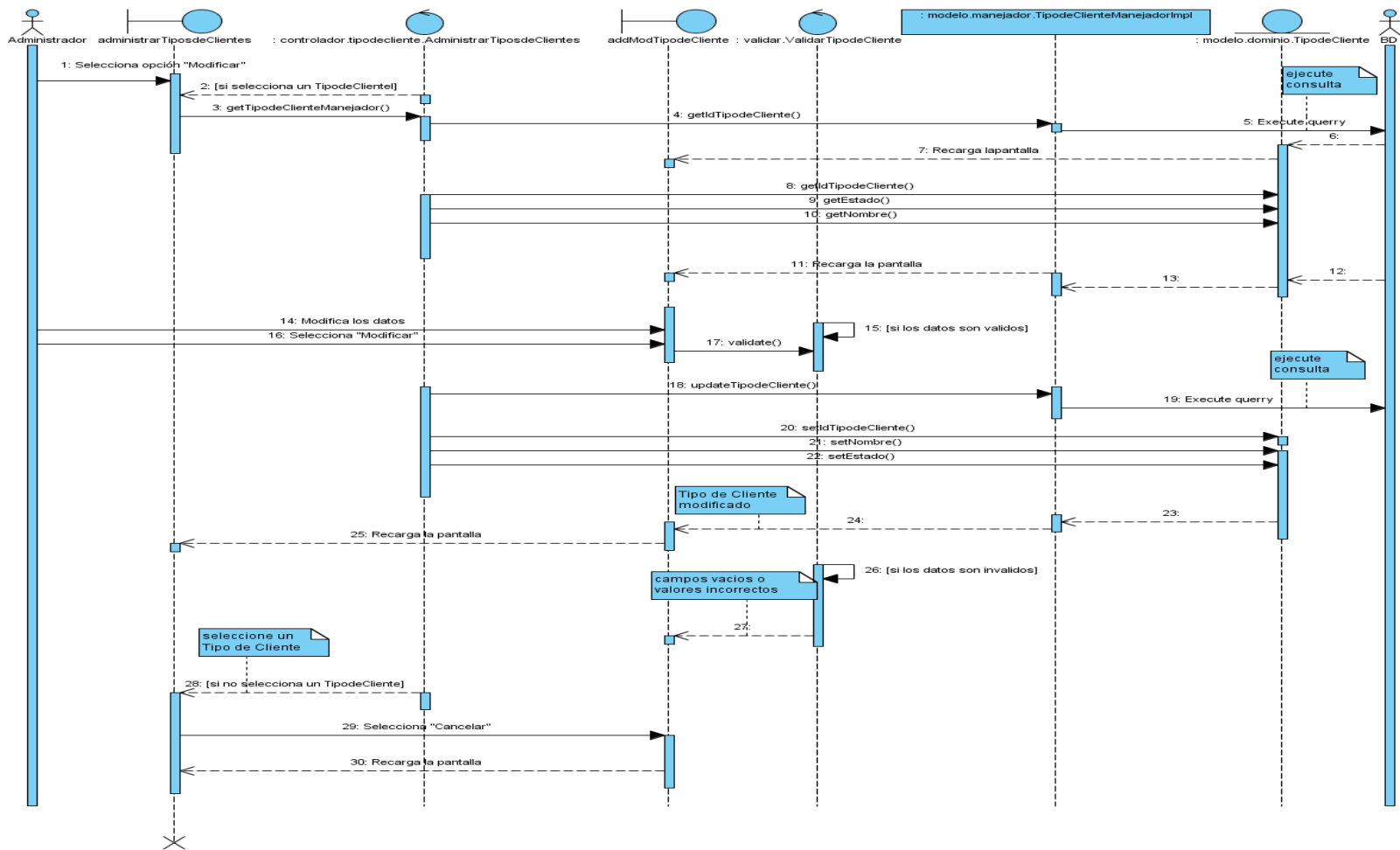


Figura 167: Diagrama de Secuencia-Modificar Tipo de Cliente

Diagrama de secuencia: C.U. Eliminar Tipo de Cliente

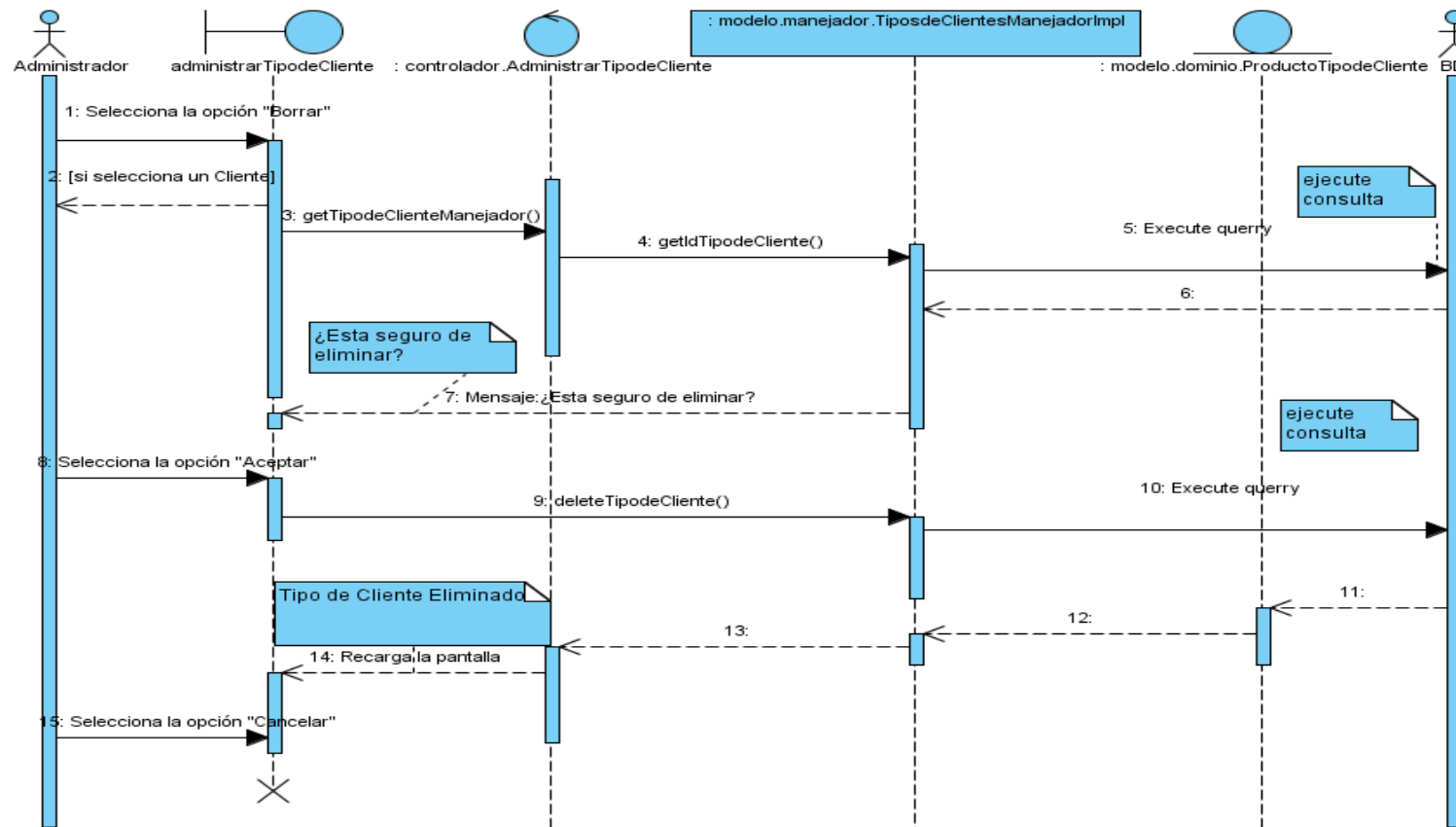


Figura 168: Diagrama de Secuencia-Eliminar Tipo de Cliente

Diagrama de secuencia: C.U. Administrar Ventas

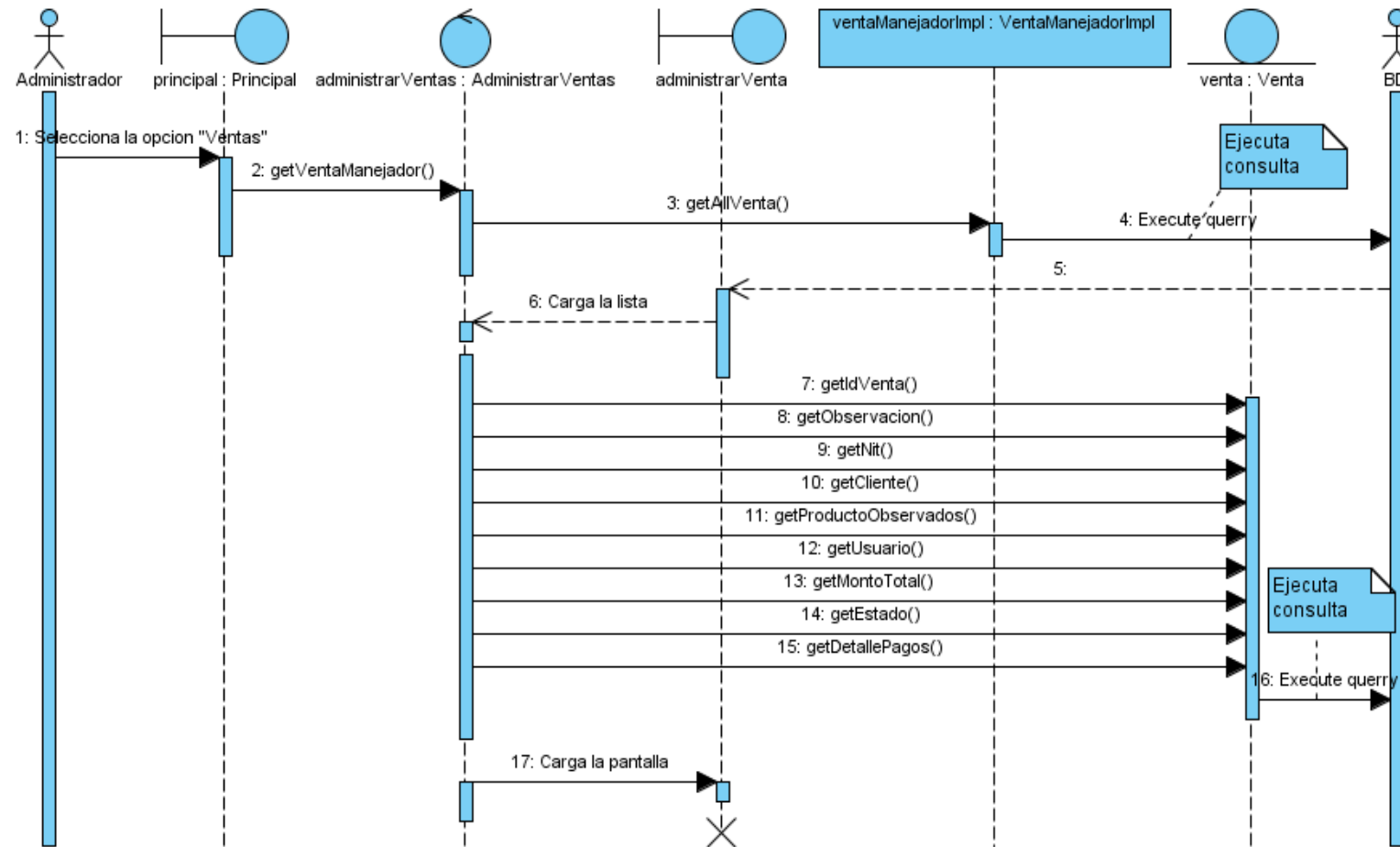


Figura 169: Diagrama de Secuencia-Administrar Ventas

Diagrama de secuencia: C.U. Adicionar Venta

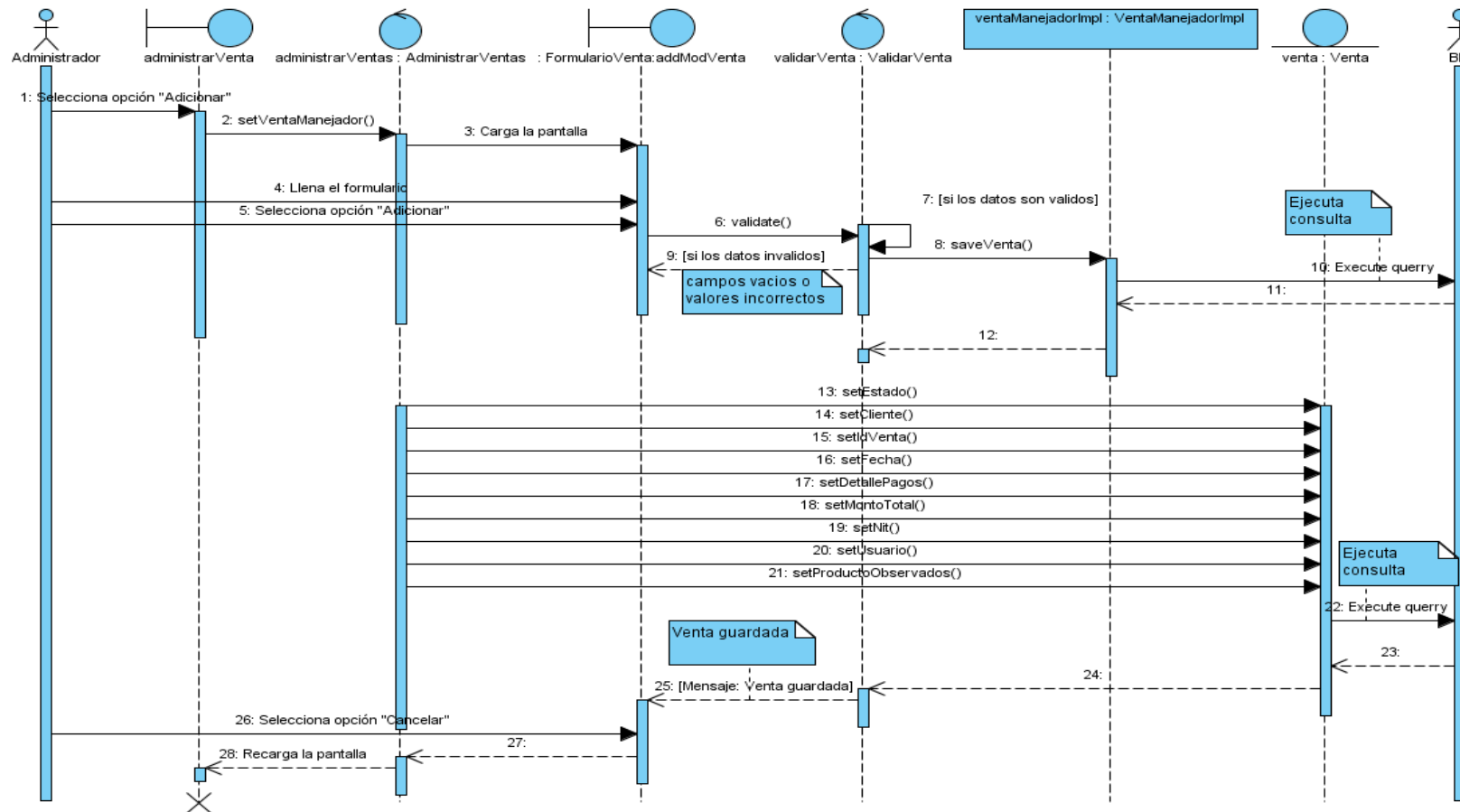


Figura 170: Diagrama de Secuencia-Adicionar Venta

Diagrama de secuencia: C.U. Administrar Tipos de Ventas

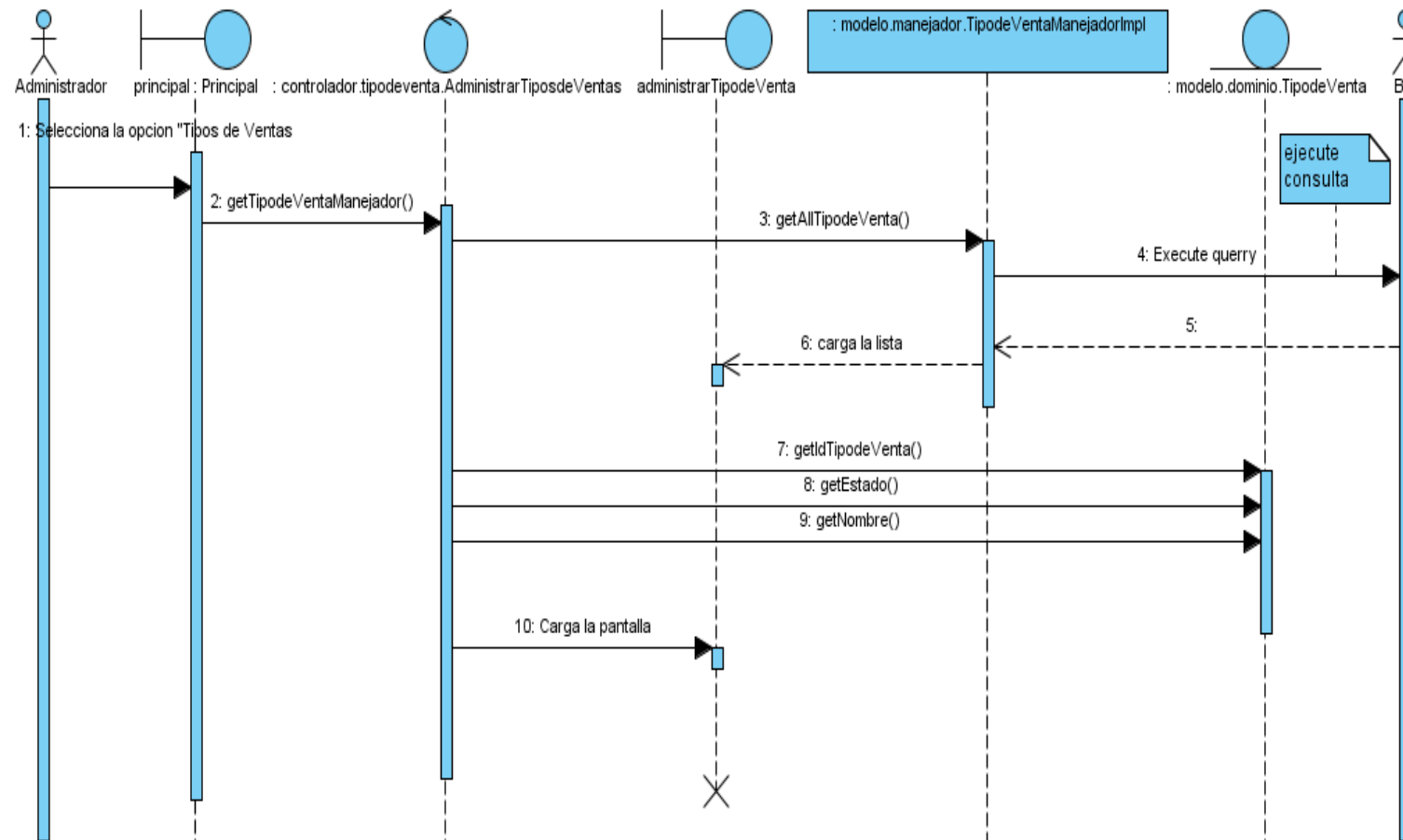


Figura 171: Diagrama de Secuencia-Administrar Tipos de Ventas

Diagrama de secuencia: C.U. Adicionar Tipo de Venta

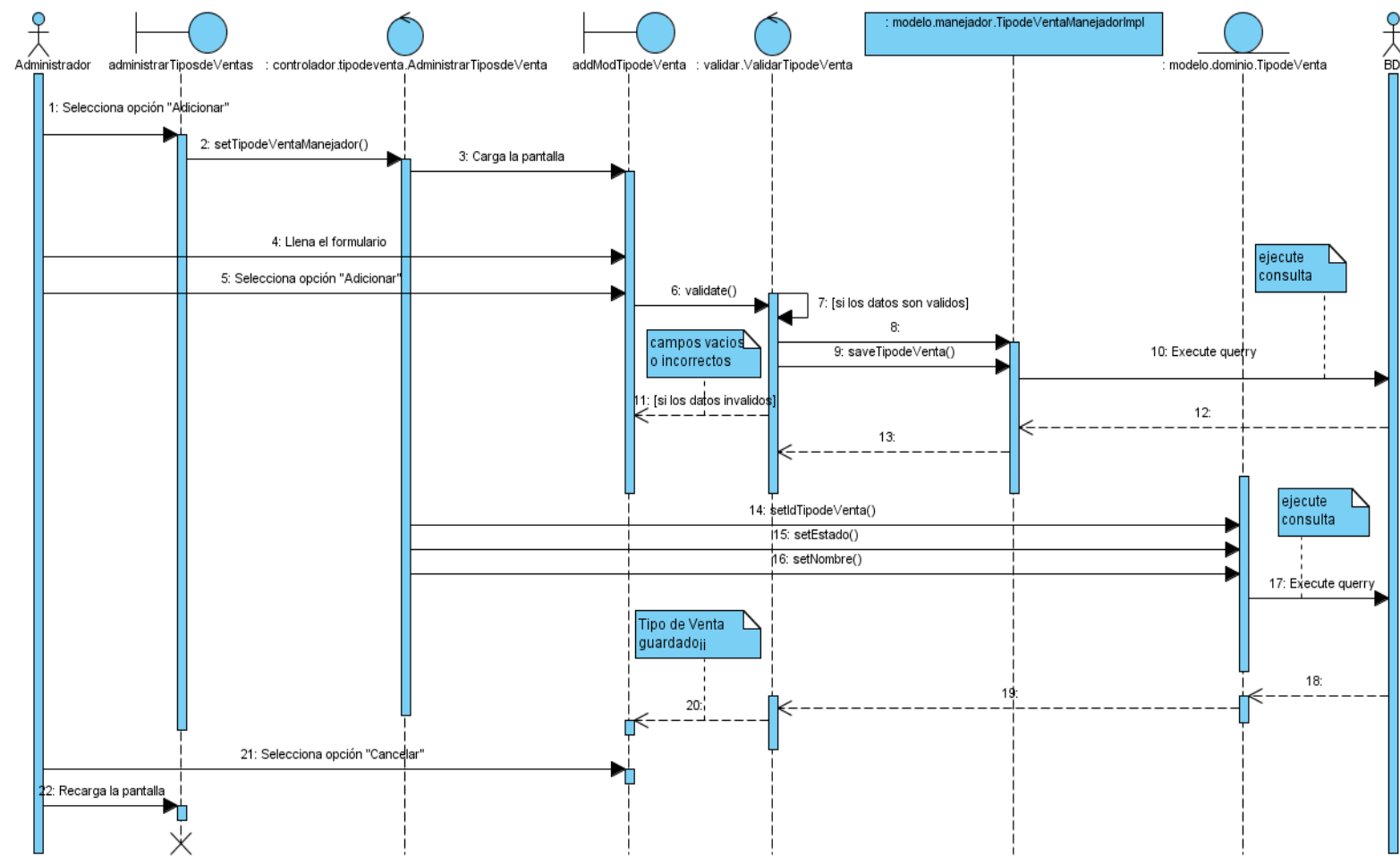


Figura 172: Diagrama de Secuencia-Adicionar Tipos de Ventas

Diagrama de secuencia: C.U. Modificar Tipo de Venta

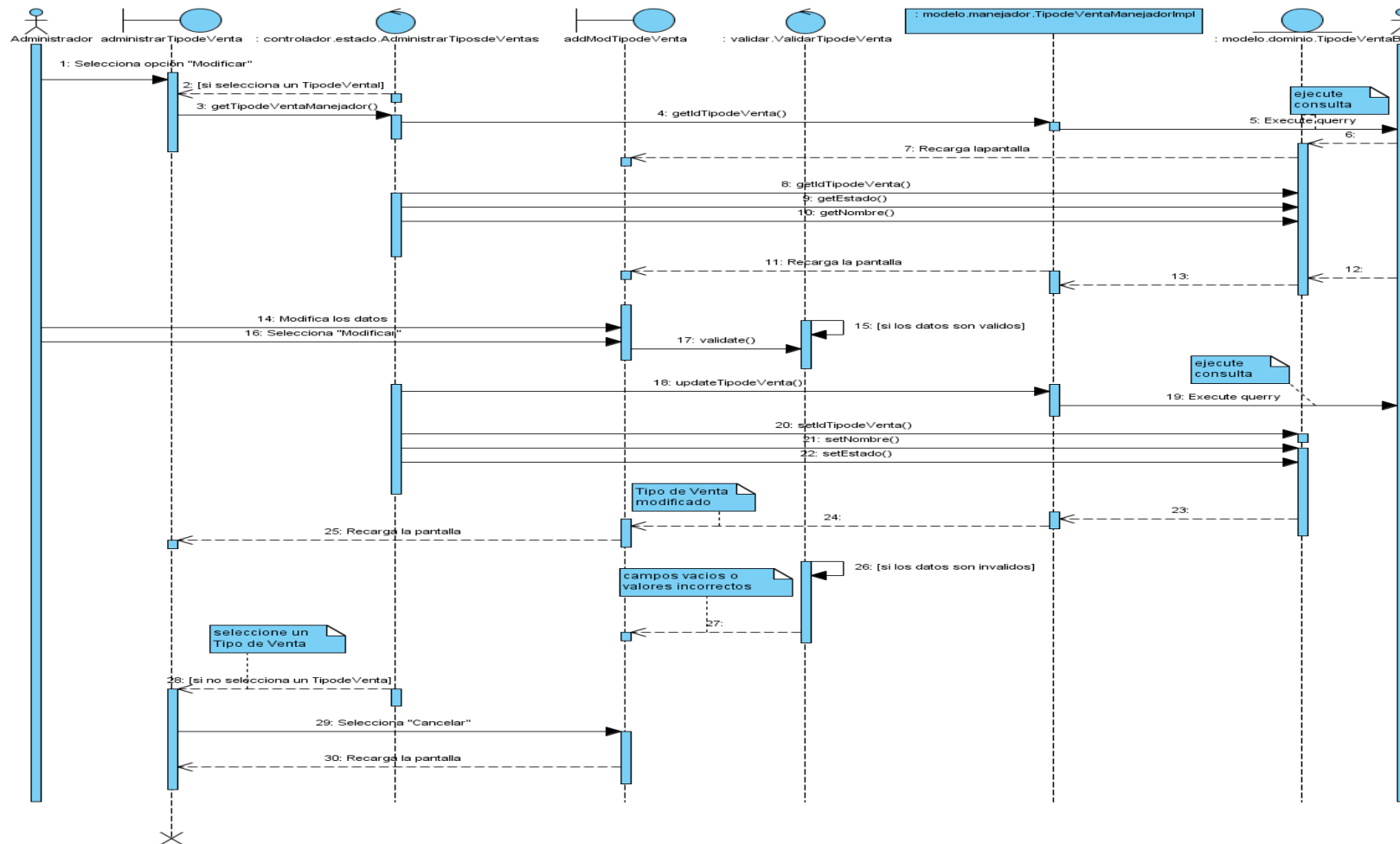


Figura 173: Diagrama de Secuencia-Modificar Tipos de Ventas

Diagrama de secuencia: C.U. Eliminar Tipo de Venta

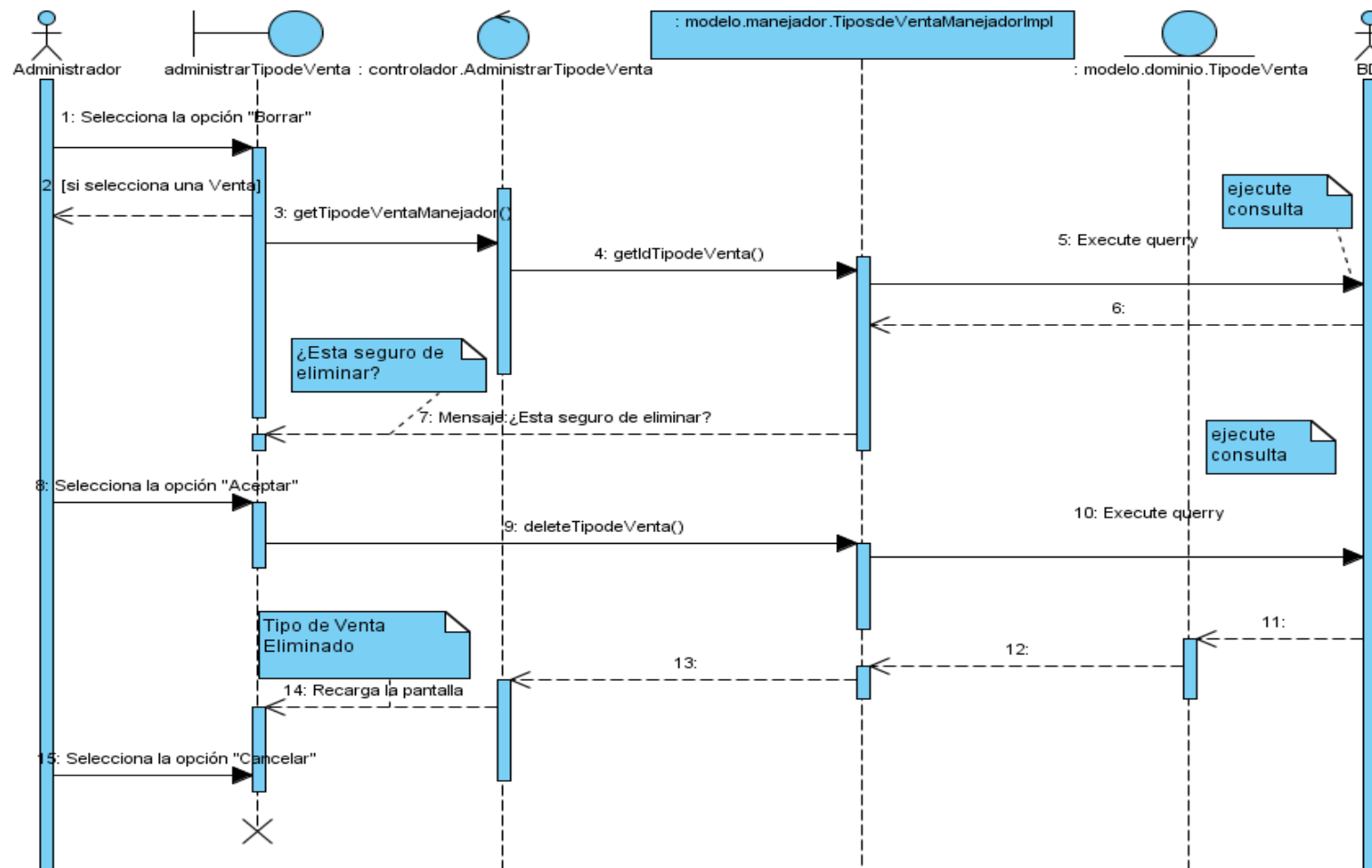


Figura 174: Diagrama de Secuencia-Eliminar Tipos de Ventas

Diagrama de Secuencia: Ver Plan de Pagos

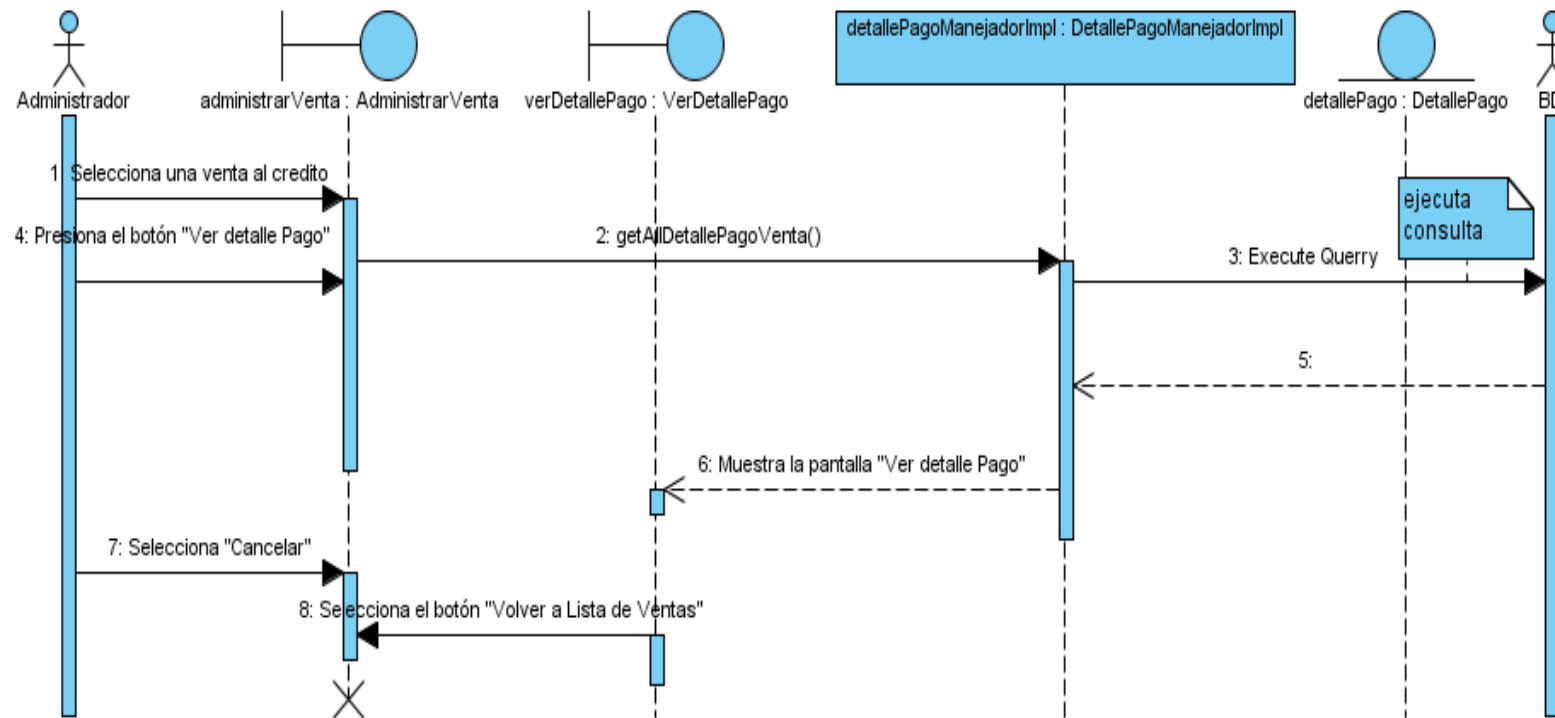


Figura 175: Diagrama de Secuencia-Ver Plan de Pagos

Diagrama de secuencia: Pagar cuota de Plan de Pago:

