SISTEMA DE INFORMACION WEB PARA LA

GESTION DE MEDICAMENTOS Y PEDIDOS A

DOMICILIO PARA LA FARMACIA “SAN PEDRO”

**INTEGRANTES**



Rubén Darío Toconas Romero Henry Mercado Parada.

Alexander Yasmani Cárdenas Chusco

# MATERIA

Tecnología Web **CARRERA**

Ingeniería De Sistemas

# DOCENTE

Ing. Edwin Calle Terrazas

Montero – Santa Cruz - Bolivia

SISTEMA DE INFORMACION WEB PARA LA GESTION DE MEDICAMENTOS Y PEDIDOS A DOMICILIO:

“SAN PEDRO”

En este documento la planificación y gestión del proyecto utiliza las fases que plantea la metodología del Proceso Unificado de Desarrollo de Software (PUDS) para el desarrollo del producto se utiliza el Lenguaje Unificado de Modelado (UML) para documentar los artefactos del sistema.

**DOCENTE**

Ing. Edwin Calle Terrazas

Nombres y Apellidos Registro

Yasmani Alexander Cárdenas Chusco 215091809

Henry Mercado Parada 215096096

Rubén Darío Toconas Romero 215173589

Montero - Santa Cruz – Bolivia

## INDICE

Página

CAPÍTULO I: PERFIL DEL PROYECTO ................................................................................. 1

1.1. INTRODUCCION ................................................................................................................ 1

1.2. ANTECEDENTES ............................................................................................................... 1

1.3. DESCRIPCION DEL PROBLEMA .................................................................................... 2

1.4. SITUACION PROBLEMÁTICA ........................................................................................ 2

1.5. JUSTIFICACION ................................................................................................................. 3

1.5.1. Personal ............................................................................................................................. 3

1.5.2. Técnica............................................................................................................................... 3

1.5.3. Teórica ............................................................................................................................... 3

1.6. OBJETIVOS ......................................................................................................................... 3

1.6.1. Objetivo General................................................................................................................ 3

1.6.2. Objetivos Específicos ........................................................................................................ 3

1.7. METODOLOGÍA ................................................................................................................. 4

1.8. ALCANCE ........................................................................................................................... 4

1.9. CRONOGRAMA ................................................................................................................. 5 CAPITULO 2: CAPTURA DE REQUISITOS ........................................................................... 6

2.1. Requerimientos del sistema .................................................................................................. 6

2.1.1. Lista de requerimientos ..................................................................................................... 6

2.1.1.1. Requerimientos funcionales ........................................................................................... 6

2.1.1.2. Requerimientos no funcionales .................................................................................... 11

2.1.1.2.1. Tiempo de aprendizaje............................................................................................... 11

2.1.1.2.2. Identificación del usuario propio de la aplicación ..................................................... 11

2.1.1.2.3. Confiabilidad ............................................................................................................. 11

2.1.1.2.4 Desempeño ................................................................................................................. 11

2.1.1.2.5. Restricciones de diseño ............................................................................................. 12

2.1.1.2.6. Interfaces ................................................................................................................... 12

2.2. Captura de requisitos como casos de uso ........................................................................... 13

2.2.1. Modelo de especificación de casos de uso ...................................................................... 13

2.2.2. Actores ............................................................................................................................. 14

2.2.2.1. Identificación de los actores del sistema ...................................................................... 14

2.2.3. Tabla de priorización de casos de uso ............................................................................. 15

2.3. Casos de uso ....................................................................................................................... 16

2.1.2.5. Modelo general de casos de uso ................................................................................... 43

CAPITULO 3: ANALISIS DEL SISTEMA ............................................................................. 44

3.1. Introducción ........................................................................................................................ 44

3.2. Identificación de paquetes de análisis ................................................................................ 44

3.3. Realización de casos de uso (diag. de colaboración).......................................................... 45 Capitulo 4: DISEÑO DEL SISTEMA ....................................................................................... 50

4.1. Introducción ........................................................................................................................ 50

4.2. Arquitectura del sistema (diag. de paquetes e interfaces) .................................................. 50

4.2.1. Diagrama de paquetes organizado en capas. ................................................................... 50

4.2.2. Interfaces ......................................................................................................................... 51

4.3. Diagrama de despliegue (hardware) ................................................................................... 51