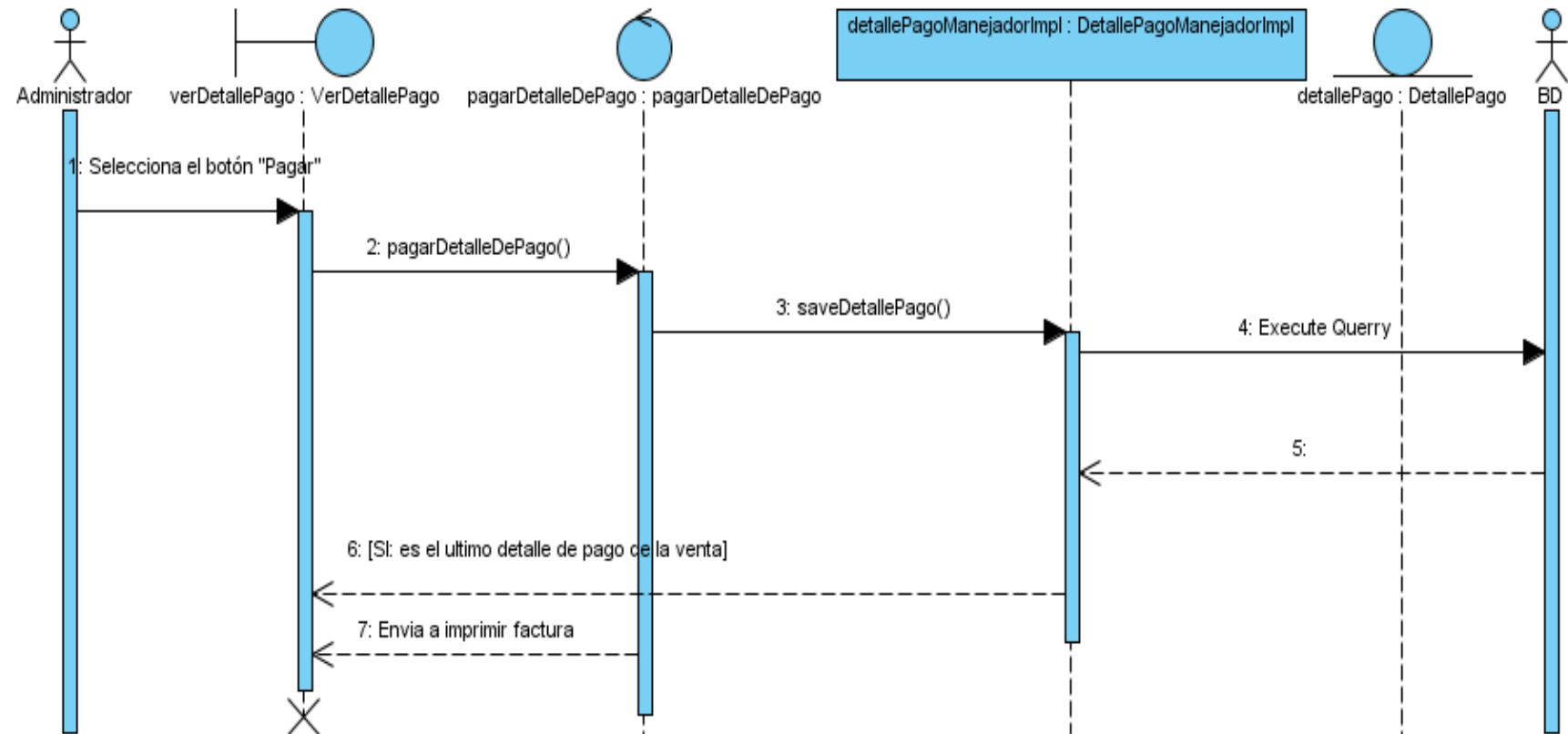


Figura 176: Diagrama de Secuencia-Pagar cuota de Plan de Pago

Diagrama de secuencia: C.U. Administrar Producto

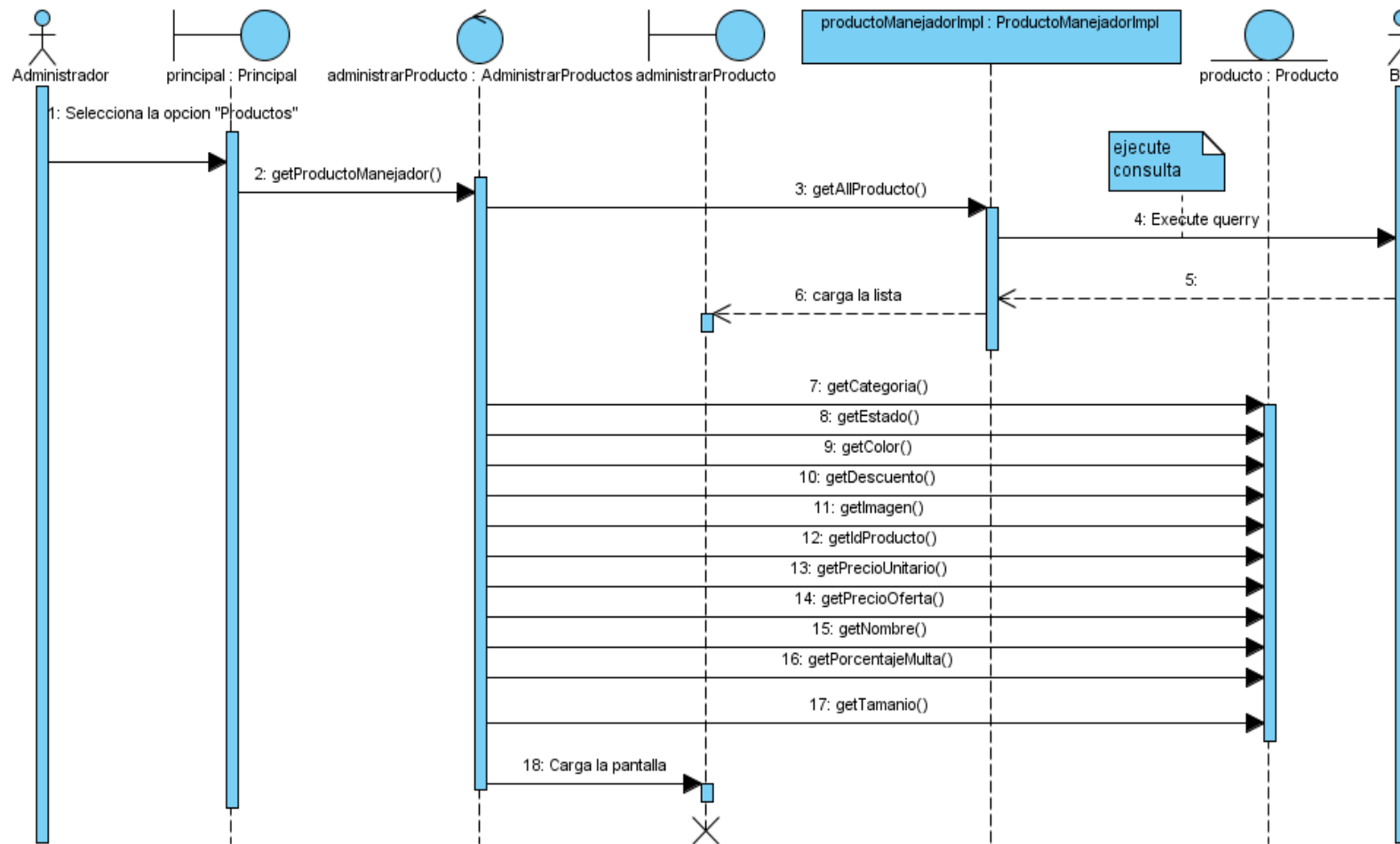


Figura 177: Diagrama de Secuencia-Administrar Producto

Diagrama de secuencia: C.U. Adicionar Producto

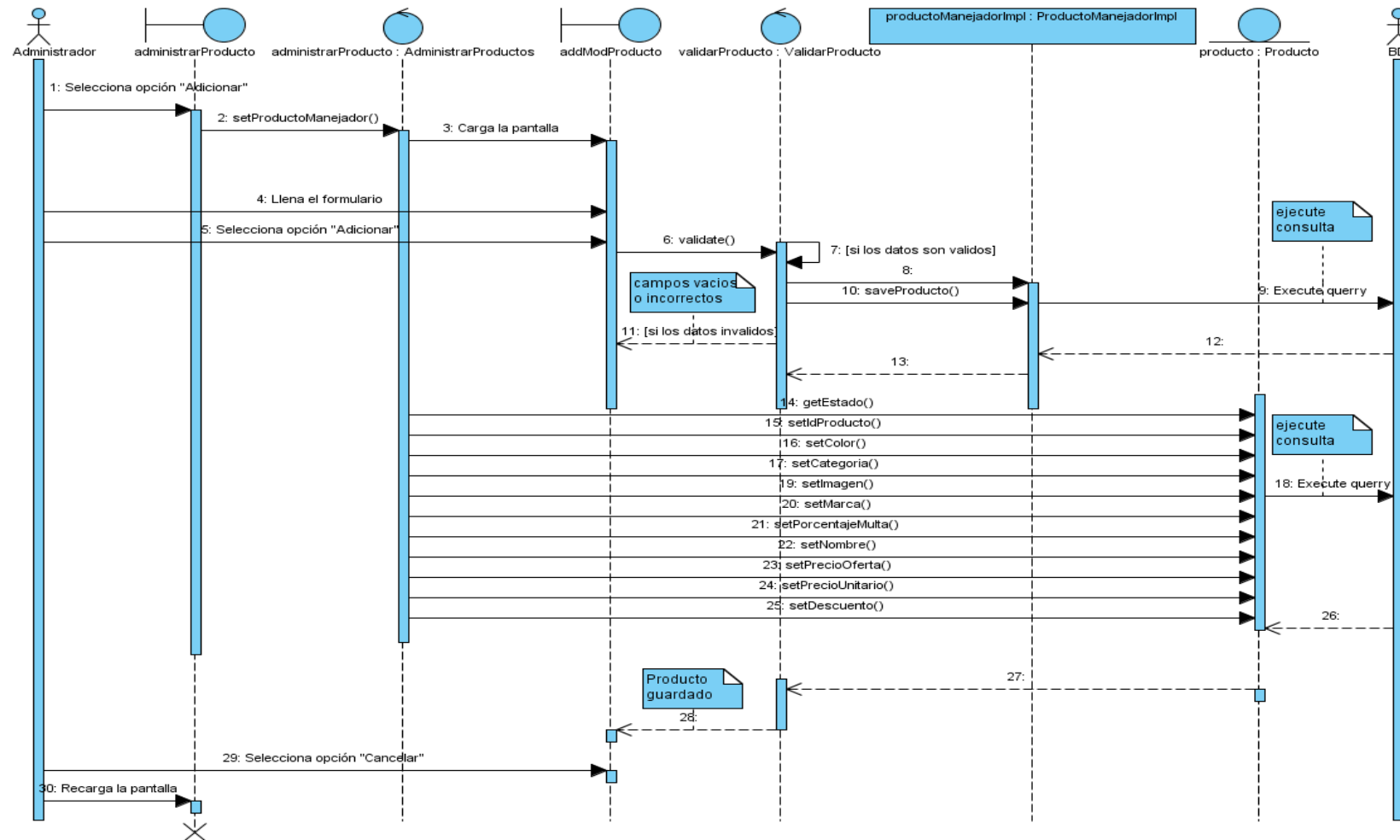


Figura 178: Diagrama de Secuencia-Adicionar Producto

Diagrama de secuencia: C.U. Modificar Producto

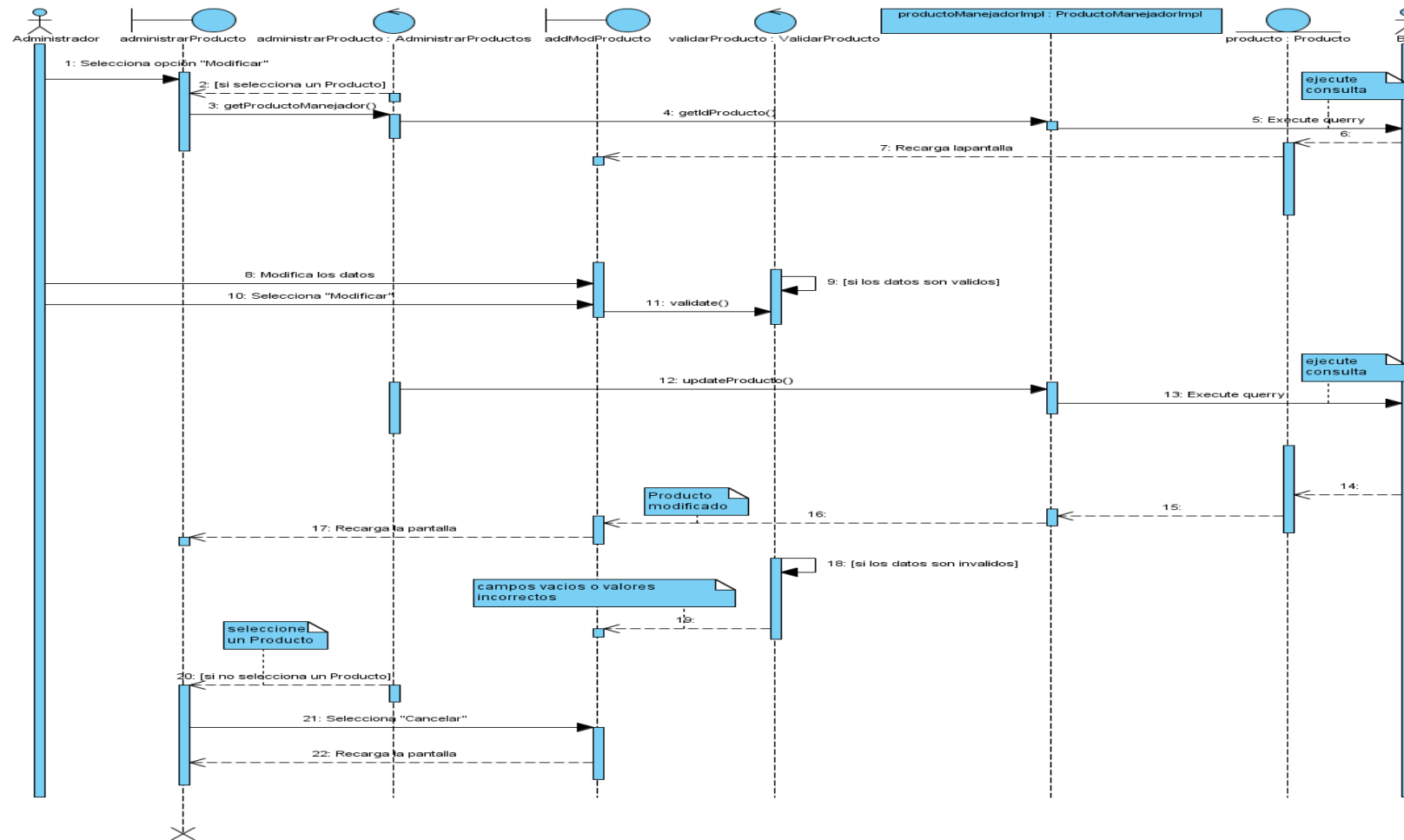


Figura 179: Diagrama de Secuencia-Modificar Producto

Diagrama de secuencia: C.U. Eliminar Producto

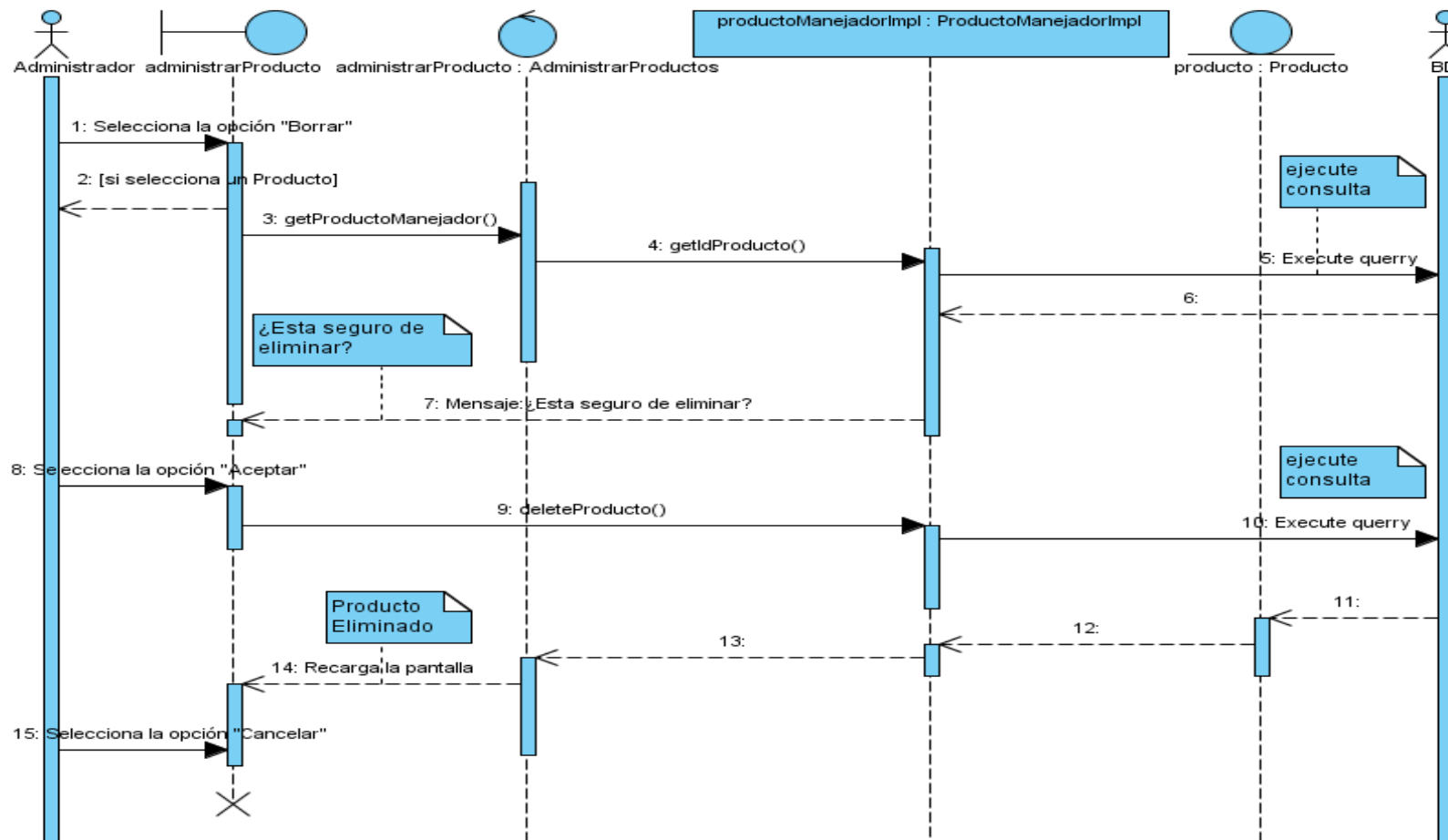


Figura 180: Diagrama de Secuencia-Eliminar Producto

Diagrama de secuencia: C.U. Administrar Almacén

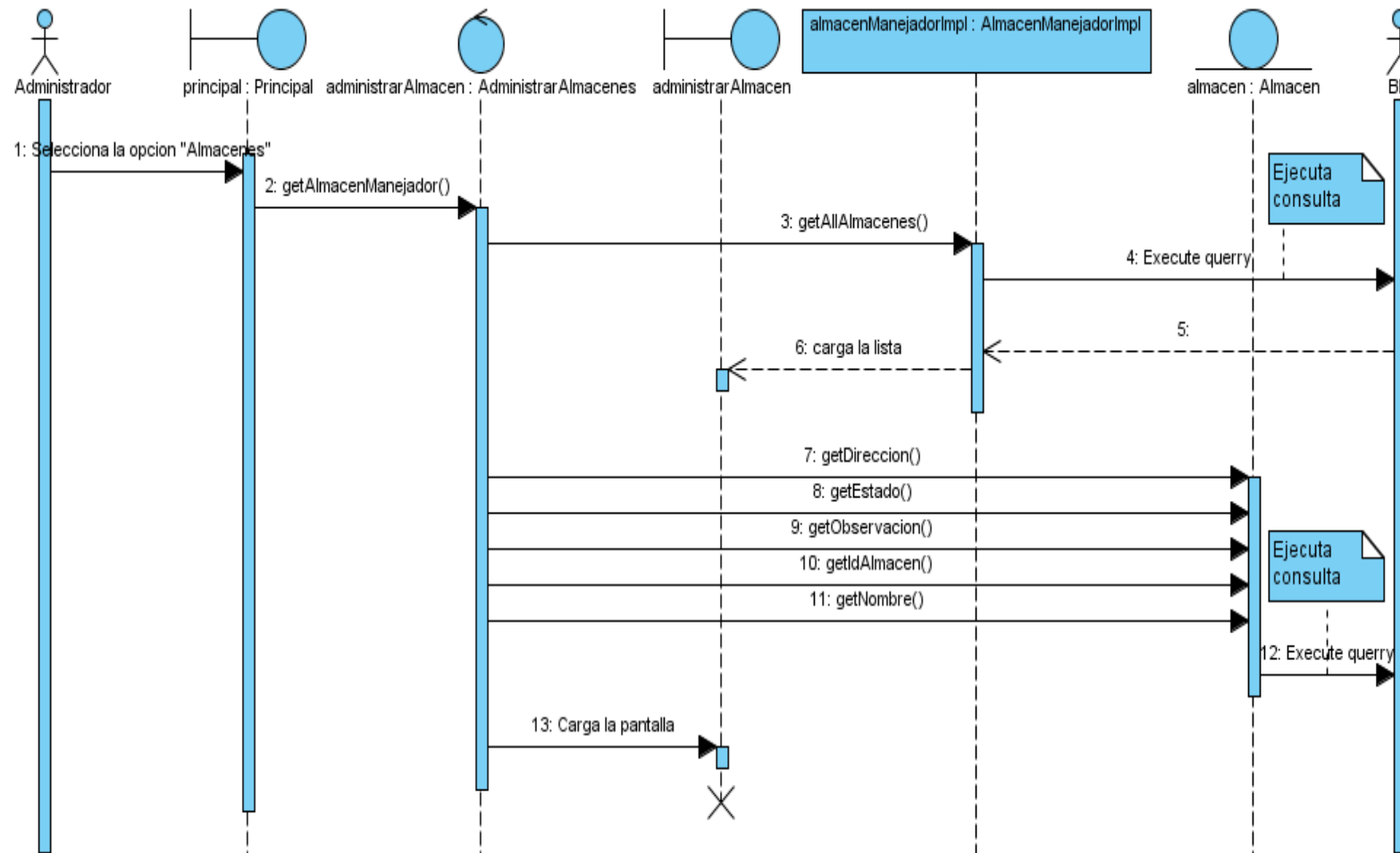


Figura 181: Diagrama de Secuencia-Administrar Almacén

Diagrama de secuencia: C.U. Adicionar Almacén

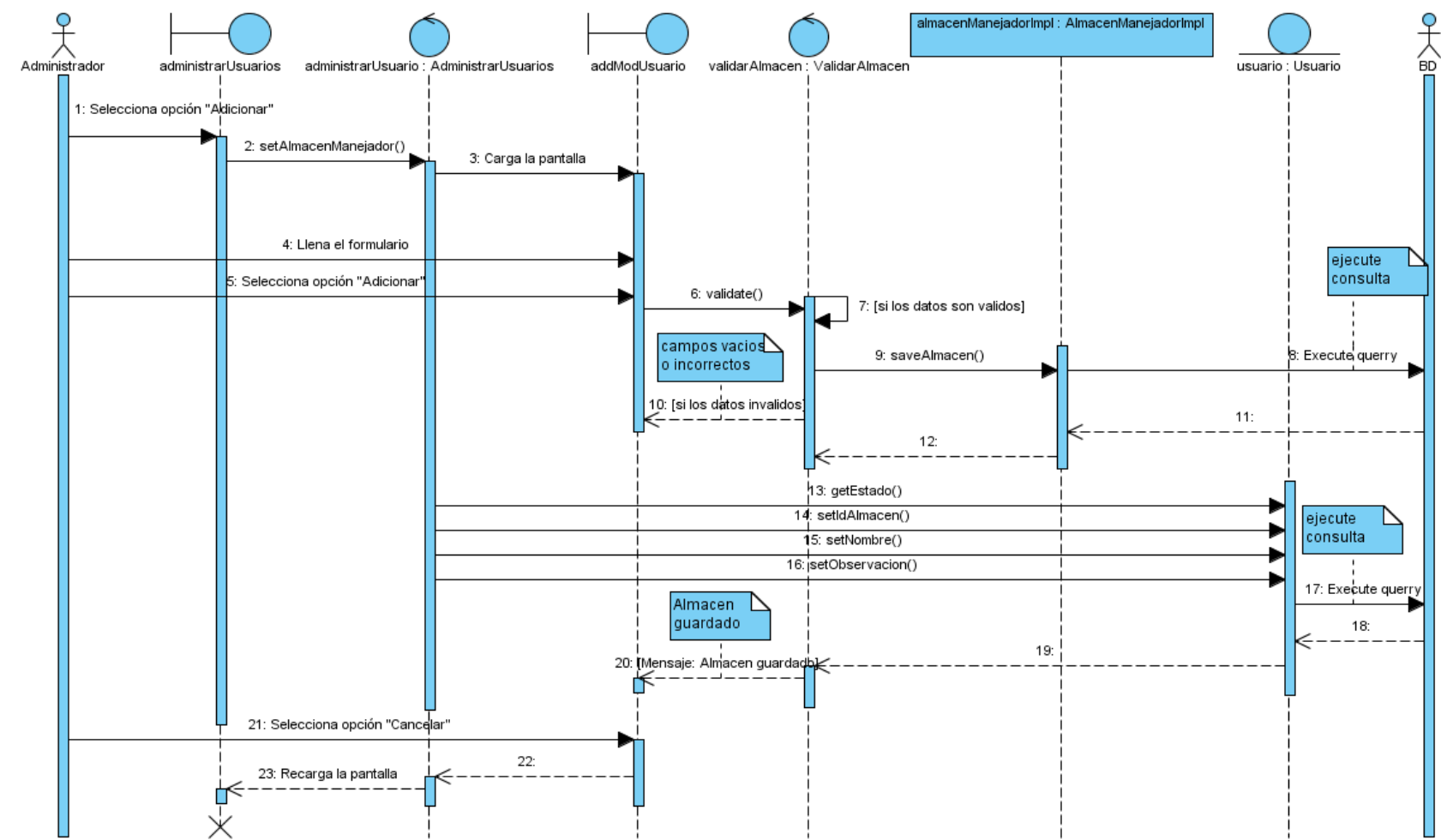


Figura 182: Diagrama de Secuencia-Adicionar Almacén

Diagrama de secuencia: C.U. Modificar Almacén

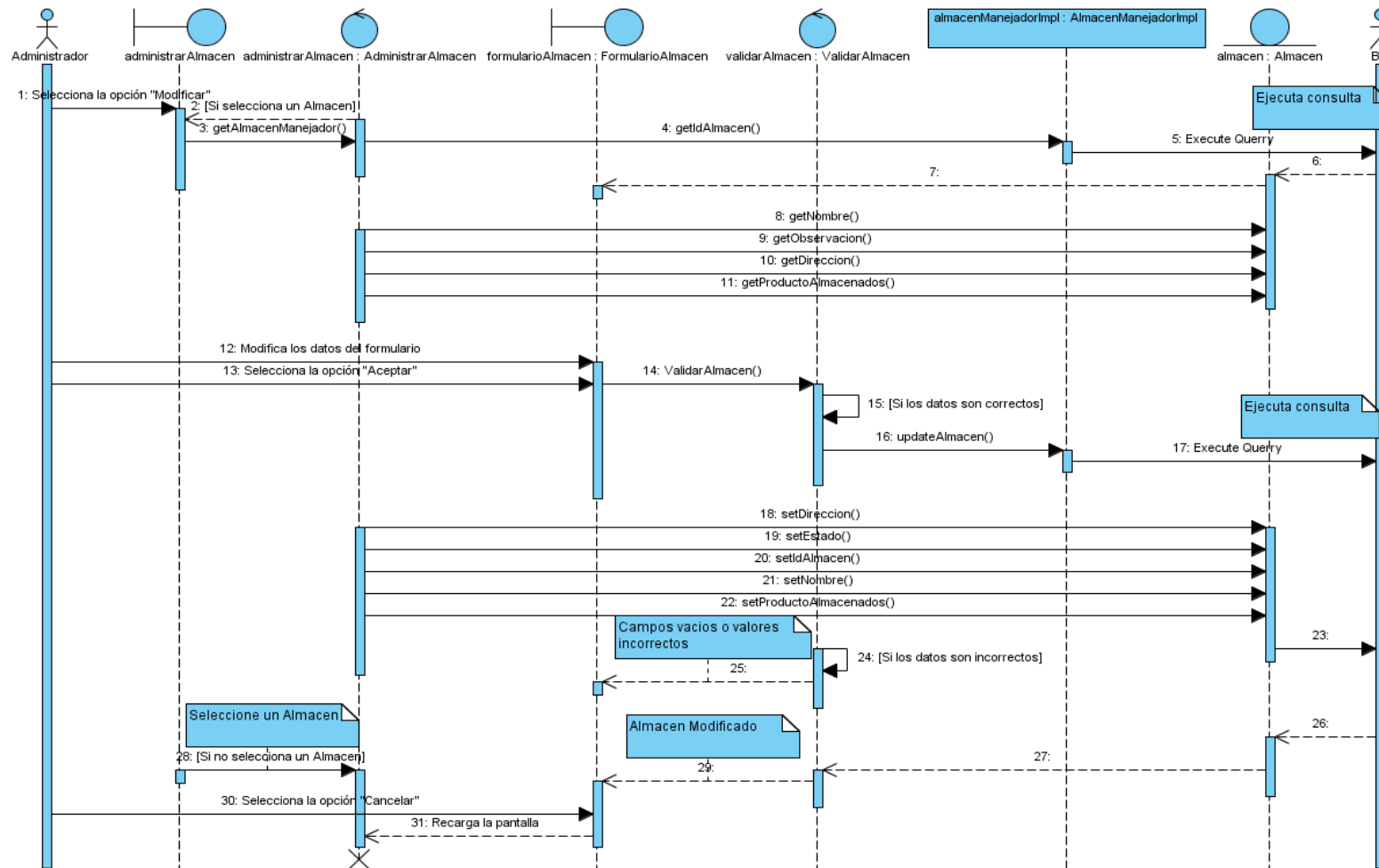


Figura 183: Diagrama de Secuencia-Modificar Almacén

Diagrama de secuencia: C.U. Eliminar Almacén

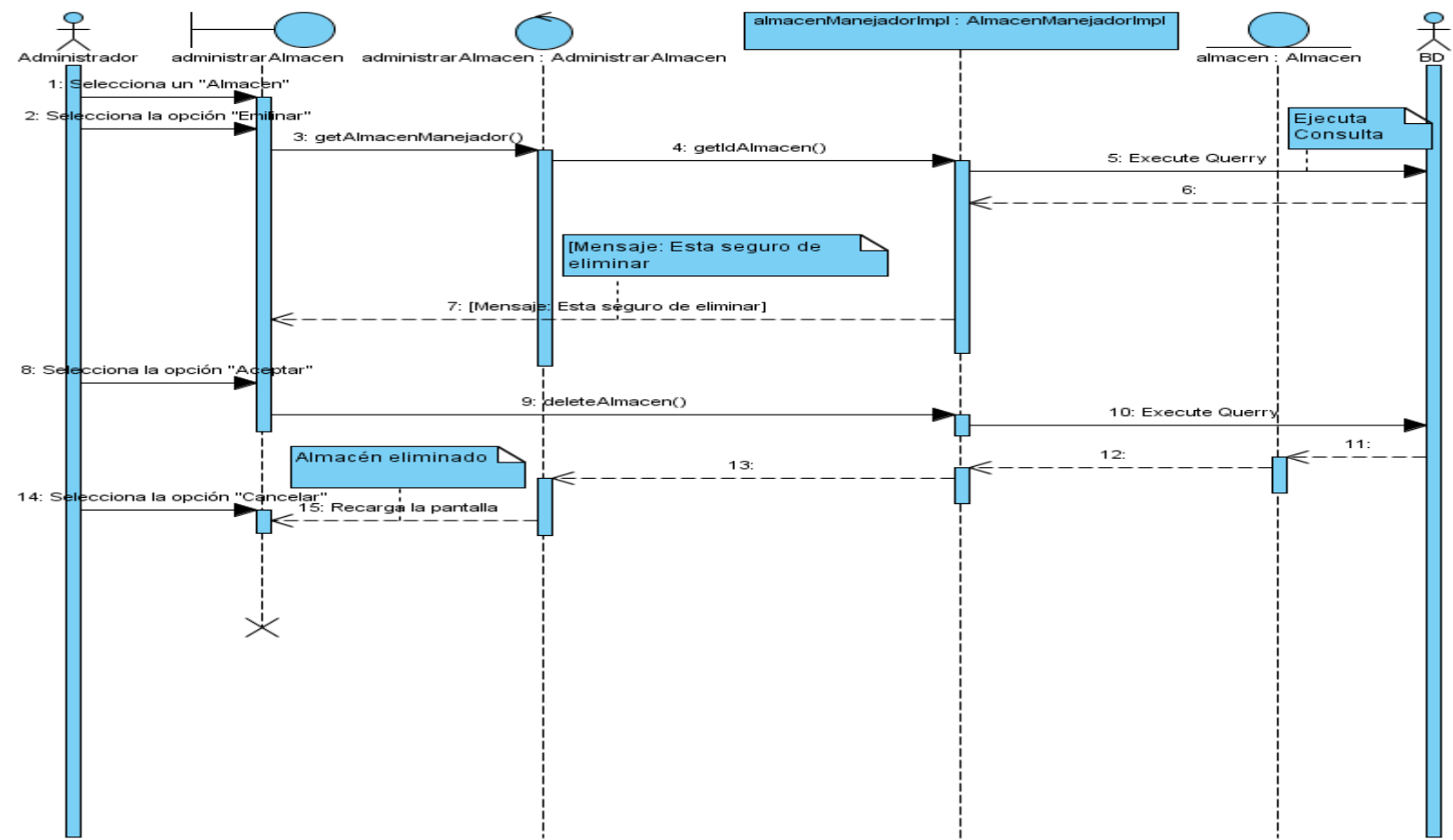


Figura 184: Diagrama de Secuencia-Eliminar Almacén

Diagrama de secuencia: C.U. Administrar Proveedores

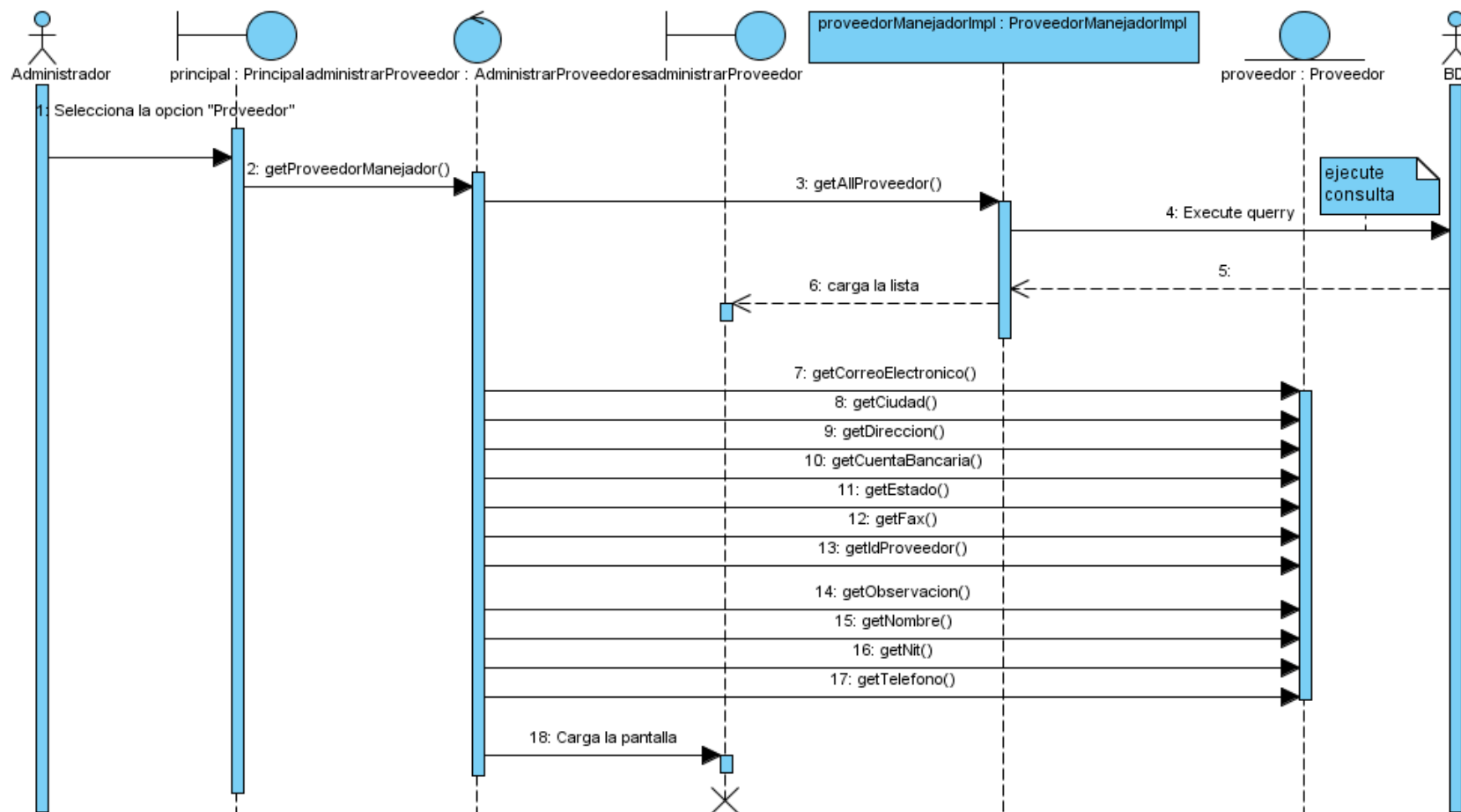


Figura 185: Diagrama de Secuencia-Administrar Proveedores

Diagrama de secuencia: Adicionar Proveedor

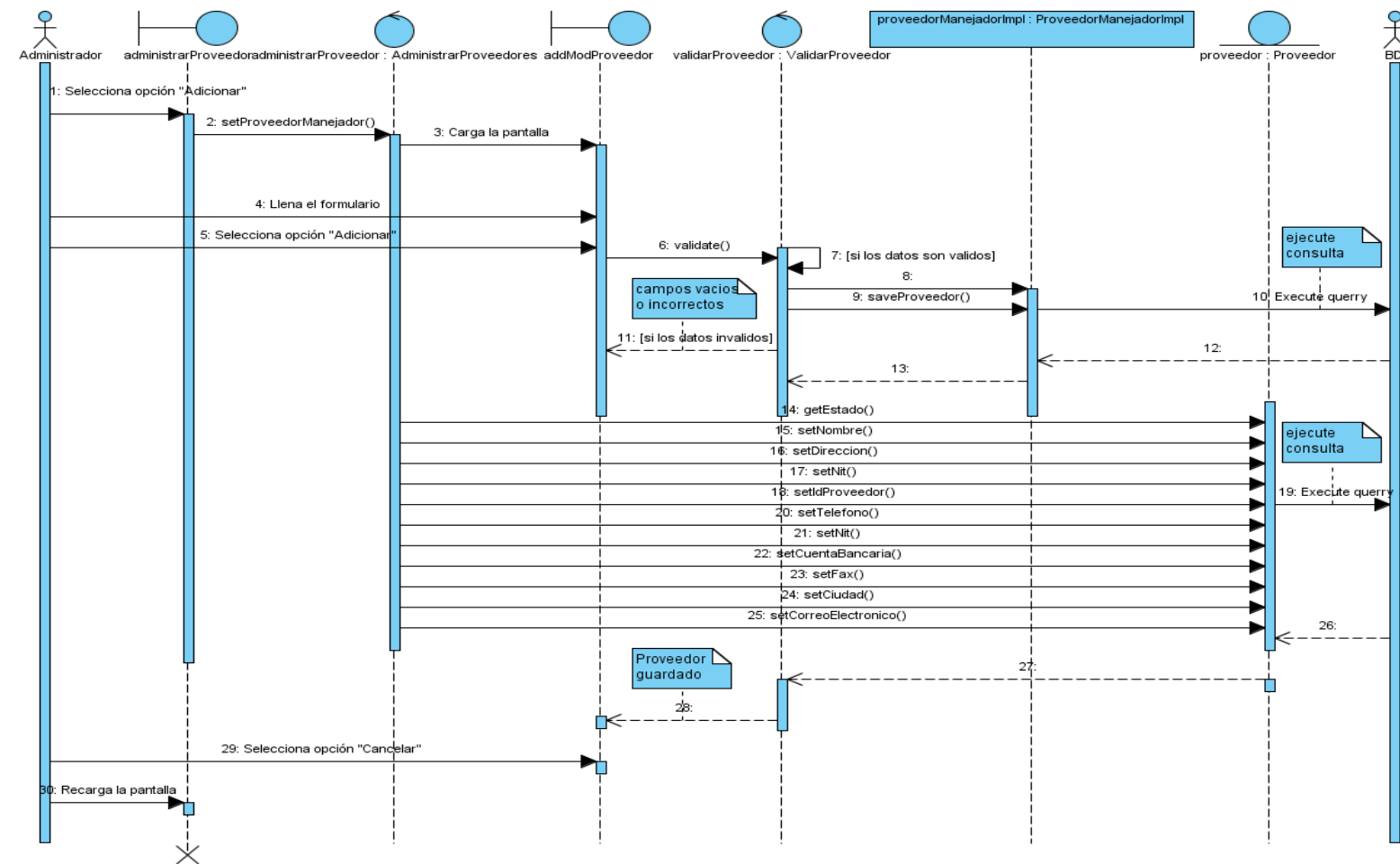


Figura 186: Diagrama de Secuencia-Adicionar Proveedor

Diagrama de secuencia: C.U. Modificar Proveedor

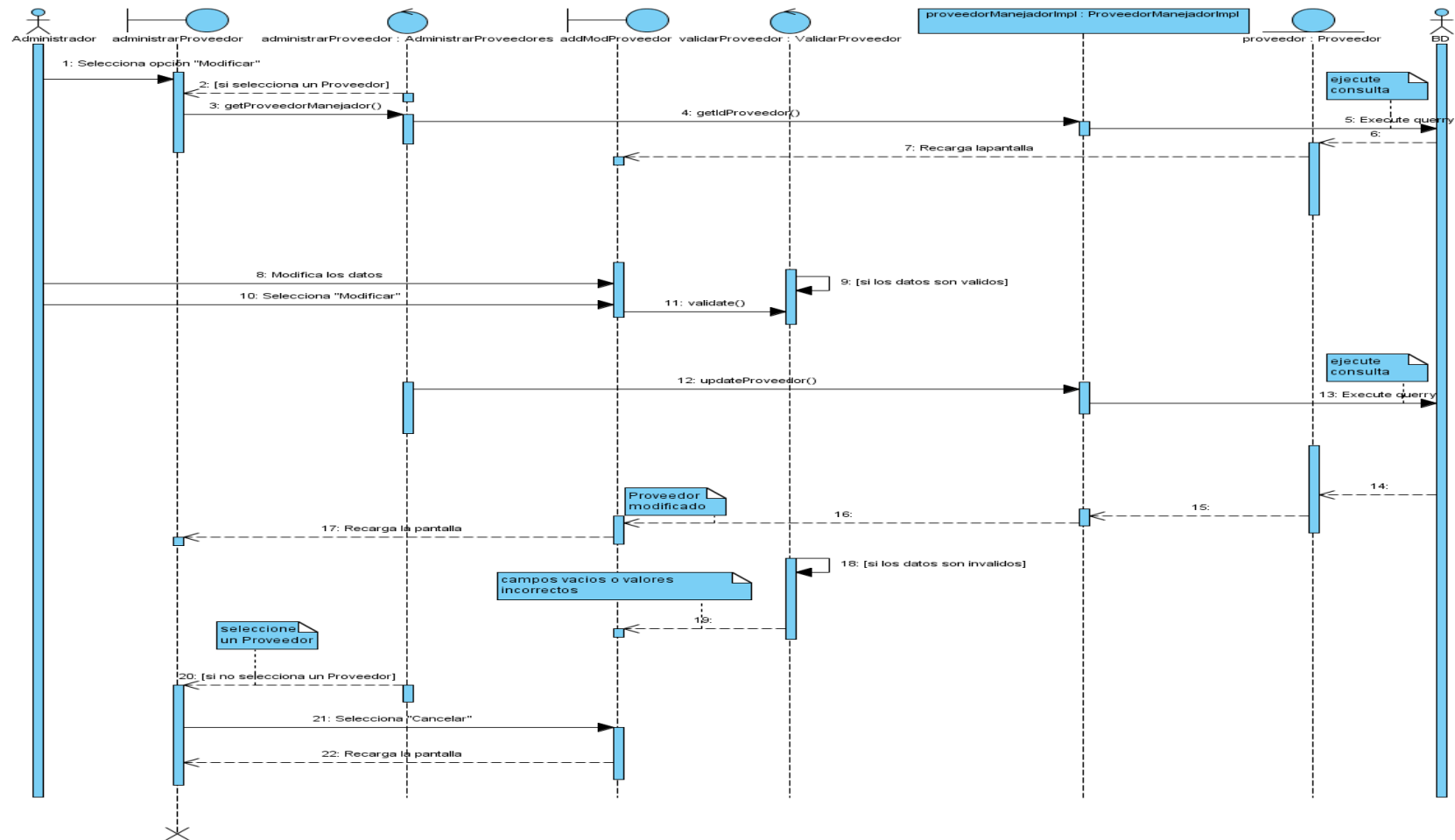


Figura 187: Diagrama de Secuencia-Modificar Proveedor

Diagrama de secuencia: C.U. Eliminar Proveedor

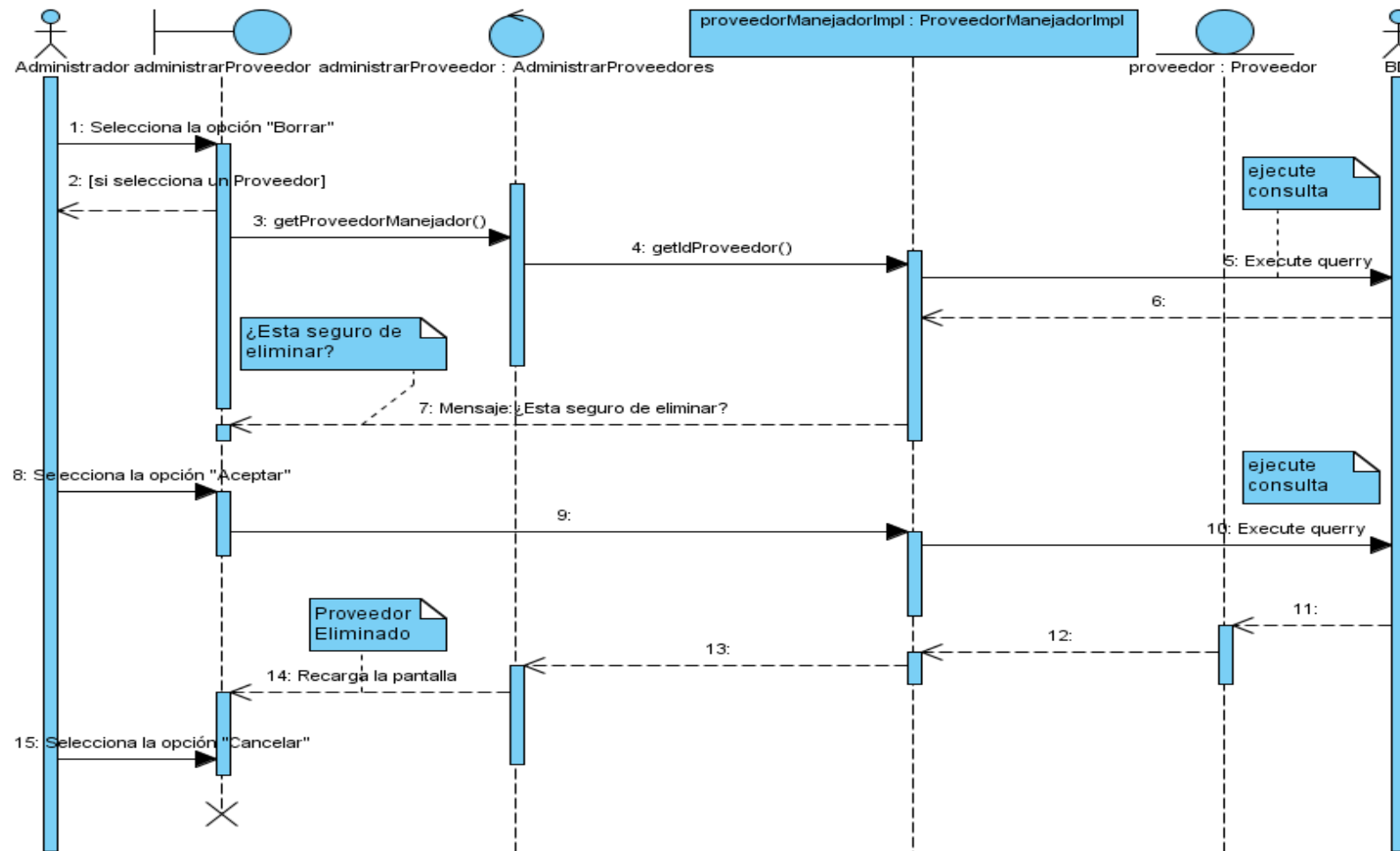


Figura 188: Diagrama de Secuencia-Eliminar Proveedor

Diagrama de secuencia: C.U. Administrar Estados

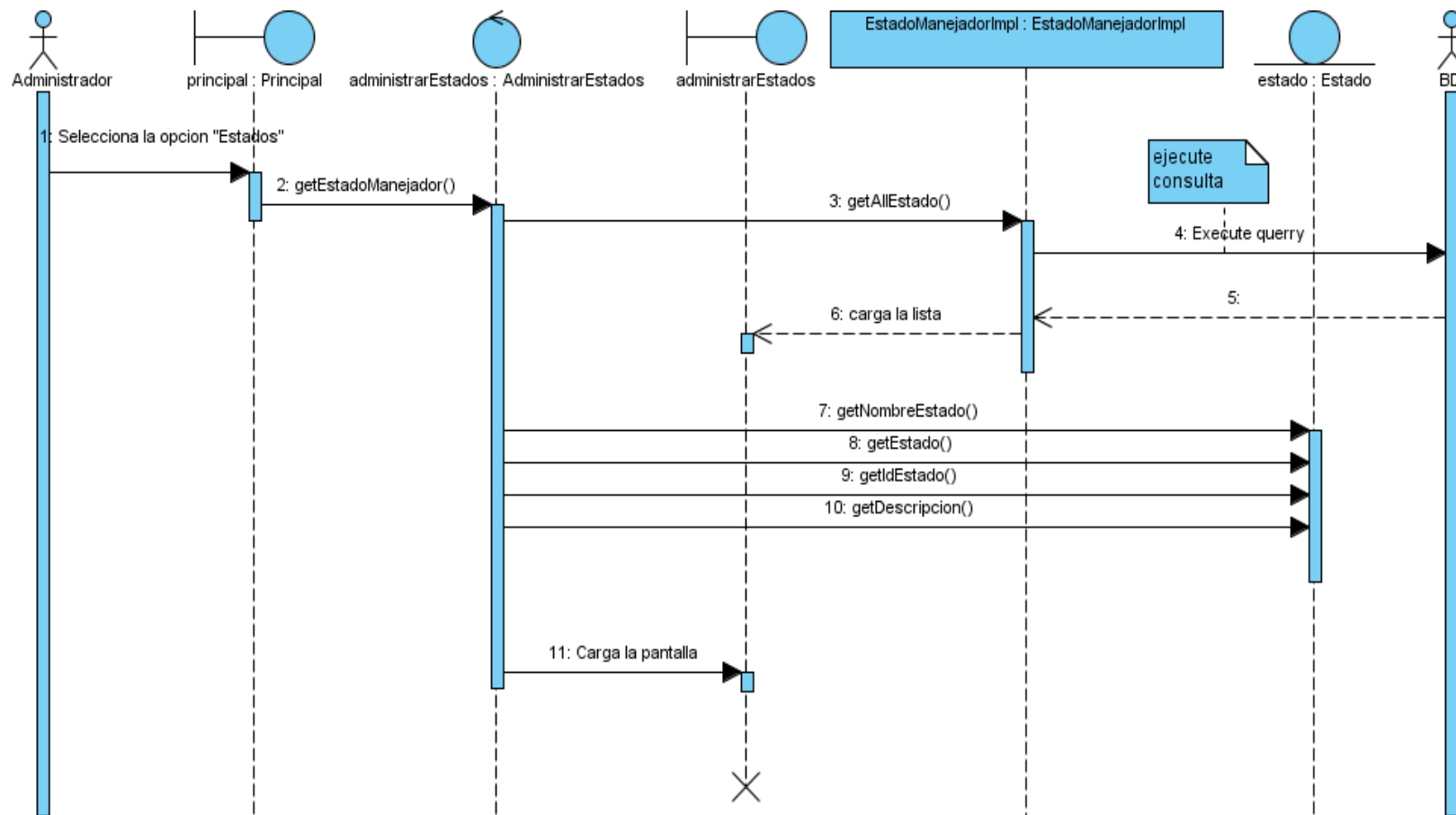


Figura 189: Diagrama de Secuencia-Administrar Estados

Diagrama de secuencia: C.U. Adicionar Estado

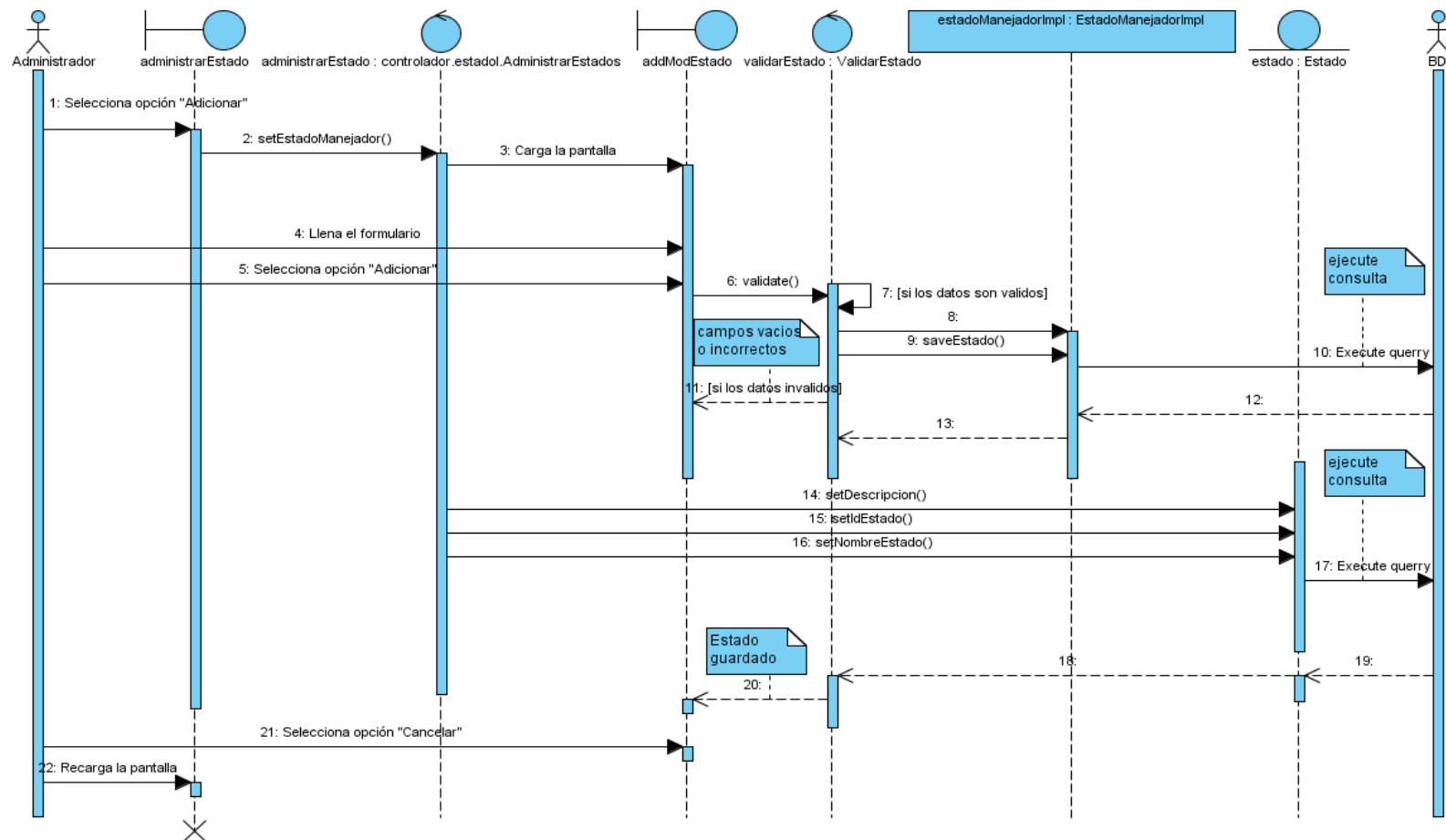


Figura 190: Diagrama de Secuencia-Adicionar Estado

```
sequenceDiagram
    actor Admin as Administrador
    participant Controller as administrarEstado
    participant Service as administrarEstados
    participant DAO as addModEstado
    participant Validator as validarEstado
    participant Entity as estado : Estado
    participant DB as BD

    Admin->>Controller: 1: Selecciona opción "Modificar"
    activate Controller
    Controller->>Service: 2: [si selecciona un Estado]
    activate Service
    Service->>DAO: 3: getEstadoManejador()
    activate DAO
    DAO->>Entity: 4: getIdEstado()
    activate Entity
    Entity->>DB: 5: Execute query
    activate DB
    DB-->>Entity: 6:
    deactivate DB
    Entity-->>DAO: 7: Recarga la pantalla
    deactivate Entity
    deactivate DAO
    Service->>Entity: 8: getDescripcion()
    Service->>Entity: 9: getIdEstado()
    Service->>Entity: 10: setNombreEstado()
    activate Entity
    Entity-->>DAO: 11: Recarga la pantalla
    deactivate Entity
    DAO-->>Controller: 12:
    deactivate DAO
    Controller->>DAO: 14: Modifica los datos
    Controller->>DAO: 16: Selecciona "Modificar"
    activate DAO
    DAO->>Validator: 17: validate()
    activate Validator
    Validator->>Entity: 15: [si los datos son validos]
    activate Entity
    Entity->>DB: 19: Execute query
    activate DB
    DB-->>Entity: 20:
    deactivate DB
    Entity-->>DAO: 21: setNombreEstado()
    Entity-->>DAO: 22: setDescripcion()
    Entity-->>DAO: 23: setIdEstado()
    activate DAO
    DAO-->>Controller: 24: Estado modificado
    deactivate DAO
    Controller-->>Admin: 25: Recarga la pantalla
    deactivate Controller
    Admin->>Controller: 26: [si no selecciona un Estado]
    activate Controller
    Controller->>DAO: 29: Selecciona "Cancelar"
    activate DAO
    DAO-->>Admin: 30: Recarga la pantalla
    deactivate DAO
    deactivate Controller
    destroy Admin
```

Figura 191: Diagrama de Secuencia-Modificar Estado

Diagrama de secuencia: C.U. Eliminar Estado

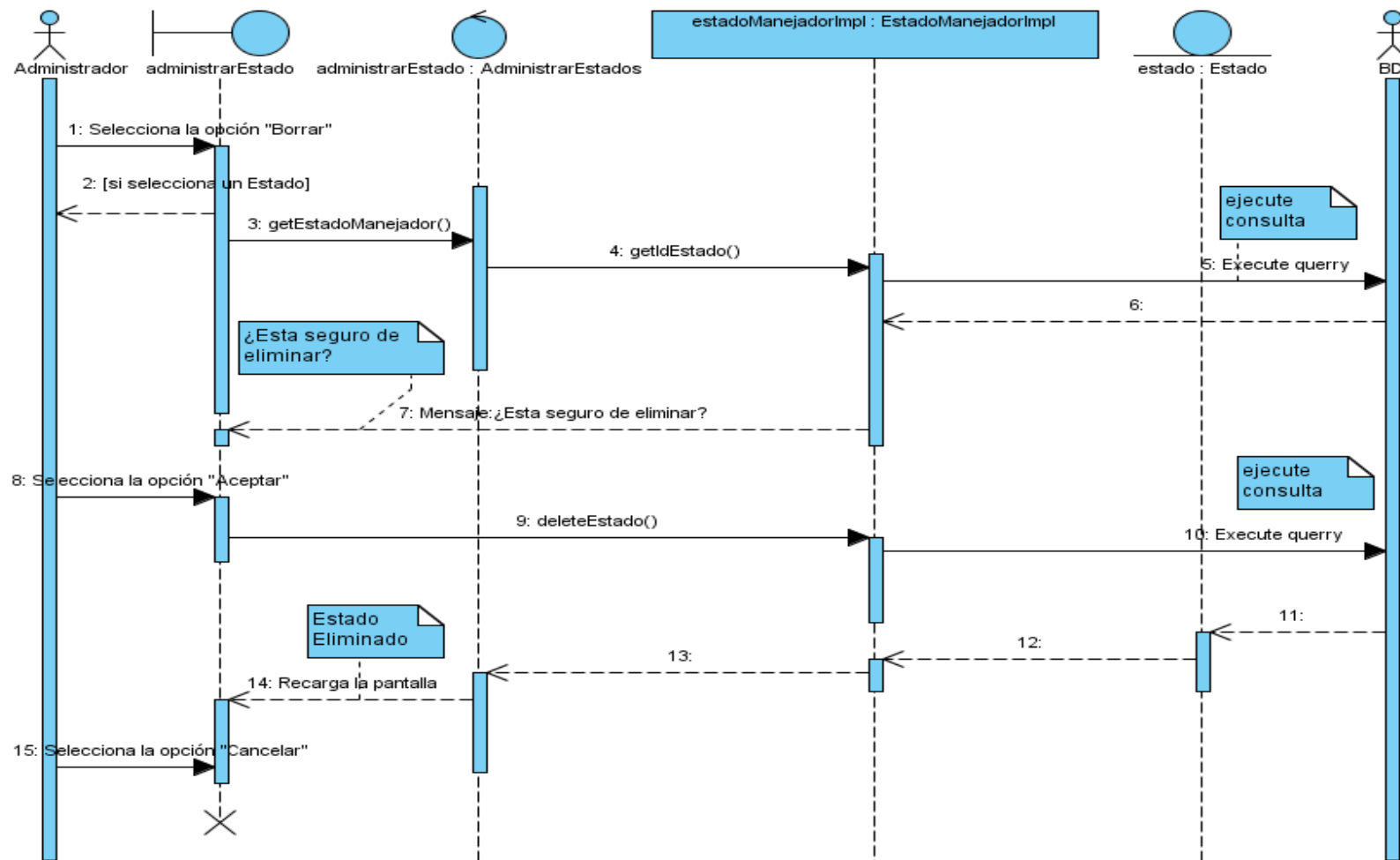


Figura 192: Diagrama de Secuencia-Eliminar Estado

Diagrama de secuencia: C.U. Generar Reporte

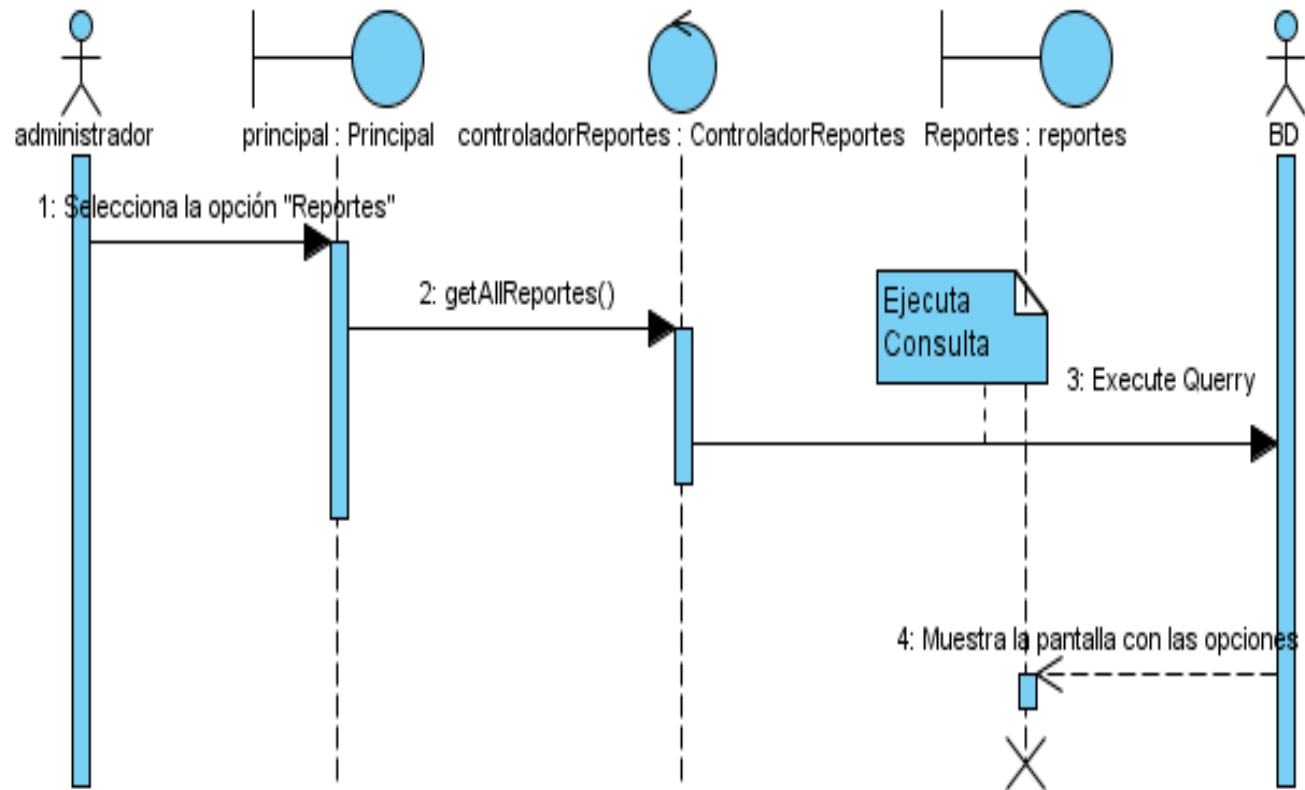


Figura 193: Diagrama de Secuencia-Generar Reportes

Diagrama de secuencia: C.U. Reporte Producto Almacenado

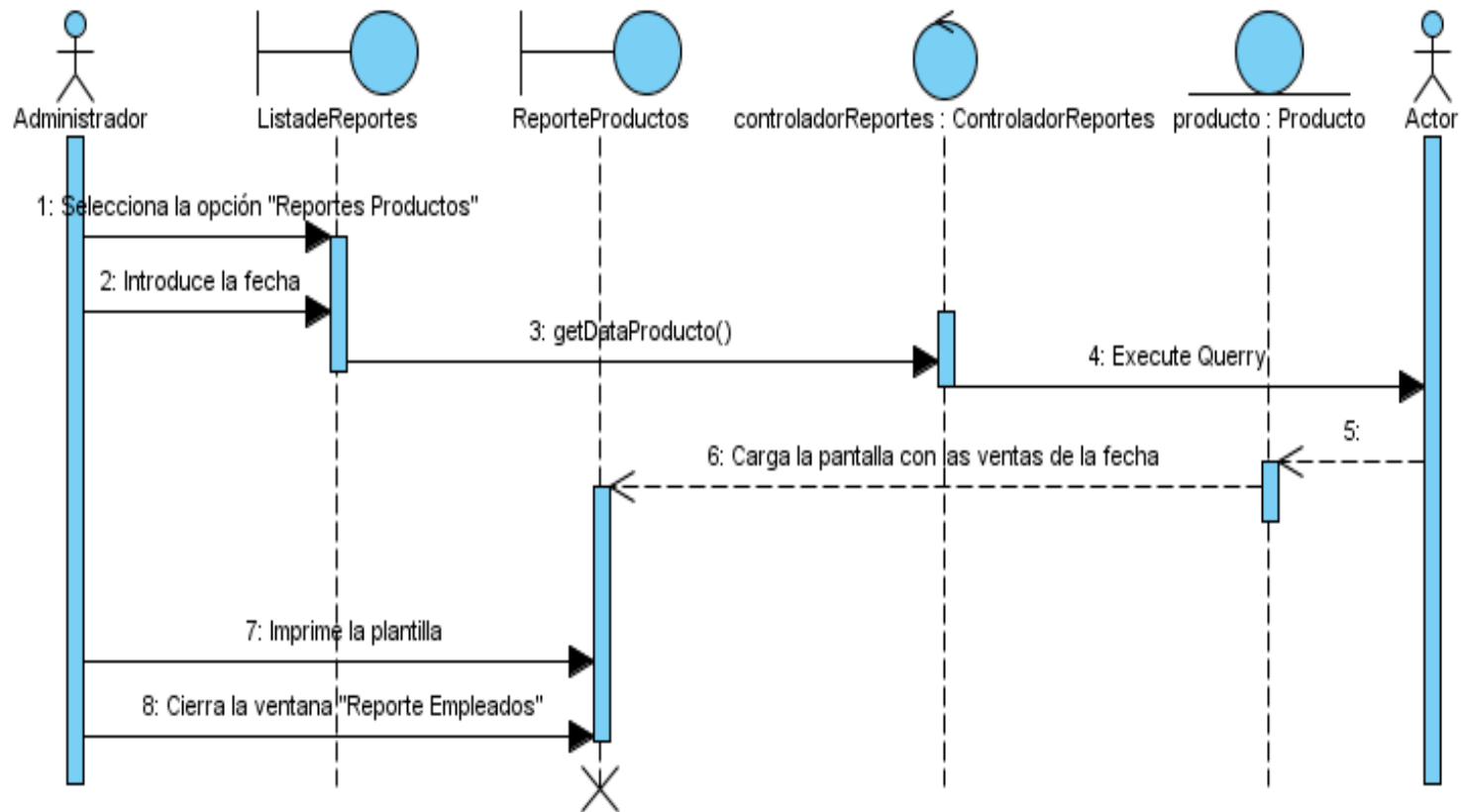


Figura 194: Diagrama de Secuencia- Reporte Producto Almacenado

Diagrama de secuencia: C.U. Reporte Cantidad de Compras Clientes

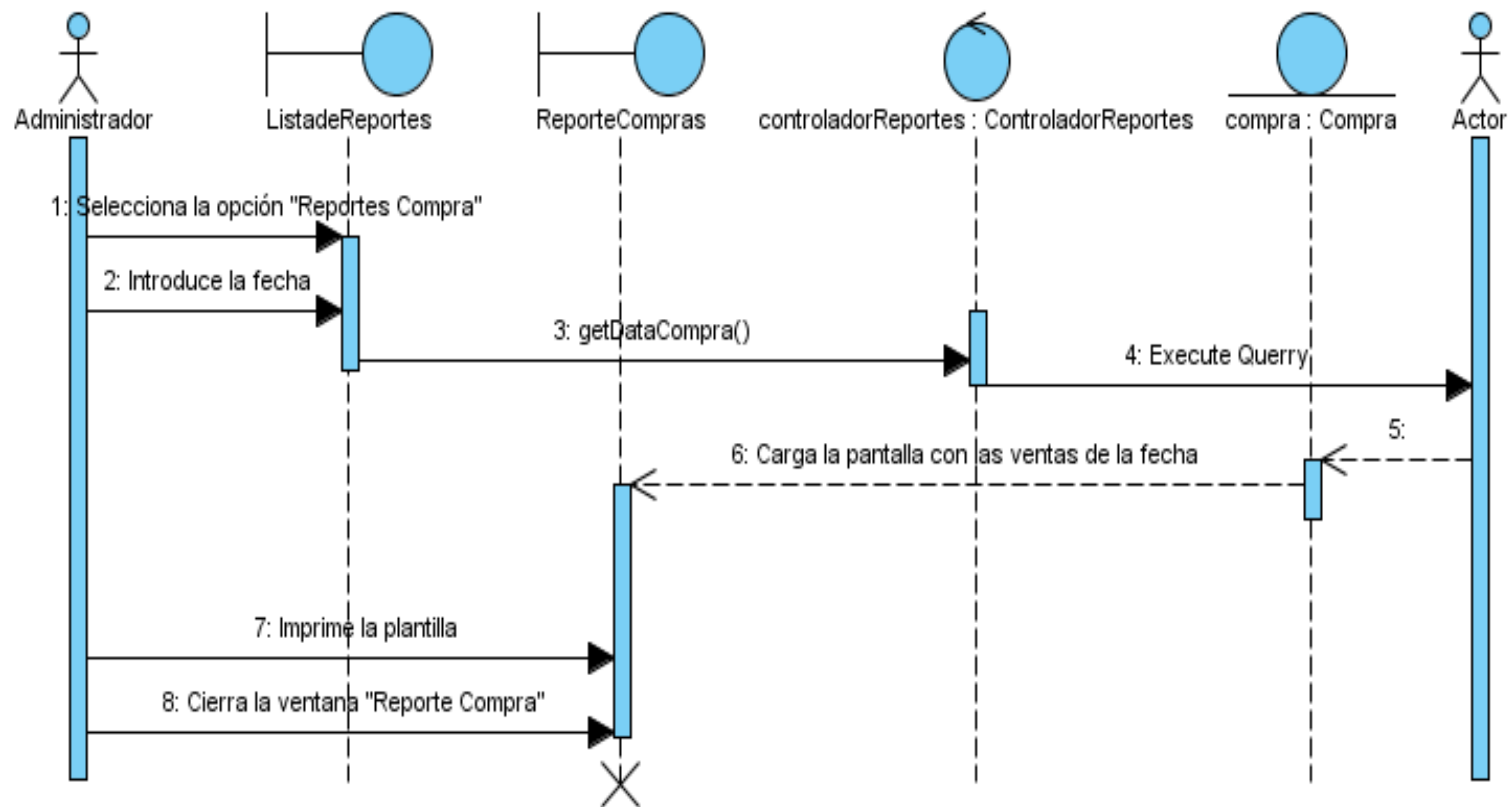


Figura 195: Diagrama de Secuencia- Reporte Compra

Diagrama de secuencia: C.U. Reporte de Ventas

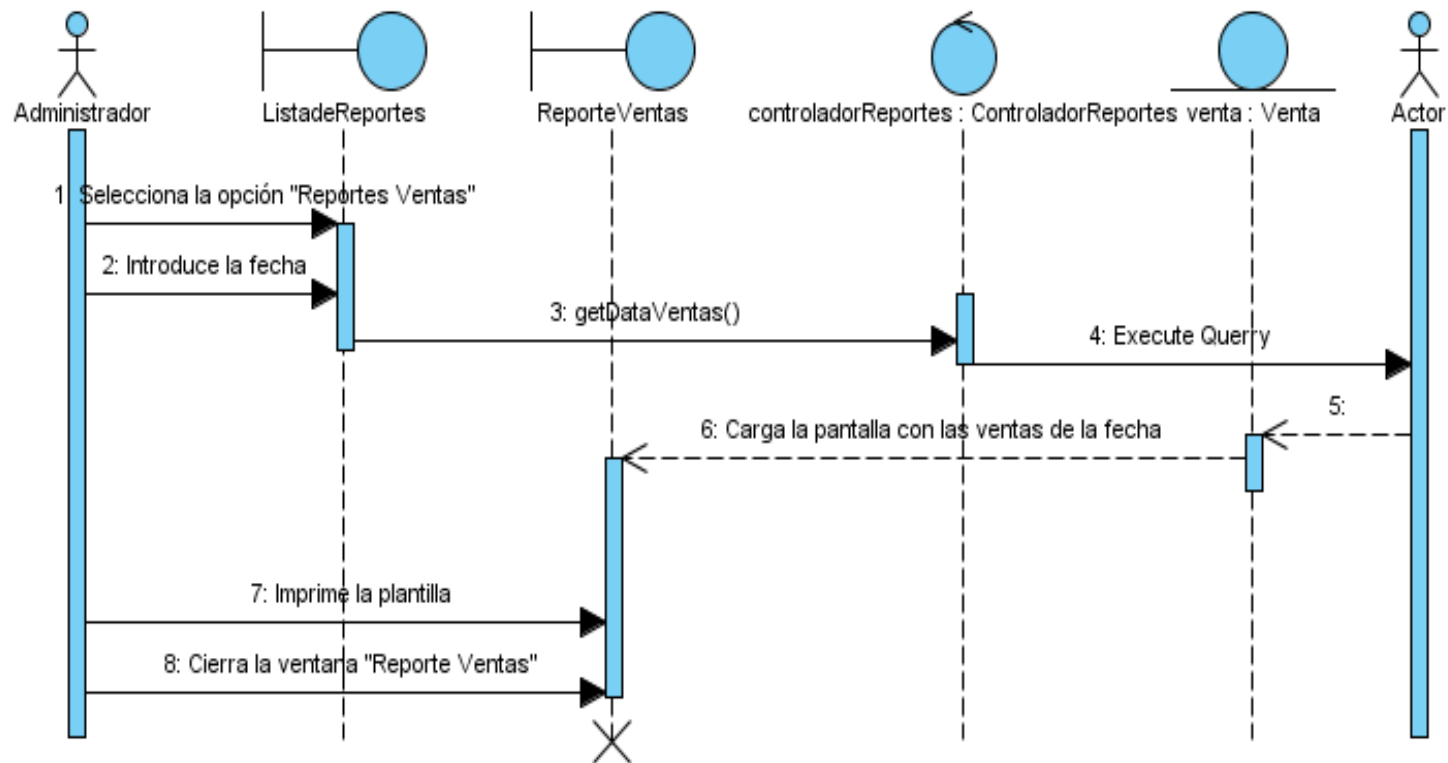


Figura 196: Diagrama de Secuencia- Reporte Venta

2.1.2.2.3.1.1 DIAGRAMAS DE COLABORACIÓN

Diagrama de colaboración: C.U. Ingresar al Sistema

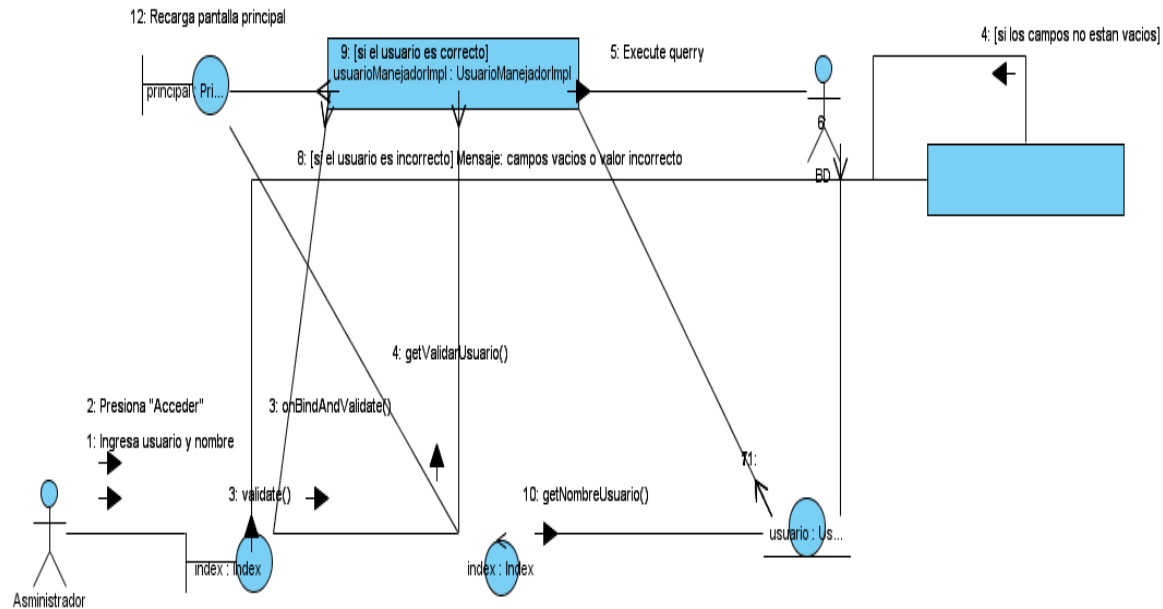


Figura 197: Diagrama de Colaboración-Ingresar al Sistema

Diagrama de colaboración: C.U. Administrar Usuario

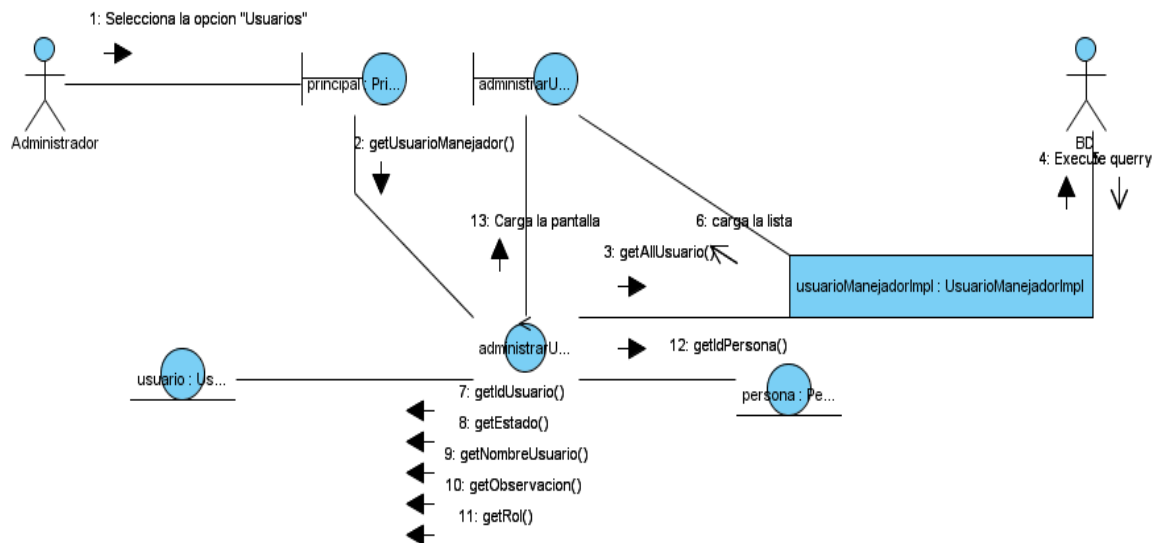


Figura 198: Diagrama de Colaboración-Administrar Usuarios

Diagrama de colaboración: C.U. Adicionar Usuario

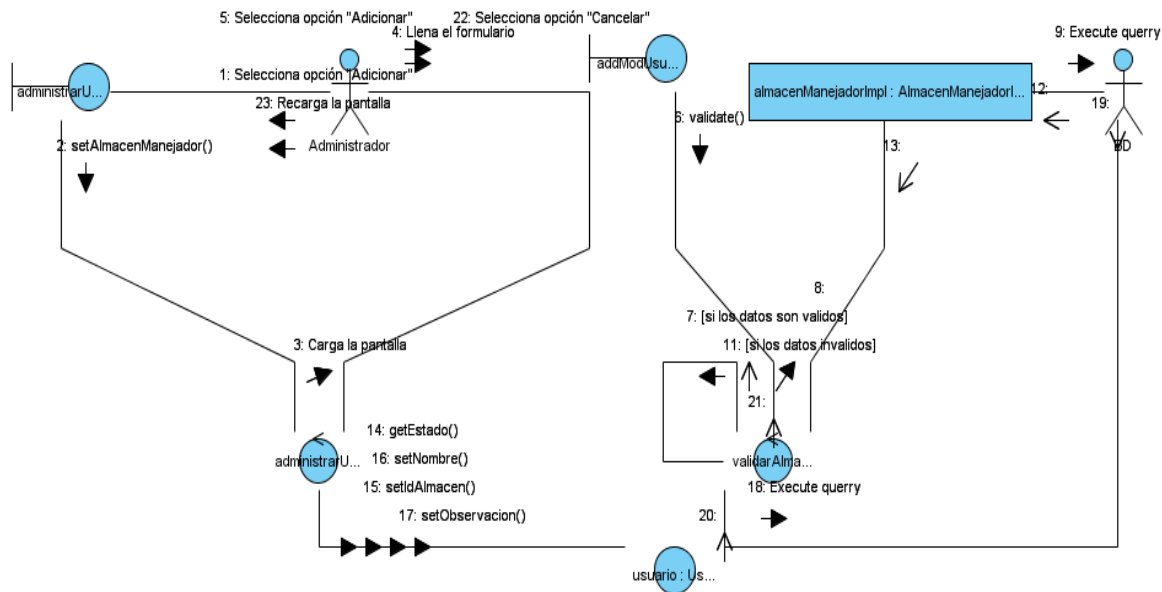


Figura 199: Diagrama de Colaboración-Adicionar Usuario

Diagrama de colaboración: C.U. Modificar Usuario

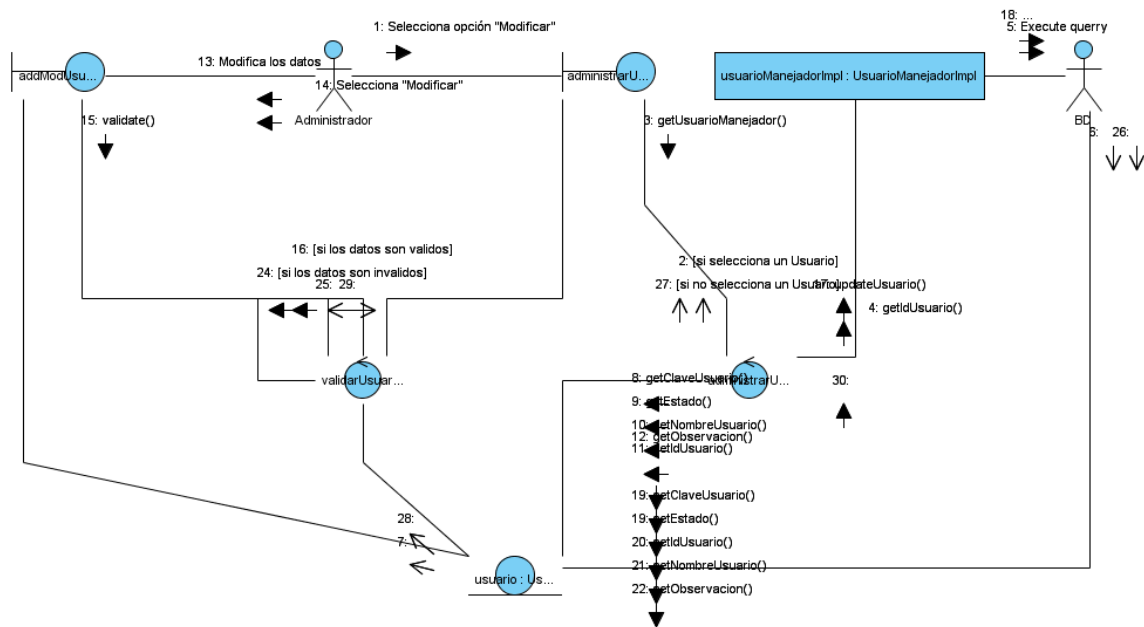


Figura 200: Diagrama de Colaboración-Modificar Usuario

Diagrama de colaboración: C.U. Eliminar Usuario

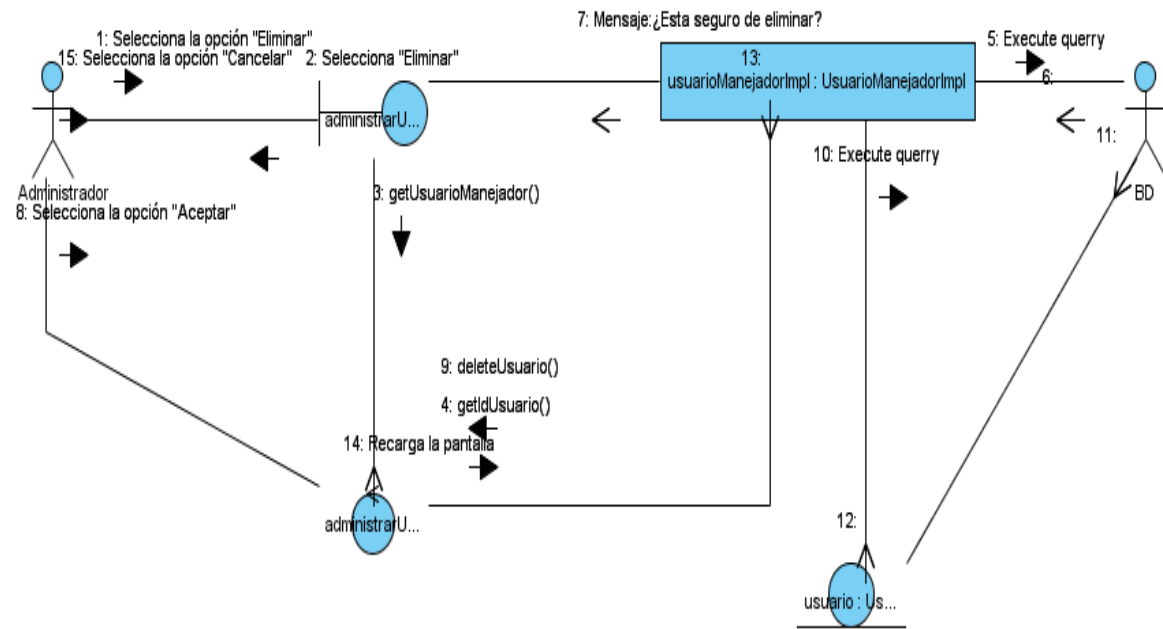


Figura 201: Diagrama de Colaboración-Eliminar Usuario

Diagrama de colaboración: C.U. Administrar Almacenes

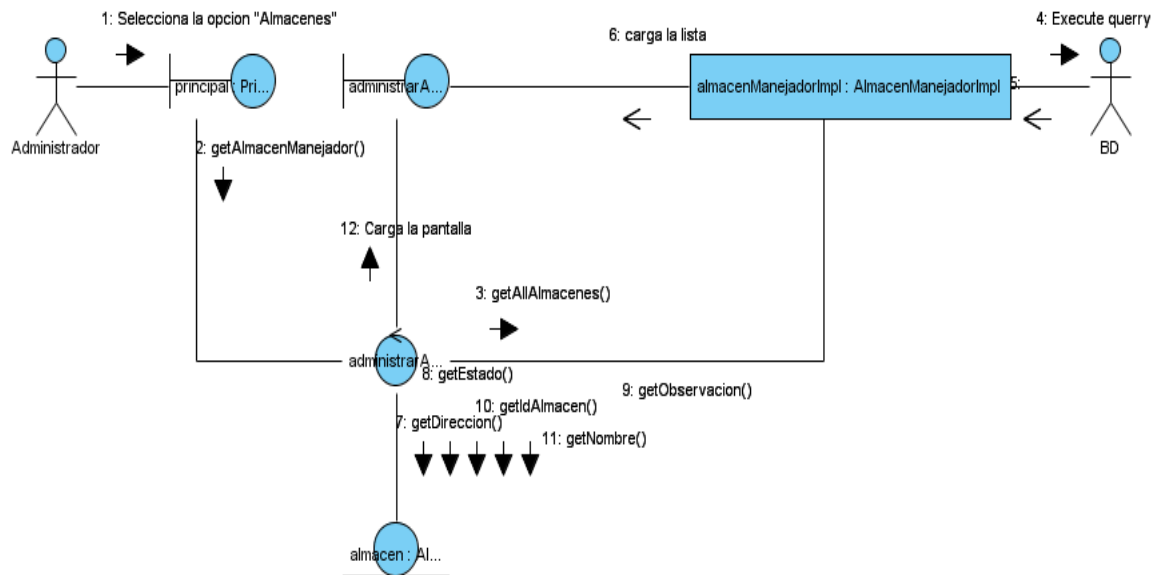


Figura 202: Diagrama de Colaboración-Administrar Almacenes

Diagrama de colaboración: C.U. Adicionar Almacén

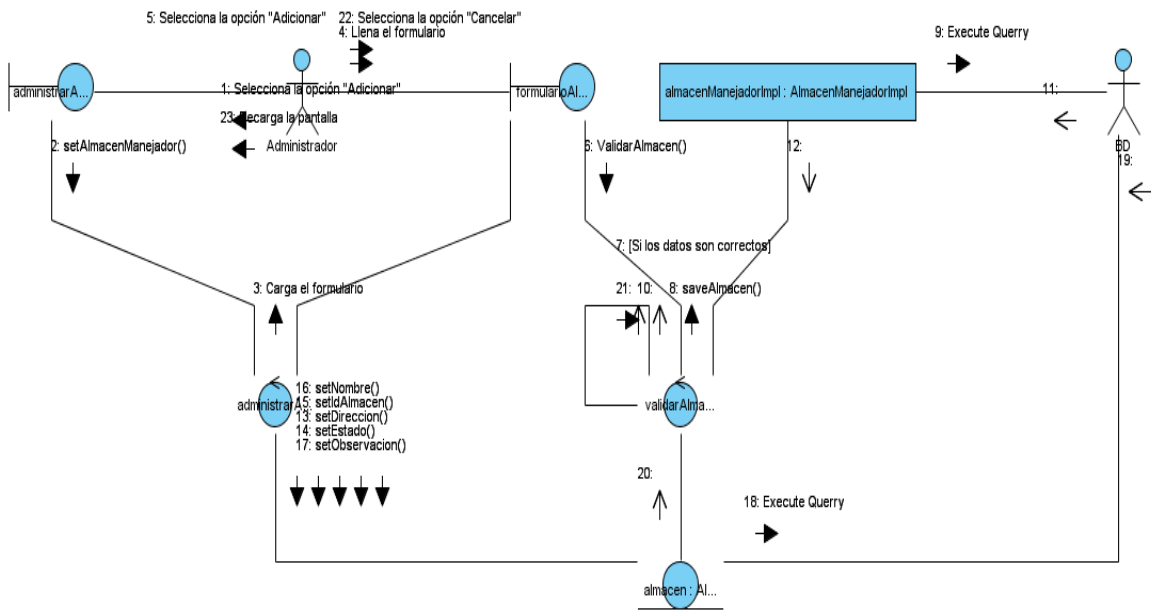


Figura 203: Diagrama de Colaboración-Adicionar Almacén

Diagrama de colaboración: C.U. Modificar Almacén

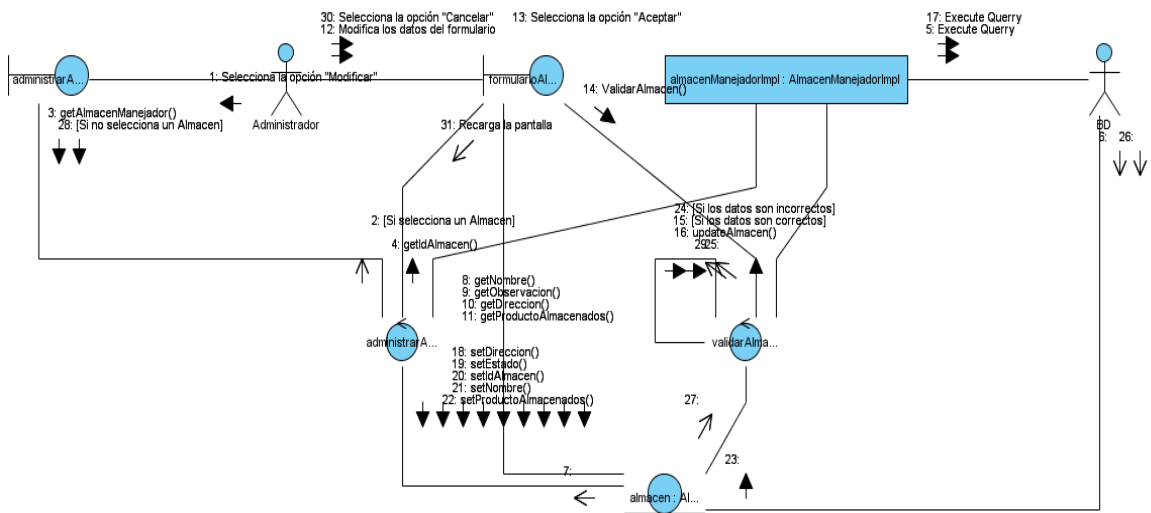


Figura 204: Diagrama de Colaboración-Modificar Almacén

Diagrama de colaboración: C.U. Eliminar Almacén

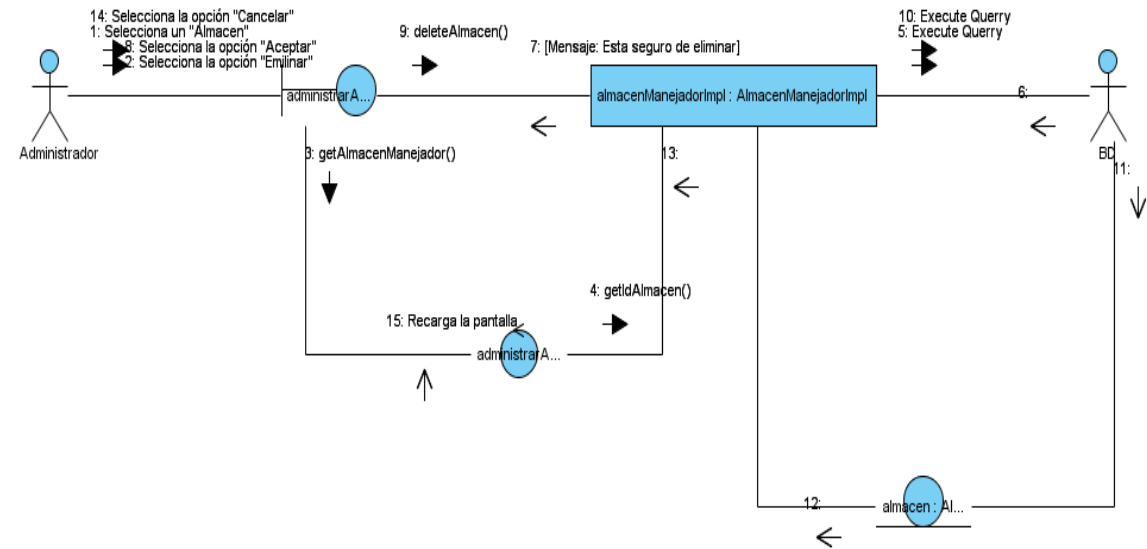


Figura 205: Diagrama de Colaboración-Eliminar Almacén

Diagrama de colaboración: C.U. Administrar Cargos

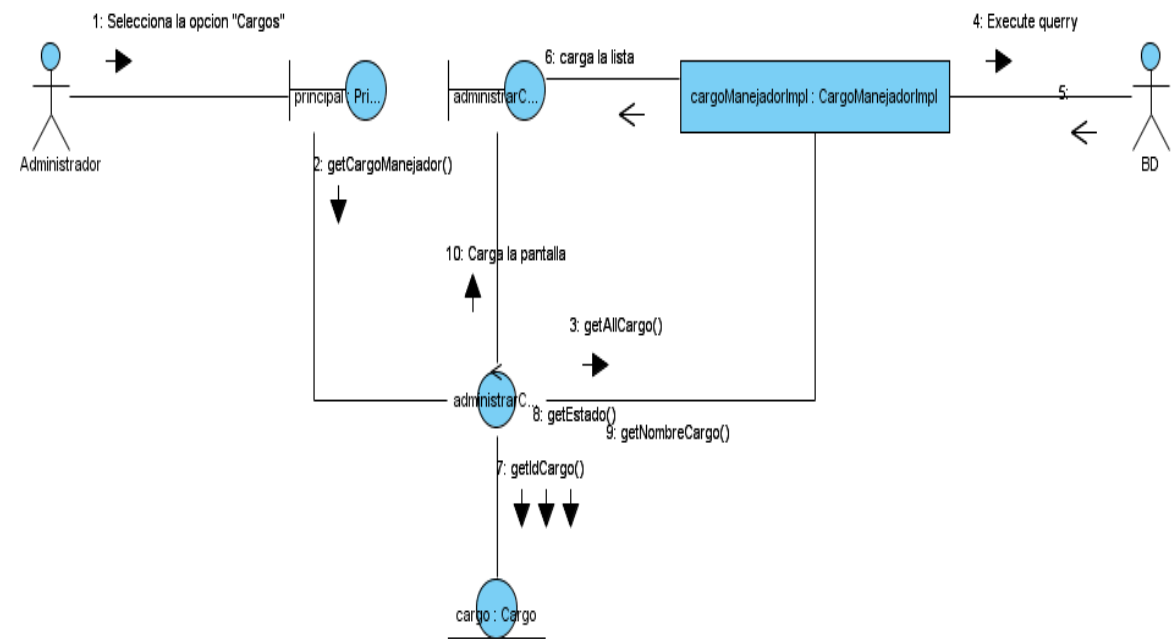


Figura 206: Diagrama de Colaboración-Administrar Cargos

Diagrama de colaboración: C.U. Adicionar Cargo

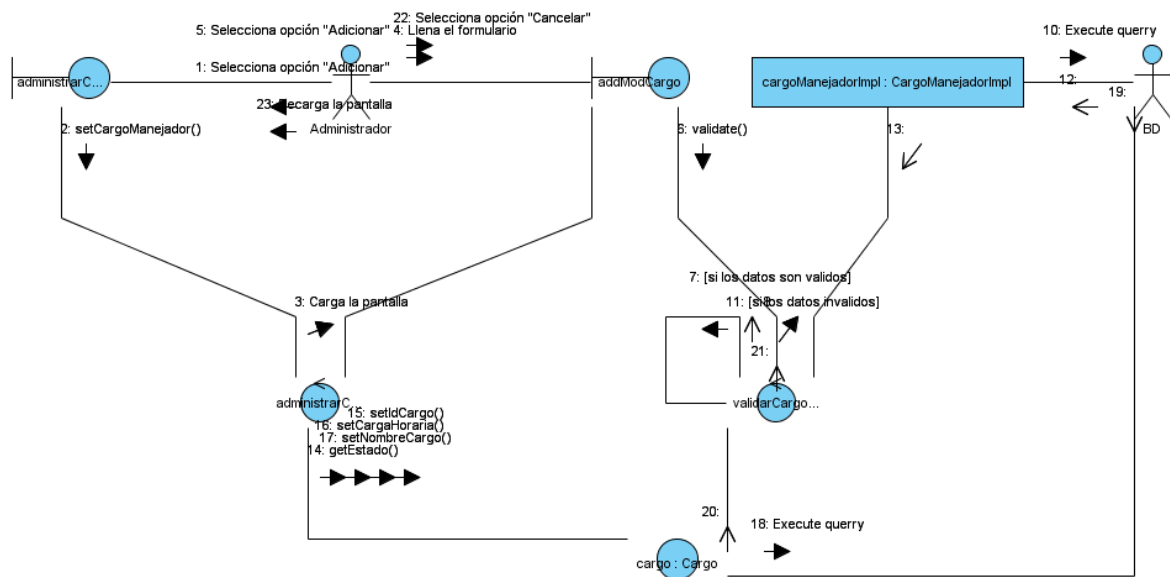


Figura 207: Diagrama de Colaboración-Adicionar Cargo

Diagrama de colaboración: C.U. Modificar Cargo

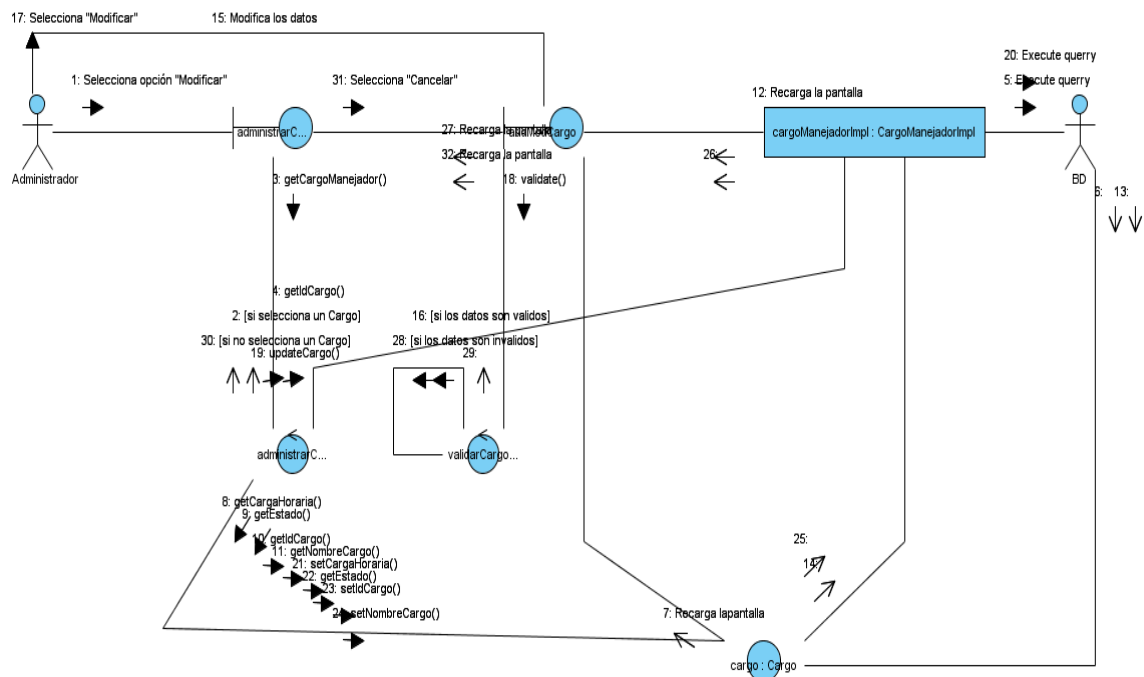


Figura 208: Diagrama de Colaboración-Modificar Cargo

Diagrama de colaboración: C.U. Eliminar Cargo

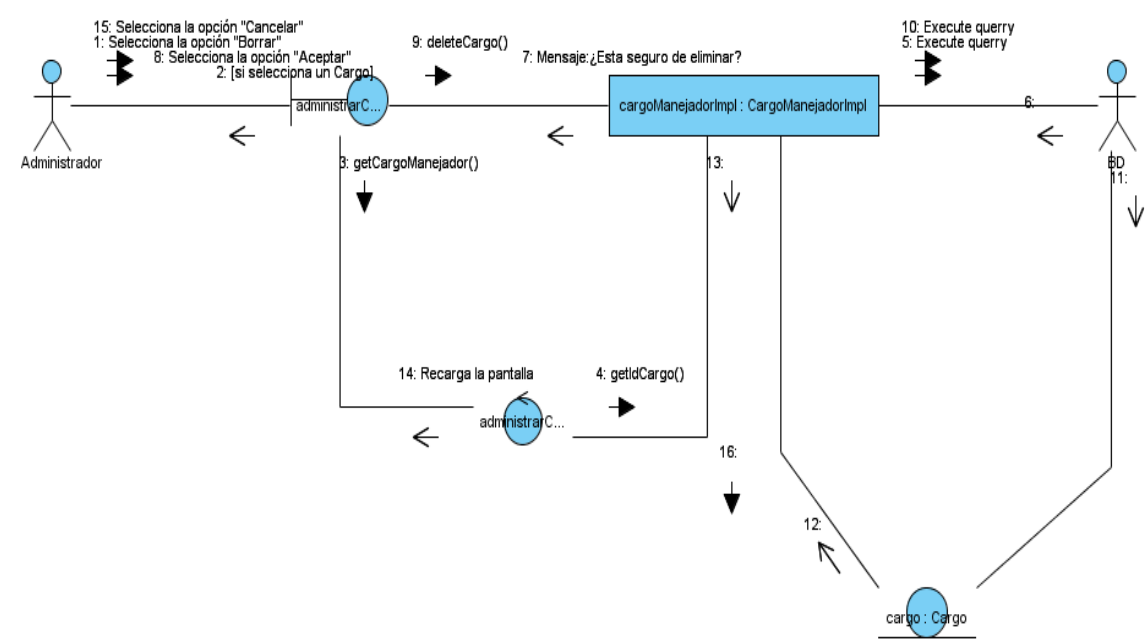


Figura 209: Diagrama de Colaboración-Eliminar Cargo

Diagrama de colaboración: C.U. Administrar Empleados

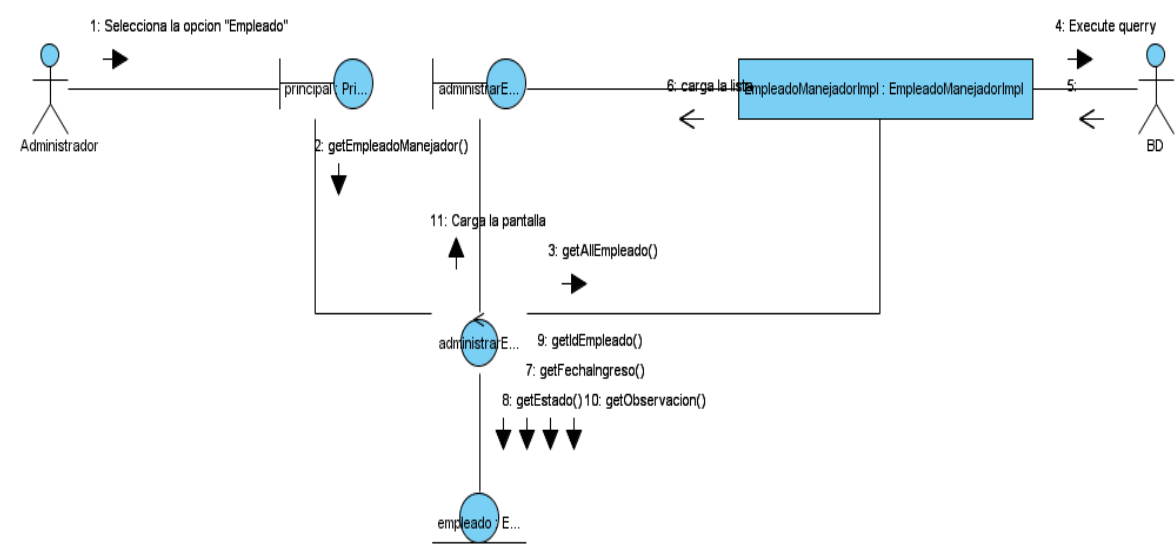


Figura 210: Diagrama de Colaboración-Administrar Empleados

Diagrama de colaboración: C.U. Agregar empleado

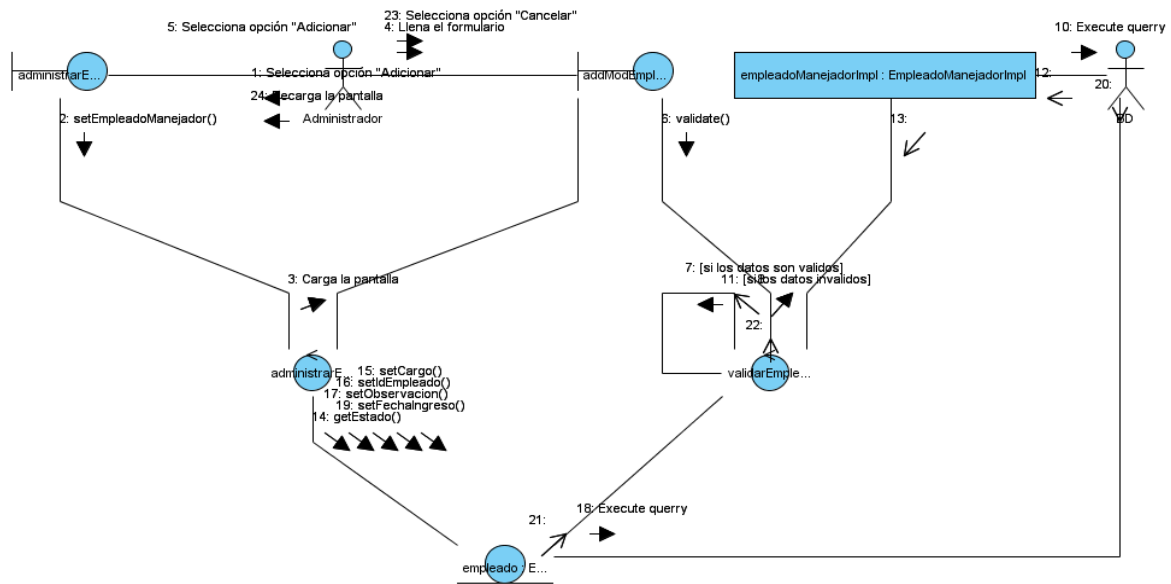


Figura 211: Diagrama de Colaboración-Adicionar Empleado

Diagrama de colaboración: C.U. Modificar Empleado

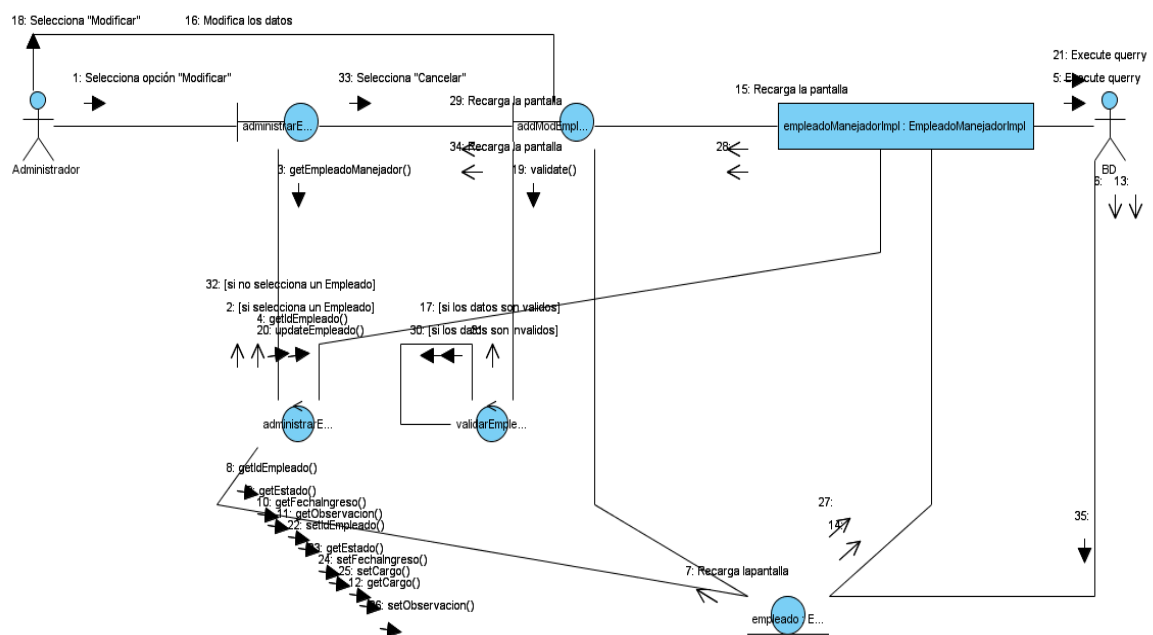


Figura 212: Diagrama de Colaboración-Modificar Empleado

Diagrama de colaboración: C.U. Eliminar Empleado

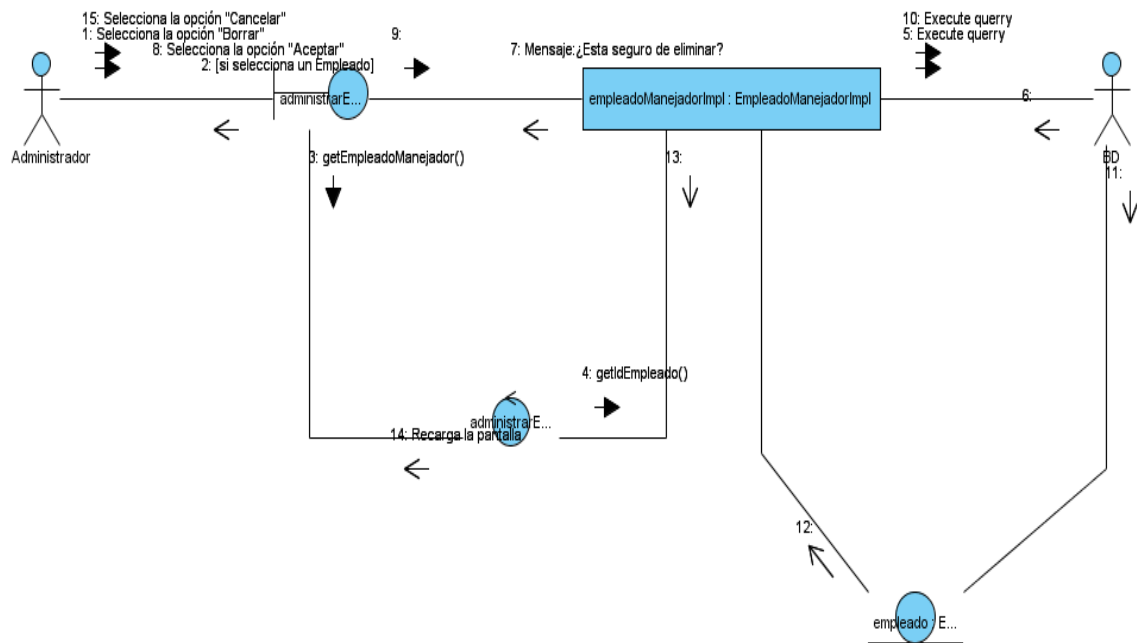


Figura 213: Diagrama de Colaboración-Eliminar Empleado

Diagrama de colaboración: C.U. Administrar Estados

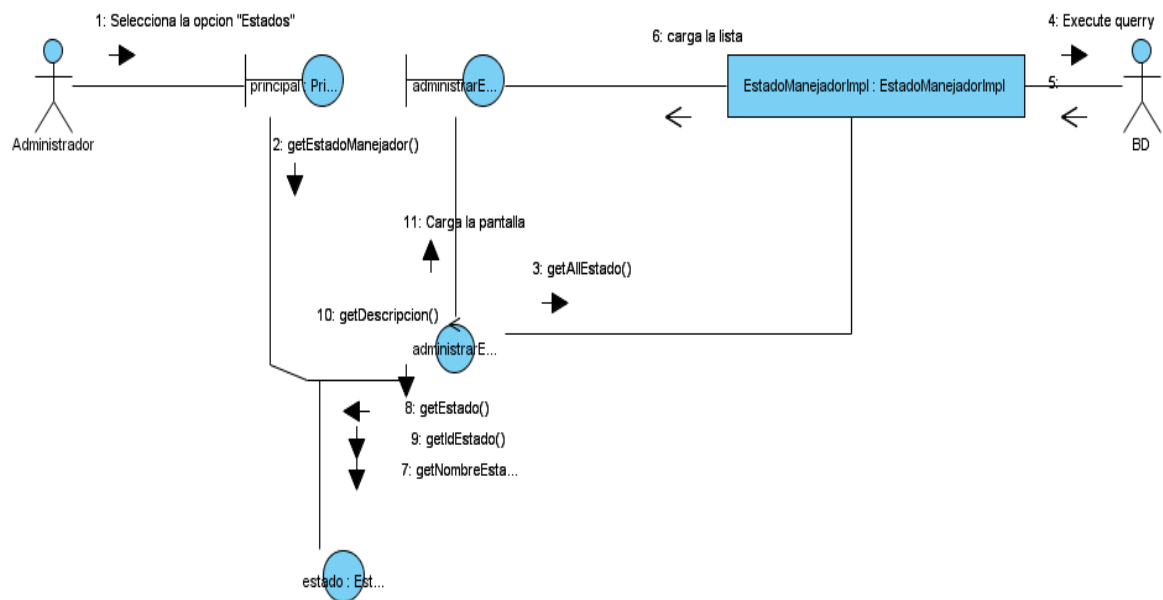


Figura 214: Diagrama de Colaboración-Administrar Estado

Diagrama de colaboración: C.U. Adicionar Estado

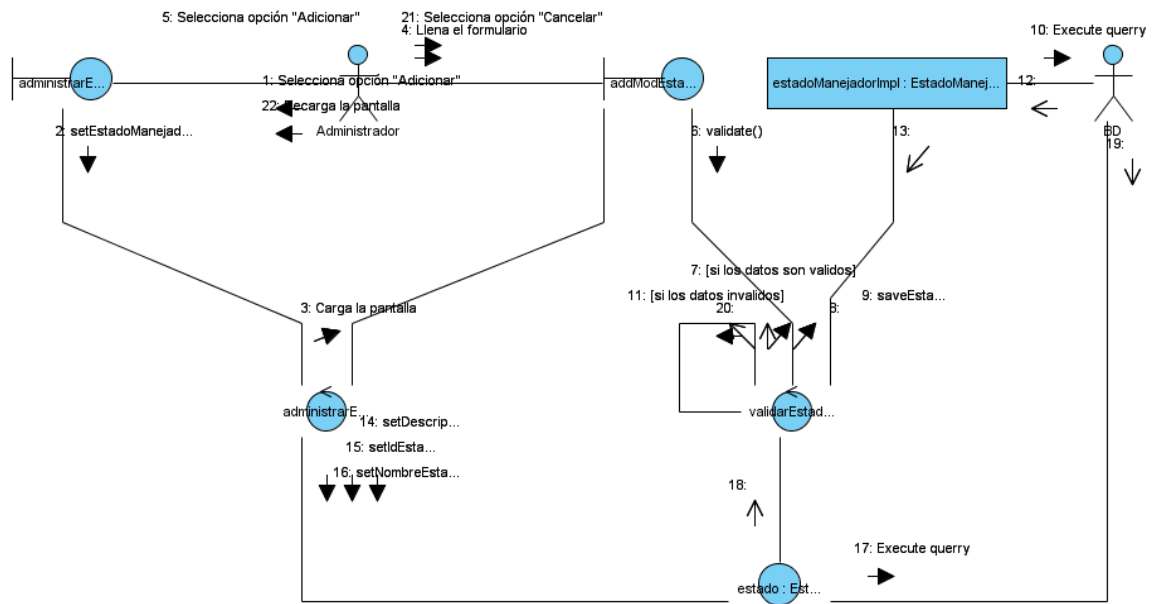


Figura 215: Diagrama de Colaboración-Adicionar Estado

Diagrama de colaboración: C.U. Modificar Estado

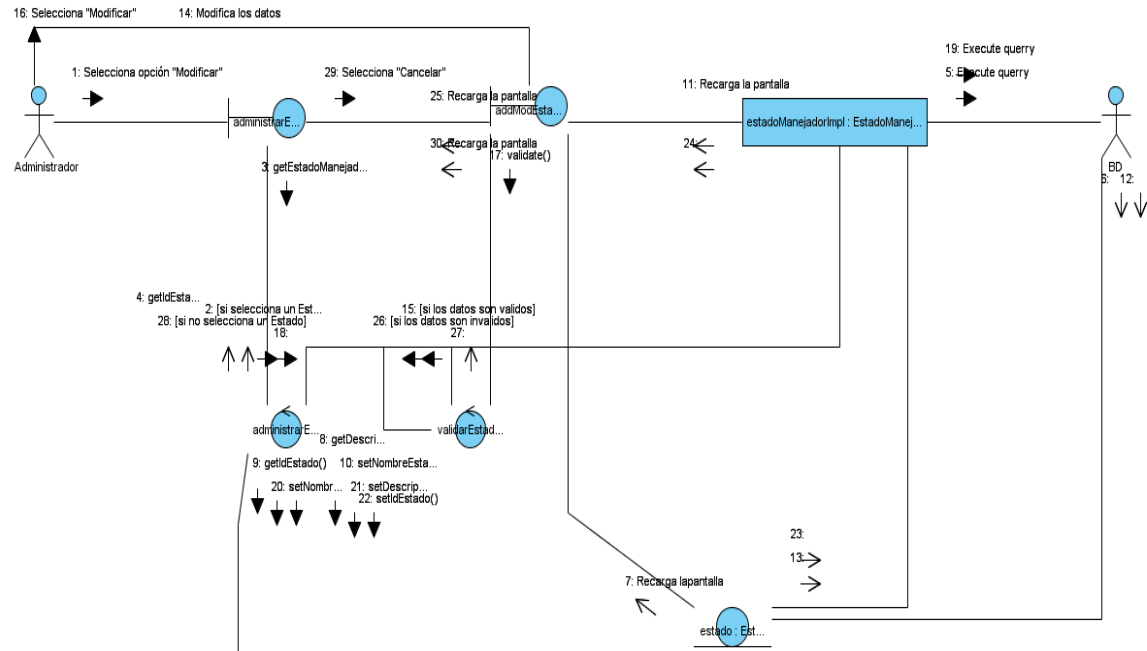


Figura 216: Diagrama de Colaboración-Modificar Estado

Diagrama de colaboración: C.U. Eliminar Estado

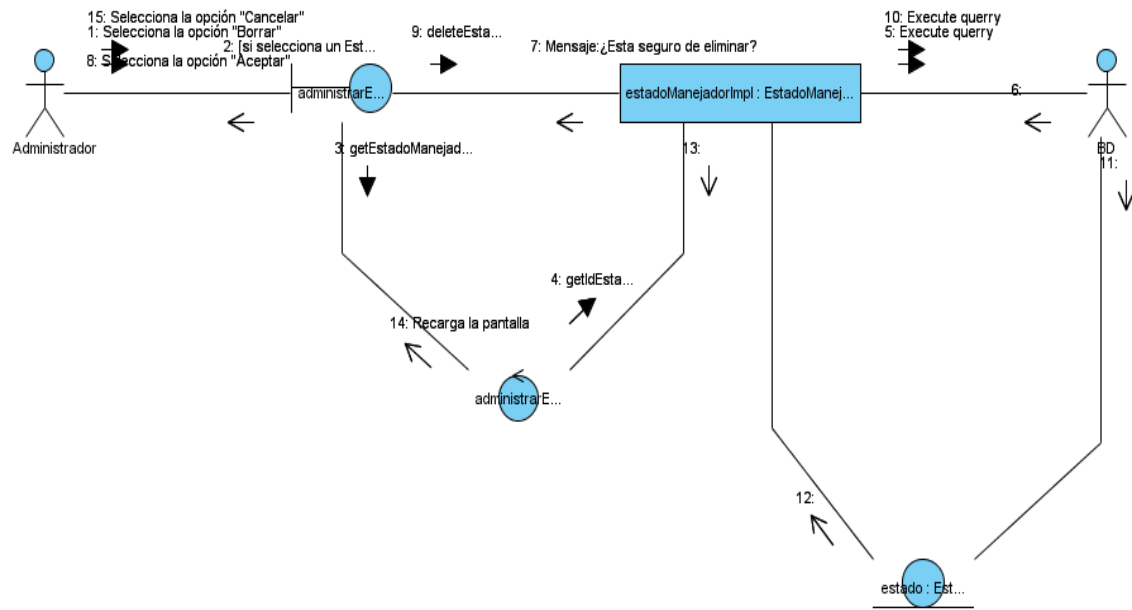


Figura 217: Diagrama de Colaboración-Eliminar Estado

Diagrama de colaboración: C.U. Administrar Productos

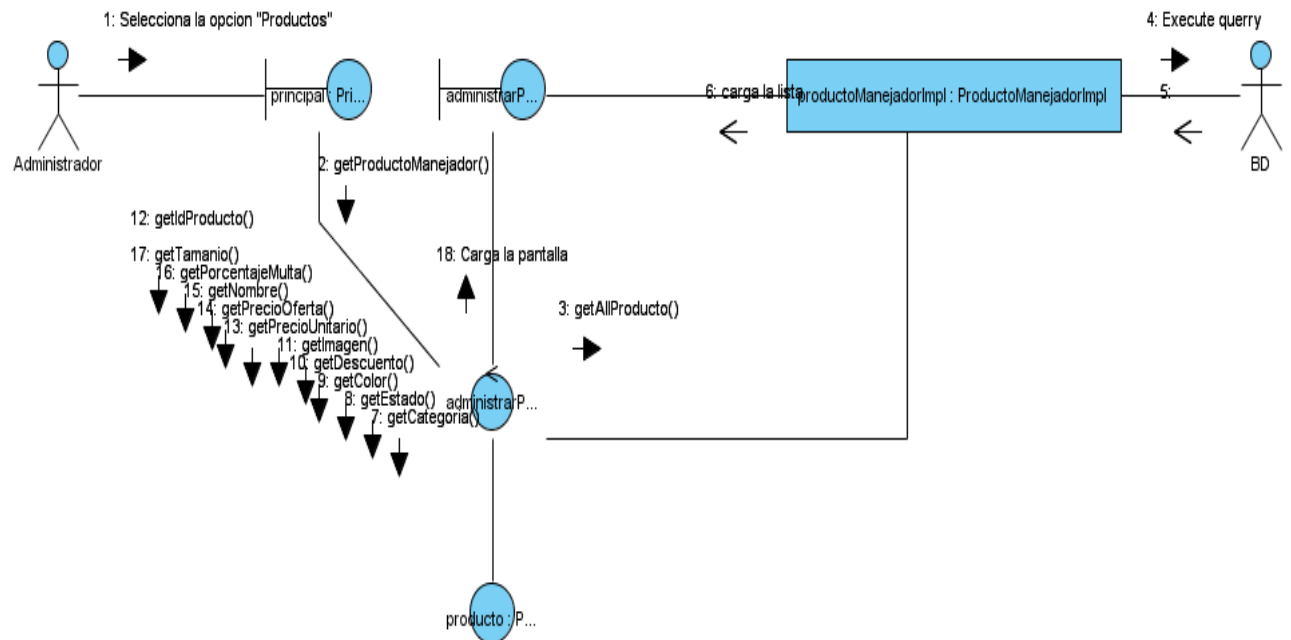


Figura 218: Diagrama de Colaboración-Administrar Productos

Diagrama de colaboración: C.U. Adicionar Producto

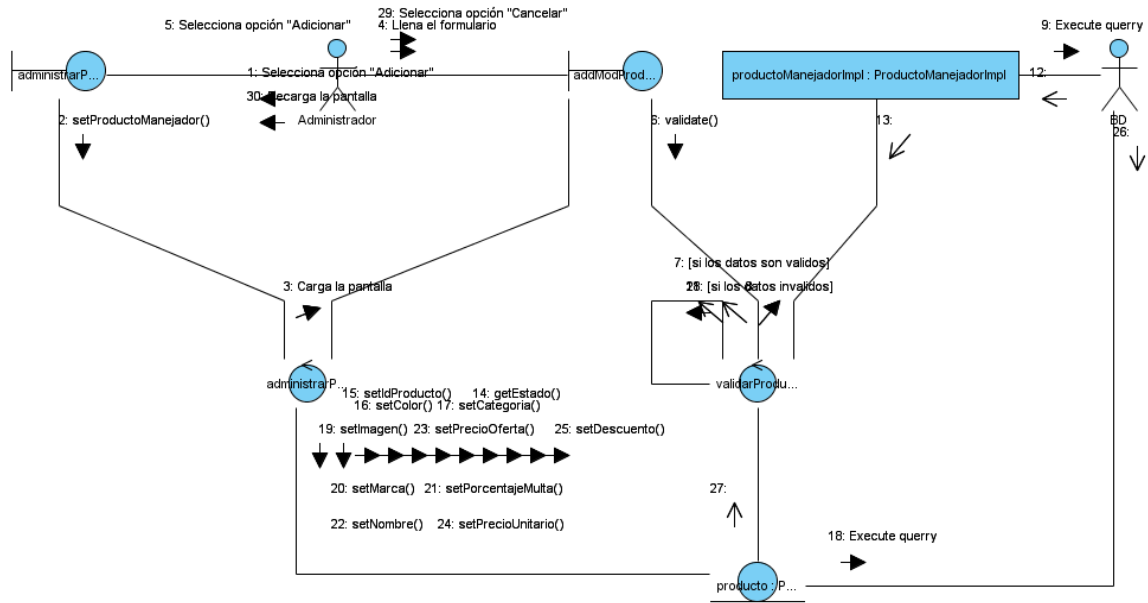


Figura 219: Diagrama de Colaboración-Adicionar Producto

Diagrama de colaboración: C.U. Modificar Producto

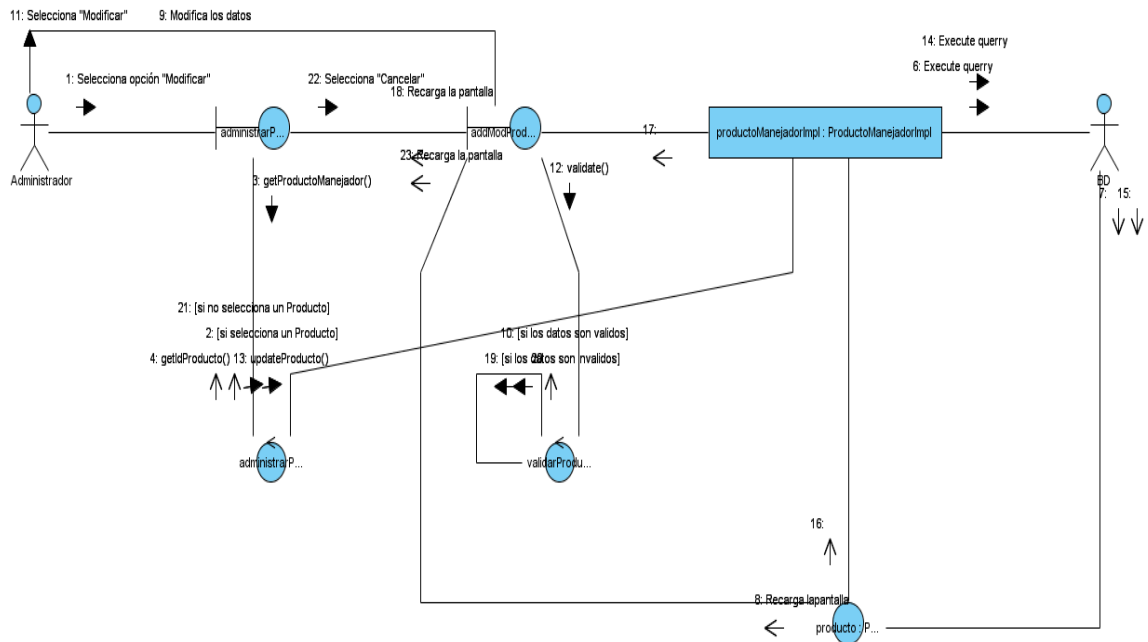


Figura 220: Diagrama de Colaboración-Modificar Productos

Diagrama de colaboración: C.U. Eliminar Producto

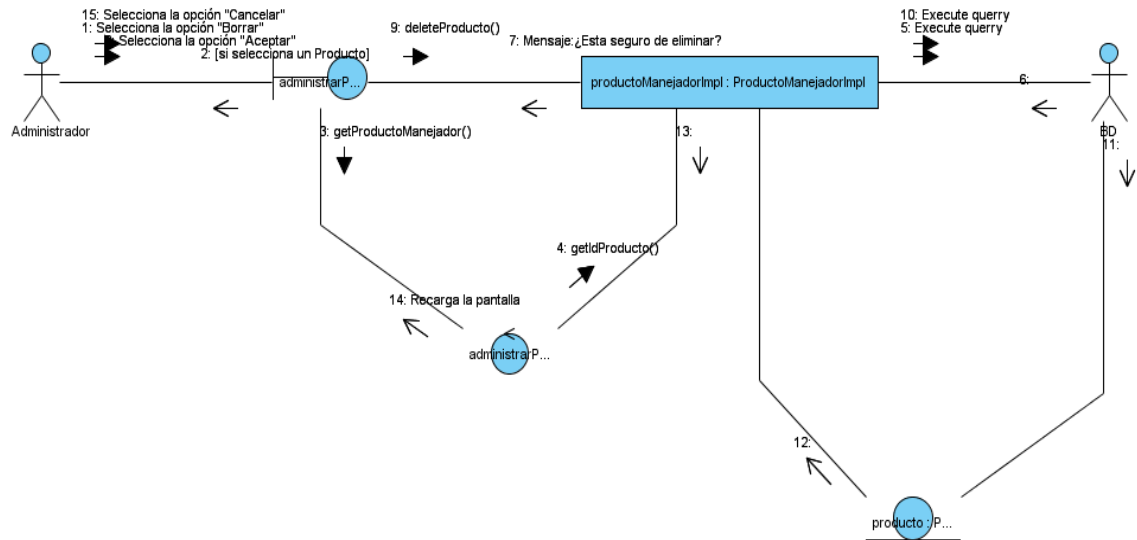


Figura 221: Diagrama de Colaboración-Eliminar Producto

Diagrama de colaboración: C.U. Administrar Proveedores

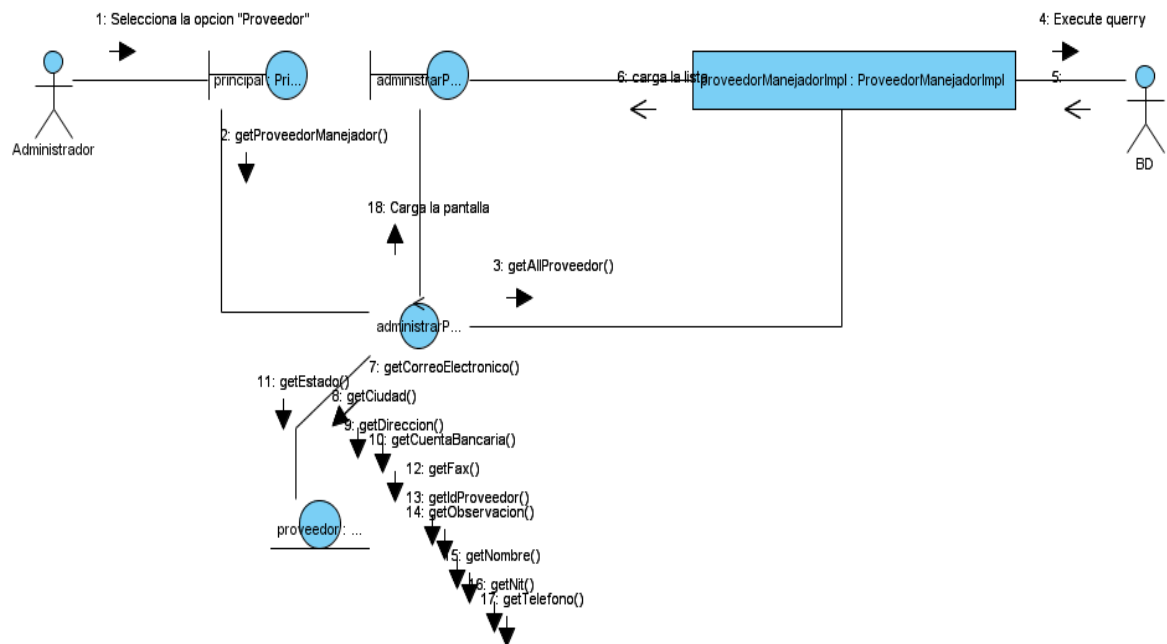


Figura 222: Diagrama de Colaboración-Administrar Proveedores

Diagrama de colaboración: C.U. Adicionar Proveedor

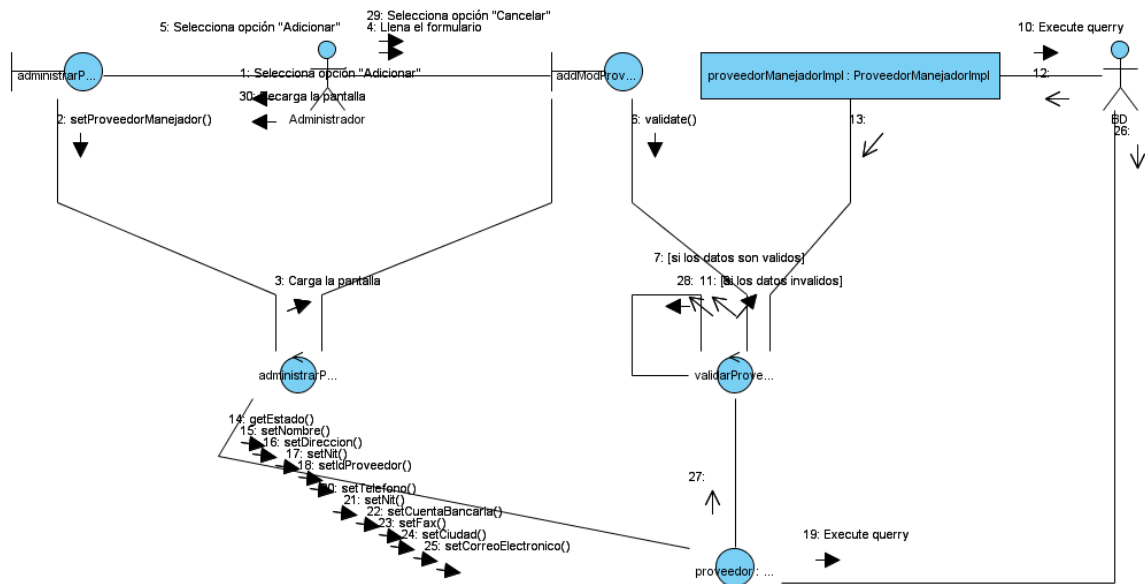


Figura 223: Diagrama de Colaboración-Adicionar Proveedores

Diagrama de colaboración: C.U. Modificar Proveedor

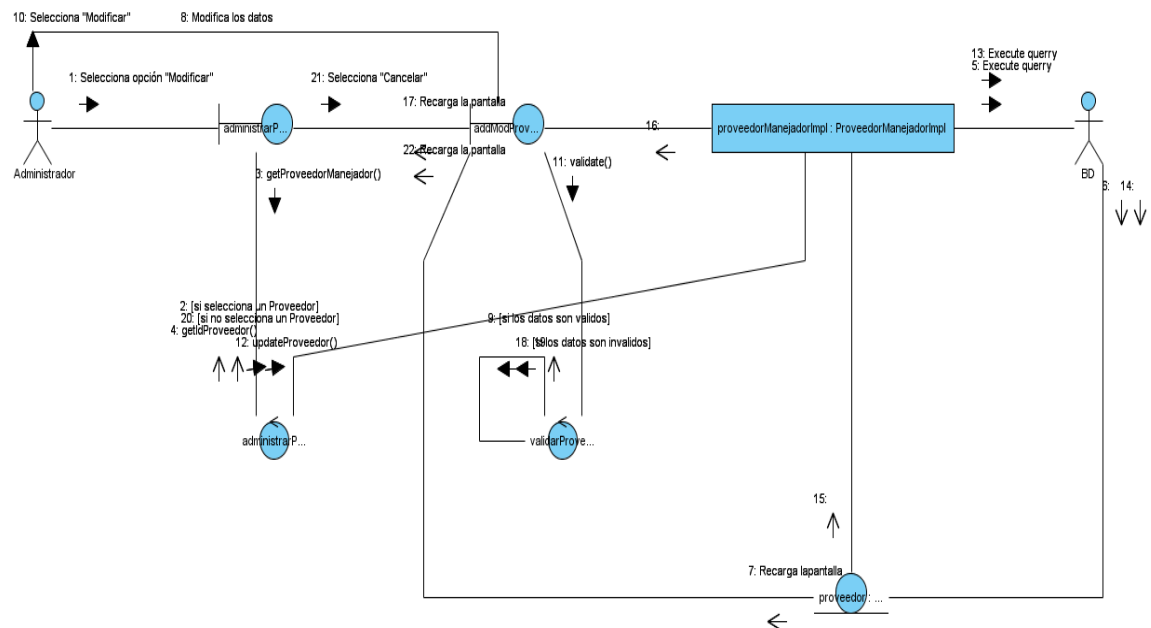


Figura 224: Diagrama de Colaboración-Adicionar Proveedor