4.4. Diseño de los casos de uso ................................................................................................. 52

4.4.1. Diagramas de secuencia para cada caso de uso ............................................................... 52

4.5 Diagrama de clases .............................................................................................................. 61

4.6. Diseño Lógico de la Base de Datos (campos) .................................................................... 61

4.7. Diseño Físico de la base de datos (tablas y campos) .......................................................... 63

1. IMPLEMENTACION ........................................................................................................... 68 Selección del entorno de desarrollo para la implementación .................................................... 69
   1. CONCLUSIONES .............................................................................................................. 73
   2. RECOMENDACIONES .................................................................................................... 74

## CAPÍTULO I: PERFIL DEL PROYECTO

### 1.1. INTRODUCCION

Los sistemas de información hoy en día son fundamentales para las empresas porque existe un mejor control de la información de las actividades que realizan. En este trabajo se buscó mejorar el funcionamiento de una empresa que no contaba con este sistema web.

Hoy en día la venta de productos farmacéuticos es importante para todas las personas y empresas ya que los mismos ayudan a curar, proteger y mejorar unos de los bienes más preciados de la humanidad, la salud.

Los productos farmacéuticos son una de las herramientas terapéuticas que más se usa en la actualidad, el uso adecuado de los mismos permite a la población conseguir enormes beneficios en cuanto al alivio o prevención de enfermedades, mejorando de esta forma el estado de salud de las personas que están enfermas, o sufren alguna dolencia.

Este trabajo plantea un sistema web que ayude a la empresa a tener un control sobre las actividades de compra, venta, y pedido a domicilio por parte del cliente. Cuando se tiene un buen control se mejora el rendimiento de la empresa.

Para conseguir esto, es necesario realizar una investigación sobre las actividades de la empresa mediante entrevistas, cuestionarios y la observación para obtener información que nos sirva para desarrollar un sistema que ayude en el control de compra y venta de la empresa.

Se busca que el sistema a implementar cumpla con los requerimientos necesarios y a la vez sea fácil de usar de manera que todo el personal autorizado pueda usarlo de una mejor manera.

Se coordinó con el propietario y el personal para obtener información de cómo quieren que sea el sistema para hacer más cómodo su uso adaptándolo a las necesidades de la empresa en el ámbito de compra y venta, esto con el fin de obtener una satisfacción total por parte del cliente.

### 1.2. ANTECEDENTES

Es de gran importancia e interés conocer cómo la empresa farmacéutica SAN PEDRO ha llegado hasta donde está hoy en día.

Esta empresa fue fundada el 5 de marzo del año 1996 en MONTERO por la Dra. Elizabeth

Rebollo Gutiérrez. Esta se encuentra ubicada en el barrio Fabril a inmediaciones del mercado Villa Verde, esta se encuentra a una cuadra de la rotonda sur carretera a Santa Cruz.

Cuando la empresa empezó a funcionar solo contaba con un empleado y el gerente que sería el fundador.

La empresa lleva funcionando 23 años, su actual dueño es el primer propietario y fundador “Dra. Elizabeth Rebollo Gutiérrez”. Hoy en día la empresa cuenta con 5 empleados incluyendo al gerente propietario.

Esta empresa realiza ventas de productos farmacéuticos a la población en general.

Actualmente se encuentra ubicada por inmediaciones de la Av. Fabril, a una cuadra de la rotonda sur. En todo el tiempo que lleva funcionando no ha tenido ningún antecedente de incumplimiento según la ley establecida por lo cual la empresa se ha mantenido activa hasta la fecha.

### 1.3. DESCRIPCION DEL PROBLEMA

La empresa cuenta con los siguientes problemas:

* La empresa solo cuenta con un sistema para registrar el ingreso y el costo de la mercadería, la cual se realiza de forma manual.
* No cuenta con ningún sistema de software para registrar las compras y ventas de sus productos.
* No cuenta con ningún sistema de software para gestionar los pedidos de productos por parte de los clientes.
* Dada la falta de información acerca de los productos que se tiene en la empresa, existe un deficiente control del stock de mercadería que se compra, se vende.
* La empresa no tiene registros de las ganancias por producto en un tiempo determinado.
* No se lleva un registro de los almacenes en los que se guardan los productos.

### 1.4. SITUACION PROBLEMÁTICA

La empresa tiene el siguiente problema:

No cuenta con un sistema web para registrar las actividades de compra, venta y pedidos a domicilio por parte de los clientes.

### 1.5. JUSTIFICACION

#### 1.5.1. Personal

Las razones que nos impulsan a realizar este proyecto son las siguientes: Adquirir conocimiento sobre el desarrollo de un sistema de información web haciendo uso de la metodología PUDS, obtener experiencia en la realización de un proyecto de desarrollo de software, mejorar el rendimiento de la empresa, y aprobar la materia Tecnología Web.

#### 1.5.2. Técnica

El presente proyecto tiene como finalidad el desarrollo de un sistema de información web que servirá para gestionar las actividades de compra, venta y pedidos de productos farmacéuticos por parte de los clientes.

Se desea potenciar el desempeño de la empresa modernizando los procesos de trabajo relacionados con la compra y venta, el control eficiente de la información nos permite elevar el nivel competitivo para que en el futuro la empresa pueda tener un mejor rendimiento y un crecimiento constante, esto se pretende lograr aplicando los conocimientos adquiridos en la materia Tecnología Web.

#### 1.5.3. Teórica

Para el presente proyecto nos basamos en la metodología PUDS, tomando como referencia lo aprendido en la materia Tecnología Web, diapositivas referentes a la materia, libros, tutoriales entre otros.

### 1.6. OBJETIVOS

#### 1.6.1. Objetivo General

Desarrollar un sistema de información web para gestionar la compra, venta y pedidos de productos farmacéuticos para la farmacia “San Pedro” ubicada en la ciudad de Montero.

#### 1.6.2. Objetivos Específicos

* Elaborar entrevistas al administrador y a los empleados para conocer las necesidades y los problemas que tiene la empresa en las áreas de compra, venta y pedidos.
* Analizar toda la información recaudada para conocer las necesidades de la empresa y determinar los requerimientos del sistema.
* Elaborar el diseño conceptual, lógico y físico de la base de datos.
* Desarrollar el sistema web usando la técnica de programación en tres capas; capa de datos, capa de negocio y capa de presentación
* Aplicar pruebas al sistema para encontrar posibles errores.

### 1.7. METODOLOGÍA

La metodología podemos definirla como un conjunto de métodos que se siguen en una investigación científica.

El presente proyecto está basado en la metodología del proceso unificado de desarrollo de software (PUDS) y el Lenguaje Unificado de Modelado (U.M.L.) para documentar los artefactos del sistema.

### 1.8. ALCANCE

El sistema de información abarcará los siguientes campos.

Sistema De Información

Web

Venta

Reg. Cli

ente

ntas

Reg. Ve

Reg.

Emplea

do

Compra

Reg.

ores

Proveed

e

Reg. D

a

compr

Reg. De

o

Emplead

Producto

Reg. De

Productos

n

Administracio

de usuario

Reg.

usuario

Pedidos

Reg. De cliente

web

Reg. De

empleado

Reg. Del

destino del

pedido

Reg. De

pedidos

### 1.9. CRONOGRAMA

|  |  |
| --- | --- |
| ***25 de abril de 2019*** | Se fue a conocer la empresa y preguntar al administrador si contaba con un sistema informático que le permita registrar sus compras y ventas de productos. |
| ***27 de abril de 2019*** | Se realizó una entrevista al administrador para obtener mayor información sobre las actividades de la empresa y los problemas que tenía la empresa en el área de compra y venta, también se le consultó si realizaba ventas a pedido con entrega a domicilio. |
| ***2 de mayo de 2019*** | Se empezó a analizar la información recaudada esencialmente en el ámbito de compra y venta y pedidos. |
| ***03 de mayo de 2019*** | Se realizó el diseño conceptual y el diseño lógico de la base de datos |
| ***04 de mayo de 2019*** | Se realizó el diseño físico de la base de datos |
| ***23 de mayo de 2019*** | Se modificó el diseño físico de la base de datos |

## CAPITULO 2: CAPTURA DE REQUISITOS

### 2.1. REQUERIMIENTOS DEL SISTEMA

#### 2.1.1. Lista de requerimientos

##### 2.1.1.1. Requerimientos funcionales

Para mayor claridad los requerimientos funcionales se han agrupado por áreas, las mismas se presentan en la siguiente tabla:

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.1 | Gestión de Usuario |
| RF.2 | Gestión de Categoría |
| RF.3 | Gestión de Laboratorio |
| RF.4 | Gestión de Producto |
| RF.5 | Gestión de stock de producto |
| RF.6 | Gestión de Empleado |
| RF.7 | Gestión de Cliente |
| RF.8 | Gestión de Proveedor |
| RF.9 | Gestión de Compra |
| RF.10 | Gestión del Detalle de Compra |
| RF.11 | Gestión de Venta |
| RF.12 | Gestión del Detalle de Venta |
| RF.13 | Gestión de Pedido |
| RF.14 | Gestión del Detalle Pedido |

### Gestión de Usuario

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.1.1 | Ingresar nuevo usuario |
| RF.1.2 | Iniciar sesión de usuario |
| RF.1.3 | Mostrar usuario registrado |
| RF.1.4 | Modificar usuario registrado |
| RF.1.5 | Eliminar usuario registrado |

### Gestión de Categoría

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.2.1 | Ingresar nueva categoría |
| RF.2.2 | Mostrar categoría registrada |
| RF.2.3 | Modificar categoría registrada |
| RF.2.4 | Eliminar categoría registrada |

### Gestión de Laboratorio

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.3.1 | Ingresar nuevo laboratorio |
| RF.3.2 | Mostrar laboratorio registrado |
| RF.3.3 | Modificar laboratorio registrado |
| RF.3.4 | Eliminar laboratorio registrado |

### Gestión de Producto

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.4.1 | Ingresar nuevo producto |
| RF.4.2 | Mostrar producto registrado |
| RF.4.3 | Modificar producto registrado |
| RF.4.4 | Eliminar producto registrado |
| RF.4.5 | Ingresar nueva categoría |
| RF.4.6 | Modificar categoría |

### Gestión de Stock de Producto

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.6.1 | Registro de ingreso de producto |
| RF.6.2 | Registro de egreso de producto |

### Gestión de Empleado

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.7.1 | Ingresar nuevo empleado |
| RF.7.2 | Mostrar empleado registrado |
| RF.7.3 | Modificar empleado registrado |
| RF.7.4 | Eliminar empleado registrado |

### Gestión de Cliente

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.8.1 | Ingresar nuevo cliente |
| RF.8.2 | Mostrar cliente registrado |
| RF.8.3 | Modificar cliente registrado |
| RF.8.4 | Eliminar cliente registrado |

### Gestión de Proveedor

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.9.1 | Ingresar nuevo proveedor |
| RF.9.2 | Mostrar proveedor registrado |
| RF.9.3 | Modificar proveedor registrado |
| RF.9.4 | Eliminar proveedor registrado |

### Gestión de Compra

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.10.1 | Ingresar nueva compra |
| RF.10.2 | Mostrar compra registrada |
| RF.10.3 | Modificar compra registrada |
| RF.10.4 | Eliminar compra registrada |

### Gestión del Detalle de Compra

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.11.1 | Registro de la cantidad del producto a comprar y el precio total del mismo |

### Gestión de Venta

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.12.1 | Ingresar nueva venta |
| RF.12.2 | Mostrar venta registrada |
| RF.12.3 | Modificar venta registrada |
| RF.12.4 | Eliminar venta registrada |

### Gestión del Detalle de Venta

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.13.1 | Registrar la cantidad de producto a vender y el precio total del mismo |

### Gestión del Pedido

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.14.1 | Ingresar nuevo pedido |
| RF.14.2 | Mostrar pedido registrado |
| RF.14.3 | Modificar pedido registrado |
| RF.14.4 | Eliminar pedido registrado |

### Gestión del Detalle Pedido

|  |  |
| --- | --- |
| **Referencia** | **Función** |
| RF.15.1 | Registrar la cantidad de producto pedido y el precio total del mismo |

#### 2.1.1.2. Requerimientos no funcionales

##### 2.1.1.2.1. Tiempo de aprendizaje

La capacitación del personal en cuanto al manejo del software y el mantenimiento del mismo deberá considerar un tiempo considerable de por lo menos 40 horas laborales de capacitación (mínimo 1 hora por día), hasta que los lineamientos de manejo del sistema y otros hayan sido debidamente adquiridos por el personal asignado para su uso.