Diagrama de colaboración: C.U. Eliminar Proveedor

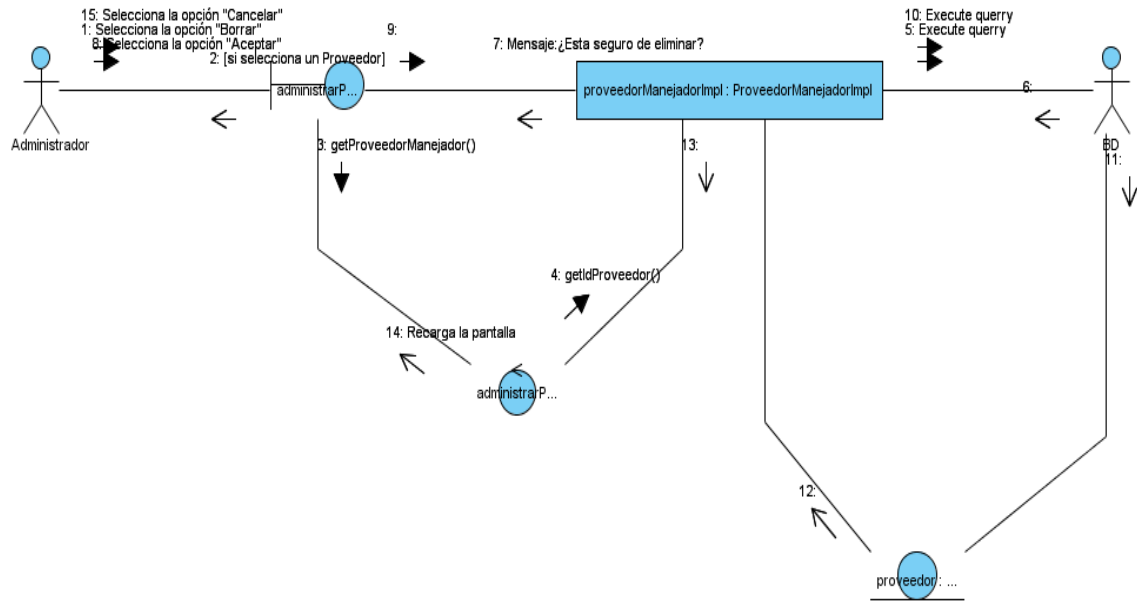


Figura 225: Diagrama de Colaboración-Eliminar Proveedor

Diagrama de colaboración: C.U. Administrar Roles

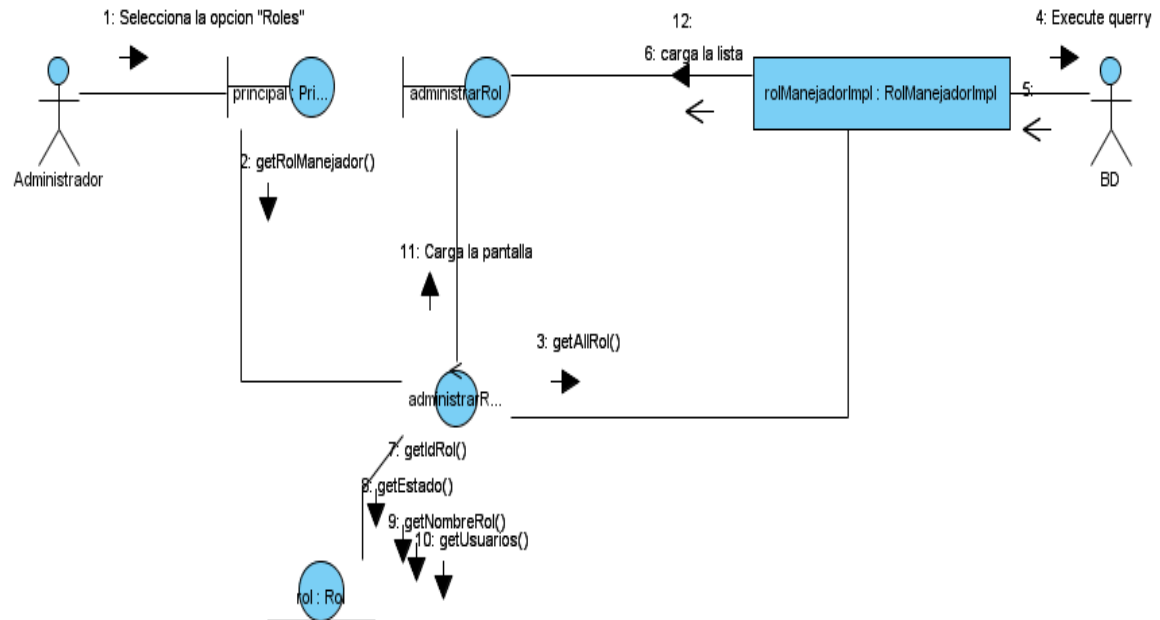


Figura 226: Diagrama de Colaboración-Administrar Roles

Diagrama de colaboración: C.U. Adicionar Rol

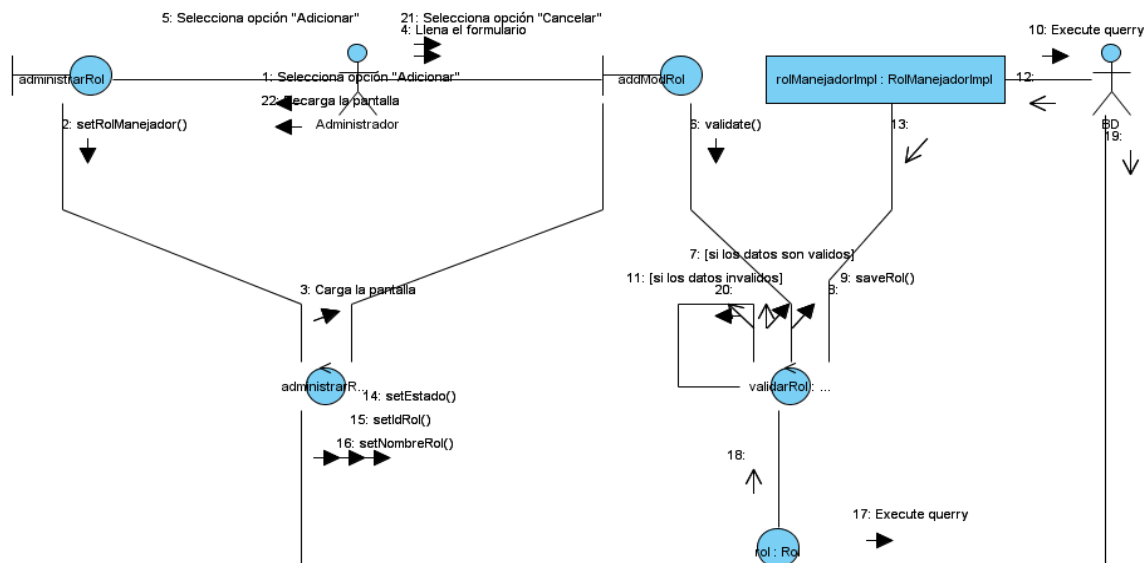


Figura 227: Diagrama de Colaboración-Adicionar Rol

Diagrama de colaboración: C.U. Modificar Rol

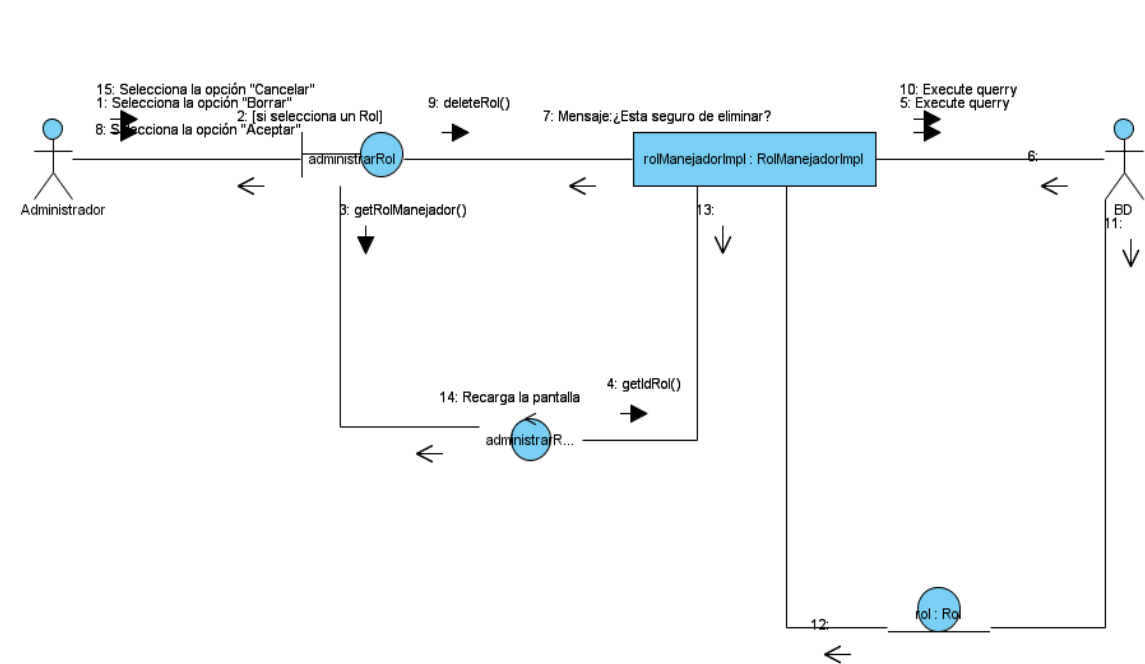


Figura 228: Diagrama de Colaboración-Administrar Rol

Diagrama de colaboración: C.U. Eliminar Rol

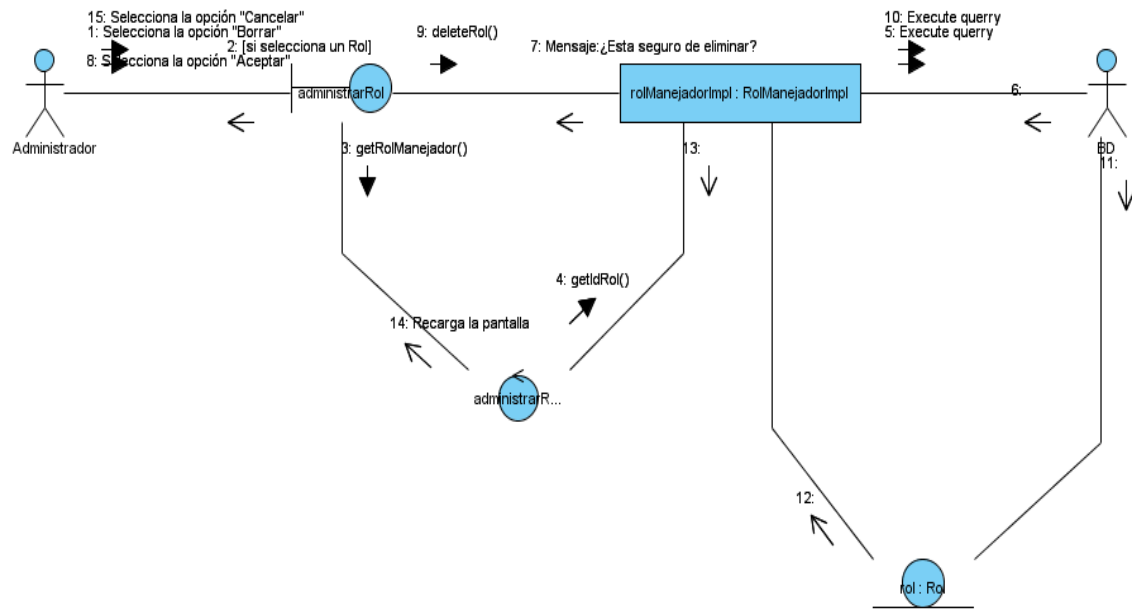


Figura 229: Diagrama de Colaboración-Eliminar Rol

Diagrama de colaboración: C.U. Administrar Clientes

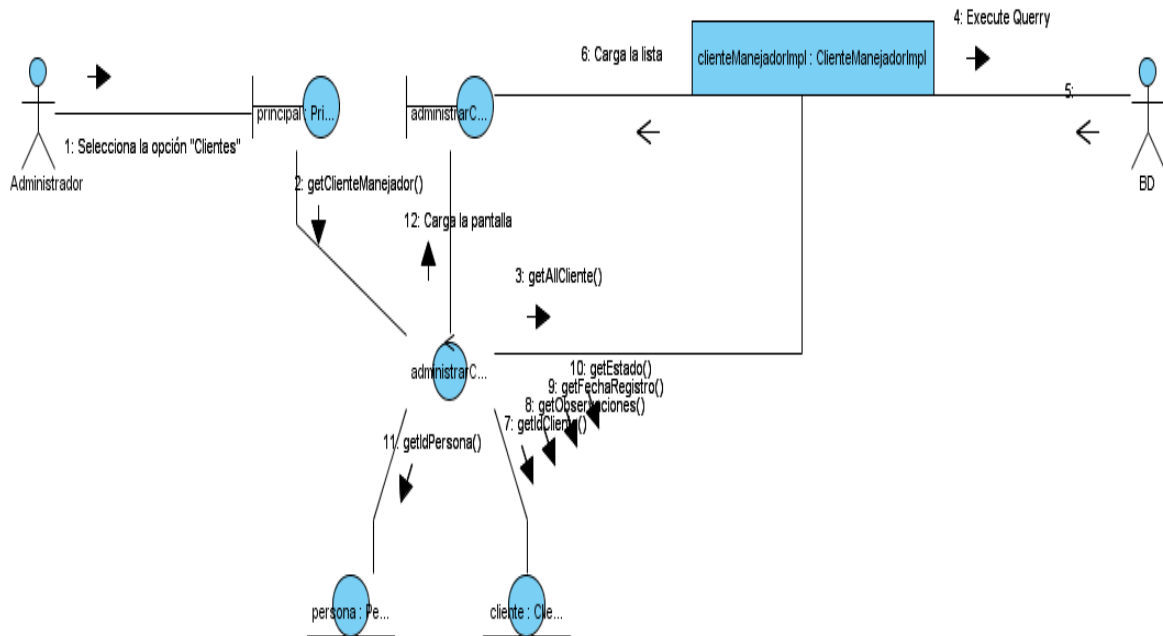


Figura 230: Diagrama de Colaboración-Administrar Clientes

Diagrama de colaboración: C.U. Agregar Cliente

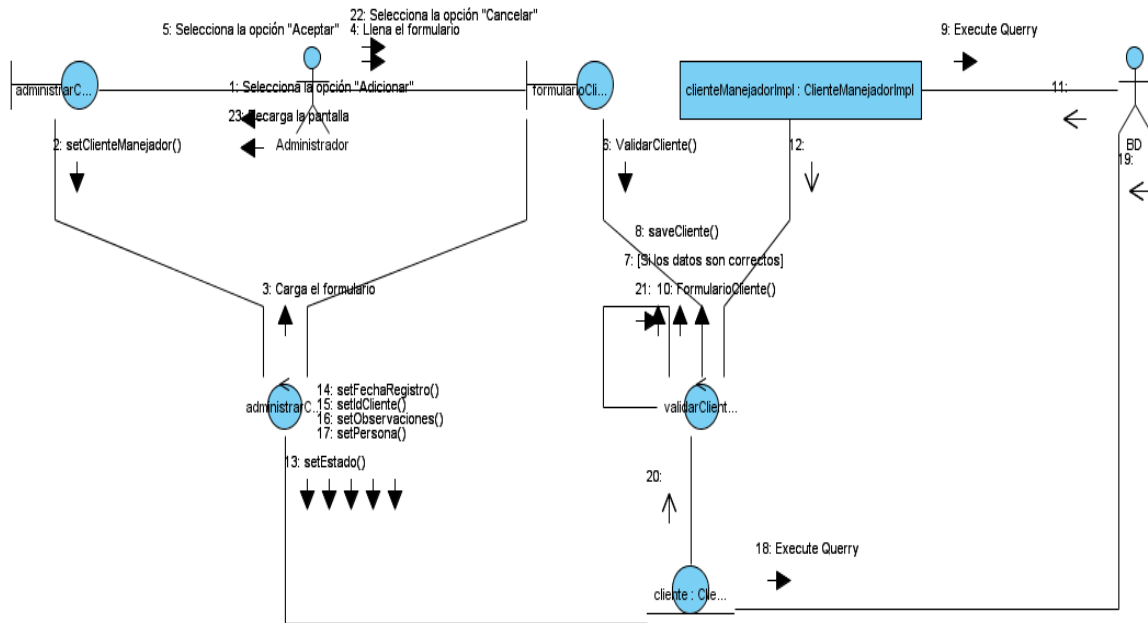


Figura 231: Diagrama de Colaboración-Adicionar Cliente

Diagrama de colaboración: C.U. Modificar Cliente

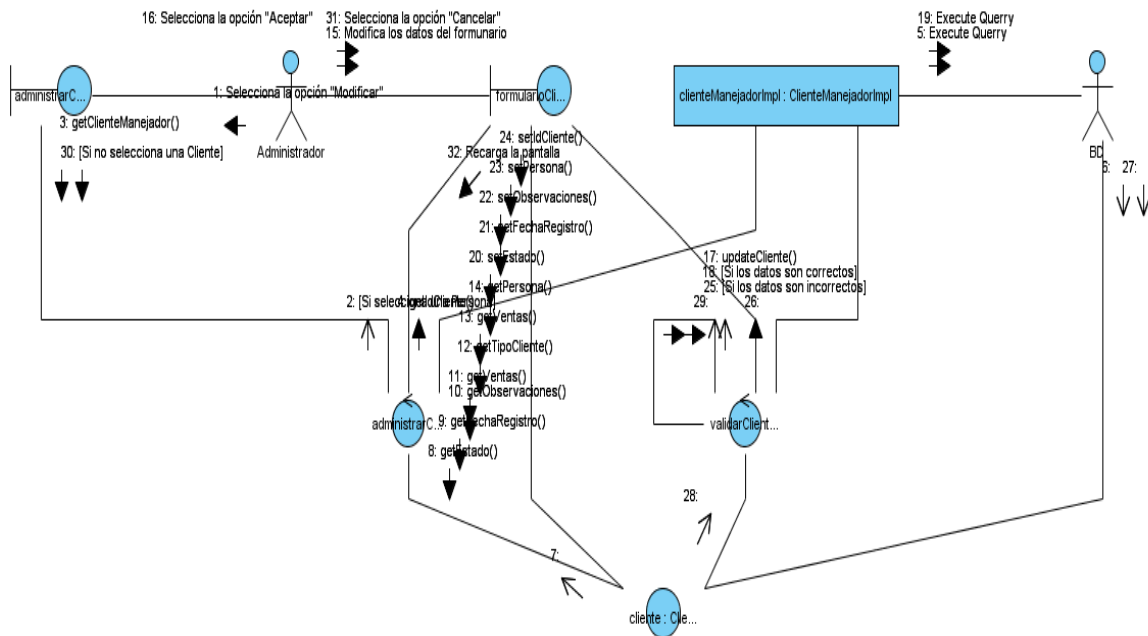


Figura 232: Diagrama de Colaboración-Modificar Cliente

Diagrama de colaboración: C.U. Eliminar Cliente

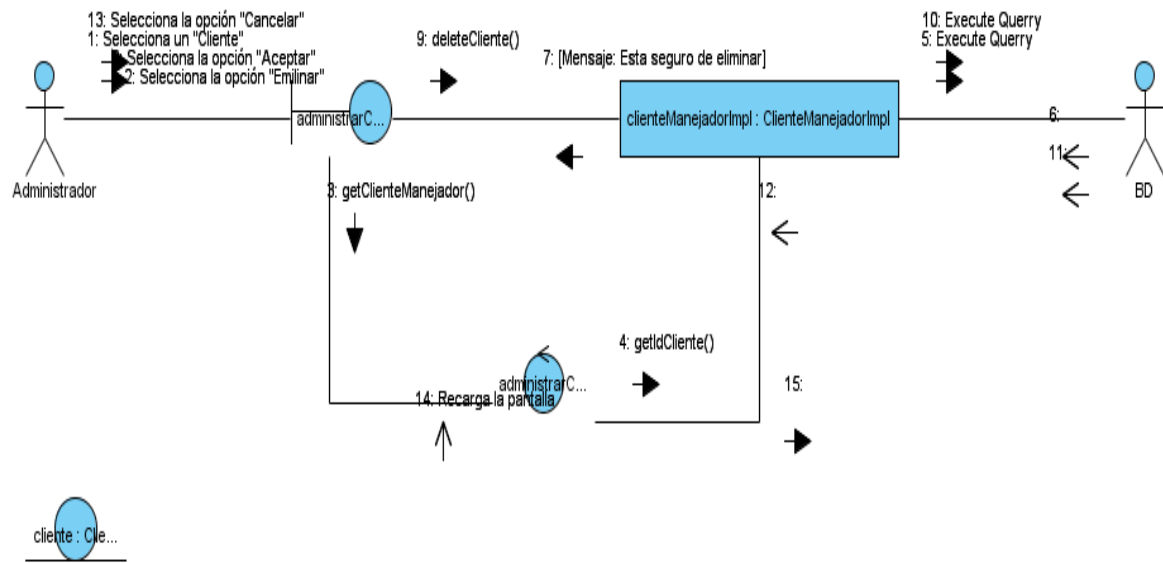


Figura 233: Diagrama de Colaboración-Eliminar Cliente

Diagrama de colaboración: C.U. Administrar Venta

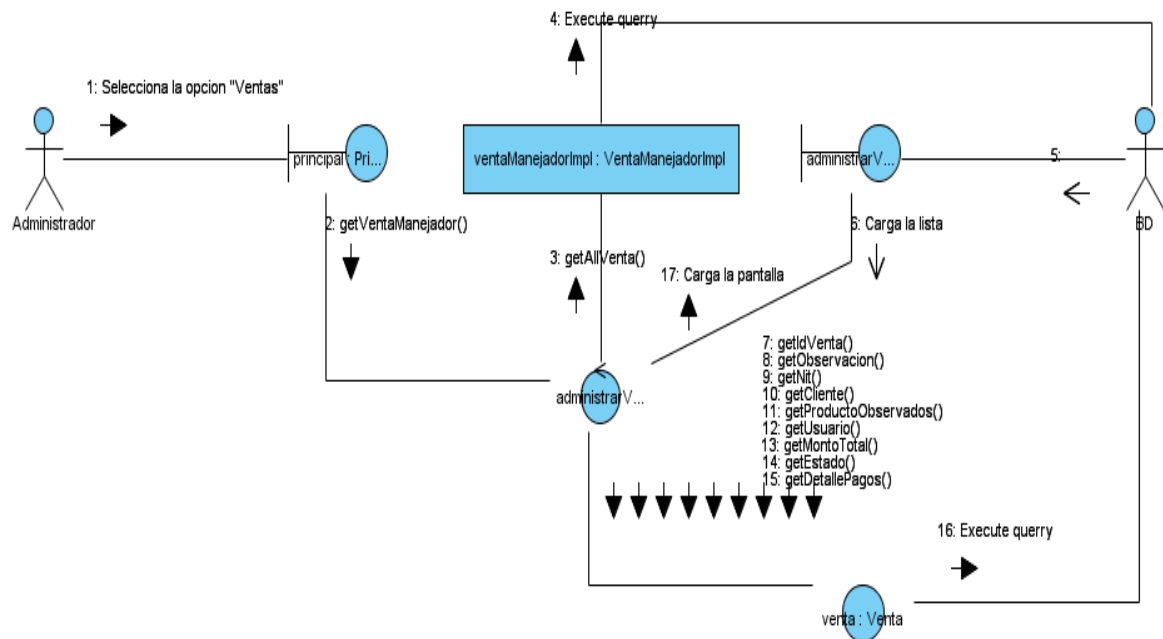


Figura 234: Diagrama de Colaboración-Administrar Ventas

Diagrama de colaboración: C.U. Adicionar Venta

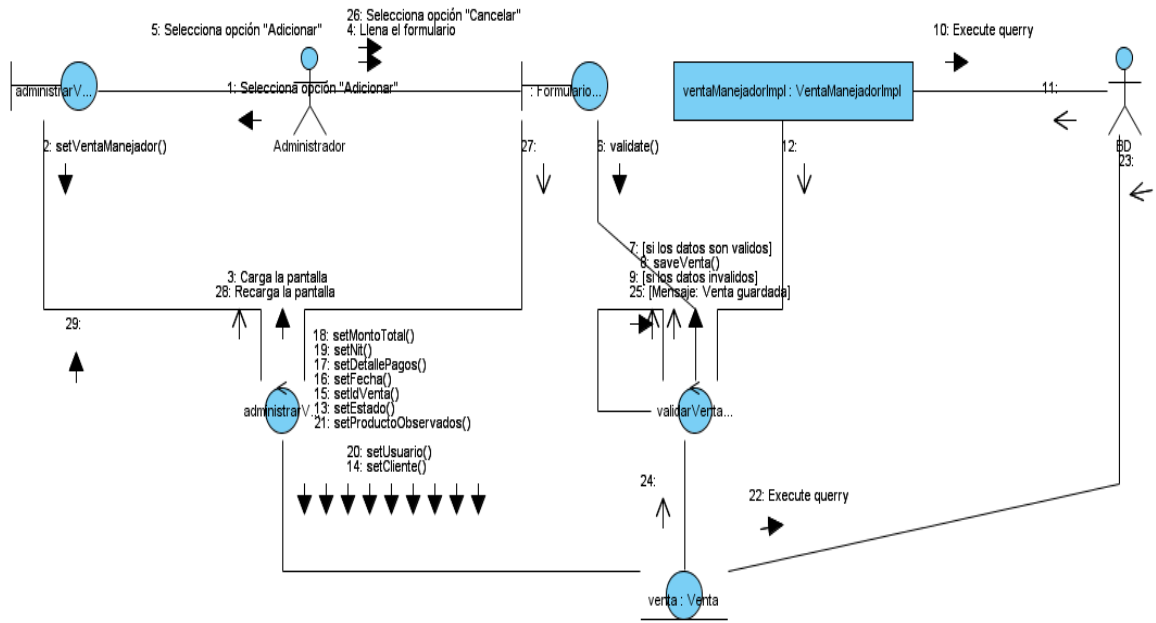


Figura 235: Diagrama de Colaboración-Administrar Ventas

Diagrama de Colaboración: C.U. Administrar Productos Observados

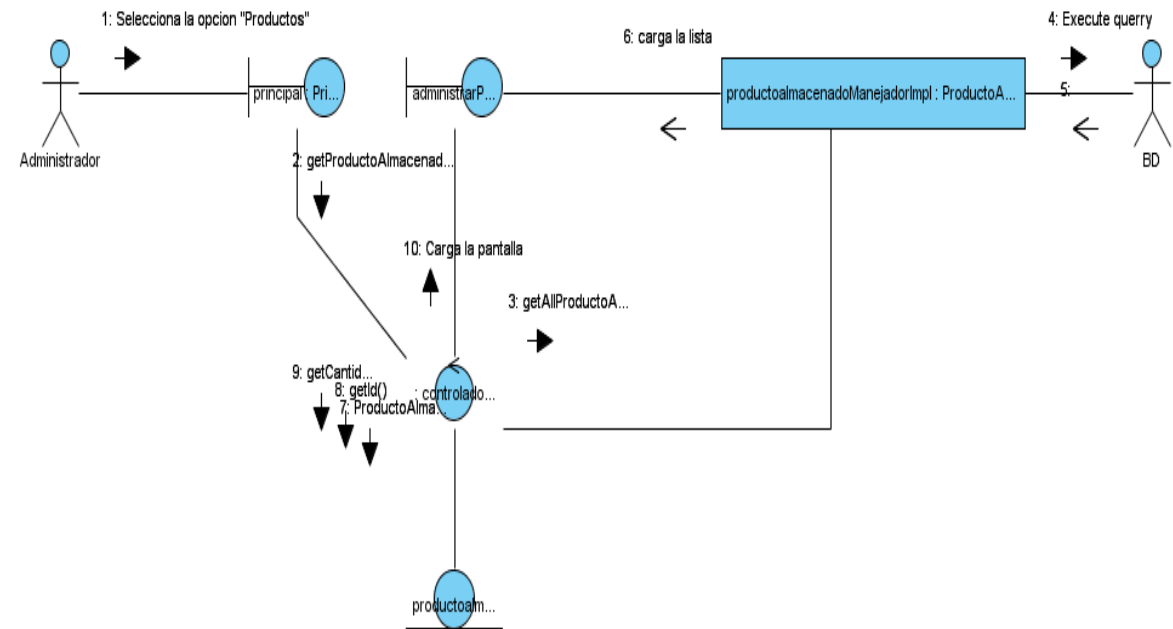


Figura 236: Diagrama de Colaboración-Administrar Productos Observados

Diagrama de Colaboración: C.U. Adicionar Producto Observado

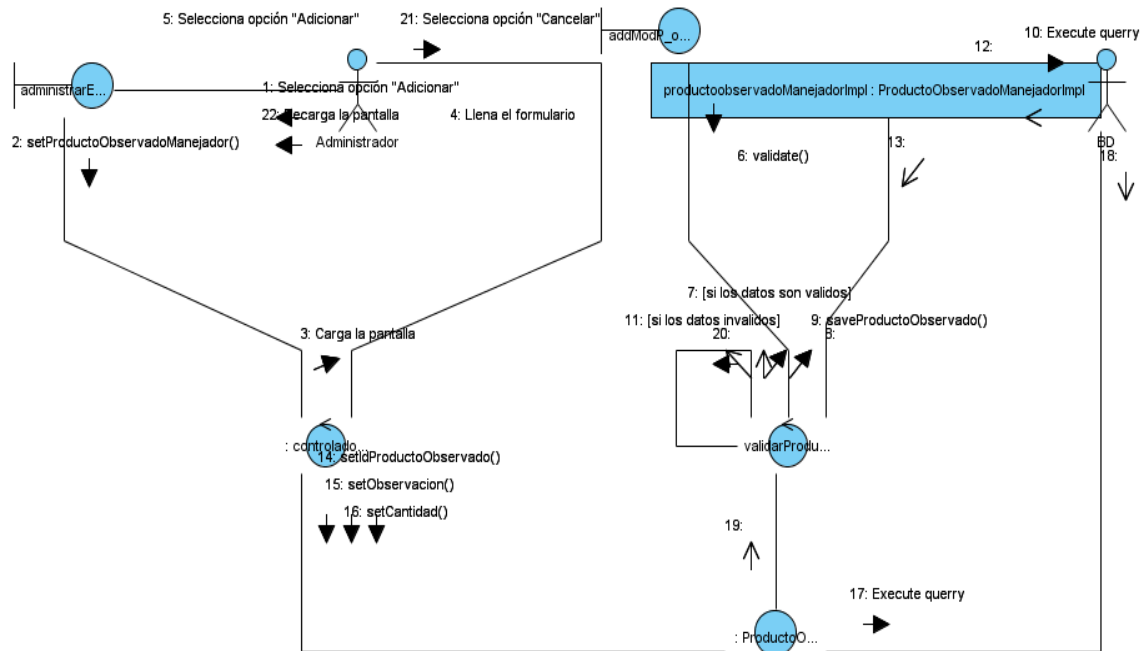


Figura 237: Diagrama de Colaboración-Adicionar Producto Observado

Diagrama de Colaboración: C.U. Modificar Producto Observado

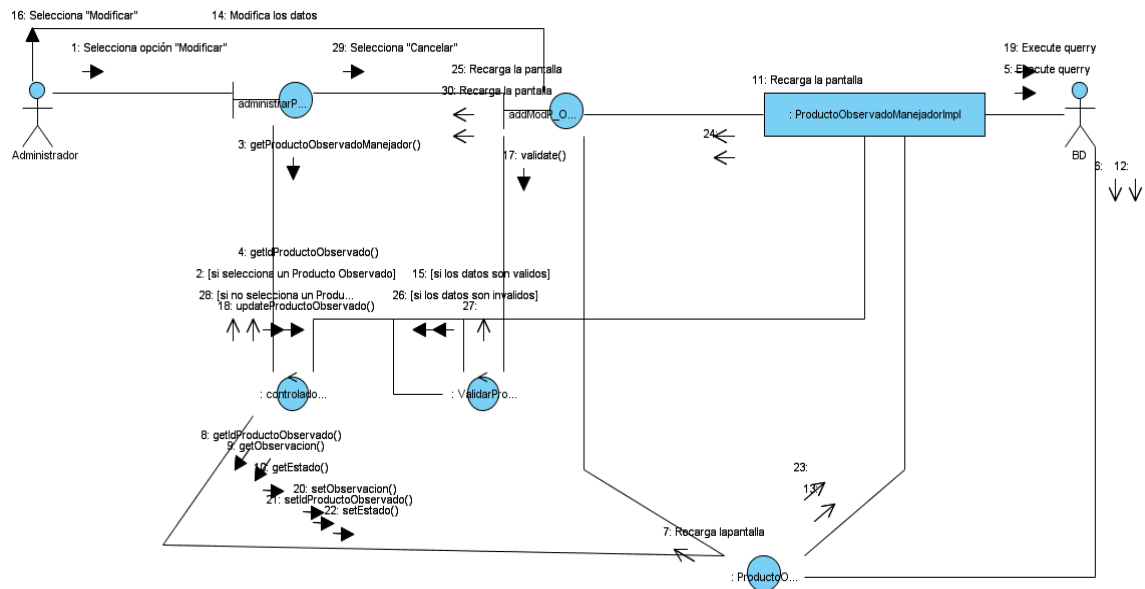


Figura 238: Diagrama de Colaboración-Modificar Producto Observado

Diagrama de Colaboración: C.U. Eliminar Producto Observado

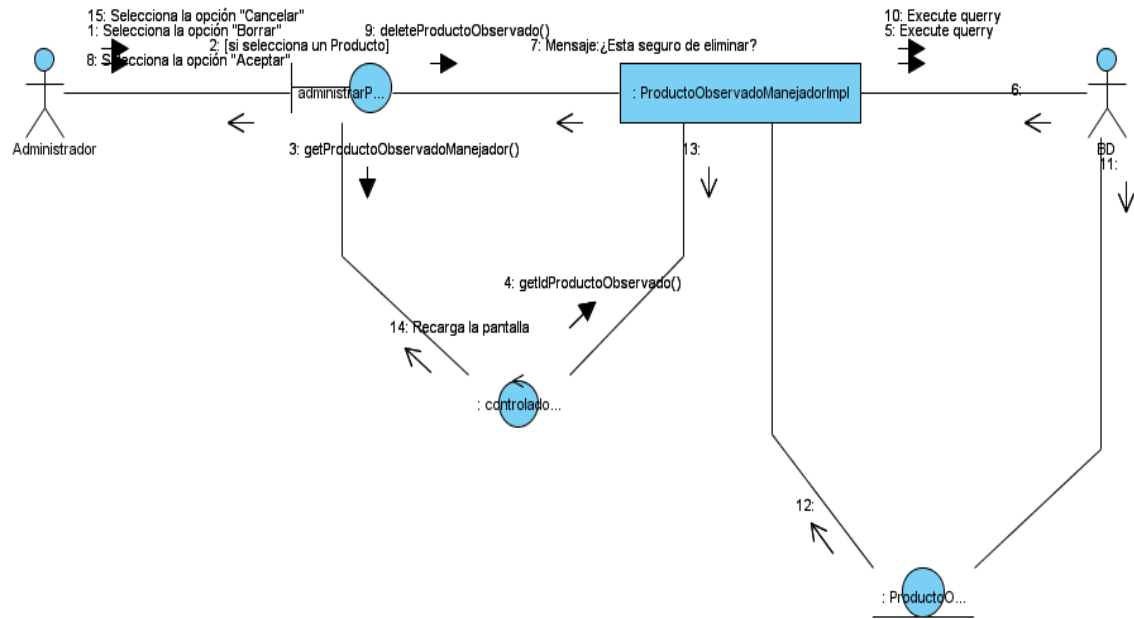


Figura 239: Diagrama de Colaboración-Eliminar Producto Observado

Diagrama de Colaboración: C.U. Administrar Tipo de Clientes

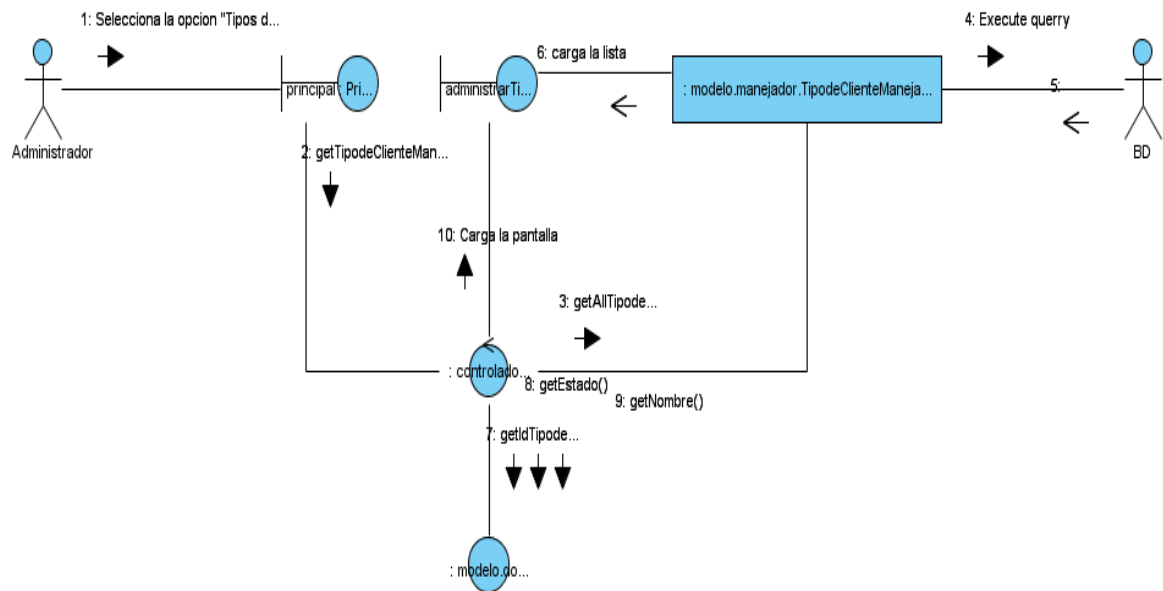


Figura 240: Diagrama de Colaboración-Administrar Tipos de Clientes

Diagrama de Colaboración: C.U. Agregar Tipo de Cliente

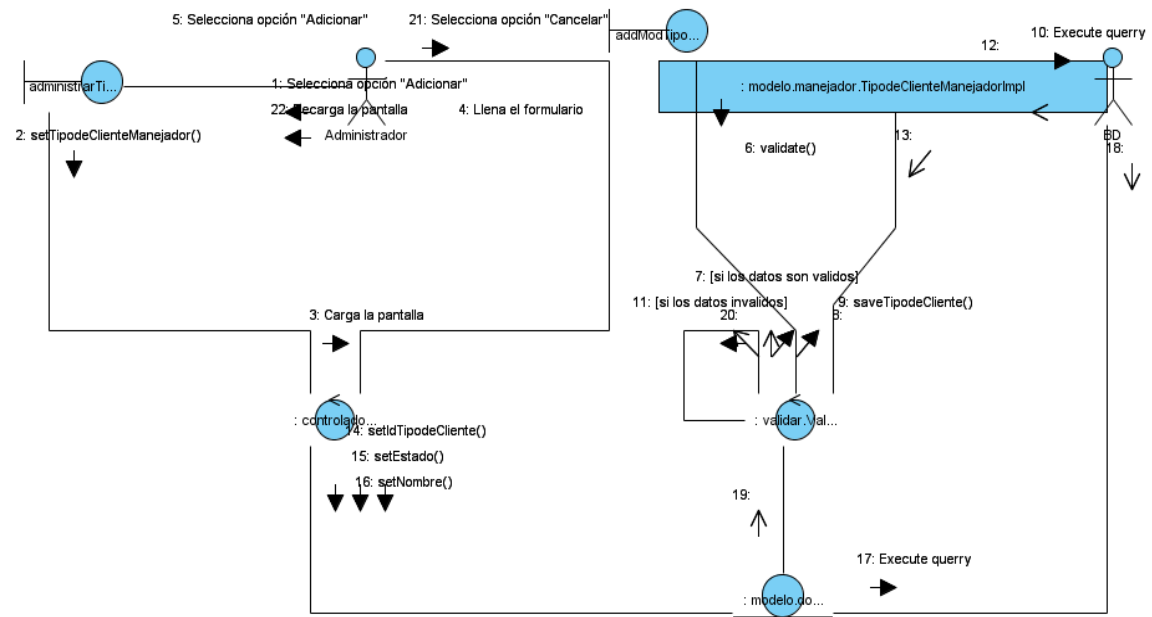


Figura 241: Diagrama de Colaboración-Adicionar Tipo de Cliente

Diagrama de Colaboración: C.U. Modificar Tipo de Cliente

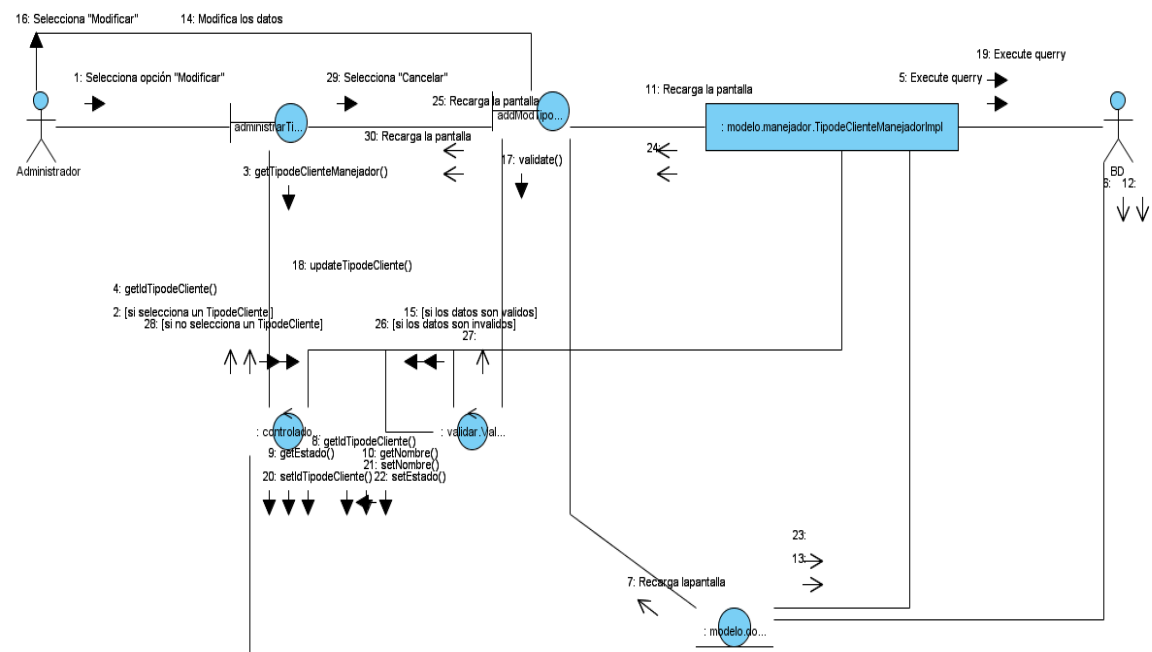


Figura 242: Diagrama de Colaboración-Modificar Tipo de Cliente

Diagrama de Colaboración: C.U. Eliminar Tipo de Cliente

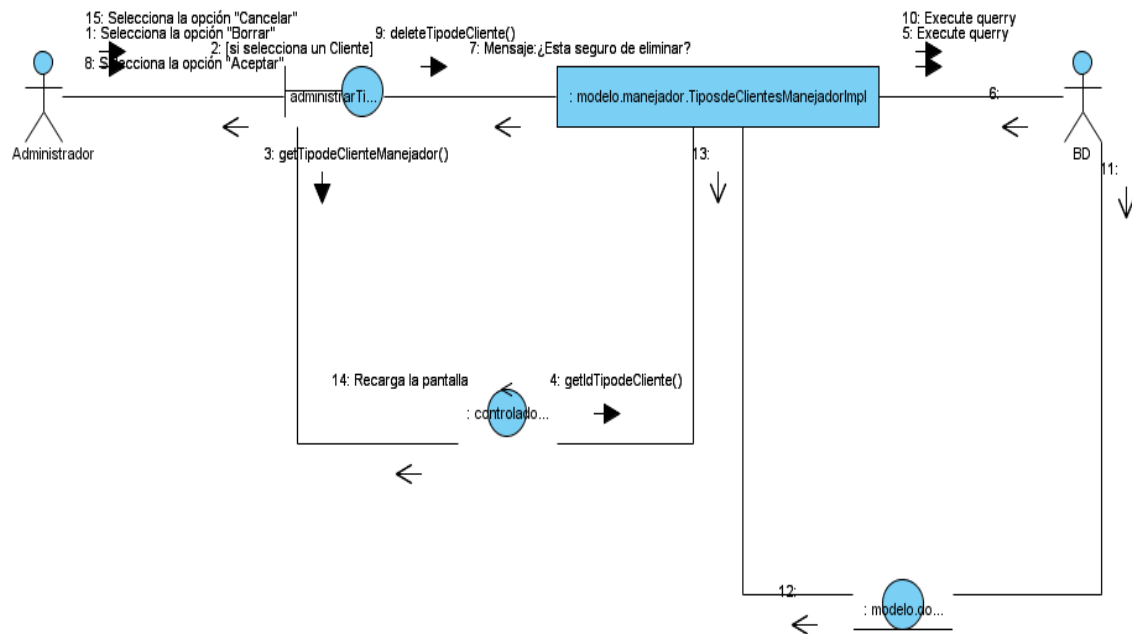


Figura 243: Diagrama de Colaboración-Eliminar Tipo de Cliente

Diagrama de Colaboración: C.U. Administrar Tipo de Ventas

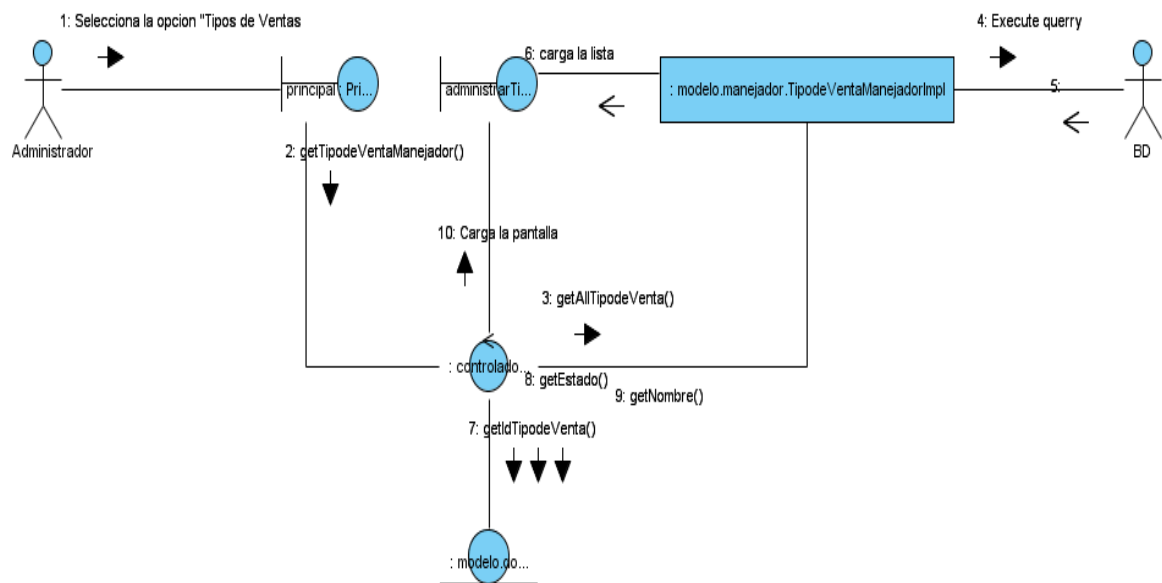


Figura 244: Diagrama de Colaboración-Administrar Tipos de Ventas

Diagrama de Colaboración: C.U. Adicionar Tipo de Venta

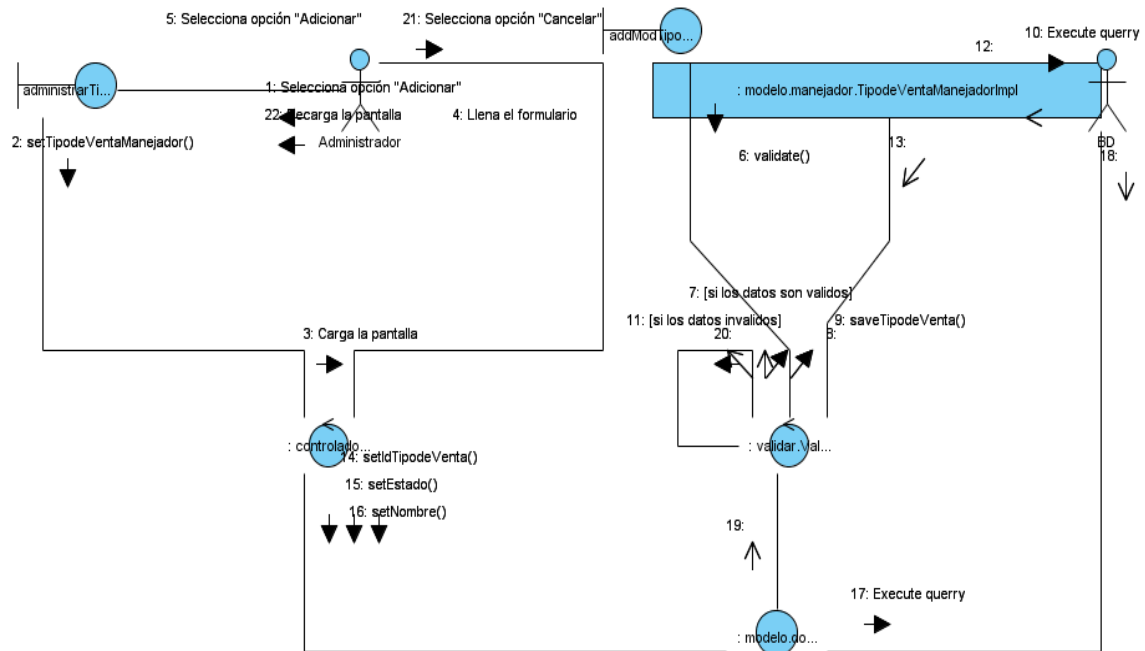


Figura 245: Diagrama de Colaboración-Adicionar Tipos de Venta

Diagrama de secuencia: C.U. Modificar Tipo de Venta

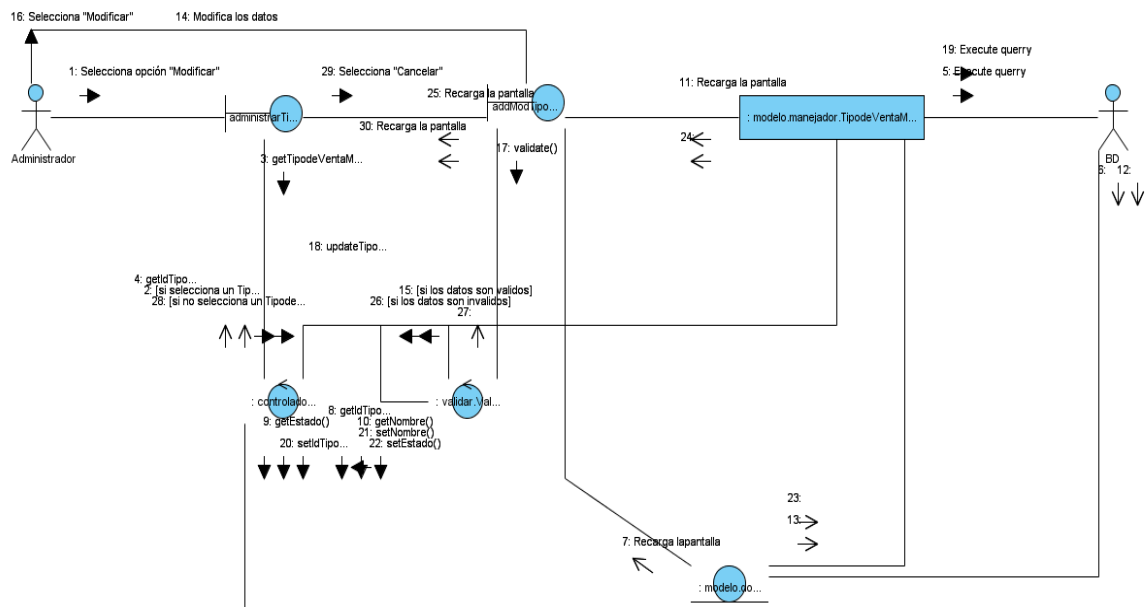


Figura 246: Diagrama de Colaboración-Modificar Tipo de Ventas

Diagrama de secuencia: C.U. Eliminar Tipo de Venta

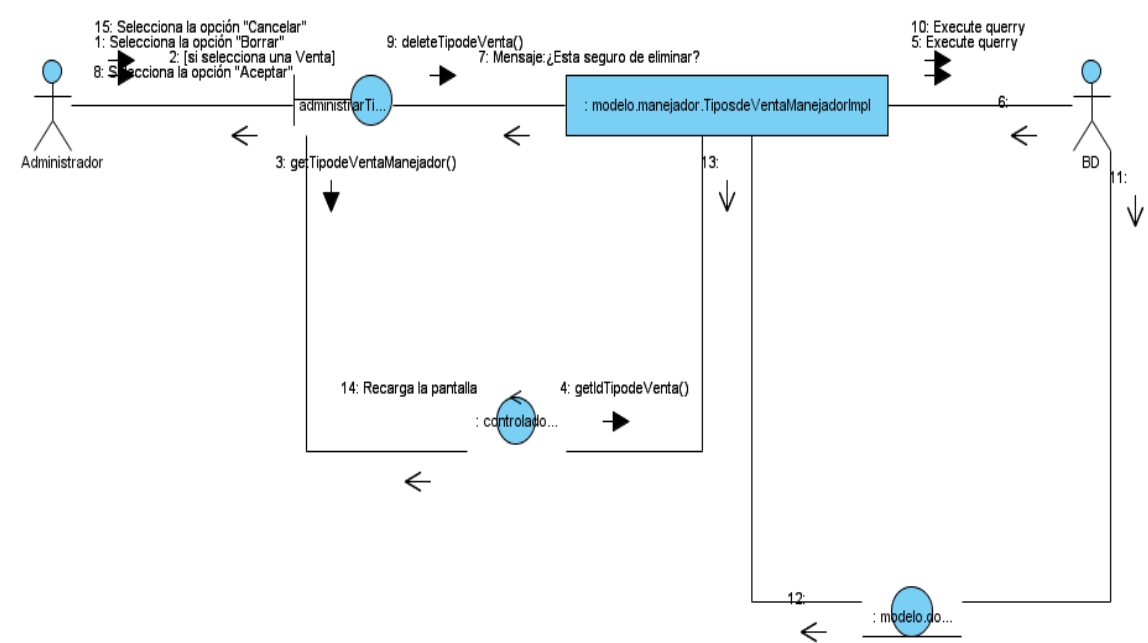


Figura 247: Diagrama de Colaboración-Eliminar Tipo de Venta

Diagrama de colaboración: C.U. Generar Reporte

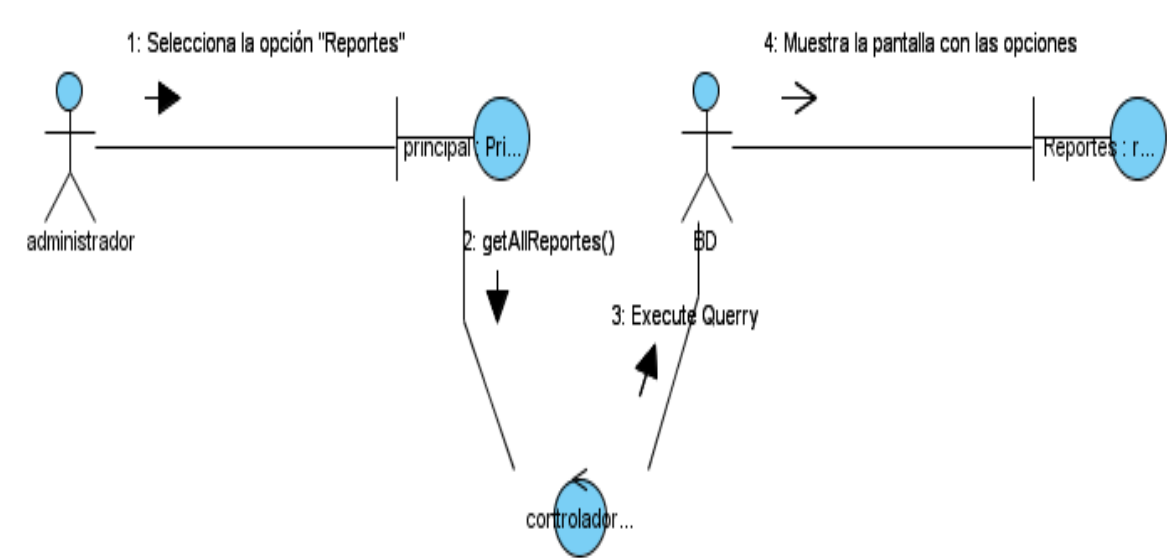


Figura 248: Diagrama de Colaboración-Generar Reporte

Diagrama de colaboración: C.U. Reporte Producto Almacenado

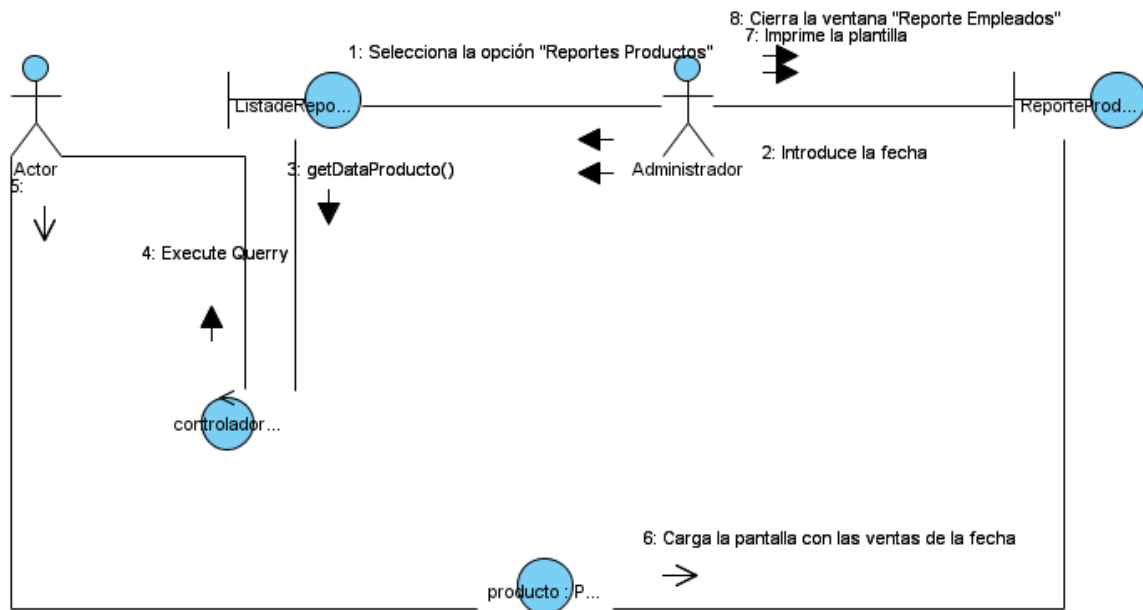


Figura 249: Diagrama de Colaboración-Reporte Producto Almacenado

Diagrama de colaboración: C.U. Reporte Venta

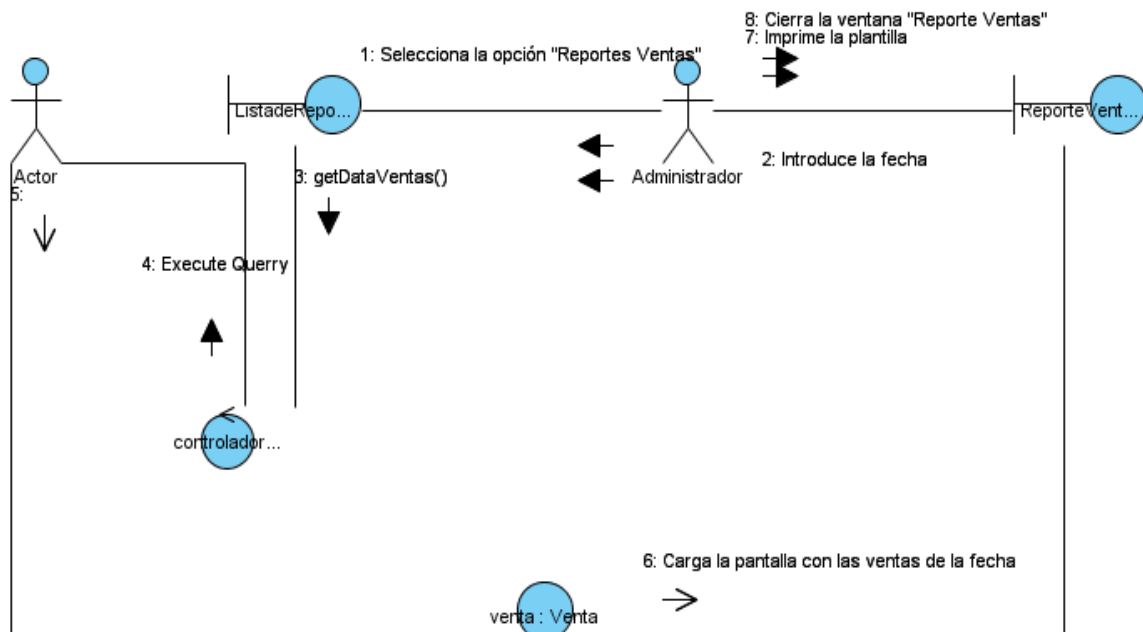


Figura 250: Diagrama de Colaboración-Reporte Ventas

Diagrama de colaboración: C.U. Reporte cantidad de Compra Clientes

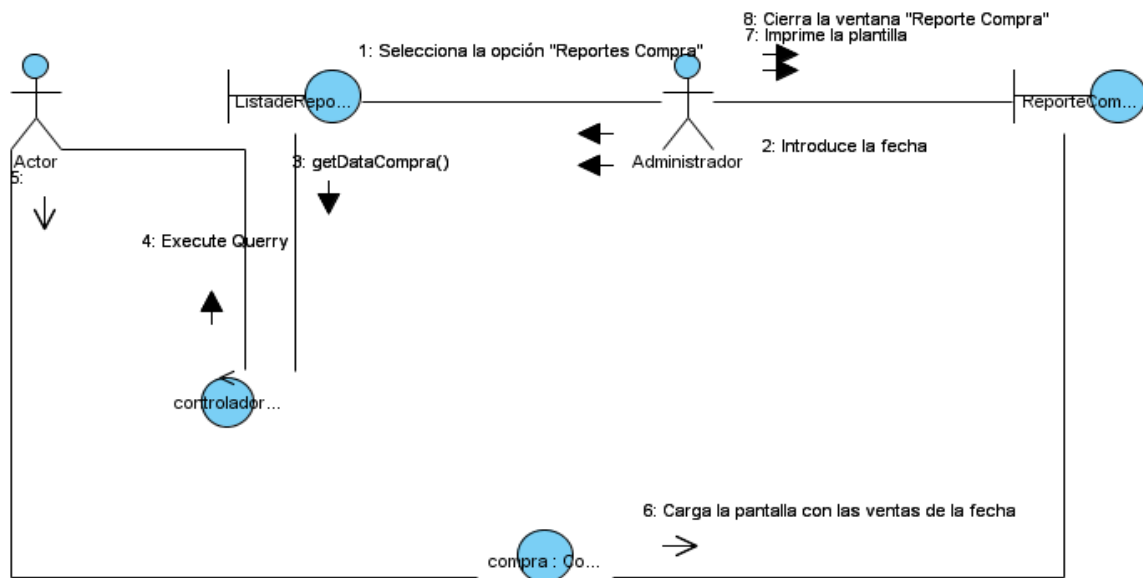


Figura 251: Diagrama de Colaboración-Reporte cantidad de Compras Clientes

2.1.2.2.3.2 MODELO DE DATOS

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

2.1.2.2.3.2.1 DIAGRAMA DE CLASES

2.1.2.2.3.2.2 DIAGRAMA DE LA BASE DE DATOS

2.1.2.2.3.2.3 SCRIPT DE LA BASE DE DATOS

```
CREATE SEQUENCE public.almacen_id_almacen_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.cargo_id_cargo_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.cliente_id_cliente_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.compra_id_compra_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.d_compra_id_d_compra_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.d_venta_id_d_venta_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.detalle_pago_id_detalle_pago_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.empleado_id_empleado_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.producto_id_producto_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.producto_observado_id_producto_observado_seq  
INCREMENT 1 MINVALUE 1 MAXVALUE 9223372036854775807 START 0  
CACHE 1;
```

```
CREATE SEQUENCE public.proveedor_id_proveedor_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.roles_id_rol_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.tipo_cliente_id_tipo_cliente_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.tipo_venta_id_tipo_venta_seq INCREMENT 1  
MINVALUE 1 MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.usuario_id_usuario_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE SEQUENCE public.venta_id_venta_seq INCREMENT 1 MINVALUE 1  
MAXVALUE 9223372036854775807 START 0 CACHE 1;
```

```
CREATE TABLE public.estado (  
  id_estado CHARACTER VARYING(3) NOT NULL,  
  nombre_estado CHARACTER VARYING(50) NOT NULL,  
  descripcion TEXT NOT NULL,  
  estado CHARACTER VARYING(3),  
  tipo CHARACTER,  
  CONSTRAINT identificador_de_estado PRIMARY KEY (id_estado)  
);
```

```
COMMENT ON COLUMN public.estado.id_estado IS 'el identificador del estado';
```

```
COMMENT ON COLUMN public.estado.nombre_estado IS 'el nombre del estado';
```

```
COMMENT ON COLUMN public.estado.descripcion IS 'la descripcion posible del  
estado';
```

```
COMMENT ON COLUMN public.estado.estado IS 'el estado del estado (avtivo,  
inactivo)';
```

```
/* ----- */  
/* Add table "public.proveedor" */  
/* ----- */
```

```
CREATE TABLE public.proveedor (  
  id_proveedor INTEGER DEFAULT nextval('proveedor_id_proveedor_seq')  
NOT NULL,  
  nit INTEGER,  
  nombre CHARACTER VARYING(50) NOT NULL,  
  direccion CHARACTER VARYING(100) NOT NULL,  
  correo_electronico CHARACTER VARYING(50) NOT NULL,  
  telefono INTEGER,  
  fax INTEGER,  
  ciudad CHARACTER VARYING(15) NOT NULL,  
  cuenta_bancaria INTEGER NOT NULL,  
  estado CHARACTER VARYING(3) DEFAULT 'A' NOT NULL,
```


observacion TEXT,
CONSTRAINT identificador_de_proveedor PRIMARY KEY (id_proveedor)
);

COMMENT ON COLUMN public.proveedor.id_proveedor IS 'el identificador del producto';

COMMENT ON COLUMN public.proveedor.nit IS 'el número de la identificacion tributaria del proveedor';

COMMENT ON COLUMN public.proveedor.nombre IS 'el nombre de la empresa o del proveedor';

COMMENT ON COLUMN public.proveedor.direccion IS 'la direccion en la que se encuentra el proveedor';

COMMENT ON COLUMN public.proveedor.correo_electronico IS 'el correo electronico del proveedor';

COMMENT ON COLUMN public.proveedor.telefono IS 'el número de telefono del proveedor';

COMMENT ON COLUMN public.proveedor.fax IS 'el número de fax del proveedor';

COMMENT ON COLUMN public.proveedor.ciudad IS 'la ciudad en la que se encuentra las instalaciones de la empresa o del del proveedor';

COMMENT ON COLUMN public.proveedor.cuenta_bancaria IS 'en número de la cuenta bancaria del proveedor';

COMMENT ON COLUMN public.proveedor.estado IS 'el estado del proveedor (activo, iactivo)';

COMMENT ON COLUMN public.proveedor.observacion IS 'la posible observacion que se tenga del proveedor (la primera adquisicion de esta empresa o de este proveedor)';

/* ----- */
/* Add table "public.rol" */
/* ----- */

CREATE TABLE public.rol (
id_rol INTEGER DEFAULT nextval('roles_id_rol_seq') NOT NULL,
nombre_rol CHARACTER VARYING(30) NOT NULL,

```
    estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,  
    CONSTRAINT identificador_rol PRIMARY KEY (id_rol)  
);
```

```
COMMENT ON COLUMN public.rol.id_rol IS 'el identificador del rol';
```

```
COMMENT ON COLUMN public.rol.nombre_rol IS 'el nombre del rol ';
```

```
COMMENT ON COLUMN public.rol.estado IS 'el estado del rol';
```

```
/* ----- */  
/* Add table "public.almacen" */  
/* ----- */
```

```
CREATE TABLE public.almacen (  
    id_almacen INTEGER DEFAULT nextval('almacen_id_almacen_seq') NOT  
NULL,  
    nombre CHARACTER(2) DEFAULT 'A1'::bpchar NOT NULL,  
    direccion CHARACTER VARYING(20) NOT NULL,  
    estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,  
    observacion TEXT,  
    CONSTRAINT identificador_almacen PRIMARY KEY (id_almacen)  
);
```

```
COMMENT ON COLUMN public.almacen.id_almacen IS 'identificador de Almacen ';
```

```
COMMENT ON COLUMN public.almacen.nombre IS 'Nombre del almacen';
```

```
COMMENT ON COLUMN public.almacen.direccion IS 'direccion actual del almacen';
```

```
COMMENT ON COLUMN public.almacen.estado IS 'estado del almacen (activo,  
inactivo)';
```

```
COMMENT ON COLUMN public.almacen.observacion IS 'observacion que se deba  
tomar en cuenta del almacen (lleno, con espacio, el más cercano)';
```

```
CREATE TABLE public.tipo_venta (  
    id_tipo_venta INTEGER DEFAULT nextval('tipo_venta_id_tipo_venta_seq')  
NOT NULL,  
    nombre CHARACTER VARYING(40) NOT NULL,  
    estado CHARACTER VARYING(3) DEFAULT 'A' NOT NULL,  
    tipo CHARACTER,  
    CONSTRAINT ideniticator_de_tipo_tabla PRIMARY KEY (id_tipo_venta)
```

);

COMMENT ON COLUMN public.tipo_venta.id_tipo_venta IS 'el identificador del tipo de venta';

COMMENT ON COLUMN public.tipo_venta.nombre IS 'el nombre del tipo de venta';

COMMENT ON COLUMN public.tipo_venta.estado IS 'el estado del tipo de venta (activo, inactivo)';

CREATE TABLE public.compra (
 id_compra INTEGER DEFAULT nextval('compra_id_compra_seq') NOT
 NULL,
 fecha DATE NOT NULL,
 monto_total REAL NOT NULL,
 id_proveedor INTEGER NOT NULL,
 observacion TEXT,
 estado CHARACTER VARYING(3) DEFAULT 'A' NOT NULL,
 CONSTRAINT identificador_de_compra PRIMARY KEY (id_compra)
);

COMMENT ON COLUMN public.compra.id_compra IS 'el identificador de la compra';

COMMENT ON COLUMN public.compra.fecha IS 'la fecha en la que se está registrando la venta';

COMMENT ON COLUMN public.compra.monto_total IS 'el monto total cancelado por la compra de los productos';

COMMENT ON COLUMN public.compra.id_proveedor IS 'el identificador del proveedor';

COMMENT ON COLUMN public.compra.observacion IS 'las observaciones posibles del proveedor (es la primera vez que se realiza una compra de esta empresa)';

COMMENT ON COLUMN public.compra.estado IS 'es estado actual de la compra (activa, inactiva)';

CREATE TABLE public.persona (
 ci BIGINT NOT NULL,
 id_persona INTEGER NOT NULL,

nombre CHARACTER VARYING(30) NOT NULL,
ap_paterno CHARACTER VARYING(30),
ap_materno CHARACTER VARYING(30) NOT NULL,
fecha_nac DATE NOT NULL,
direccion TEXT NOT NULL,
telefono CHARACTER VARYING(30) NOT NULL,
descripcion TEXT NOT NULL,
sexo CHARACTER(1) DEFAULT 'F'::bpchar NOT NULL,
ocupacion CHARACTER VARYING(10) DEFAULT 'S'::bpchar NOT NULL,
estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,
CONSTRAINT identificador_de_persona PRIMARY KEY (id_persona)
);

COMMENT ON COLUMN public.persona.ci IS 'el número de la cedula de
identificacion de la persona';

COMMENT ON COLUMN public.persona.id_persona IS 'el identificador de la
persona';

COMMENT ON COLUMN public.persona.nombre IS 'el nombre de la persona';

COMMENT ON COLUMN public.persona.ap_paterno IS 'el apellido paterno de la
persona';

COMMENT ON COLUMN public.persona.ap_materno IS 'el apellido materno de la
persona ';

COMMENT ON COLUMN public.persona.fecha_nac IS 'la fecha de nacimiento de la
persona';

COMMENT ON COLUMN public.persona.direccion IS 'la direccion actual de donde
vive la persona';

COMMENT ON COLUMN public.persona.telefono IS 'el número de telefono actual de
la persona';

COMMENT ON COLUMN public.persona.descripcion IS 'la posible observacion de la
persona (verificar posibles deudas)';

COMMENT ON COLUMN public.persona.sexo IS 'el género de la persona';

COMMENT ON COLUMN public.persona.ocupacion IS 'la ocupacion actual de la
persona (trabajador público, estudiante, ama de casa u otros)';

COMMENT ON COLUMN public.persona.estado IS 'el estado de la persona (activo, inactivo)';

```
CREATE TABLE public.producto (  
  id_producto INTEGER DEFAULT nextval('producto_id_producto_seq') NOT  
  NULL,  
  nombre CHARACTER VARYING(30) NOT NULL,  
  marca CHARACTER VARYING(20) NOT NULL,  
  precio_unitario REAL NOT NULL,  
  descuento REAL DEFAULT 0 NOT NULL,  
  precio_oferta REAL DEFAULT 0 NOT NULL,  
  estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,  
  observacion TEXT,  
  categoria CHARACTER VARYING(30) NOT NULL,  
  imagen CHARACTER VARYING(50) NOT NULL,  
  tamano CHARACTER VARYING(20),  
  color CHARACTER VARYING(15),  
  CONSTRAINT identificador_de_producto PRIMARY KEY (id_producto)  
);
```

COMMENT ON COLUMN public.producto.id_producto IS 'el identificador del producto';

COMMENT ON COLUMN public.producto.nombre IS 'el nombre del producto';

COMMENT ON COLUMN public.producto.marca IS 'la marca del producto';

COMMENT ON COLUMN public.producto.precio_unitario IS 'el precio unitario con el que se encuentra registrado para la venta el producto ';

COMMENT ON COLUMN public.producto.descuento IS 'el posible monto de descuento con el que cuenta el producto';

COMMENT ON COLUMN public.producto.precio_oferta IS 'el posible monto de oferta con el que se encuentre el producto';

COMMENT ON COLUMN public.producto.estado IS 'es estado del producto (activo, inactivo, recuperado, en reparacion y otros)';

COMMENT ON COLUMN public.producto.observacion IS 'la posible observacion del producto';

COMMENT ON COLUMN public.producto.categoria IS 'el nombre de la categoria a la que pertenece el producto';

COMMENT ON COLUMN public.producto.imagen IS 'una fotografia del producto';

COMMENT ON COLUMN public.producto.tamano IS 'el tamaño del producto (por medida, por descripcion, por cantidad y otros)';

COMMENT ON COLUMN public.producto.color IS 'el color del producto';

CREATE TABLE public.cargo (
 id_cargo INTEGER DEFAULT nextval('cargo_id_cargo_seq') NOT NULL,
 nombre_cargo CHARACTER VARYING(30) NOT NULL,
 carga_horaria CHARACTER VARYING(20) NOT NULL,
 estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,
 CONSTRAINT identificador PRIMARY KEY (id_cargo)
);

COMMENT ON COLUMN public.cargo.id_cargo IS 'identificador del cargo';

COMMENT ON COLUMN public.cargo.nombre_cargo IS 'nombre del cargo
(Vendedor(a), Secretaria(o),
Cajero(a))';

COMMENT ON COLUMN public.cargo.carga_horaria IS 'la carga horaria que se le
asignó a este cargo';

COMMENT ON COLUMN public.cargo.estado IS 'el estado actual del cargo (activo,
inactivo, prueba u otro)';

CREATE TABLE public.empleado (
 id_empleado INTEGER DEFAULT nextval('empleado_id_empleado_seq')
NOT NULL,
 id_persona INTEGER NOT NULL,
 id_cargo INTEGER NOT NULL,
 estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,
 fecha_ingreso DATE NOT NULL,
 observacion TEXT NOT NULL,
 CONSTRAINT identificador_empleado PRIMARY KEY (id_empleado)
);

COMMENT ON COLUMN public.empleado.id_empleado IS 'identificador del
empleado';

COMMENT ON COLUMN public.Empleado.id_persona IS 'el identificador de los registros de persona ';

COMMENT ON COLUMN public.Empleado.id_cargo IS 'identificador del cargo del empleado';

COMMENT ON COLUMN public.Empleado.estado IS 'el estado del cargo (Activo, Inactivo)';

COMMENT ON COLUMN public.Empleado.fecha_ingreso IS 'la fecha en la que se está registrando al empleado';

COMMENT ON COLUMN public.Empleado.observacion IS 'las posibles observaciones que se tenga del empleado (a prueba)';

**CREATE TABLE public.tipo_cliente (
 id_tipo_cliente INTEGER DEFAULT nextval('tipo_cliente_id_tipo_cliente_seq')
NOT NULL,
 estado CHARACTER VARYING(3) DEFAULT 'A' NOT NULL,
 tipo CHARACTER VARYING(20) NOT NULL,
 CONSTRAINT identificador_de_tipo_cliente PRIMARY KEY (id_tipo_cliente)
);**

COMMENT ON COLUMN public.tipo_cliente.id_tipo_cliente IS 'el identificador del tipo de cliente';

COMMENT ON COLUMN public.tipo_cliente.estado IS 'el estado del tipo de cliente (activo, inactivo)';

COMMENT ON COLUMN public.tipo_cliente.tipo IS 'el tipo de cliente (de oro, moroso, puntual, impuntual, irregular y otros)';

**CREATE TABLE public.producto_almacenado (
 id_almacen INTEGER NOT NULL,
 id_producto INTEGER NOT NULL,
 cantidad INTEGER DEFAULT 0 NOT NULL,
 CONSTRAINT identificador_productoa_almacenados (id_producto)
PRIMARY KEY (id_almacen, id_producto)
);**

**CREATE TABLE public.cliente (
 id_cliente INTEGER DEFAULT nextval('cliente_id_cliente_seq') NOT NULL,**

```
fecha_registro DATE DEFAULT now() NOT NULL,  
estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,  
observaciones TEXT,  
id_persona INTEGER NOT NULL,  
id_tipo_cliente INTEGER NOT NULL,  
CONSTRAINT identificador_cliente PRIMARY KEY (id_cliente)  
);
```

```
COMMENT ON COLUMN public.cliente.id_cliente IS 'el identificador del cliente';
```

```
COMMENT ON COLUMN public.cliente.fecha_registro IS 'la fecha en la que se está  
registrando al cliente';
```

```
COMMENT ON COLUMN public.cliente.estado IS 'es estado actual del cliente';
```

```
COMMENT ON COLUMN public.cliente.observaciones IS 'las observaciones que se  
pueda tener del cliente (no tiene vivienda propia)';
```

```
COMMENT ON COLUMN public.cliente.id_persona IS 'el identificador de persona';
```

```
COMMENT ON COLUMN public.cliente.id_tipo_cliente IS 'el identificador del tipo de  
cliente ';
```

```
CREATE TABLE public.usuario (  
  id_usuario INTEGER DEFAULT nextval('usuario_id_usuario_seq') NOT  
  NULL,  
  nombre_usuario CHARACTER VARYING(10) NOT NULL,  
  clave_usuario CHARACTER VARYING(16) NOT NULL,  
  estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,  
  observacion TEXT,  
  id_persona INTEGER NOT NULL,  
  id_rol INTEGER NOT NULL,  
  imagen_usuario CHARACTER VARYING(200),  
  CONSTRAINT identificador_de_usuario PRIMARY KEY (id_usuario)  
);
```

```
COMMENT ON COLUMN public.usuario.id_usuario IS 'el identificador del usuario';
```

```
COMMENT ON COLUMN public.usuario.nombre_usuario IS 'el nombre del usuario';
```

```
COMMENT ON COLUMN public.usuario.clave_usuario IS 'la clave del usuario';
```


COMMENT ON COLUMN public.usuario.estado IS 'el estado del usuario (activo, inactivo)';

COMMENT ON COLUMN public.usuario.observacion IS 'la posible observacion del usuario';

COMMENT ON COLUMN public.usuario.id_persona IS 'el identificador de persona (para poder tener acceso a la persona)';

COMMENT ON COLUMN public.usuario.id_rol IS 'el identificador del rol del usuario en el sistema';

COMMENT ON COLUMN public.usuario.imagen_usuario IS 'la fotografia del usuario';

CREATE TABLE public.dosificacion (
 id_dosificacion INTEGER NOT NULL,
 llave CHARACTER VARYING,
 fecha DATE,
 fecha_limite DATE,
 numero_aut CHARACTER VARYING,
 id_usuario INTEGER,
 CONSTRAINT dosificacion_pkey PRIMARY KEY (id_dosificacion)
);

CREATE TABLE public.d_compra (
 id_d_compra INTEGER DEFAULT nextval('d_compra_id_d_compra_seq')
NOT NULL,
 id_producto INTEGER NOT NULL,
 id_compra INTEGER NOT NULL,
 cantidad SMALLINT NOT NULL,
 precio_compra REAL NOT NULL,
 total REAL NOT NULL,
 observacion TEXT,
 CONSTRAINT identificador_de_d_compra PRIMARY KEY (id_d_compra)
);

COMMENT ON COLUMN public.d_compra.id_d_compra IS 'el identificador del detalle de la venta';

COMMENT ON COLUMN public.d_compra.id_producto IS 'el o los identificadores de los productos que se encuentran registrados en la venta';

COMMENT ON COLUMN public.d_compra.id_compra IS 'el identificador de la compra';

COMMENT ON COLUMN public.d_compra.cantidad IS 'la cantidad de productos que se registraron en la venta o compra';

COMMENT ON COLUMN public.d_compra.precio_compra IS 'el monto total cancelado por la compra';

COMMENT ON COLUMN public.d_compra.total IS 'el precio total de todos los productos registrados en la venta';

COMMENT ON COLUMN public.d_compra.observacion IS 'las posibles observaciones que pueda tener tanto la venta como la compra';

CREATE TABLE public.venta (
 id_venta INTEGER DEFAULT nextval('venta_id_venta_seq') NOT NULL,
 fecha DATE NOT NULL,
 monto_total CHARACTER VARYING NOT NULL,
 id_cliente INTEGER NOT NULL,
 id_usuario INTEGER NOT NULL,
 estado CHARACTER VARYING(3) DEFAULT 'A'::bpchar NOT NULL,
 observacion TEXT,
 nit INTEGER,
 id_tipo_venta INTEGER,
 CONSTRAINT identificador_de_venta (id_venta) PRIMARY KEY (id_venta)
);

COMMENT ON COLUMN public.venta.id_venta IS 'el identificador de la venta';

COMMENT ON COLUMN public.venta.fecha IS 'la fecha en la que se está registrando la venta';

COMMENT ON COLUMN public.venta.monto_total IS 'el monto total de la venta';

COMMENT ON COLUMN public.venta.id_cliente IS 'el identificador del cliente';

COMMENT ON COLUMN public.venta.id_usuario IS 'el identificador del usuario que está registrando la venta';

COMMENT ON COLUMN public.venta.estado IS 'el estado de la venta';

COMMENT ON COLUMN public.venta.observacion IS 'la posibles obsevacion de la venta';

COMMENT ON COLUMN public.venta.nit IS 'el número de nit del cliente';

COMMENT ON COLUMN public.venta.id_tipo_venta IS 'el identificador del tipo de venta (directa o con plan de pagos)';

```
CREATE TABLE public.d_venta (  
  id_d_venta INTEGER DEFAULT nextval('d_venta_id_d_venta_seq') NOT  
NULL,  
  id_producto INTEGER NOT NULL,  
  id_venta INTEGER NOT NULL,  
  cantidad INTEGER NOT NULL,  
  precio_venta REAL NOT NULL,  
  total REAL NOT NULL,  
  CONSTRAINT identificador_de_d_venta PRIMARY KEY (id_d_venta)  
);
```

COMMENT ON COLUMN public.d_venta.id_d_venta IS 'el identificador de la venta';

COMMENT ON COLUMN public.d_venta.id_producto IS 'el identificador del producto';

COMMENT ON COLUMN public.d_venta.id_venta IS 'el identificador de la venta';

COMMENT ON COLUMN public.d_venta.cantidad IS 'la cantidad de productos registrados en la venta';

COMMENT ON COLUMN public.d_venta.precio_venta IS 'el precio de cada producto para el registro de la venta';

COMMENT ON COLUMN public.d_venta.total IS 'el monto total de todos los productos registrados en la venta';

```
CREATE TABLE public.detalle_pago (  
  id_detalle_pago INTEGER DEFAULT  
nextval('detalle_pago_id_detalle_pago_seq') NOT NULL,  
  fecha_pago DATE,  
  monto_pago REAL NOT NULL,  
  nro_de_pago INTEGER NOT NULL,  
  estado CHARACTER VARYING(3) DEFAULT 'A' NOT NULL,  
  id_venta INTEGER NOT NULL,  
  observacion TEXT NOT NULL,  
  fecha_pagar DATE,
```

**CONSTRAINT identificador_de_detalle_pago PRIMARY KEY
(id_detalle_pago)
);**

COMMENT ON COLUMN public.detalle_pago.id_detalle_pago IS 'el identificador de las ventas con plan de pagos';

COMMENT ON COLUMN public.detalle_pago.fecha_pago IS 'la fecha en la que se registra el pago inicial';

COMMENT ON COLUMN public.detalle_pago.monto_pago IS 'el monto inicial cancelado (y los demas)';

COMMENT ON COLUMN public.detalle_pago.nro_de_pago IS 'la cantidad de pagos que se realizara para cancelar el monto total de la venta';

COMMENT ON COLUMN public.detalle_pago.estado IS 'el estado actual del detalle de pago de una venta con plan de pagos (activo, inactivo)';

COMMENT ON COLUMN public.detalle_pago.id_venta IS 'el identificador de la venta con plan de pago a la que pertenece el detalle de pago';

COMMENT ON COLUMN public.detalle_pago.observacion IS 'las posibles observaciones de la venta con plan de pagos (consideracion especial, por ser un cliente preferencial)';

**CREATE TABLE public.producto_observado (
 id_producto_observado INTEGER DEFAULT
nextval('producto_observado_id_producto_observado_seq') NOT NULL,
 id_producto INTEGER NOT NULL,
 id_venta INTEGER,
 id_compra INTEGER,
 observacion TEXT,
 estado CHARACTER VARYING(3) DEFAULT 'A' NOT NULL,
 cantidad SMALLINT DEFAULT 1 NOT NULL,
 id_almacen INTEGER NOT NULL,
 CONSTRAINT identificador_de_producto_observado PRIMARY KEY
(id_producto_observado)
);**

ALTER TABLE public.tipo_cliente ADD CONSTRAINT
identificador_tipocliente_estado

FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE
SET DEFAULT ON UPDATE CASCADE;

```
ALTER TABLE public.venta ADD CONSTRAINT identificador_cliente1 (id_cliente)  
FOREIGN KEY (id_cliente) REFERENCES public.cliente (id_cliente);
```

```
ALTER TABLE public.venta ADD CONSTRAINT identificador_tipo_de_venta  
FOREIGN KEY (id_tipo_venta) REFERENCES public.tipo_venta (id_tipo_venta);
```

```
ALTER TABLE public.venta ADD CONSTRAINT identificador_usuario15  
FOREIGN KEY (id_usuario) REFERENCES public.usuario (id_usuario);
```

```
ALTER TABLE public.venta ADD CONSTRAINT identificador_venta_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.estado ADD CONSTRAINT identificador_estad_de_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET NULL ON UPDATE CASCADE;
```

```
ALTER TABLE public.proveedor ADD CONSTRAINT  
identificador_proveedor_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.rol ADD CONSTRAINT identificador_rol_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.producto_almacenado ADD CONSTRAINT  
identificador_productos_almacenados12 (id_almacen)  
FOREIGN KEY (id_almacen) REFERENCES public.almacen (id_almacen) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.producto_almacenado ADD CONSTRAINT  
identificador_productos_almacenados13 (id_producto)  
FOREIGN KEY (id_producto) REFERENCES public.producto (id_producto) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.d_venta ADD CONSTRAINT identificador1_de_producto  
FOREIGN KEY (id_producto) REFERENCES public.producto (id_producto) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.d_venta ADD CONSTRAINT identificador_de_venta7  
FOREIGN KEY (id_venta) REFERENCES public.venta (id_venta) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.cliente ADD CONSTRAINT identificador_cliente_estado  
    FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.cliente ADD CONSTRAINT identificador_de_tipo_cliente4  
    FOREIGN KEY (id_tipo_cliente) REFERENCES public.tipo_cliente  
(id_tipo_cliente) ON UPDATE CASCADE;
```

```
ALTER TABLE public.cliente ADD CONSTRAINT identificador_persona11  
    FOREIGN KEY (id_persona) REFERENCES public.persona (id_persona) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.almacen ADD CONSTRAINT identificador_almacen_estado  
    FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.usuario ADD CONSTRAINT identificador_persona9  
    FOREIGN KEY (id_persona) REFERENCES public.persona (id_persona) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.usuario ADD CONSTRAINT identificador_rol14  
    FOREIGN KEY (id_rol) REFERENCES public.rol (id_rol) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.usuario ADD CONSTRAINT identificador_usuario_estado  
    FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.dosificacion ADD CONSTRAINT  
dosificacion_id_usuario_fkey  
    FOREIGN KEY (id_usuario) REFERENCES public.usuario (id_usuario);
```

```
ALTER TABLE public.tipo_venta ADD CONSTRAINT  
identificador_tipoventa_estado  
    FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.d_compra ADD CONSTRAINT identificador2_de_producto  
    FOREIGN KEY (id_producto) REFERENCES public.producto (id_producto);
```

```
ALTER TABLE public.d_compra ADD CONSTRAINT identificador_de1_compra  
    FOREIGN KEY (id_compra) REFERENCES public.compra (id_compra);
```

```
ALTER TABLE public.detalle_pago ADD CONSTRAINT identificador_de_venta8  
FOREIGN KEY (id_venta) REFERENCES public.venta (id_venta) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.detalle_pago ADD CONSTRAINT  
identificador_delledepago_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.compra ADD CONSTRAINT identificador_compra_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.compra ADD CONSTRAINT  
identificador_de_proveedor_en_compra  
FOREIGN KEY (id_proveedor) REFERENCES public.proveedor (id_proveedor);
```

```
ALTER TABLE public.persona ADD CONSTRAINT identificador_persona_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.producto_observado ADD CONSTRAINT  
identificador_compra_en_productoobservado  
FOREIGN KEY (id_compra) REFERENCES public.compra (id_compra);
```

```
ALTER TABLE public.producto_observado ADD CONSTRAINT  
identificador_de_almacen1  
FOREIGN KEY (id_almacen) REFERENCES public.almacen (id_almacen) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE public.producto_observado ADD CONSTRAINT  
identificador_idventa_en_productoobservado  
FOREIGN KEY (id_venta) REFERENCES public.venta (id_venta);
```

```
ALTER TABLE public.producto_observado ADD CONSTRAINT  
identificador_producto_en_productoobservado  
FOREIGN KEY (id_producto) REFERENCES public.producto (id_producto);
```

```
ALTER TABLE public.producto_observado ADD CONSTRAINT  
identificador_productoobservado_estado  
FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE  
SET DEFAULT ON UPDATE CASCADE;
```

```
ALTER TABLE public.producto ADD CONSTRAINT identificador_producto_estado
```

FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE SET DEFAULT ON UPDATE CASCADE;

ALTER TABLE public.cargo ADD CONSTRAINT identificador_cargo_estado FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE SET DEFAULT ON UPDATE CASCADE;

ALTER TABLE public.empleado ADD CONSTRAINT identificador_cargo1 FOREIGN KEY (id_cargo) REFERENCES public.cargo (id_cargo) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE public.empleado ADD CONSTRAINT identificador_empleado_estado FOREIGN KEY (estado) REFERENCES public.estado (id_estado) ON DELETE SET DEFAULT ON UPDATE CASCADE;

ALTER TABLE public.empleado ADD CONSTRAINT identificador_persona10 FOREIGN KEY (id_persona) REFERENCES public.persona (id_persona) ON DELETE CASCADE ON UPDATE CASCADE;

2.1.2.2.3.2.4 ESPECIFICACIÓN DE MÉTODOS

A continuación se detalla especificación de los métodos de los objetos utilizados.

1. Objeto Persona:

OBJETO:	Persona
MÉTODO:	<code>getAllPersona</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(cantidad).list(); return resp;</pre>	
DESCRIPCIÓN	
El método <code>getAllPersona</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista y <code>estado</code> para listar Personas (activas o inactivas) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Persona
MÉTODO:	getAllPersonas
ARGUMENTO:	Buscar
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).list(); return resp;</pre>	
DESCRIPCIÓN	
El método <code>getAllPersonas</code>, necesita el argumento <code>Buscar</code> para mostrar los datos requeridos por pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Persona
MÉTODO:	getIdPersona
ARGUMENTO:	idPersona
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = getSessionFactory().getCurrentSession(); return (Persona) session.get(Persona.class, idPersona);</pre>	
DESCRIPCIÓN	
El método <code>getIdPersona</code>, necesita como argumento <code>idPersona</code>, para obtener el objeto Persona de la Base de Datos.	

2. Objeto Empleado:

OBJETO:	Empleado
MÉTODO:	getAllRegistroEmpleado
ARGUMENTO:	inicio, cantidad y estado
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(</pre>	

<code>inicio).setMaxResults(cantidad).list();</code>
DESCRIPCIÓN
El método <code>getAllRegistroEmpleado</code> , necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Empleados (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code> , es una función de hibernate para realizar consultas a la base de datos.

OBJETO:	Empleado
MÉTODO:	<code>getCantidadRegistroEmpleados</code>
ARGUMENTO:	<code>buscar</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); return Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método <code>getCantidadRegistroEmpleados</code> , necesita el argumento <code>Buscar</code> para mostrar los datos requeridos por pantalla. <code>CreateQuery</code> , es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Empleado
MÉTODO:	<code>saveRegistroEmpleado</code>
ARGUMENTO:	<code>(RegistroEmpleado registroempleado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(registroempleado);</pre>	
DESCRIPCIÓN	
El método <code>saveRegistroEmpleado</code> , necesita como argumento el objeto <code>registroempleado</code> para guardar en la Base de Datos.	

OBJETO:	Empleado

MÉTODO:	updateRegistroEmpleado
ARGUMENTO:	(RegistroEmpleado registroempleado
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(registroempleado);</pre>	
DESCRIPCIÓN	
El método updateRegistroEmpleado, necesita como argumento (registroempleado), para modificar en la Base de Datos.	

OBJETO:	Empleado
MÉTODO:	deleteRegistroEmpleado
ARGUMENTO:	RegistroEmpleado registroempleado)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.delete(((RegistroEmpleado) session.get(RegistroEmpleado.class, registroempleado.getIdRegistroEmpleado())));</pre>	
DESCRIPCIÓN	
El método deleteRegistroEmpleado, necesita como argumento RegistroEmpleado), para poder eliminar en la Base de datos.	

OBJETO:	Empleado
MÉTODO:	getIdRegistroEmpleado
ARGUMENTO:	idRegistroEmpleado
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>return (RegistroEmpleado) session.get(RegistroEmpleado.class, idRegistroEmpleado);</pre>	
DESCRIPCIÓN	
El método getIdRegistroEmpleado, necesita como argumento idRegistroEmpleado	

para obtener el objeto Empleado de la Base de Datos.

3. Ver Plan de Pago:

OBJETO:	Ver Plan de Pago
MÉTODO:	<code>getAllPlanDePagos</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(cantidad).list(); return resp;</pre>	
DESCRIPCIÓN	
El método <code>getAllPlanDePagos</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Plan de Pagos (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

4. Objeto Detalle de Compra:

OBJETO:	Detalle de Compra
MÉTODO:	<code>getAllDCompra</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(c antidad).list(); return resp;</pre>	
DESCRIPCIÓN	
El método <code>getAllDCompra</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Detalles de Compras, (activas o inactivas) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Detalle de Compra
MÉTODO:	getCantidadDCompras
ARGUMENTO:	buscar
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); Return new Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método <code>getCantidadDCompras</code>, necesita el argumento <code>buscar</code>, para mostrar los datos requeridos por pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Detalle de Compra
MÉTODO:	saveDCompra
ARGUMENTO:	(DCompra dcompra)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(dcompra);</pre>	
DESCRIPCIÓN	
El método <code>saveDCompra</code>, necesita como argumento <code>dcompra</code>, para guardar en la Base de Datos.	

OBJETO:	Detalle de Compra
MÉTODO:	updateDCompra
ARGUMENTO:	(DCompra dcompra)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(dcompra);</pre>	

DESCRIPCIÓN	
El método <code>updateDCompra</code> , necesita como argumento <code>(dcompra)</code> , para modificar en la Base de Datos.	

OBJETO:	Detalle de Compra
MÉTODO:	<code>deleteDCompra</code>
ARGUMENTO:	<code>(DCompra dcompra)</code>
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.delete((DCompra) session.get(DCompra.class, dcompra.getIdDCompra()));</pre>	
DESCRIPCIÓN	
El método <code>deleteDCompra</code> , necesita como argumento <code>dcompra</code> , para poder eliminar en la Base de datos.	

OBJETO:	Detalle de Compra
MÉTODO:	<code>getIdDCompra</code>
ARGUMENTO:	<code>idDCompra</code>
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>return (DCompra) session.get(DCompra.class, idDCompra);</pre>	
DESCRIPCIÓN	
El método <code>getIdDCompra</code> , necesita como argumento <code>idDCompra</code> , para obtener el objeto Detalle de Compra, de la Base de Datos.	

5. Objeto Detalle de Venta:

OBJETO:	Detalle de Venta
MÉTODO:	<code>getAllDVenta()</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	public

ESPECIFICACIÓN
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(c antidad).list(); return resp;</pre>
DESCRIPCIÓN
El método <code>getAllDVenta</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Detalles de Ventas (activas o inactivas) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.

OBJETO:	Detalle de Venta
MÉTODO:	getCantidadDVentas
ARGUMENTO:	buscar
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); Return new Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método getCantidadDVentas, necesita el argumento buscar, para mostrar los datos requeridos por pantalla. CreateQuery, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Detalle de Venta
MÉTODO:	saveDVenta
ARGUMENTO:	(DVenta dventa)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(dventa);</pre>	
DESCRIPCIÓN	
El método saveDVenta, necesita como argumento dventa, para guardar en la Base de Datos.	

OBJETO:	Detalle de Venta
MÉTODO:	updateDVenta
ARGUMENTO:	(DVenta dventa)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(dventa);</pre>	
DESCRIPCIÓN	
El método updateDVenta, necesita como argumento dventa, para modificar en la Base de Datos.	

OBJETO:	Detalle de Venta
MÉTODO:	deleteDVenta
ARGUMENTO:	(DVenta dventa)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.delete((DVenta) session.get(DVenta.class, dventa.getIdDVenta())));</pre>	
DESCRIPCIÓN	
El método deleteDVenta, necesita como argumento dventa, para poder eliminar en la Base de datos.	

OBJETO:	Detalle de Venta
MÉTODO:	getIdDVenta
ARGUMENTO:	idDVenta
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>return (DVenta) session.get(DVenta.class, idDVenta);</pre>	
DESCRIPCIÓN	
El método getIdDVenta, necesita como argumento idDVenta, para obtener el objeto	

Detalle de Venta, de la Base de Datos.

6. Objeto Cliente:

OBJETO:	Cliente
MÉTODO:	<code>getAllCliente</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(cantidad).list(); return resp;</pre>	
DESCRIPCIÓN	
El método <code>getAllCliente</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Clientes (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Cliente
MÉTODO:	<code>getCantidadClientes</code>
ARGUMENTO:	<code>buscar</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); Return new Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método <code>getCantidadClientes</code>, necesita el argumento <code>buscar</code>, para mostrar los datos requeridos por pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Cliente
MÉTODO:	<code>saveCliente</code>

ARGUMENTO:	(Cliente cliente)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(cliente);</pre>	
DESCRIPCIÓN	
El método <code>saveCliente</code> , necesita como argumento <code>cliente</code> , para guardar en la Base de Datos.	

OBJETO:	Cliente
MÉTODO:	updateCliente(
ARGUMENTO:	(Cliente cliente)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(cliente);</pre>	

7. Objeto Usuario:

OBJETO:	Usuario
MÉTODO:	getAllUsuario
ARGUMENTO:	inicio, cantidad y estado
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResult s(cantidad).list(); return resp;</pre>	
DESCRIPCIÓN	
El método <code>getAllUsuario</code> , necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Usuarios (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code> , es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Usuario
MÉTODO:	getCantidadUsuarios
ARGUMENTO:	buscar
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); Return new Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método <code>getCantidadUsuarios</code>, necesita el argumento <code>buscar</code>, para mostrar los datos requeridos por pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Usuario
MÉTODO:	saveUsuario
ARGUMENTO:	(Usuario usuario)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(usuario);</pre>	
DESCRIPCIÓN	
El método <code>saveUsuario</code>, necesita como argumento <code>(usuario)</code>, para guardar en la Base de Datos.	

OBJETO:	Usuario
MÉTODO:	updateUsuario
ARGUMENTO:	(Usuario usuario)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(usuario);</pre>	
DESCRIPCIÓN	

El método `updateUsuario`, necesita como argumento `(usuario)`, para modificar en la Base de Datos.

OBJETO:	Usuario
MÉTODO:	<code>deleteUsuario</code>
ARGUMENTO:	<code>(Usuario usuario)</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.delete((Usuario) session.get(Usuario.class, usuario.getIdUsuario()));</pre>	
DESCRIPCIÓN	
El método <code>deleteUsuario</code> , necesita como argumento <code>Usuario</code> , para poder eliminar en la Base de datos.	

OBJETO:	Usuario
MÉTODO:	<code>getIdUsuario</code>
ARGUMENTO:	<code>idUsuario</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>return (Usuario) session.get(Usuario.class, idUsuario);</pre>	
DESCRIPCIÓN	

8. Objeto Estado:

OBJETO:	Estado
MÉTODO:	<code>getAllEstado</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(cantidad).list();</pre>	

<code>return resp;</code>
DESCRIPCIÓN
El método <code>getAllEstado</code> , necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Estados (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code> , es una función de hibernate para realizar consultas a la base de datos.

OBJETO:	Estado
MÉTODO:	<code>getCantidadEstados</code>
ARGUMENTO:	<code>buscar</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); Return new Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método <code>getCantidadEstados</code> , necesita el argumento <code>buscar</code> , para mostrar los datos requeridos por pantalla. <code>CreateQuery</code> , es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	Estado
MÉTODO:	<code>saveEstado</code>
ARGUMENTO:	<code>(Estado estado)</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(estado);</pre>	
DESCRIPCIÓN	
El método <code>saveEstado</code> , necesita como argumento <code>estado</code> , para guardar en la Base de Datos.	

OBJETO:	Estado
MÉTODO:	<code>updateEstado</code>

ARGUMENTO:	(Estado estado)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(estado);</pre>	
DESCRIPCIÓN	
El método <code>updateEstado</code> , necesita como argumento <code>(estado)</code> , para modificar en la Base de Datos.	

OBJETO:	Estado
MÉTODO:	<code>deleteEstado</code>
ARGUMENTO:	(Estado estado)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.delete((Estado) session.get(Estado.class, estado.getIdEstado()));</pre>	
DESCRIPCIÓN	
El método <code>deleteEstado</code> , necesita como argumento <code>(Estado)</code> , para poder eliminar en la Base de datos.	

OBJETO:	Estado
MÉTODO:	<code>getIdEstado</code>
ARGUMENTO:	<code>idEstado</code>
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>return (Estado) session.get(Estado.class, idEstado);</pre>	
DESCRIPCIÓN	
El método <code>getIdEstado</code> , necesita como argumento <code>idEstado</code> , para obtener el objeto Estado, de la Base de Datos.	

9. Objeto Tipo de Cliente:

OBJETO:	TipoCliente
MÉTODO:	getAllTipoCliente
ARGUMENTO:	inicio, cantidad y estado
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>query += "ORDER BY idTipoCliente, tipo"; resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(cantidad).list(); return resp; }</pre>	
DESCRIPCIÓN	
El método <code>getAllTipoCliente</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Estados (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	TipoCliente
MÉTODO:	getCantidadTipoClientes
ARGUMENTO:	buscar
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); Return new Integer(session.createQuery(query).list().size());</pre>	
DESCRIPCIÓN	
El método <code>getCantidadTipoClientes</code>, necesita el argumento <code>buscar</code>, para mostrar los datos requeridos por pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

OBJETO:	TipoCliente
MÉTODO:	saveTipoCliente
ARGUMENTO:	(TipoCliente tipocliente)
CONTROL DE EXPORTACIÓN:	public

ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(tipocliente);</pre>	
DESCRIPCIÓN	
El método <code>saveTipoCliente</code> , necesita como argumento <code>TipoCliente</code> , para guardar en la Base de Datos.	

OBJETO:	TipoCliente
MÉTODO:	<code>updateTipoCliente</code>
ARGUMENTO:	<code>(TipoCliente tipocliente)</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.update(tipocliente);</pre>	
DESCRIPCIÓN	
El método <code>updateTipoCliente</code> , necesita como argumento <code>(tipocliente)</code> , para modificar en la Base de Datos.	

OBJETO:	TipoCliente
MÉTODO:	<code>deleteTipoCliente</code>
ARGUMENTO:	<code>(TipoCliente tipocliente)</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.delete(((TipoCliente) session.get(TipoCliente.class, tipocliente.getIdTipoCliente())));</pre>	
DESCRIPCIÓN	
El método <code>deleteTipoCliente</code> , necesita como argumento <code>(tipocliente)</code> , para poder eliminar en la Base de datos.	

10. Dosificación:

OBJETO:	Dosificacion
MÉTODO:	<code>getAllDosificacion</code>
ARGUMENTO:	<code>inicio</code> , <code>cantidad</code> y <code>estado</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>query += "ORDER BY idDosificacion desc"; List<Dosificacion> resp = new ArrayList<Dosificacion>; Session session = sessionFactory.getCurrentSession(); resp = session.createQuery(query).setFirstResult(inicio).setMaxResults(cantidad).list(); return resp; }</pre>	
DESCRIPCIÓN	
El método <code>getAllDosificacion</code>, necesita los argumentos, <code>inicio</code> para mostrar la lista desde este valor, <code>cantidad</code> para la cantidad a mostrar en la lista, y <code>estado</code> para listar Estados (activos o inactivos) y mostrar en pantalla. <code>CreateQuery</code>, es una función de hibernate para realizar consultas a la base de datos.	

11. Adicionar dosificación:

OBJETO:	Dosificacion
MÉTODO:	<code>saveDosificacion</code>
ARGUMENTO:	<code>(Dosificacion dosificacion)</code>
CONTROL DE EXPORTACIÓN:	<code>public</code>
ESPECIFICACIÓN	
<pre>Session session = sessionFactory.getCurrentSession(); session.save(dosificacion);</pre>	
DESCRIPCIÓN	
El método <code>saveDosificacion</code>, necesita como argumento <code>estado</code>, para guardar en la Base de Datos.	

2.1.2.2.3.3 MODELO DE IMPLEMENTACIÓN

2.1.2.2.3.3.1 DIAGRAMA DE COMPONENTES

Diagrama de Componente: Ingreso a Sistema

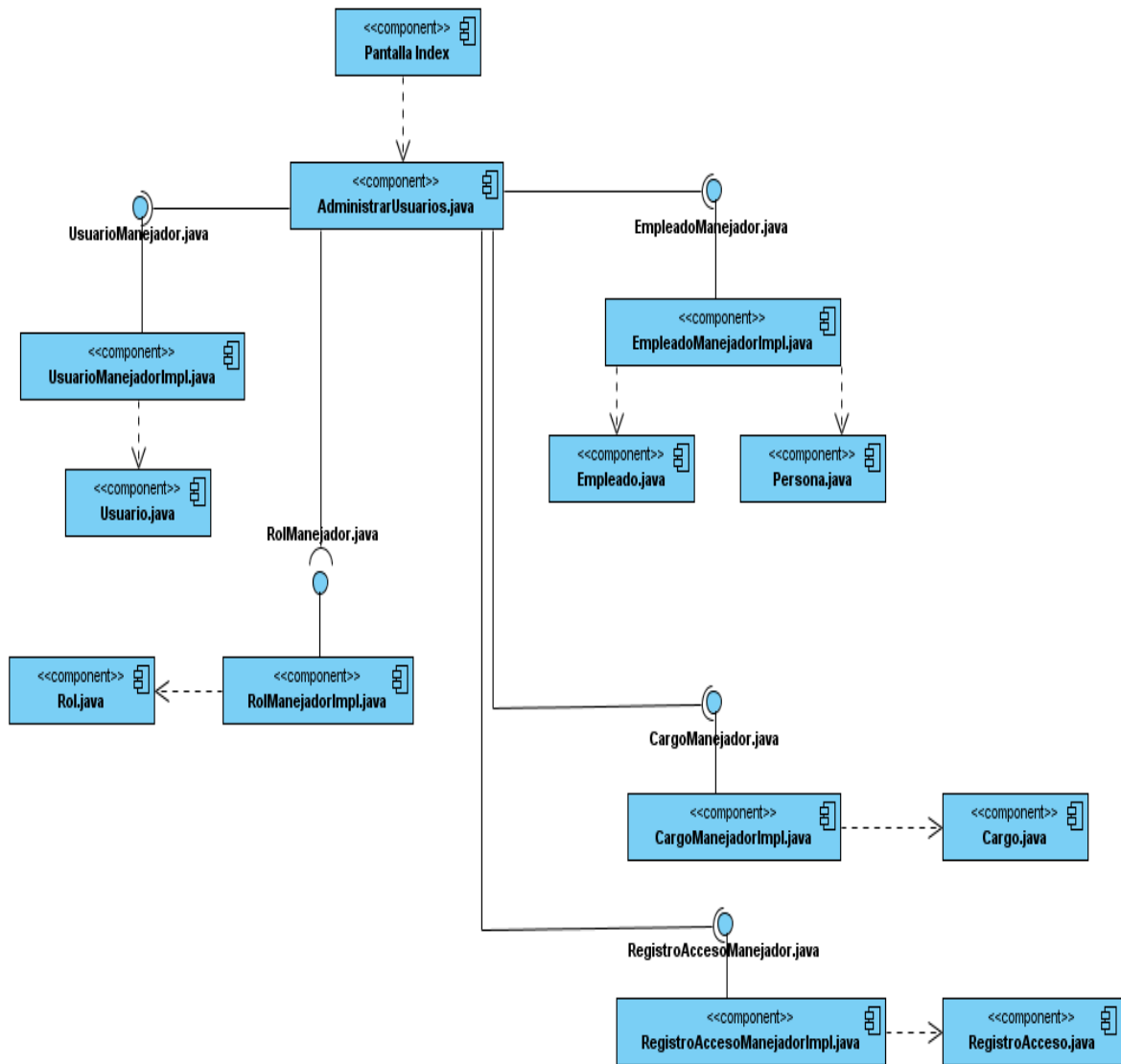


Figura 252: Diagrama de Componente-Ingreso al sistema

Diagrama de Componente: Administrar Usuarios

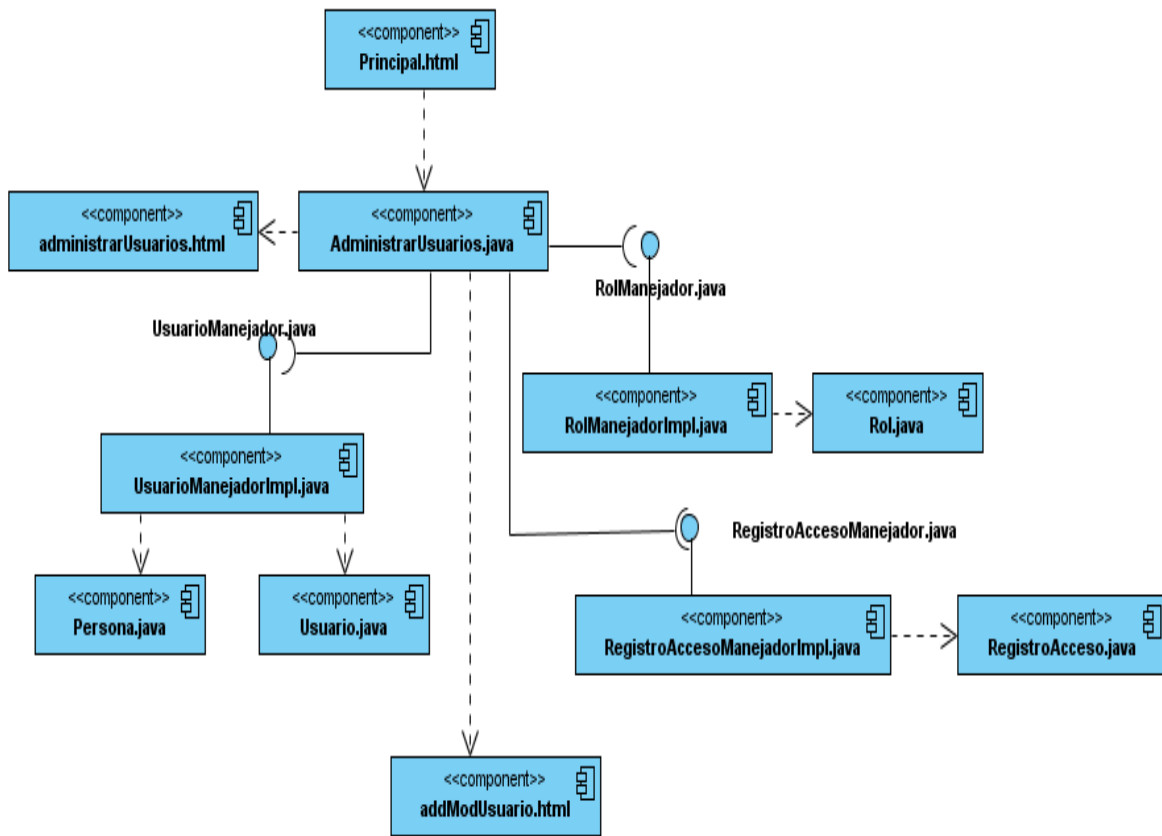


Figura 253: Diagrama de Componente-Administrar Usuarios

Diagrama de Componente: Administrar Empleados

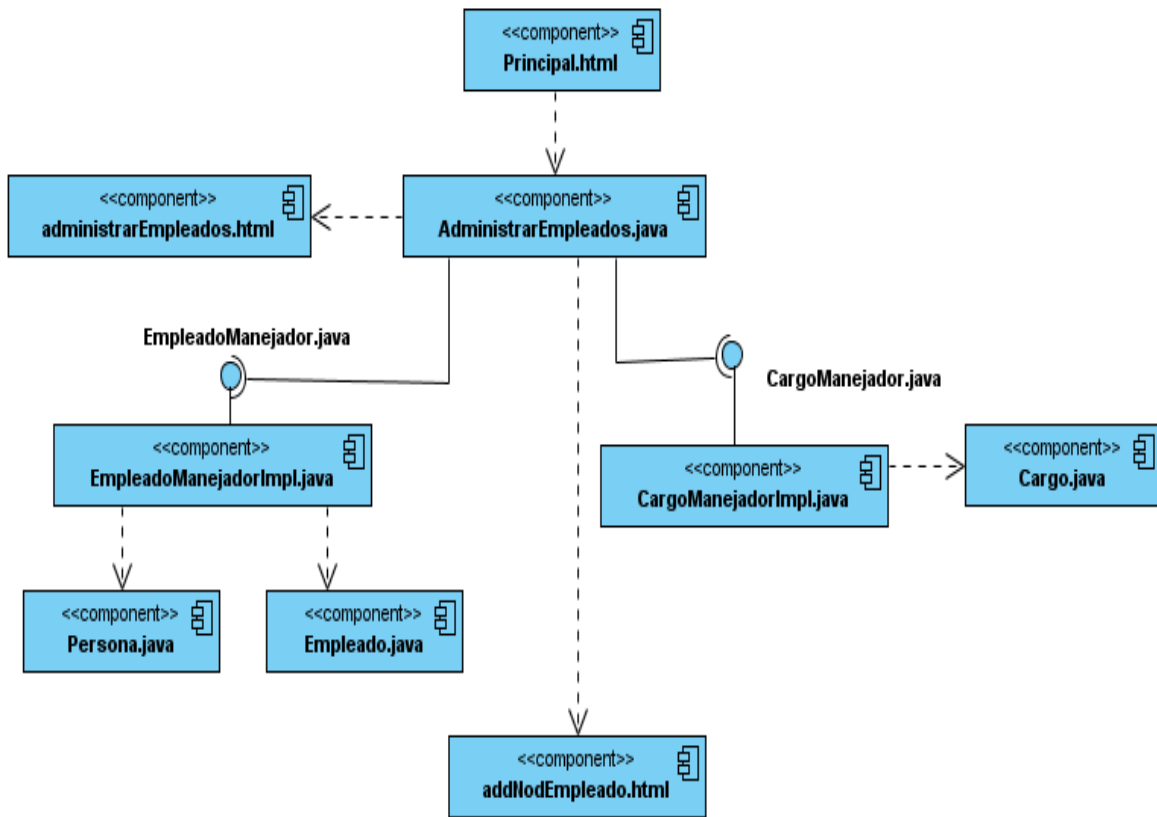


Figura 254: Diagrama de Componente-Administrar Empleados

Diagrama de Componente: Administrar Proveedores

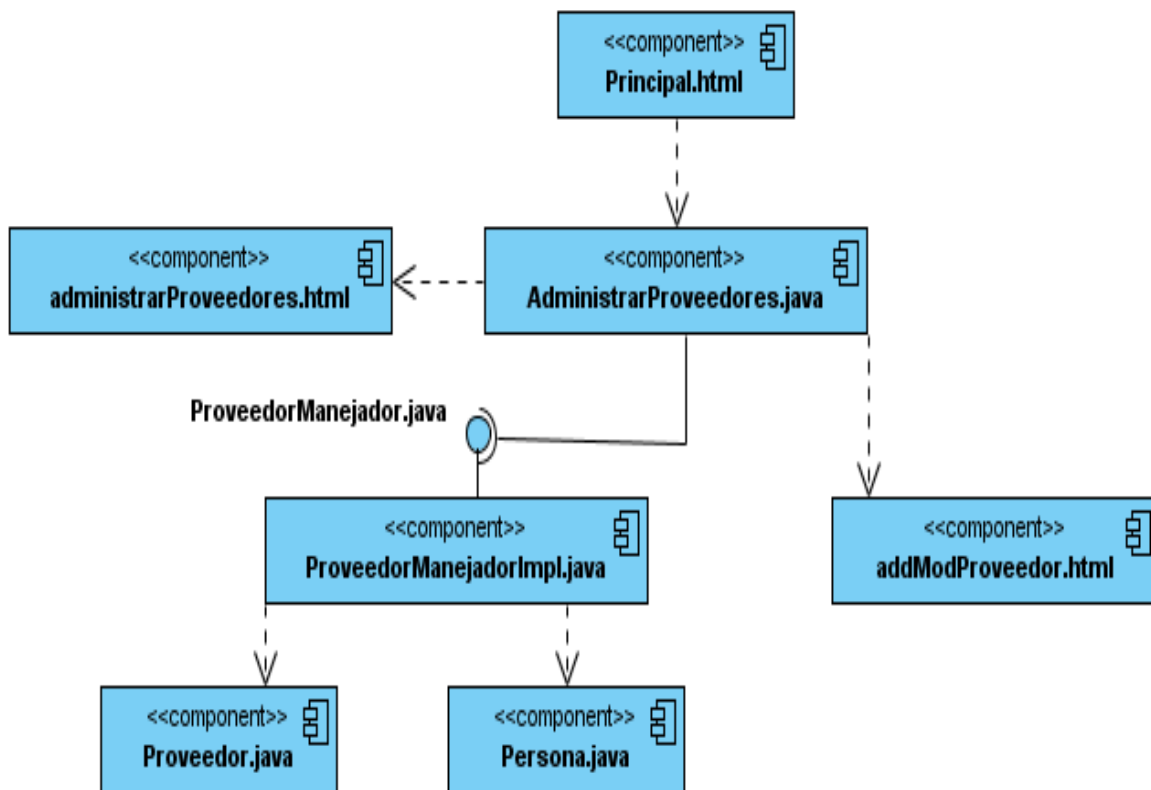


Figura 255: Diagrama de Componente-Administrar Proveedores

Diagrama de Componente: Administrar Clientes

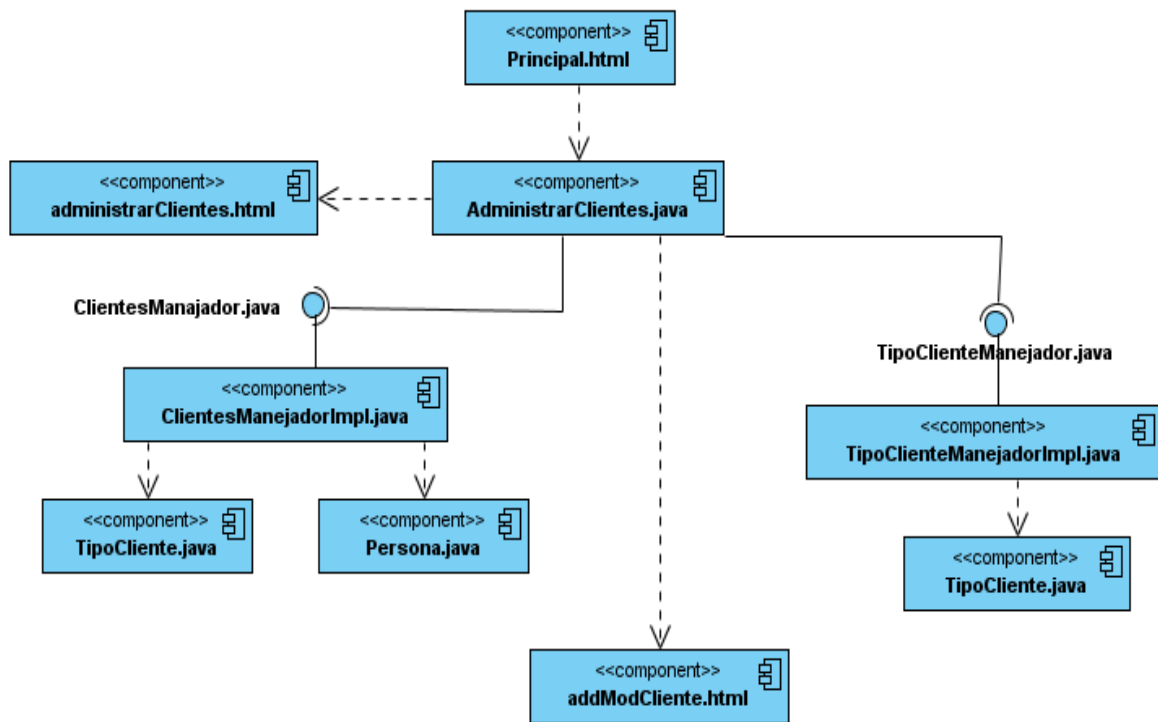


Figura 256: Diagrama de Componente-Administrar Clientes

Diagrama de Componente: Administrar Almacenes

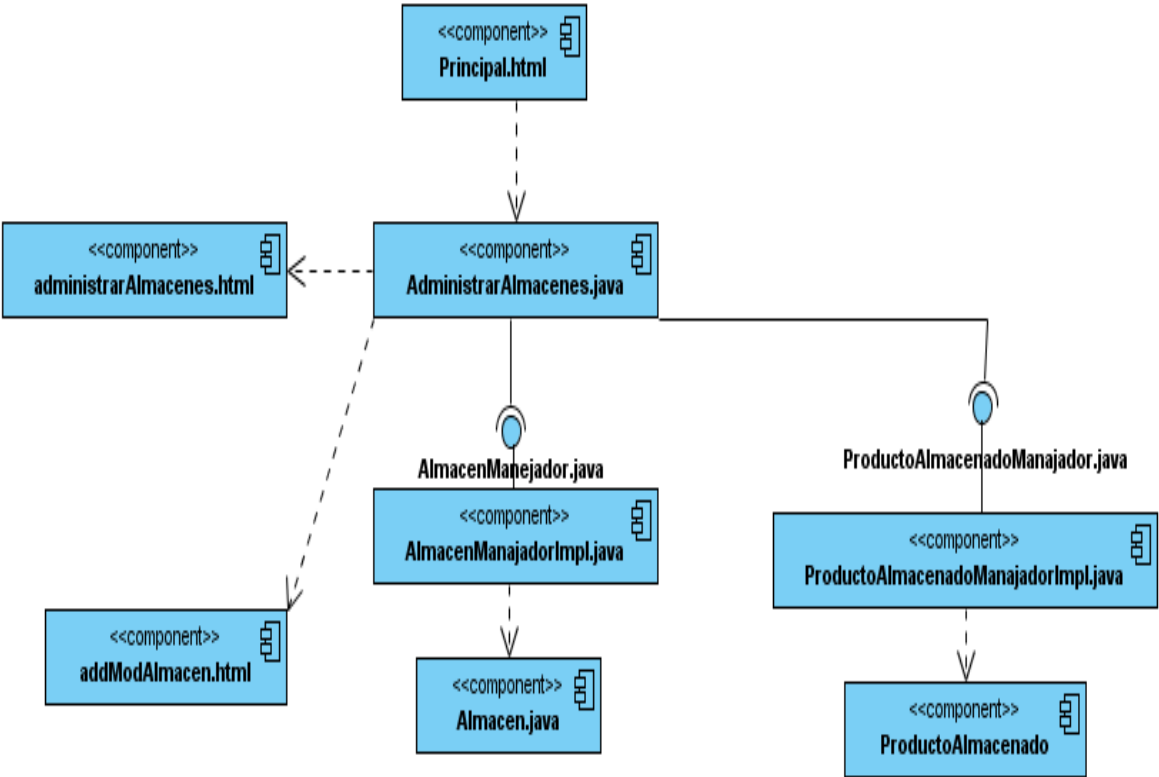


Figura 257: Diagrama de Componente-Administrar Almacenes

Diagrama de Componente: Administrar Productos

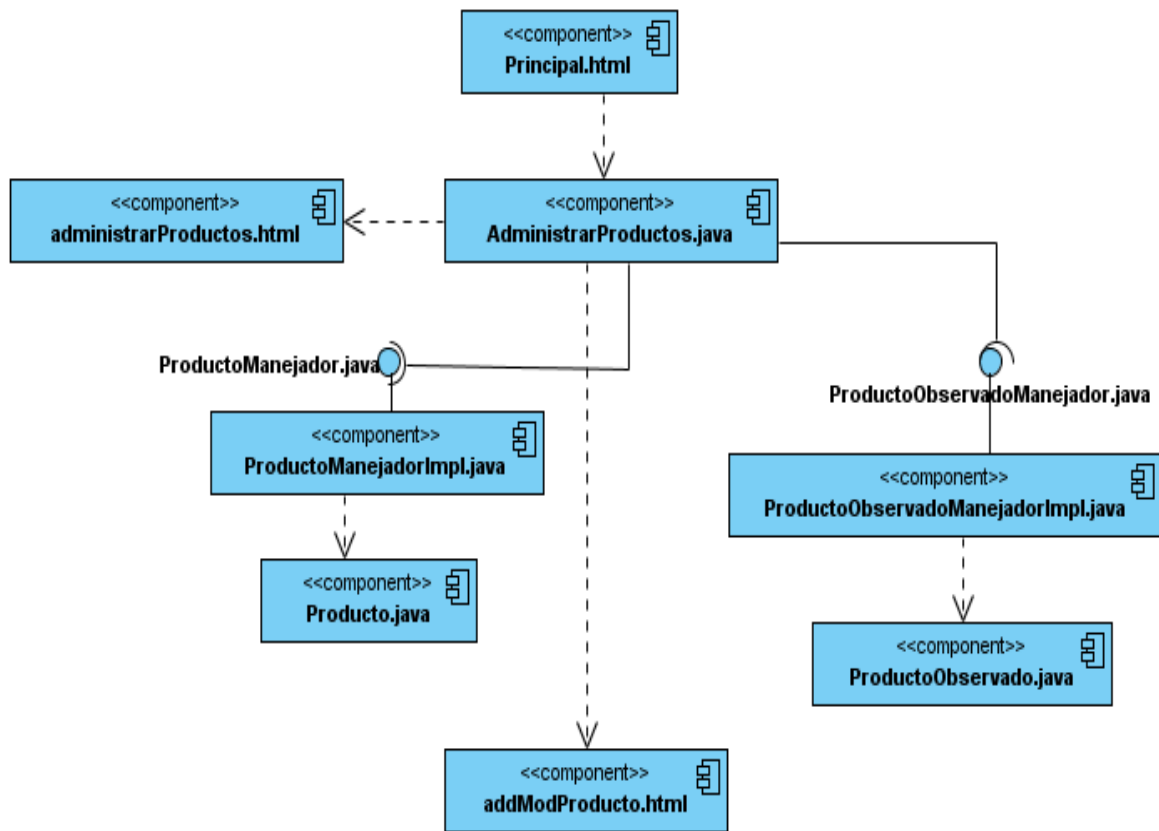


Figura 258: Diagrama de Componente-Administrar Productos

Diagrama de Componente: Administrar Compras

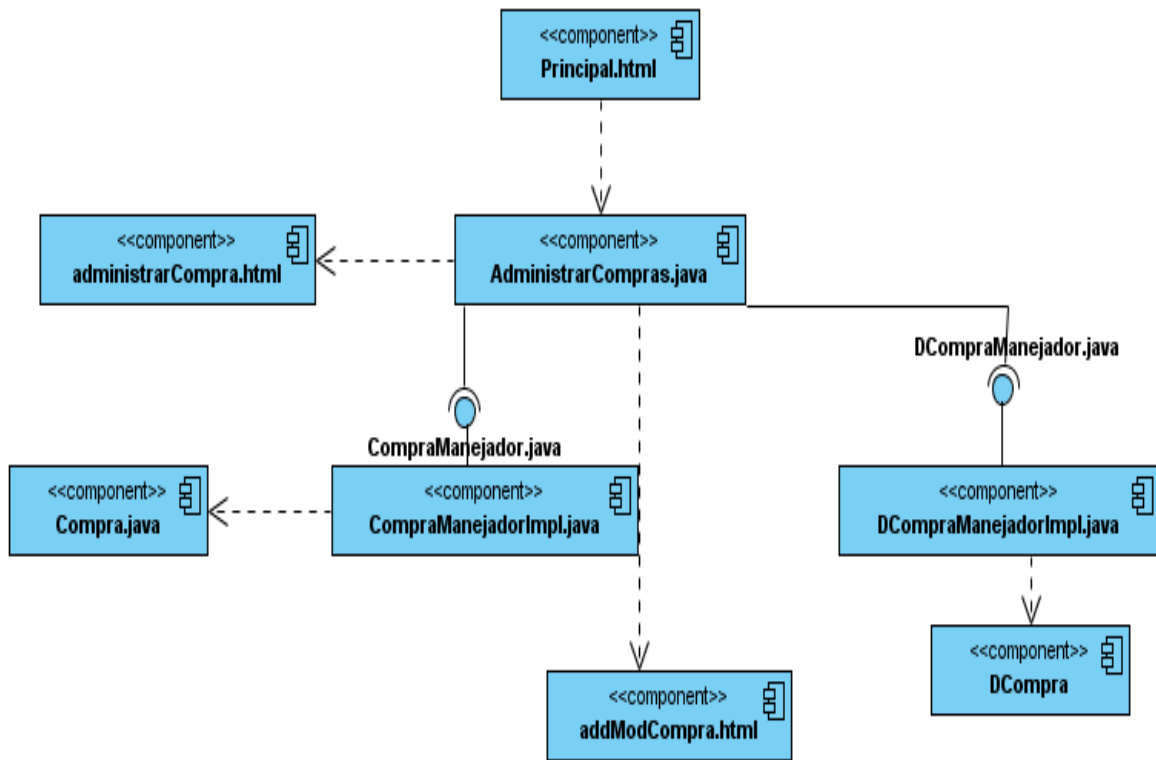


Figura 259: Diagrama de Componente-Administrar Compras

Diagrama de Componente: Administrar Plan de Pagos

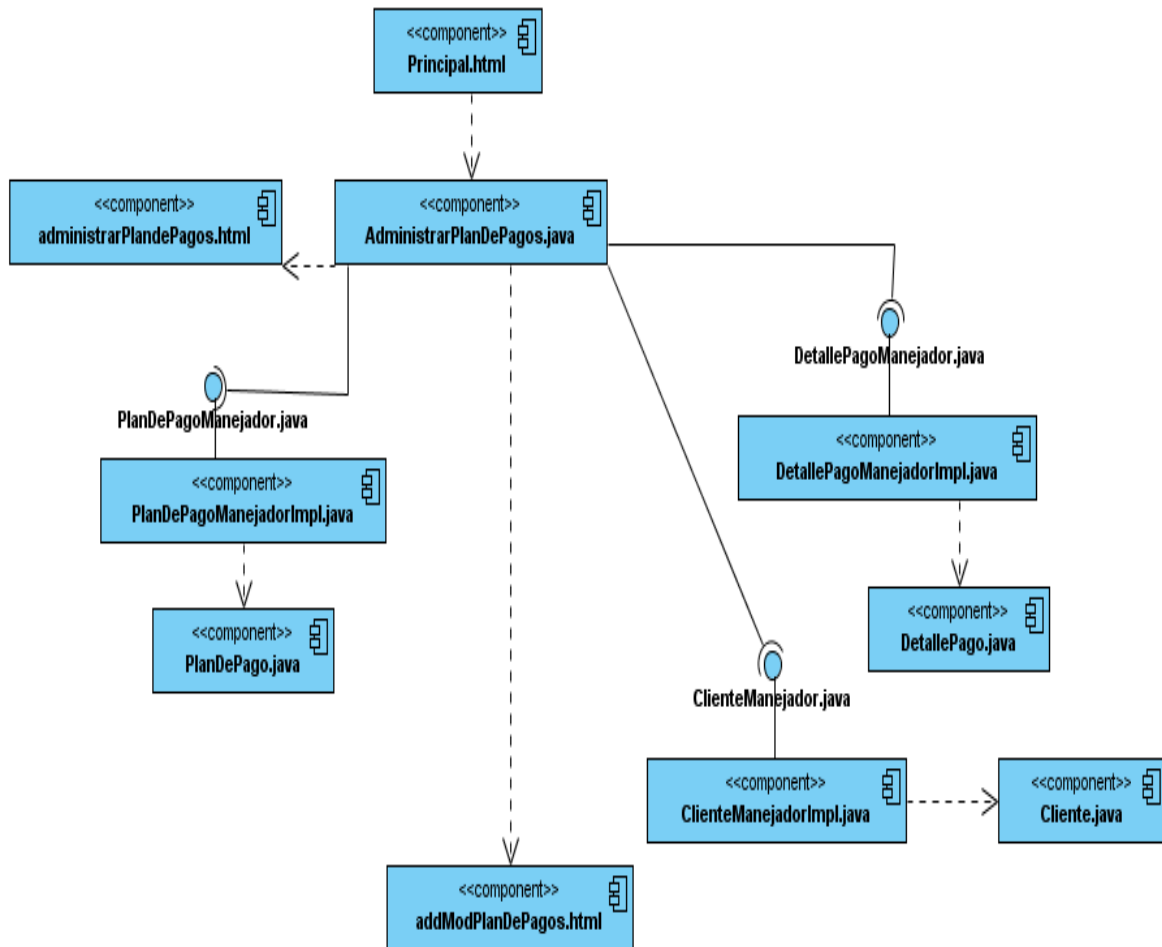


Figura 260: Diagrama de Componente-Administrar Plan de Pagos

Diagrama de Componente: Administrar Ventas

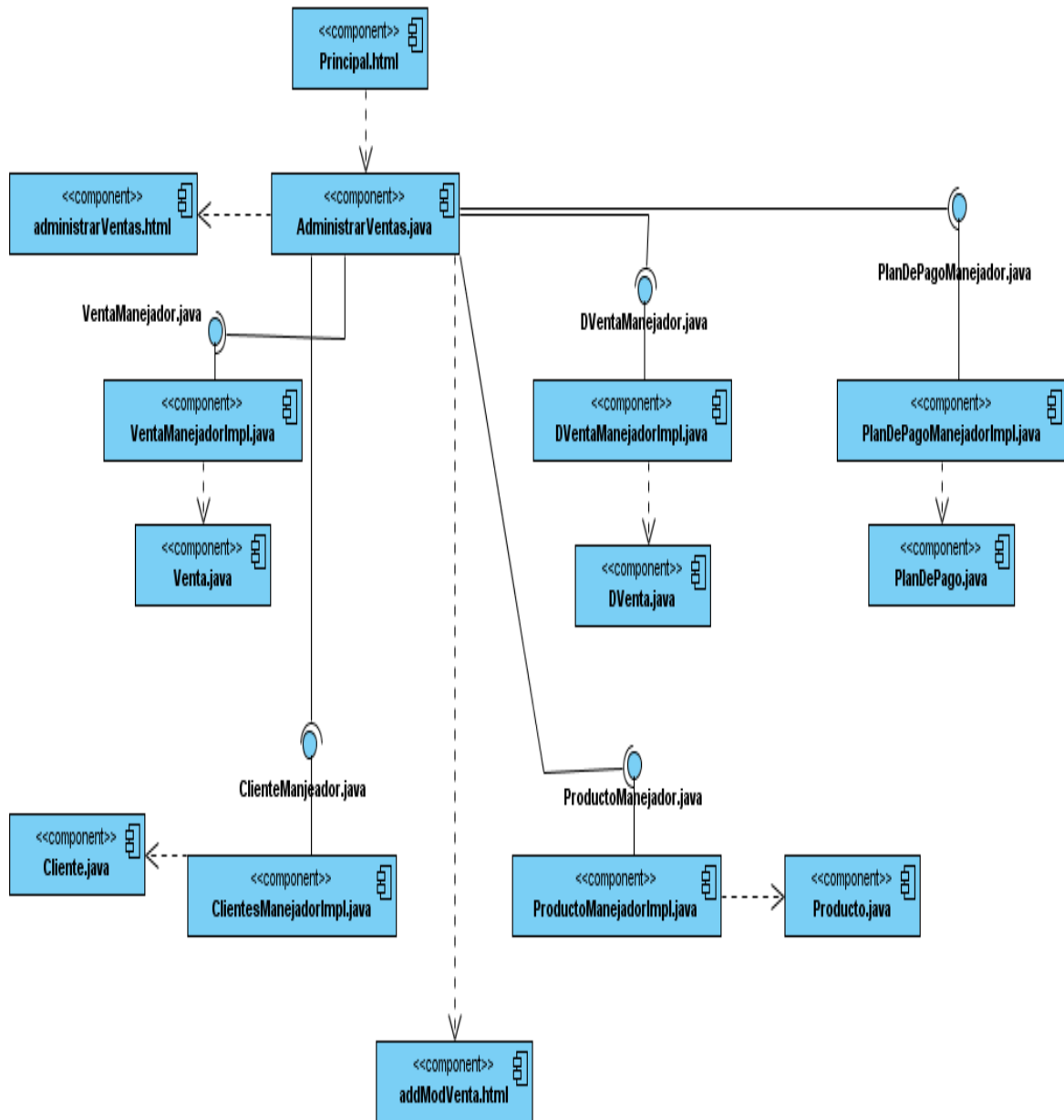


Figura 261: Diagrama de Componente-Administrar Compras

Diagrama de Componente: El Sistema SICOMFIG

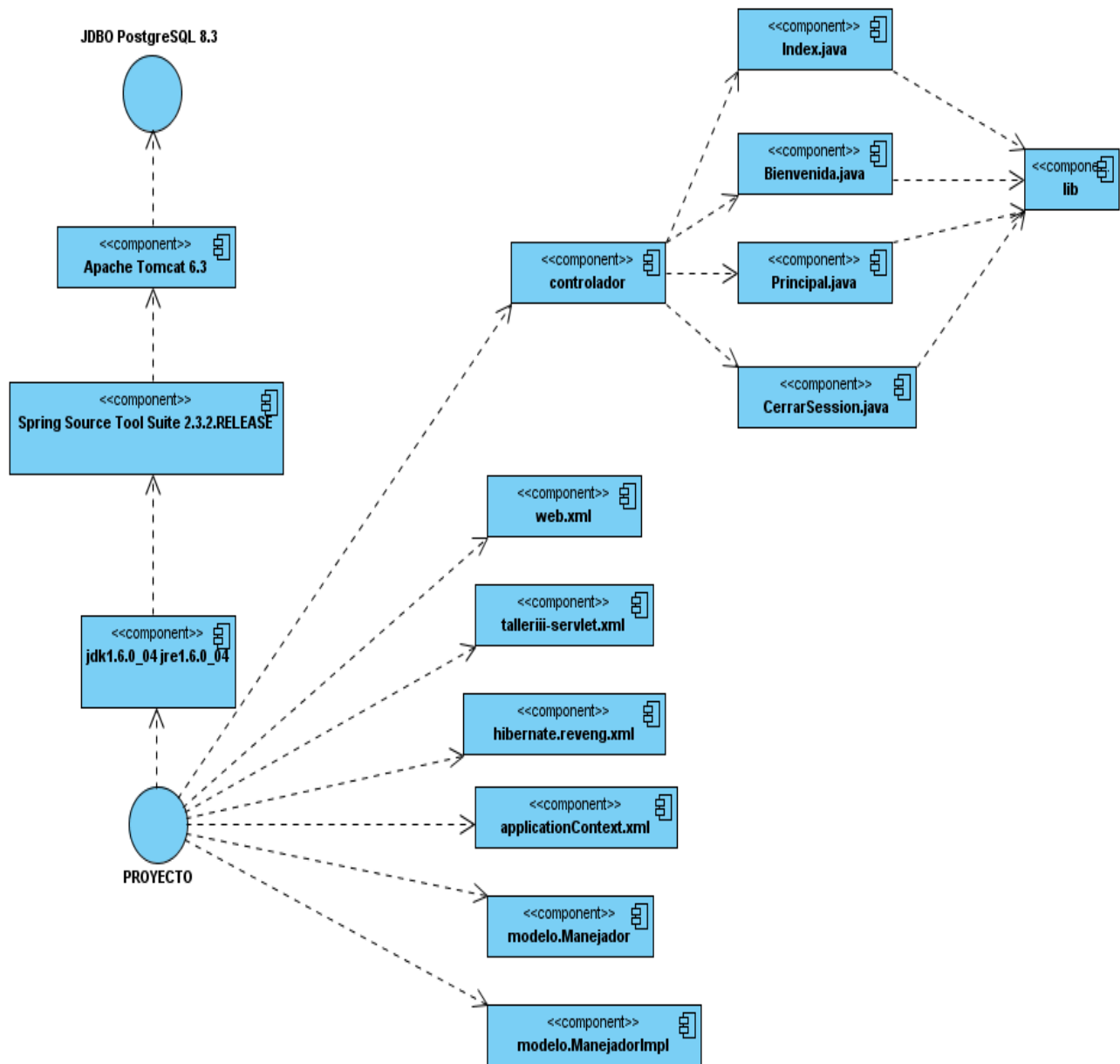


Figura 262: Diagrama de componente-El sistema SICOMFIG

Diagrama de Componente: modelo.Manejador

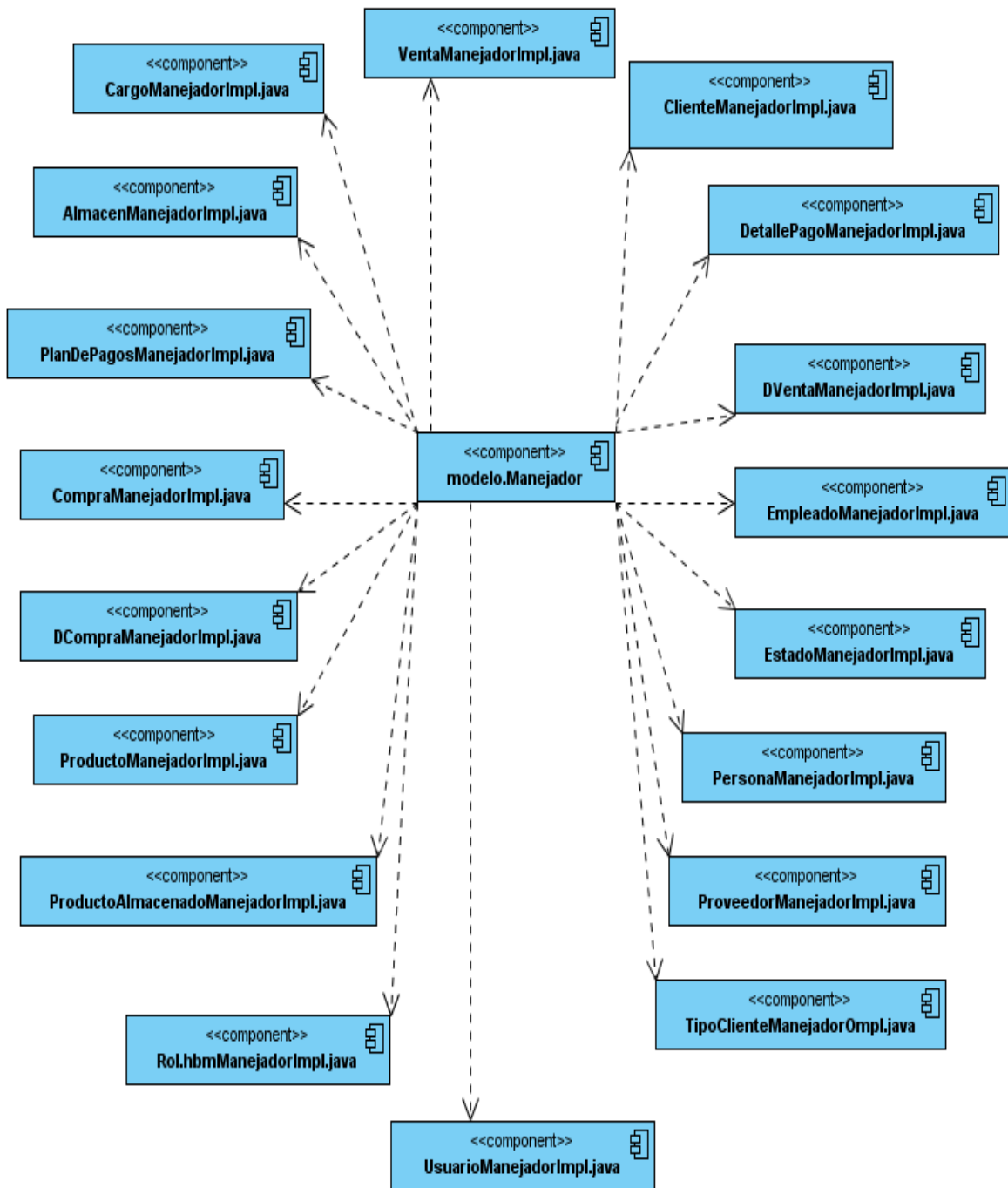


Figura 263: Diagrama de Componente-Modelo.Manejador

Diagrama de Componente: modelo.ManejadorImpl

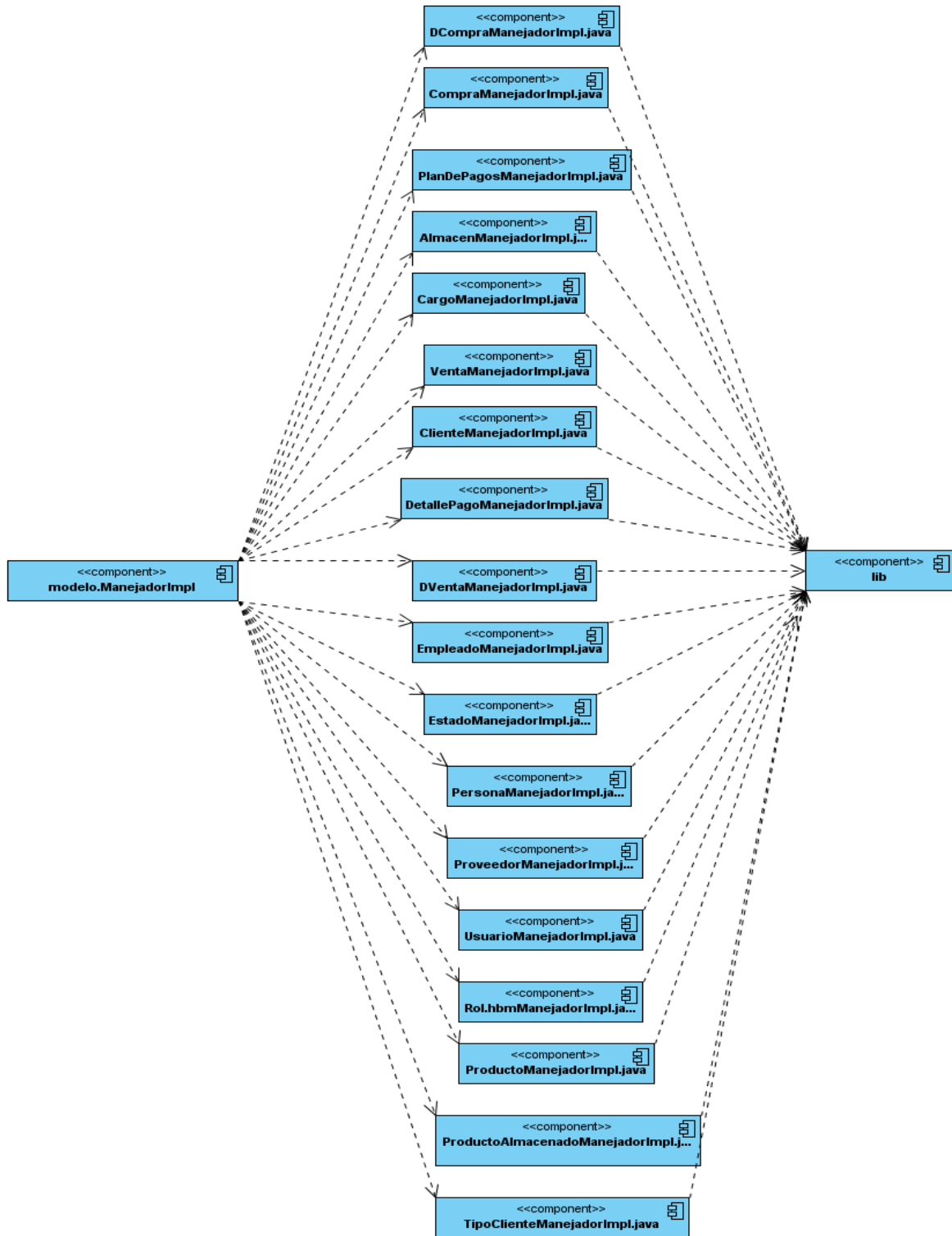


Figura 264: Diagrama de Componente-Modelo.ManejadorImpl

Diagrama de Componente: Librerías primera parte (lib)

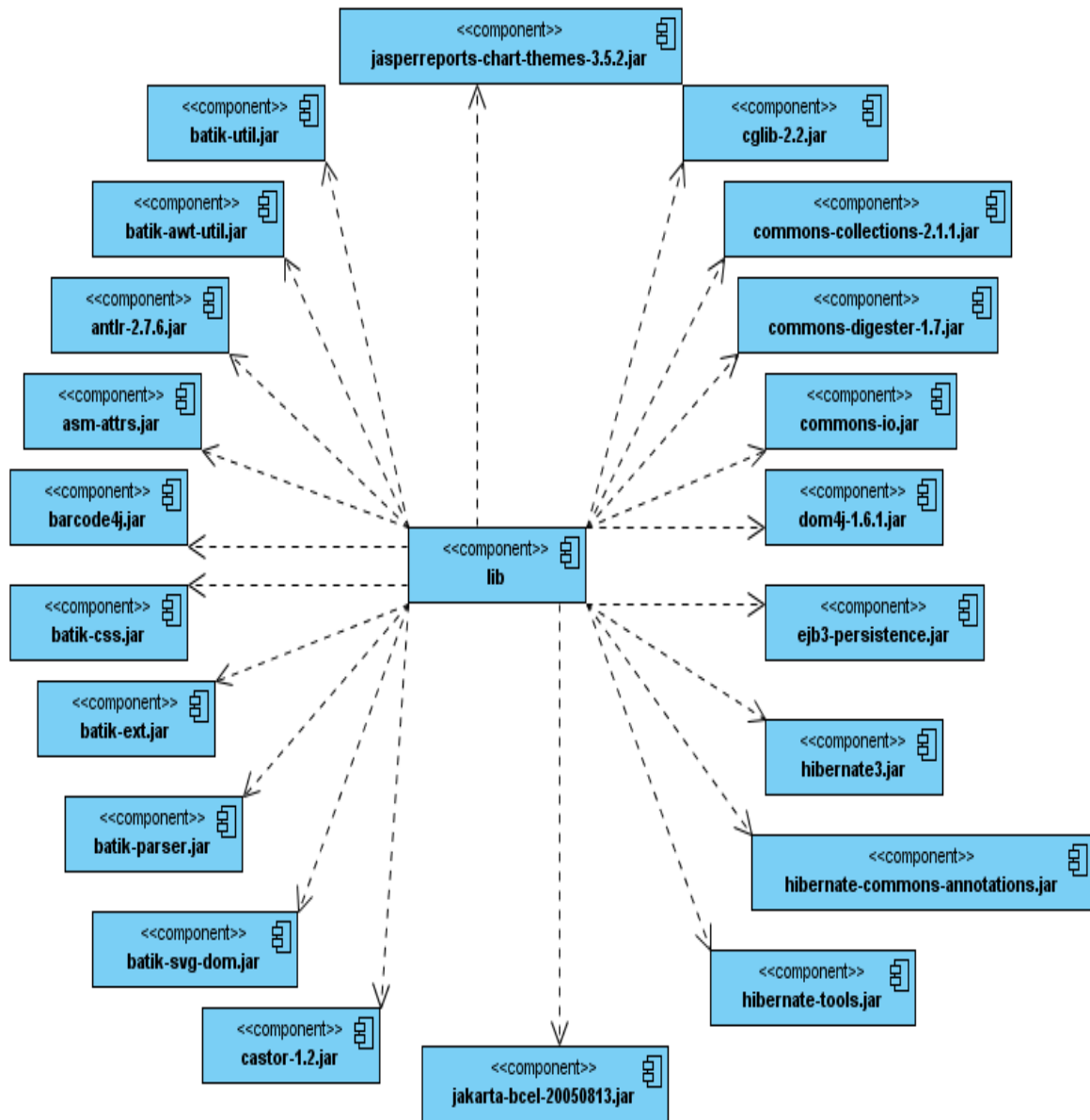


Figura 265: Diagrama de Componente-Lib. Parte 1

Diagrama de Componente: Librerías segunda parte (lib)

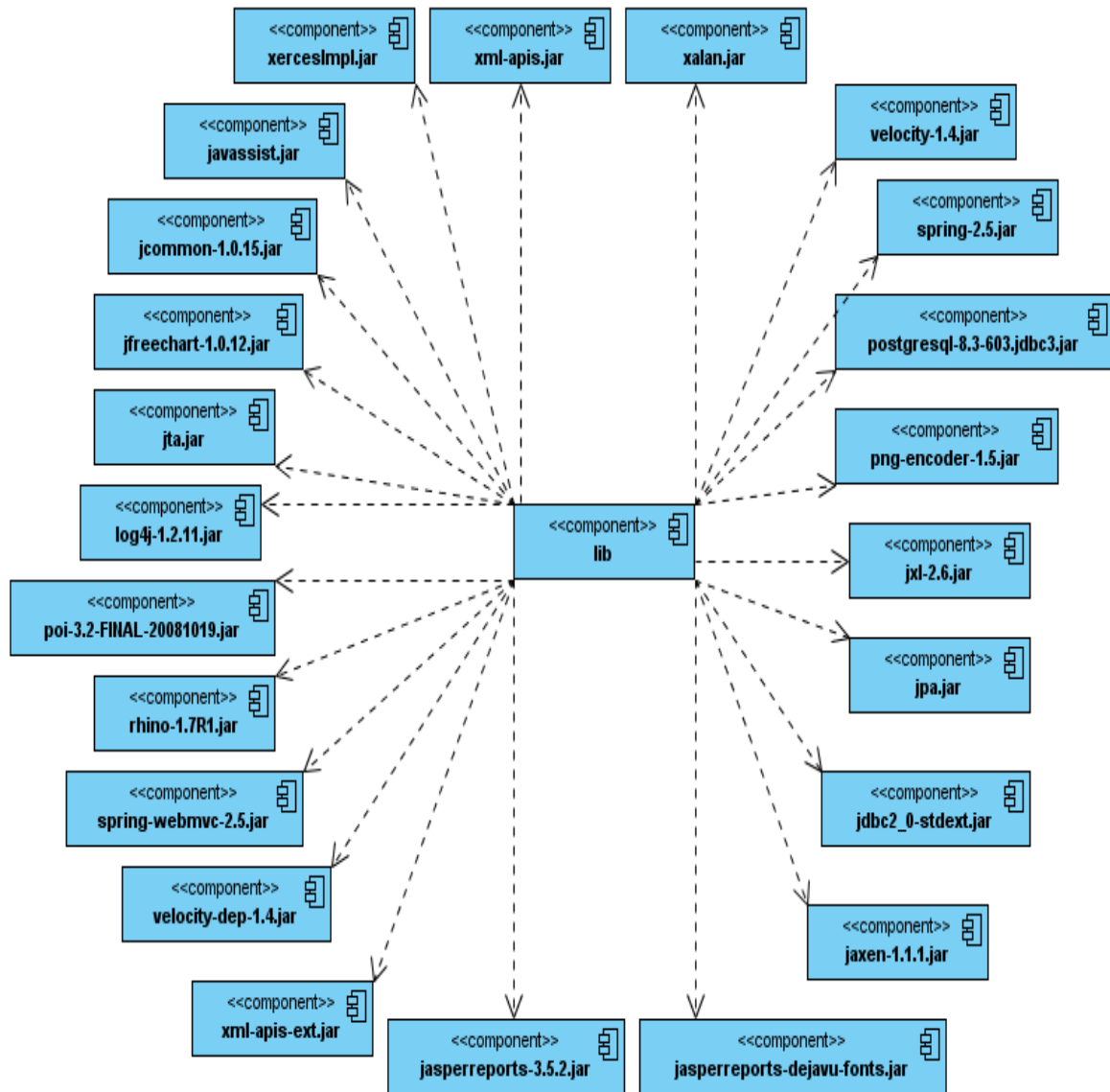


Figura 266: Diagrama de Componente-Lib. Parte 2

Diagrama de Componente: Librerías tercera parte (lib)

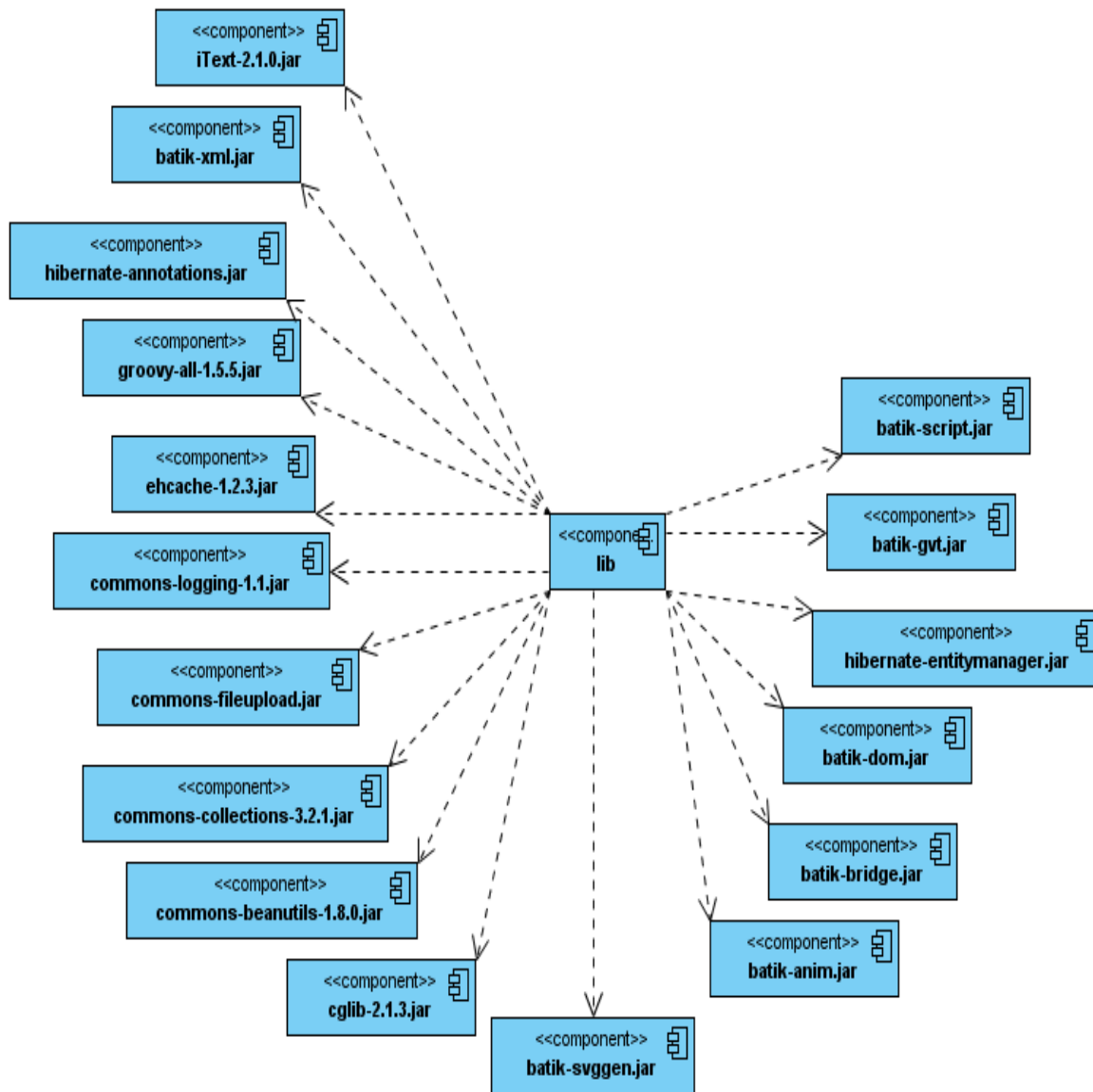


Figura 267: Diagrama de Componente-Lib. Parte 3

2.1.2.2.3.3.2 DIAGRAMA DE PAQUETES

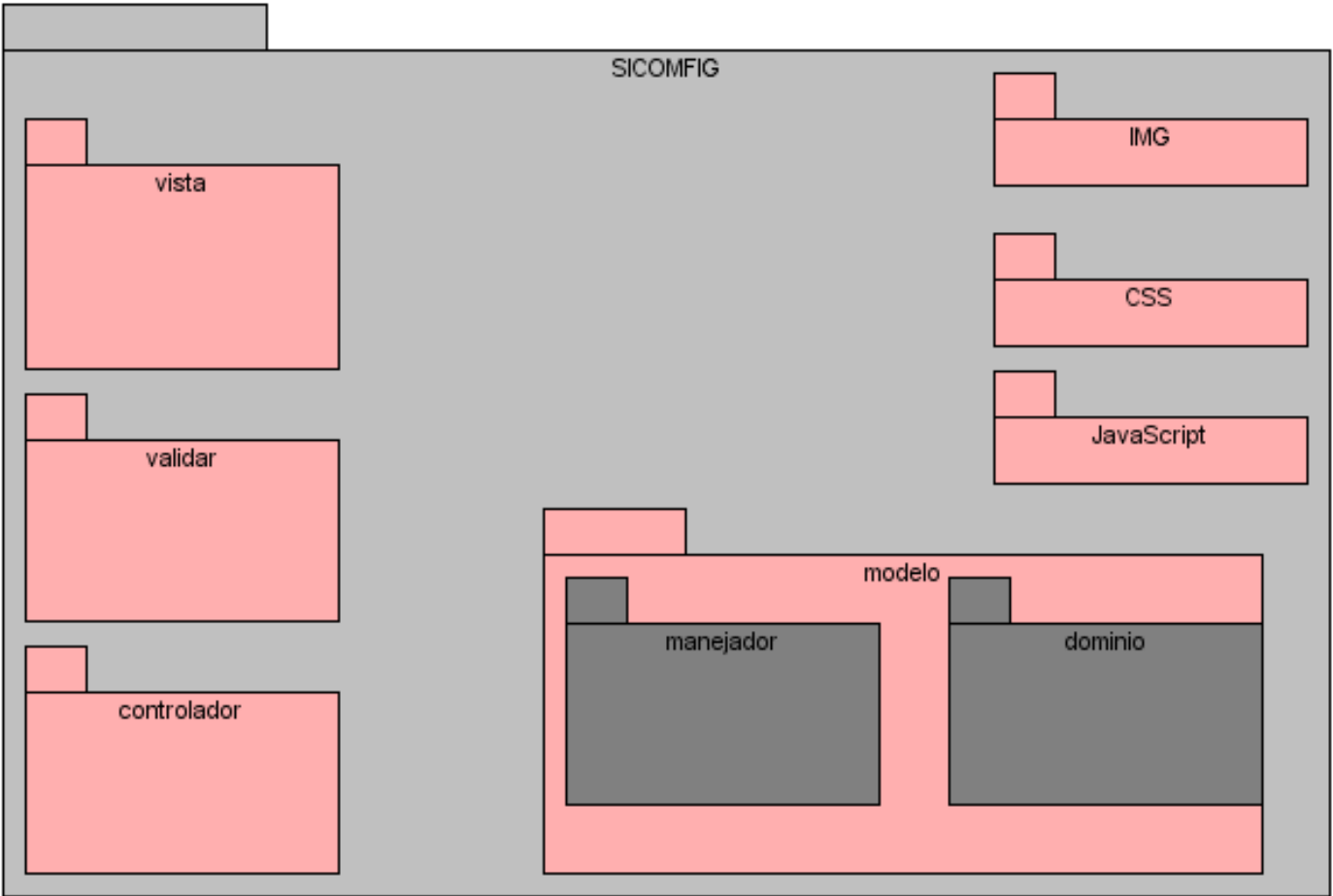


Figura 268: Diagrama de Paquetes

2.1.2.2.3.4 DIAGRAMA DE DESPLIEGUE

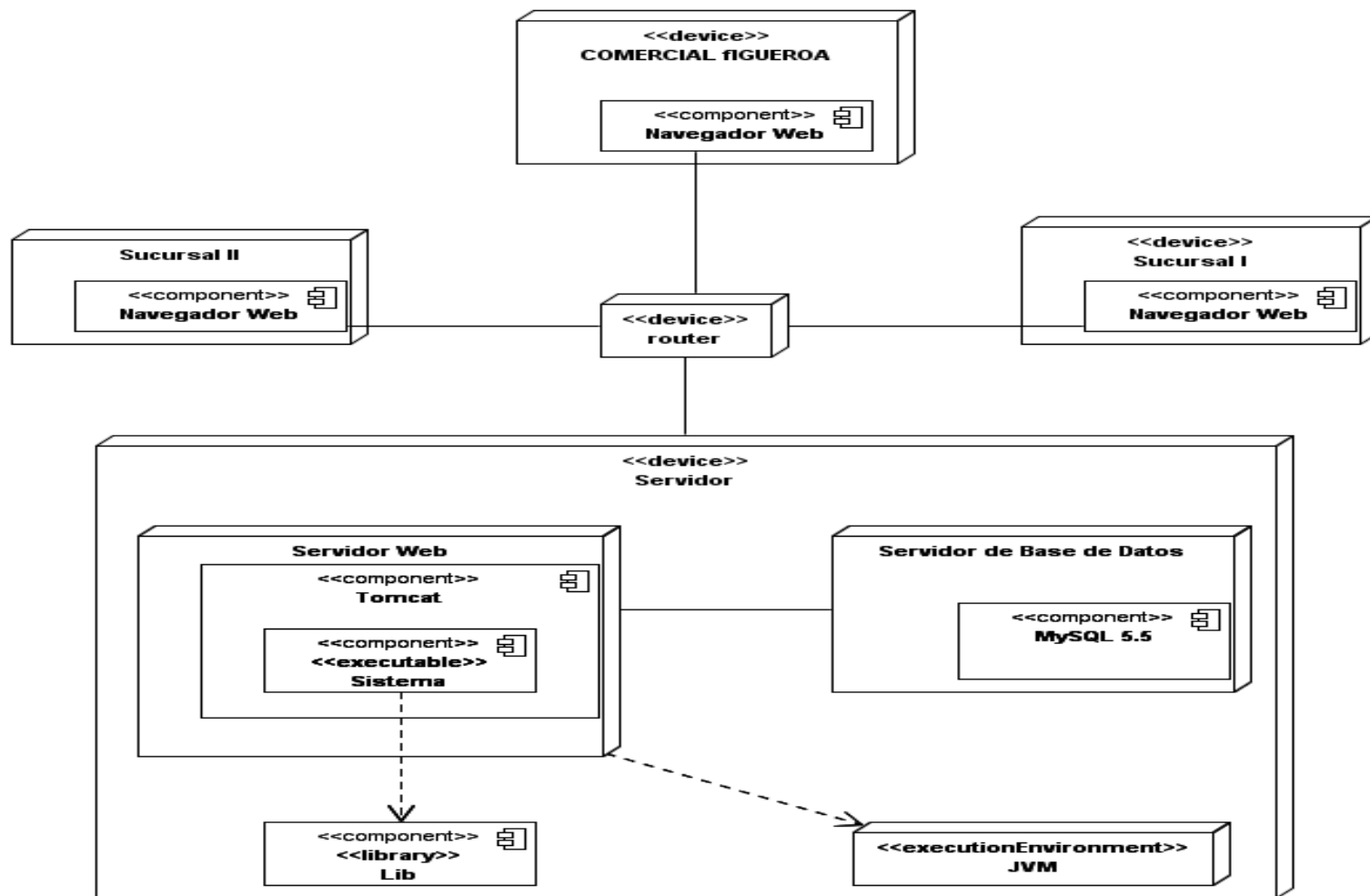


Figura 269: Diagrama de Despliegue

2.1.2.2.3.5 CASOS DE PRUEBA

El desarrollo de Sistemas de software implica la realización de una serie de actividades predispuestas a cometer errores (en la etapa de Requerimientos, de diseño, de desarrollo,...).

Debido a que estos errores se deben a nuestra habilidad natural de provocar errores, es necesario incorporar una actividad que garantice la calidad del Software.

Objetivos de las pruebas:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces
- Una prueba tiene éxito si descubre un error no detectado hasta entonces

La prueba no puede asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software.

2.1.2.2.3.5.1 PRUEBAS DE CAJA NEGRA

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software.

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa.

Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Las pruebas de caja negra pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.

Errores de inicialización y de terminación.

Tipo de Prueba de Caja Negra

- Prueba de partición equivalente
- Prueba de análisis de valores límites

Utilizaremos el método de **partición equivalente**.

2.1.2.2.3.5.2 PRUEBA DE PARTICIÓN EQUIVALENTE

Interfaz: Ingreso a sistema

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Usuario	Rango Lógico	1. Cualquier valor alfanumérico mayor a 5 caracteres y menor a 10	2. Usuario < 5 caracteres 3. Usuario > 10 caracteres 4. blanco
Clave	Rango Lógico	5. Cualquier valor alfanumérico mayor a 5 caracteres y menor a 10	6. Clave < 5 caracteres 7. Clave > 10 caracteres 8. blanco

Casos de prueba:

CP1
Condición de Entrada: Usuario

a) Fernando	b) 122222	c) 123fernando12345678910
Clase Equivalencia Válida		Clase de Equivalencia No validas
a y b cumple: 1		c cumple: 3

CP2		
Condición de Entrada: Clave		
a) Mar123/	b) Maria04872	c)
Clase Equivalencia Válida		Clase de Equivalencia No validas
a, b cumple: 1		c cumple: 4

Interfaz: Adicionar Empleado

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
CI	Rango Lógico	1. 7 > cadena < 10 caracteres	2. cadena < 5 caracteres 3. cadena > 45 caracteres 4. blanco
Nombre	Rango Lógico	5. 3 > cadena < 30 caracteres	6. cadena < 1 caracteres 7. cadena > 30 caracteres 8. blanco

Ap. Paterno	Rango Lógico	9. 0 > cadena < 30 caracteres 10. blanco	11. cadena > 30 caracteres
Ap. Materno	Rango Lógico	12. 5 > cadena < 30 caracteres	13. cadena <5 caracteres 14. cadena > 30 caracteres 15. blanco
Fecha de Nacimiento	Valor Lógico	16. formato “yyyy-mm-dd”	17. formato “yy/mm/dd” 18. formato (mm-dd- yyyy) 19. blanco
Teléfono	Rango Lógico	20. 7> numeral<15	21. numeral <7 números 22. numeral > 15 números 23. cadena 24. blanco
Sexo	Conjunto de valores	25. “Masculino” “Femenino”	26. Cadena distinta a las válidas
Dirección Actual	Rango	27. 10 < cadena > 30	28. cadena< 10 caracteres 29. cadena > 30 caracteres 30. blanco
Descripción	Lógico	31. cadena < 100 caracteres	32. Cadena > 100 caracteres 33. blanco 34. imagen

Cargo	Conjunto de valores	35. “Vendedor(a)” “Gerente General” “Secretario(a)” “Cajero(a)”	36. Cadena distinta a las válidas
Fecha de registro	Valor	37. formato “yyyy-mm-dd”	38. formato “mm/dd/yyyy” 39. formato (yyyy/ mm/dd)
Observaciones	Lógico	40. cadena < 100 caracteres	41. cadena > 100 caracteres 42. blanco 43. imagen

CP1		
Condición de Entrada: CI		
a) 5047061	b) 12	c) Kkkkkkkkkkkkkkkk123456987123456987145625855554
Clase Equivalencia Válida		Clase de Equivalencia No validas
a cumple: 1		b cumple: 2 c cumple: 3

CP2
Condición de Entrada: Nombre

a) María	b) Kkkkkkkkkkkkkkkk123456987123456987145625855 554	c)
Clase Equivalencia Válida		Clase de Equivalencia No validas
a cumple: 6	b cumple: 7 c cumple: 8	

CP3		
Condición de Entrada: Ap. Paterno		
a)	b) Castillo	c) aaaaaaaaaaaaaaasd12345698712345698714562585554
Clase Equivalencia Válida		Clase de Equivalencia No validas
a cumple 10 b cumple: 9	c cumple: 11	

CP4		
Condición de Entrada: Ap. Materno		
a)	b) Soto	c) Cstelleasdfs sd fsafdfasdf dfasdfasdfasfsfsdfsd ffd
d) Castellanos	e) Sdfasd//*/*	f)
Clase Equivalencia Válida		Clase de Equivalencia No validas

b, d y e cumple: 10	c cumple: 14 a y f cumple: 15

CP5			
Condición de Entrada: Fecha de Nacimiento			
a) 28-04-1987	b) 1987-12-12	c) 2000/10/02	d)
Clase Equivalencia Válida		Clase de Equivalencia No validas	
b cumple: 16		a cumple: 17 c cumple: 18 d cumple: 19	

CP6			
Condición de Entrada: Teléfono			
a) 12	b) 7239544	c) 3215458777897877797878779787797	
d) ldkdssdvssdbgdsgf	e)		
Clase Equivalencia Válida		Clase de Equivalencia No validas	
b cumple: 20		c cumple: 21 d cumple: 22	

	e cumple: 23

CP7		
Condición de Entrada: Sexo		
a)	b) Masculino	c) Femenino
Clase Equivalencia Válida		Clase de Equivalencia No validas
b y c cumple: 25		a cumple: 26

CP8		
Condición de Entrada: Cargo		
a) Cajero(a)	b) Secretario(a)	c)
Clase Equivalencia Válida		Clase de Equivalencia No validas
a y b cumple: 35		c cumple: 36

CP9		
Condición de Entrada: Fecha de registro		
a) 12/03/2010	b) 2000-10-05	c) 1987-12-25

Clase Equivalencia Válida	Clase de Equivalencia No validas
c cumple: 37	a cumple: 38 b cumple: 39

CP10		
Condición de Entrada: Observaciones		
a) fjf	b) Este empleado es temporal.	c)
Clase Equivalencia Válida	Clase de Equivalencia No validas	
a y b cumple: 40	a cumple: c cumple: 42	

Interfaz: Modificar Cargo

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Nombre	Valor	1. 5 < cadena > 30	2. Cadena < 5 caracteres

			3. Cadena > 30 caracteres
Carga Horaria	Valor	4. 5 < cadena > 20	5. Cadena < 5 caracteres 6. Cadena > 20 caracteres
Estado	Conjunto de valores	7. “Activo”, “Inactivo”, “Reemplazo”	8. Cadena distinta a las validas
Cargo	Valor	9. un cargo seleccionado	10. cargo no seleccionado 11. varios cargos seleccionados

CP1		
Condición de Entrada: Nombre		
a) ab	b) Vendedor(a)	c) wwwwwwwwwwwwjjjjjjjjjjjjjjjjjjjjioooooooooobbbbb
Clase Equivalencia Válida	Clase de Equivalencia No validas	
b cumple: 1	a cumple: 2 c cumple: 3	

CP2		
Condición de Entrada: Carga Horaria		
a) ab	b) de 8:00 a 12:00	c) 212121sddddddddddddddgggg*****
Clase Equivalencia Válida	Clase de Equivalencia No validas	

b cumple: 4	a cumple: 5 c cumple: 6

CP3		
Condición de Entrada: Estado		
a) Activo	b) Inactivo	c)
Clase Equivalencia Válida		Clase de Equivalencia No validas
a y b cumple: 7		c cumple: 8

CP4		
Condición de Entrada: Cargo		
a) seleccionado	b) varios seleccionados	c) No seleccionado
Clase Equivalencia Válida		Clase de Equivalencia No validas
a cumple: 9		b cumple: 10 c cumple: 11

Interfaz: Adicionar Proveedor

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
NIT	Valor	1. 9 < cadena > 15	2. Cadena < 9 caracteres 3. Cadena > 15 caracteres 4. blanco
Nombre	Valor	5. 5 < cadena > 50	6. Cadena < 5 caracteres 7. Cadena > 50 caracteres 8. blanco
Correo Electrónico	Valor	9. Formato “.....@...com”	10. Formato “@.....com” 11. Formato “.....@”
Cuenta bancaria	Valor	12. Un número entero 13. Blanco	14. Un número decimal
Ciudad	Valor	15. 5< cadena >15	16. Cadena < 5 caracteres 17. Cadena > 15 caracteres
Dirección	Valor	18. 5 < cadena > 100	19. Cadena < 5 caracteres 20. Cadena > 100 caracteres 21. Blanco
Observaciones	Lógico	22. Cadena < 100	23. Cadena > 100 caracteres 24. Una imagen

			25. Blanco

CP1		
Condición de Entrada: NIT		
a) 5047061	b) Ab5567	c)
Clase Equivalencia Válida	Clase de Equivalencia No validas	
a cumple: 1	b cumple: 2 c cumple: 4	

CP2		
Condición de Entrada: Nombre		
a) Liliana	b) Ab	c)
Clase Equivalencia Válida	Clase de Equivalencia No validas	
a cumple: 5	b cumple: 6 c cumple: 8	

CP3

Condición de Entrada: correo electrónico		
a) @Liliana.com	b) grys_@hotmail.com	c) Liliana@
Clase Equivalencia Válida	Clase de Equivalencia No validas	
a cumple: 10	b cumple: 9 c cumple: 11	

CP4		
Condición de Entrada: Cuenta bancaria		
a) 1235687	b)	c) 235,12587
Clase Equivalencia Válida	Clase de Equivalencia No validas	
a cumple: 12 b cumple: 13	c cumple: 14	

CP5		
Condición de Entrada: Ciudad		
a) lldvjdavjjdvkjdbgfgjjjkxkhkk	b) Santa Cruz	c) bj
Clase Equivalencia Válida	Clase de Equivalencia No validas	

b cumple: 15	c cumple: 16 a cumple: 17

CP6		
Condición de Entrada: Dirección		
a) c/ La Paz número 325	b) yt	c)
Clase Equivalencia Válida	Clase de Equivalencia No validas	
a cumple: 18	b cumple: 19 c cumple: 21	

CP7	
Condición de Entrada: Observaciones	
a) Mini Empresa	b)
Clase Equivalencia Válida	Clase de Equivalencia No validas
a cumple:22	b cumple: 25

2.1.2.2.3.6 PLAN DE ITERACIÓN

FASES	Objetivos de cada Iteración	Fecha Inicio	Fecha Fin
INICIO (I ₁)	Especificación de requerimientos (norma 830).	13-03-2014	30-03-2014
	Realización de Casos de uso del Negocio.	01-04-2014	05-04-2014
	Modelado de Procesos del Negocio.	06-04-2014	10-04-2014
	Realización de Casos de Uso	11-04-2014	25-04-2014
	Especificación de Casos de Uso.	26-04-2014	30-04-2014
ELABORACION (I ₂ Y I ₃)	Análisis de Riesgos	01-05-2014	03-05-2014
	Completar Diagramas de Casos de Uso.	04-05-2014	20-05-2014
	Documento Visión	21-05-2014	23-05-2014
	Modelo de Análisis y diseño		
	- Diagramas de Actividad	24-05-2014	03-06-2014
	- Diagramas de Secuencia	04-05-2014	12-06-2014
	- Diagramas de Colaboración.	13-05-2014	16-06-2014
	- Modelado de datos preliminar.	17-06-2014	19-06-2014
	Especificaciones Adicionales	20-06-2014	23-06-2014
	Diseñar el prototipo de interfaces de usuario	24-06-2014	25-07-2014
	Revisar y terminar modelado de datos.	26-07-2014	27-07-2014

CONSTRUCCION (I4 Y I5)	Terminar Elaboración de Prototipos	28-07-2014	05-08-2014
	Programación de software	06-08-2014	30-10-2014
	Realizar el modelo de implementación		
	- Diagrama de Componentes	01-11-2014	02-11-2014
	- Diagrama de paquetes	04-11-2014	05-11-2014
	- Diagrama de Despliegue	06-11-2014	08-11-2014
	Realizar los casos de prueba (Caja Negra)	09-11-2014	11-11-2014
	• Realizar los manuales de Instalación y Usuario	12-11-2014	20-11-2014

Tabla 58: Plan de Iteración

2.1.2.2.3.7EVALUACIÓN DE ITERACIÓN

Fase de inicio

Primera iteración (I1): Se logro alcanzas y Culminar con los objetivos planeados para esta iteración con algunos correcciones en la especificación de requisitos y casos de uso. Esta Iteración fue revisada y validada.

Fase de elaboración

Segunda iteración (I2): Se completo y revisó de casos de uso y la realización del análisis y diseño con toda normalidad sin ningún contratiempo a excepto de algunas correcciones en el diagrama de secuencia. Esta Iteración fue revisada y validada.

Tercera iteración (**I3**): Se presento algunos inconvenientes en el diseño de prototipos que posteriormente fueron corregidos, revisión y corrección en el modelado de datos (diagrama de clases). Iteración revisada y validada.

Fase de construcción

Cuarta iteración (**I4**): Algunos dificultades en la programación especialmente en la elaboración de reportes por lo cual se tuvo falta de tiempo en esta iteración por lo cual se tuvo que realizar un reajuste en la siguiente Iteración revisada y validada.

Quinta iteración (**I5**): Iteración terminada con toda normalidad sin mayores inconvenientes. Iteración revisada y validada.

2.1.2.2.3.8 LISTA DE RIESGOS

Riesgos del proyecto

Valoración de Riesgos

0% - 10%	10% - 25%	25% - 50%	50% - 75%	75% - 100%
Muy bajo	Bajo	Moderado	Alto	Muy alto.

Identificación de Riesgos		Análisis de Riesgo		Planificación de Riesgos		
Nº	Riesgo	Ponderación	Efecto	Estrategias		Plan de contingencia
				Anulación	Disminución	
1	Incumplimiento del cronograma de actividades	25%	Bajo	Establecer fechas de control interno y	Concientización de responsabilidad y	Asumir la responsabilidad

	por diferentes motivos que se pueden presentar en el desarrollo del sistema.			verificación de cumplimiento de realización de artefactos de cada iteración.	plazos de cumplimiento.	del cumplimiento del cronograma de actividades. Trabajo de horas extras.
2	Enfermedad	25%	Bajo	-	-	Reorganización del cronograma de actividad y redistribución de tareas más eficiente.
3	Falta de experiencia con las herramientas de desarrollo	25%	Bajo	Realizar un estudio de herramientas alternativas con similares características.	Asesoramiento con personas que tenga experiencia en las mismas y apoyo con tutoriales en el internet.	Aplicar o hacer uso de herramientas con similares características.
4	Posible Falla en algún dispositivo de almacenamiento físico al resguardar de documentación del proyecto	10%	Muy Bajo	Realizar respaldos y copias de seguridad en dispositivos del almacenamiento externo.	Realizar respaldos constantes.	Acudir a software de recuperación de datos.
5	Las Interfaces y el diseño de las pantallas no sean correctos.	10%	Muy Bajo	Establecer diseños alternativos.	Realizar diseño de interfaces con la aprobación del	Aplicar los diseños alternativos.

					cliente	
6	Cliente no satisfecho con el software desarrollado.	10%	Muy Bajo	Establecer un seguimiento y control con el cliente en el desarrollo del software	En cada iteración hacer un análisis meticuloso para verificar el cumplimiento de los requisitos del proyecto.	Presentación continúa de prototipos y funcionalidades del software al cliente.

2.1.2.3 ORGANIZACIÓN DEL PROYECTO

2.1.2.3.1 PARTICIPANTES DEL PROYECTO

Jefe de proyecto

Universitaria Griselda Gallardo Aparicio con cualidades de responsabilidad, capacidad en organizar y dirigir grupos de Investigación, y está capacitado para llevar a cabo la planificación, desarrollo y control del proyecto.

Analista de sistemas

Griselda Gallardo Aparicio

Conocimientos amplios en UML, Facilidad de relacionarse con su entorno social y desenvolverse a la hora de realizar entrevistas exponiendo claras ideas y soluciones posibles a los problemas, tiene una capacidad de capturar la información de manera concisa y exacta.

Analista – Programador

Griselda Gallardo Aparicio

Conocimientos básico de UML y experiencia en el lenguaje de programación java, java script, entorno web, en la tecnologías hibernate, spring y en la arquitectura modelo vista controlador (MVC).

Ingeniero de Software

Griselda Gallardo Aparicio

Capacidad de crítica constructiva, conocimiento de pruebas de software tanto de caja negra y blanca.

2.1.2.3.2 INTERFACES EXTERNAS.

El sistema cuenta con una interfaz diseñada en base a módulos para facilitar la organización del sistema, de esta manera facilitar el uso mismo, los módulos son: Administrar Personal, Ventas, compras, empleados, reportes, datos de tipos de venta y almacenes. Todas estas funcionalidades del sistema requieren de una validación de seguridad debido a que se manejan principalmente datos confidenciales de los clientes del “COMERCIAL FIGUEROA”

2.1.2.3.3 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. Cabe recalcar que el proyecto es desarrollado por una persona, al cual se le atribuye todas las responsabilidades.

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.

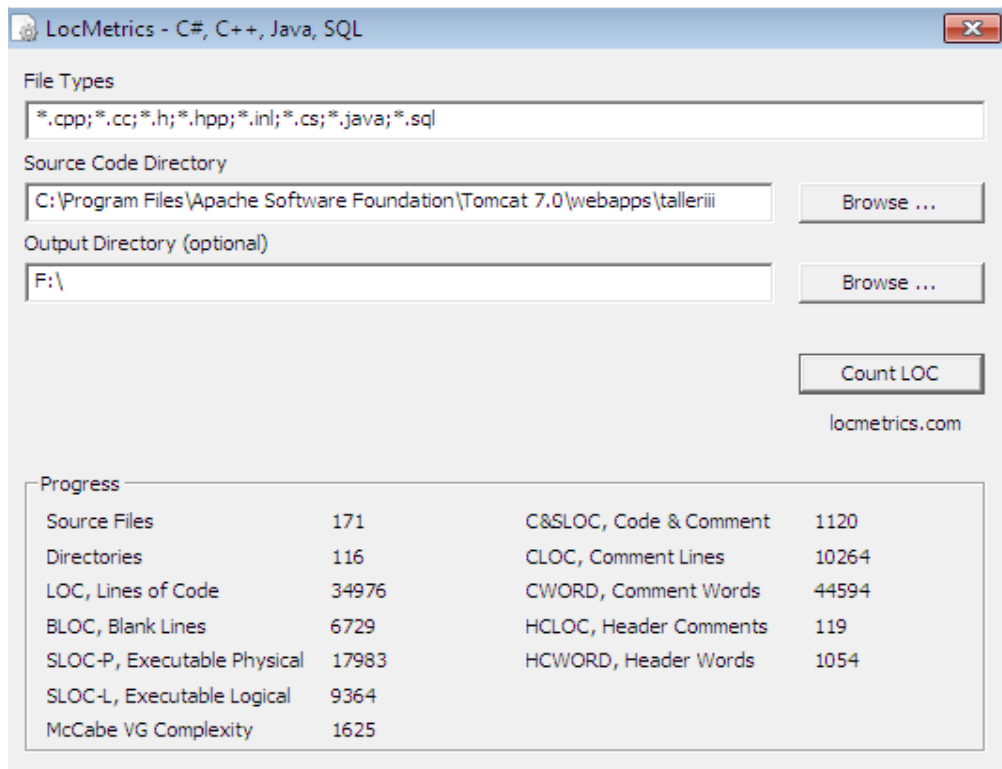
2.1.2.4 GESTIÓN DEL PROCESO

2.1.2.4.1 ESTIMACIONES DEL PROYECTO

2.1.2.4.1.1 BASADO EN LAS LÍNEAS DE CÓDIGO COCOMO BÁSICO

El tamaño del software varía de unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles de líneas (tamaño medio).

Para la estimación del esfuerzo y el tiempo de desarrollo utilizaremos el método de cócono básico para ello se tiene los siguientes datos:



LocMetrics - C#, C++, Java, SQL

File Types
.cpp;.cc;*.h;*.hpp;*.inl;*.cs;*.java;*.sql

Source Code Directory
C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\talleriii Browse ...

Output Directory (optional)
F:\ Browse ...

Count LOC

locmetrics.com

Progress

Source Files	171	C&SLOC, Code & Comment	1120
Directories	116	CLOC, Comment Lines	10264
LOC, Lines of Code	34976	CWORD, Comment Words	44594
BLOC, Blank Lines	6729	HCLOC, Header Comments	119
SLOC-P, Executable Physical	17983	HCWORD, Header Words	1054
SLOC-L, Executable Logical	9364		
McCabe VG Complexity	1625		

Figura 270: Número de Líneas de Código según LocMetrics

LDC=34976 Líneas de Código que se tiene en el proyecto (aproximadamente).

Se utilizan dos ecuaciones para determinar el esfuerzo de personal y el tiempo de desarrollo. El coste es:

Ecuación (1) $P_m = 2.4 S_k^{1.05}$

Donde:

P_M se expresa en personas-mes

S_K es el tamaño expresado en miles de líneas de código fuente.

El tiempo de desarrollo se da por:

Ecuación (2) $t_d = 2.5 P_m^{0.38}$

Donde **P_m** se obtiene de la ecuación anterior y **td** es el tiempo de desarrollo en meses.

El N° de Personas Medio se da por:

ECUACIÓN (3) $P_E = P_M / T_D$

Reemplazando datos en la ecuación (1) se tiene:

$P_M = 2.4 * (34976)^{1.05} = 45$ PERSONAS MES.

Reemplazando datos en la ecuación (2)

$T_D = 2.5 * (45)^{0.38} = 10$ MESES

$P_E = 45 / 10 = 5$ Se requiere 5 personas promedio (Aproximadamente).

Según estas cifras será necesario un equipo de 5 personas trabajando alrededor de **10** meses.

Suponiendo que cada persona tiene un sueldo de **USD 400.00** el costo de desarrollo del sistema tendría: $5 * 400.00 = 2000.00$ COSTO MENSUAL.

El costo Total: $2000.00 * 9 = \text{USD } 18000.00$

2.1.2.4.2 PLAN DEL PROYECTO

Organización en fases e iteraciones del proyecto.

2.1.2.4.2.1 PLAN DE FASES

El desarrollo se llevara a cabo según las 3 primeras fases de RUP.

FASE	ITERACIONES	DURACIÓN (SEMANAS)
INICIO	I₁	5
ELABORACIÓN	I₂, I₃	6
CONSTRUCCIÓN	I₄, I₅	20

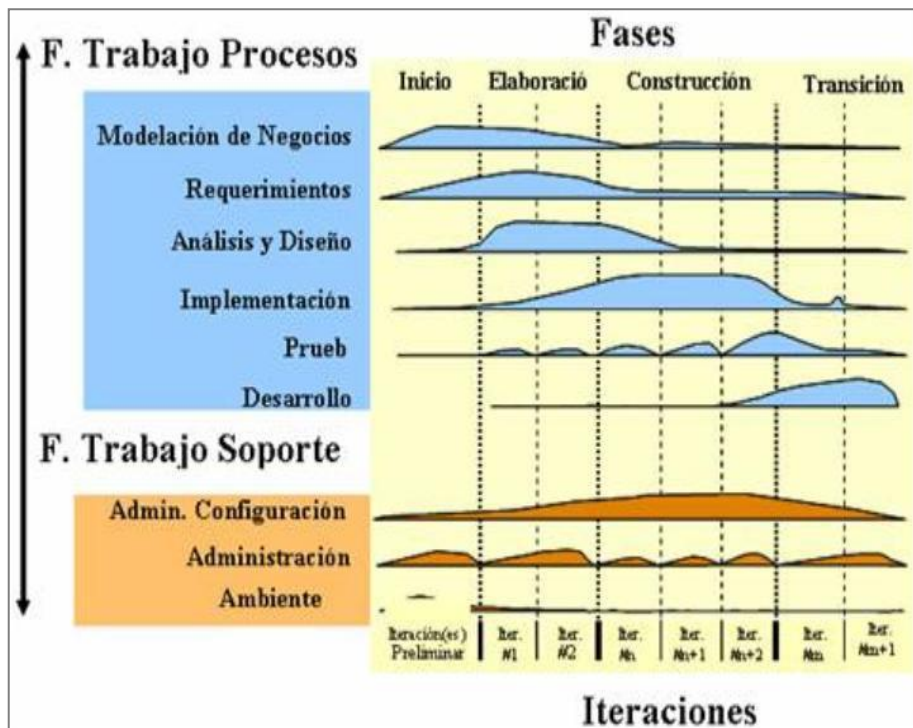
Los hitos que marcan el final de cada fase, se ven en la siguiente tabla.

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 reléase de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). 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 3 semanas.
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 3 iteraciones, cada una produciendo una reléase 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. Con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.

Tabla 59: Descripción de Hitos

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.



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.

Disciplinas / Artefactos generados o modificados durante las distintas fases	Comienzo	Aprobación
Especificación de requisitos de software (norma 830)	11-03-2014	15-04-2014
Modelo de Casos de Uso del Negocio y Modelo de Procesos del Negocio	11-03-2014	15-04-2014
Requisitos		
Glosario	20-08-2014	30-08-2014
Visión	01-08-2014	20-08-2014
Modelo de Casos de Uso	10-04-2014	15-05-2014
Especificación de Casos de Uso	10-05-2014	20-05-2014
Especificaciones Adicionales	20-05-2014	05-06-2014
Análisis/Diseño		
Modelo de Análisis/Diseño	16-05-2014	30-08-2014
Modelo de Datos		
Implementación	25-06-2014	01-11-2014
Prototipos de Interfaces de Usuario	20-04-2014	20-11-2014
Modelo de Implementación		
Pruebas de Pruebas Funcionales	12-11-2014	25-11-2014
Despliegue		
Modelo de Despliegue	10-11-2014	15-11-2014
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto	20-11-2014	30-11-2014
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	Durante todo el proyecto	
Ambiente	Durante todo el proyecto	

2.1.2.4.3 SEGUIMIENTO Y CONTROL DEL PROYECTO

2.1.2.4.3.1 GESTIÓN DE REQUISITOS

Ver anexo A (Especificación de requerimientos de Software **Norma 830**)

2.1.2.4.3.2 CONTROL DE PLAZOS

El calendario del proyecto, tendrá un seguimiento y evaluación continua por el jefe del mismo, y el control se llevara a cabo, de acuerdo las indicaciones del docente de taller III.

Fase	N° de Iteración	Inicio	Fin
Inicio	Primera	11/03/14	30/04/14
Elaboración	Segunda	01/05/14	19/06/14
	Tercera	20/06/14	03/07/14
Construcción	Cuarta	04/07/14	20/09/14
	Quinta	21/09/14	02/11/14

2.1.2.4.3.3 CONTROL DE CALIDAD

A diferencia de otras disciplinas, la ingeniería de software no está basada en leyes cuantitativas básicas, en su lugar, se hace uso de un conjunto de medidas indirectas conocidas como métricas, las cuales proporcionan una referencia de la calidad de algún producto software.

Para valorar la calidad de los productos software o los sistemas que se desarrolla, se proporciona la información adecuada sobre los datos necesarios referentes a la calidad del producto, permitiendo una visión profunda sobre el cumplimiento de los objetivos del proyecto.

Entre las varias técnicas para medir la calidad de software se encuentra el estándar ISO 9126, que identifica los siguientes atributos clave de calidad para el software: Funcionalidad, Confiabilidad, Facilidad de Mantenimiento, Usabilidad y Portabilidad.

- **Funcionalidad**

Cuantificamos el tamaño y la complejidad del sistema en términos de las funciones de usuario, valorándolo mediante una medida de funcionalidad denominada punto función.

Determinaremos las cinco características según Albretch, teniendo en cuenta las diferentes funciones que el software deberá de realizar.

Entradas de Usuario: Representamos las entradas de control del usuario que proporciona diferentes datos a la aplicación.

Salidas de Usuario: Representa cada salida de información referente a la aplicación (informes, pantallas, mensajes de error).

Consultas de Usuario: Representa a cada combinación única existente de entrada salida, donde una entrada genera una salida.

Archivos: Representa a cada archivo parte de la base de datos o a un archivo independiente.

Interfaces Externas: Interfaces legibles por el ordenador, que son utilizados para transmitir la información a otro sistema.

La siguiente tabla muestra las cinco características con factor de ponderación medio para el cálculo del punto función.

Parámetro	Cuenta	Factor de Ponderación			Total
		Simple	Medio	Complejo	
Número de entradas de usuario	80	3	4	6	320
Número de salidas de usuario	70	4	5	7	350
Número de peticiones de usuario	5	3	4	6	20
Número de archivos	0	7	10	15	0
Número de interfaces	0	5	7	10	0

externas					
Cuenta Total					690

Tabla 60: Coeficientes de Ajustes de la complejidad de las funciones

Seguidamente, es necesario calcular la complejidad del proyecto, con respecto a **14** características que pueden o no estar presentes en el mismo, y que se valoran de 0 a, dependiendo de su influencia en el proyecto. Un valor cero indica que la característica no está presente o no tiene influencia y un valor 5 indica una fuerte influencia.

Valores de Ajuste de complejidad:

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

Nº	Parámetros a Evaluar	Valor
1	Comunicación de datos	5
2	Funciones distribuidas	0
3	Rendimiento	4
4	Ejecución del sistema en un entorno operativo utilitario	4
5	Transacciones de datos sobre múltiples entradas	0
6	Entrada de datos	4
7	Copia de seguridad y recuperación de datos fiable	5
8	Actualización de archivos en forma interactiva	1
9	Complejidad de procesamiento interno	3
10	Reusabilidad de código	4
11	Facilidad de instalación	2
12	Facilidad operacional	5
13	Soporte de múltiples instalaciones	3

14	Facilidad de cambio y manejo	3
FACTOR DE AJUSTE DE COMPLEJIDAD (ΣF_i)		43

Tabla 61: Valores de ajuste de complejidad

Calculamos el Punto Función con la siguiente relación:

$$PF = \text{CuentaTotal (tabla 213)} * (0.65 + 0.01 * \Sigma Fi)$$

$$PF=690* (0.65 + 0.01 *43)$$

$$\mathbf{PF= 745}$$

Hallamos el Punto Función máximo para comparar los valores de funcionalidad del sistema:

$$PF_{\text{MAX}} = 745 * (0.65 + 0.01 * 70)$$

$$\mathbf{PF_{\text{MAX}}=1005}$$

La funcionalidad real entonces del sistema es:

$$\text{Funcionalidad} = \frac{745}{1005} * 100\% = 74 \%$$

Por lo tanto el sistema de Optimizar el control y administración de la información del “COMERCIAL FIGUEROA” tiene una capacidad del 74% para proporcionar funcionalidades que satisfagan las necesidades específicas.

- **Confiabilidad**

Es la probabilidad de operación libre de fallos de un programa en un entorno determinado y durante un tiempo específico, en este caso el sistema no realizó la implementación del mismo por lo cual no se pudo realizar una estimación exacta su confiabilidad.

- **Mantenibilidad**

El índice de madurez de software (IMS) proporciona una indicación de la estabilidad de un producto de software y nos sirve de métrica de la calidad del mantenimiento del sistema.

Calculamos el índice de Madurez de software con la siguiente relación:

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

Donde:

Mt: número de módulos en la versión actual.

Fc: número de módulos en la versión actual que se han modificado.

Fa: número de módulos en la versión actual que se han añadido.

Fe: número de módulos en la versión anterior que se han eliminado en la versión actual.

Por lo tanto se tiene que el índice de madurez de software es:

$$IMS = [6 - (2 + 0 + 0)] / 6$$

$$IMS = 0.67$$

$$IMS = 67\%$$

Por lo tanto el sistema empieza a estabilizarse en un 67%.

- **Usabilidad**

Usabilidad es el esfuerzo necesario para aprender a utilizar el sistema. El sistema cuenta con una interfaz amigable e intuitiva lo cual hace fácil su utilización.

La usabilidad se determina calculando el porcentaje de una encuesta elaborada a un cierto número de usuarios finales que hacen uso del sistema, como el sistema no se implemento no se pudo hacer dicha encuesta y realizar una medición exacta.

2.1.3 MEDIOS DE VERIFICACIÓN DEL COMPONENTE DEL PROYECTO “OPTIMIZAR EL CONTROL Y ADMINISTRACIÓN DE LA INFORMACIÓN DEL COMERCIAL FIGUEROA”

II. COMPONENTE 2: CAPACITACIÓN SOBRE LA IMPORTANCIA DE LAS TIC Y EL USO DEL SISTEMA

2.2.1 MARCO TEÓRICO

2.2.1.1 MÉTODOS Y TÉCNICAS DE ENSEÑANZA

2.2.1.1.1 INTRODUCCIÓN

Métodos y técnicas que enseñanza: constituyen recursos necesarios para la enseñanza; son los vehículos de realización ordenada, metódica y adecuada de la misma. Los métodos y técnicas tienen por objeto hacer más eficiente la dirección del aprendizaje.

2.2.1.1.2 PRINCIPIO DE LAS TÉCNICAS Y MÉTODOS

- **Principio de proximidad:** Integrar la enseñanza lo más cerca posible en la vida cotidiana del educando.
- **Principio de dirección:** Tornar claros y precisos los objetivos a alcanzar.
- **Principio de marcha propia y continúa:** Procura respetar las diferencias individuales, no exigiendo la misma realización de todos los educandos.
- **Principio de ordenamiento:** Con el establecimiento de un orden se busca facilitar la tarea de aprendizaje.
- **Principio de adecuación:** Es necesario que las tareas y objetivos de la enseñanza sean acordes con las necesidades del educando.
- **Principio de eficiencia:** El ideal: mínimo esfuerzo máxima eficiencia en el aprendizaje.
- **Principio de realidad psicológica:** Previene que no se debe perder de vista la edad evolutiva de los alumnos, así como tampoco sus diferencias individuales.
- **Principio de dificultad o esfuerzo:** Es preciso tener el cuidado de no colocar al educando ante situaciones de las que tenga Posibilidades de salir bien. Pues el fracaso continuado es peor veneno para la criatura humana.
- **Principio de participación:** El educando es parte activa y dinámica del proceso.
- **Principio de espontaneidad:** Cualquier proceso emprendido debe favorecer las manifestaciones naturales del educando.
- **Principio de transparencia:** El conocimiento aprendido debe replicarse en otras situaciones de la vida diaria.
- **Principio de evaluación:** Con un proceso continuo de evaluación, el docente podrá identificar a tiempo dificultades en el proceso de aprendizaje.
- **Principio reflexión:** Inducir al pensamiento reflexivo en el alumno como parte integral de actuar del ser humano.
- **Principio de responsabilidad:** Encaminar todo el proceso de enseñanza de modo que el

educando madure en cuanto a comportamiento responsable.

2.2.1.1.3 DIRECTIVAS DIDÁCTICAS

Es el conjunto de recomendaciones que el profesor o docente debe tener en cuenta siempre que trabaje con un grupo de alumnos.

Dentro de estas directivas didácticas están: tener en cuenta las ideas de los alumnos, incentivar la expresión libre y los debates al interior del grupo, cultivar la confianza con los alumnos dentro y fuera de la clase, manejo de los ritmos de clase y estar atento a la fatiga de los alumnos, crear ambiente agradable en clase, ser puntuales con la clase, atender con eficiencia las inquietudes estudiantiles, buscar la comunicación adecuada con los alumnos, manejar con sabiduría tanto a los alumnos mediocres como a los llamados adelantados, etc.

2.2.1.1.4 MÉTODOS DE ENSEÑANZA

2.2.1.1.4.1 MÉTODOS DE ENSEÑANZA INDIVIDUALIZADA

Su objetivo máximo es ofrecer oportunidades de desenvolvimiento individual más eficiente, y llevar al educando a un completo desarrollo de sus posibilidades personales.

- **Método de Proyectos:** Tiene la finalidad de llevar al alumno a realizar algo. Es un método esencialmente activo, cuyo propósito es hacer que el alumno realice, actúe.
- **Plan Dalton:** Se basa en la actividad, individualidad y libertad, y su objetivo principal consiste en desenvolver la vida intelectual. Cultiva también la iniciativa toda vez que deja al alumno la oportunidad de escoger los trabajos y los momentos de realizarlos. Dos de sus principales inconvenientes son: acentúa exageradamente la individualidad y su carácter es esencialmente intelectual. Otras particularidades de este método son: Conferencias, Boletines Murales y Hoja de Tareas.
- **Técnica Winnetka:** Procura conjugar las ventajas del trabajo individualizado con las del trabajo colectivo, Contiene medidas que permiten al alumno estudiar solo y controlarse a sí mismo. Al finalizar la unidad el alumno es sometido a un test de control y de acuerdo a los resultados continuará

adelantando en los estudios o hará estudios suplementarios para vencer las deficiencias comprobadas.

- **Enseñanza por Unidades:** Las fases del plan empiezan por la exploración, donde se buscan los conocimientos, luego una presentación donde se expone el tema, sigue la fase de asimilación, en la cual el alumno amplía su conocimiento y continua con una fase de organización, que es el trabajo de integración para culminar con una recitación, que es la presentación oral de la unidad.

2.2.1.1.4.2 MÉTODOS DE ENSEÑANZA SOCIALIZADA

Consiste en proporcionar trabajos en grupos e individuales procurando, también, atender a las preferencias de los educandos. Puede presentar dos modalidades:

- **Primera Modalidad:** Consiste en seis pasos: Presentación, Organización de Estudios, Estudio propiamente dicho, Discusión, Verificación del Aprendizaje e Individualización. Es aplicable sobre todo en los últimos años de la escuela primaria en secundaria.
- **Segunda Modalidad:** Comprende siete pasos que son los siguientes: Presentación Informal, Planeamiento, Estudio Sistemático, Presentación y Discusión, Elaboración Personal, Verificación del Aprendizaje e Individualización. Destinado sobre todo a los últimos años de colegio y a la enseñanza superior.

2.2.1.1.4.2.1 MÉTODOS

- **Método de la Discusión:** Consiste en orientar a la clase para que ella realice, en forma de cooperación intelectual, el estudio de una unidad o de un tema. Hace hincapié en la comprensión, la crítica y la cooperación.
- **Método de Asamblea:** Consiste en hacer que los alumnos estudien un tema y los discutan en clase, como si ésta fuese cuerpo colegiado gubernamental. Este método es más aplicable en el estudio de temas controvertidos o que pueden

provocar diferentes interpretaciones.

- **Método del Panel:** Consiste en la reunión de varias personas especialistas o bien informadas acerca de determinado asunto y que van a exponer sus ideas delante de un auditorio, de manera informal, patrocinando punto de vista divergentes, pero sin actitud polémica. El panel consta de un coordinador, los componentes del panel y el auditorio.

2.2.1.1.5 TÉCNICAS DE ENSEÑANZA

- **Técnica expositiva:** Consiste en la exposición oral, por parte del profesor; esta debe estimular la participación del alumno en los trabajos de la clase, requiere una buena motivación para atraer la atención de los educandos. Esta técnica favorece el desenvolvimiento del autodomínio, y el lenguaje.
- **Técnica del dictado:** Consiste en que el profesor hable pausadamente en tanto los alumnos van tomando nota de lo que él dice. Este constituye una marcada pérdida de tiempo, ya que mientras el alumno escribe no puede reflexionar sobre lo que registra en sus notas
- **Técnica biográfica:** Consiste en exponer los hechos o problemas a través del relato de las vidas que participan en ellos o que contribuyen para su estudio. Es más común en la historia, filosofía y la literatura.
- **Técnica exegetica:** Consiste en la lectura comentada de textos relacionados con el asunto en estudio, requiere la consulta de obras de autores. Su finalidad consiste en acostumbrar a leer las obras representativas de un autor, de un tema o una disciplina.
- **Técnica cronológica:** Esta técnica puede ser progresiva o regresiva-progresiva cuando los hechos Son abordados partiendo desde el pasado hasta llegar al presente. Regresiva cuando esos mismos hechos parten desde el presente en sentido inverso hacia el pasado.
- **Técnica de los círculos concéntricos:** Consiste en examinar diversas veces toda la esfera de un asunto o una disciplina y, en cada vez, ampliar y profundizar el estudio anterior.
- **Técnica de las efemérides:** Efemérides se refiere a hechos importantes, personalidades y

fechas significativas. Por tanto pequeños trabajos o investigaciones relativas a esas fechas pueden ayudar al aprendizaje.

- **Técnica del interrogatorio:** Uno de los mejores instrumentos del campo didáctico como auxiliar en la acción de educar, este permite conocer al alumno y resaltar sus aspectos positivos.
- **Técnica de la argumentación:** Forma de interrogatorio destinada a comprobar lo que el alumno debería saber. Requiere fundamentalmente de la participación del alumno.
- **Técnica del diálogo:** El gran objetivo del diálogo es el de orientar al alumno para que reflexione, piense y se convenza que puede investigar valiéndose del razonamiento.
- **Técnica catequística:** Consiste en la organización del asunto o tema de la lección, en forma de preguntas y las respectivas respuestas.
- **Técnica de la discusión:** Consiste en la discusión de un tema, por parte de los alumnos, bajo la dirección del profesor y requiere preparación anticipada.
- **Técnica de la demostración:** Es el procedimiento más deductivo y puede asociarse a cualquier otra técnica de enseñanza cuando sea necesario comprobar afirmaciones no muy evidentes o ver cómo funciona, en la práctica, lo que fue estudiado teóricamente.

ESTA TÉCNICA TIENE POR OBJETIVOS:

- ✓ Confirmar explicaciones orales o escritas.
 - ✓ Ilustrar lo que fue expuesto teóricamente.
 - ✓ Iniciar teóricamente una técnica para evitar errores.
 - ✓ Propiciar un esquema de acción correcto para la ejecución de una tarea.
 - ✓ Convencer racionalmente en cuanto a la veracidad de proposiciones abstractas.
- **Técnica del redescubrimiento:** Técnica activa. Especial para cuando el alumno posee poca información sobre el tema. Uso en mayor medida en áreas de las ciencias, pero en general se puede trabajar en todas las materias. Implica el uso de tiempo extra y de áreas

especiales de experimentación (laboratorios).

- **Técnica del estudio dirigido:** Es una forma de uso en especial en las universidades, por la dedicación, esfuerzo y compromiso requerido para llevar a cabo esta técnica. El profesor puede dar una explicación inicial y el alumno sigue trabajando bajo la dirección del docente en conocimientos o temas complementarios al estudio.

2.2.1.1.6 ESTILOS DE APRENDIZAJE

- **Divergentes:** (Combina la experiencia concreta y la observación reflexiva)
 - Habilidad imaginativa, son buenos generando ideas.
 - Pueden ver las situaciones desde diferentes perspectivas.
 - Emotivos, se interesan por la gente.
 - Se caracterizan por ser individuos con un trasfondo en Artes Liberales o Humanidades.
 - Características de individuos en el área de Consejería, Administración de Personal y Especialista en Desarrollo Organizacional.
- **Asimiladores:** (Combina la conceptualización abstracta y la observación reflexiva)
 - Habilidad para crear modelos teóricos.
 - Razonamiento inductivo.
 - Les preocupa más los conceptos que las personas, menos interesados en el uso práctico de las teorías.
 - Característico de individuos en el área de Ciencias, Planificación e Investigación.
- **Convergentes:** (Combina la conceptualización abstracta y la experimentación activa)
 - Son buenos en la aplicación práctica de las ideas.
 - Son buenos en situaciones donde hay más de una contestación.
 - No son emotivos, prefieren las cosas a las personas
 - Intereses técnicos.
 - Característicos de individuos en Ingeniería.

- **Acomodadores:** (Combina la experiencia concreta y la experimentación activa)
 - Habilidad para llevar a cabo planes, orientados a la acción.
 - Les gustan nuevas experiencias, son arriesgados.
 - Se adaptan a las circunstancias inmediatas.
 - Intuitivos, aprenden por tanteo y error.
 - Característicos de individuos en el área de los Negocios

2.2.2 PROPUESTA DE SOLUCIÓN PARA EL CURSO DE CAPACITACIÓN

2.2.2.1 INTRODUCCIÓN

Para una capacitación efectiva depende del estilo, metodología, materiales utilizados y el contenido para impartición de la misma.

El ambiente en el cual se llevará a cabo la capacitación tiene gran importancia para la obtención de los objetivos propuesto para la misma, por lo cual, la capacitación se realizara en un ambiente preparado en el domicilio del gerente general del “COMERCIAL FIGUEROA”.

Con lo expuesto en el anterior párrafo, la capacitación se realizar del 27/11/14 al 04/12/14, en la cual participaran todos los empleados y el gerente general, los cuales serán parte del contenido de la capacitación, que será sobre la importancia de las TIC, y el uso del sistema.

2.2.2.1.1 OBJETIVO

Capacitación a todos los empleados del “Comercial Figueroa”, para el registro correcto de las ventas, y de las otras opciones que permite el sistema.

2.2.2.1.2 METODOLOGÍA

En la capacitación se hará se realizara de acuerdo al siguiente método y técnicas:

✓ Método de Enseñanza socializada

Dirigido a todos los médicos especialmente generales y odontólogos y enfermeras como nos indica este método, si es necesario aplicando la enseñanza individual.

✓ Técnica expositiva

Como nos explica la técnica se realizará una explicación oral sobre la importancia de las TIC en la medicina y la instalación, uso o manejo del sistema.

2.2.2.1.3 MEDIOS A UTILIZAR

- ✓ Computadora con el Sistema Desarrollado.
- ✓ Diapositivas de Exposición hechas en PowerPoint.

2.2.2.1.4 CONTENIDO SOBRE EL CONTENIDO DE LAS TIC

Que son las TIC

Son las tecnologías de la Información y Comunicación, es decir, son aquellas herramientas computacionales e informáticas que procesan, sintetizan, recuperan y presentan información representada de la más variada forma. Es un conjunto de herramienta, soportes y canales para el tratamiento y acceso a la información, para dar forma, registrar, almacenar y difundir contenidos digitalizados

Importancia de las TIC

Las Tecnologías de la Información y la Comunicación han transformado nuestra manera de trabajar y gestionar recursos. Las TIC son un elemento clave para hacer que nuestro trabajo sea más productivo: agilizando las comunicaciones, sustentando el trabajo en equipo, gestionando las existencias, realizando análisis financieros, y promocionando nuestros productos en el mercado.

Las TIC en las empresas

Las TIC permiten a las empresas producir más cantidad, más rápido, de mejor calidad, y en menos tiempo. Por ende permite a las mismas ser más competitivas en el mercado.

Las TIC facilitan el trabajo de presentar el producto a los clientes, y conseguir ventas de muchas maneras distintas. Por ejemplo:

- ✓ El **correo electrónico** permite a las empresa enviar todo tipo de información y comunicados a sus clientes. Por ejemplo puede enviar un catálogo de productos, una felicitación de navidad o un boletín de noticias, sin prácticamente ningún coste. Las TIC lo hacen posible.
- ✓ Una **página web** donde exponer sus productos permite que los clientes interesados encuentren los productos fácilmente en Internet y contacten con las empresas. También permite transmitir al mundo nuestra filosofía de empresa.
- ✓ Un **sistema de gestión** de clientes informatizado, permite conocer mejor a sus clientes, analizando sus hábitos y su historial de compras. Así podemos planificar mejor nuestras acciones de venta y también gestionar de forma eficaz de las diferentes áreas de negocio de la empresa.

2.2.2.1.5 ESTRUCTURA DEL CURSO DE CAPACITACIÓN

Fecha Realizar	a	Módulos de Aprendizaje del Curso	Tiempo	Nº de Sesiones	Dirigido A
27/11/14		Importancia de las TIC (en las empresas)	40 min.	1	Gerente General y Empleados
		Totales	40 min	1 Sesión	
Fecha Realizar	a	Módulos de Aprendizaje del Curso	Tiempo	Nº de Sesiones	Dirigido A
04/12/14		Presentación y capacitación del sistema.	40 min.	1	Gerente General y Empleados
		Totales	40 min	1 Sesión	

**2.2.3 MEDIOS DE VERIFICACIÓN PARA EL COMPONENTE DE CAPACITACIÓN
SOBRE LA IMPORTANCIA DE LAS TIC Y EL USO DEL SISTEMA (VER ANEXO
B)**

III. CONCLUSIONES Y RECOMENDACIONES (VER ANEXO 4)