##### 2.1.1.2.2. Identificación del usuario propio de la aplicación

El usuario ingresara al sistema con su nombre y contraseña, las mismas serán validadas por el sistema según el rol que tiene asignado.

Para el registro de la contraseña de los usuarios deberá asegurarse que la longitud de la contraseña tenga un mínimo de 6 caracteres con un máximo de 15 caracteres.

**2.1.1.2.3. Confiabilidad**

### Tiempo de disponibilidad del sistema

La aplicación puede estar disponible para su uso a cualquier hora del día, cualquier día de la semana, durante todo el año, esto estará sujeto al correcto funcionamiento de los equipos en donde el sistema se esté ejecutando, como ser: buen rendimiento del servidor, calidad del enlace y calidad de conexión de red local.

### Tiempo fuera de servicio

El tiempo de fuera de operación del sistema dependerá del funcionamiento del servidor, hardware del ordenador, base de datos y la infraestructura de red.

#### 2.1.1.2.4 Desempeño

**Acceso de usuarios al sistema:**

Los usuarios pueden acceder a los datos en tiempo real.

### Calidad de atención al usuario

El sistema debe poder atender normalmente 2 o más usuarios al mismo tiempo.

**2.1.1.2.5. Restricciones de diseño**

### Estándares de arquitectura

* Para el desarrollo de software se hará uso de la arquitectura en 3 capas.
* Para la infraestructura de red, se debe implementar la arquitectura Cliente/Servidor**.**

**Motor de base de datos**

Se utilizará el motor de base de datos MySQL server de ORACLE.

### Cliente de escritorio

La aplicación deberá ser accesible utilizando un ordenador con procesador de tercera generación o superior con sistema operativo Windows 7 o superior, o cualquier distribución de LINUX.

**Servidor de datos**

El servidor será: Windows Server Estándar Edition.

### Lenguaje de programación

La aplicación debe desarrollarse haciendo uso de los lenguajes de programación PHP, JAVASCRIPT, y otras herramientas de desarrollo web, como ser: HTML, CSS, AJAX, JSON, entre otras.

**2.1.1.2.6. Interfaces**

### Interfaz de usuario

No debe existir presencia de imágenes distorsionadas o difíciles de entender. La presentación de mensaje de error o de infracción al usuario deberá ser lo más específico posible y comunicarse con el administrador del sistema.

### Interfaz de comunicaciones

Existe una conexión entre los usuarios y el servidor donde está alojado la base de datos:

* Los usuarios internos de la empresa se conectarán al sistema mediante una conexión de red local. Esta conexión la realizarán desde su oficina en la misma infraestructura de la empresa.
* Para el entorno de red, la aplicación deberá tener la capacidad de funcionar en un entorno de red LAN.

### Interfaz de software

Cualquier usuario que desee conectarse al sistema necesitará tener un equipo con sistema operativo Windows 7 o superior, o cualquier distribución actualizada de LINUX esto para poder acceder a él.

Por otra parte, el servidor de base de datos (MySQL SERVER) se implementará bajo el sistema operativo Windows Server 2016.

### 2.2. CAPTURA DE REQUISITOS COMO CASOS DE USO

#### 2.2.1. Modelo de especificación de casos de uso

Para tener una mayor claridad en la especificación de los casos, se ha agrupado los mismos en áreas, las cuales vemos en la siguiente tabla:

|  |  |
| --- | --- |
| **Caso de Uso** | **Descripción** |
| C.U.1 | Gestionar Usuario |
| C.U.2 | Gestionar Categoría |
| C.U.3 | Gestionar Laboratorio |
| C.U.4 | Gestionar Producto |
| C.U.5 | Gestionar Empleado |
| C.U.6 | Gestionar Cliente |
| C.U.7 | Gestionar Proveedor |
| C.U.8 | Gestionar Compra |
| C.U.9 | Gestionar Detalle Compra |
| C.U.10 | Gestionar venta |
| C.U.11 | Gestionar detalle de venta |
| C.U.12 | Gestionar pedido |
| C.U.13 | Gestionar detalle pedido |

#### 2.2.2. Actores

##### 2.2.2.1. Identificación de los actores del sistema

A través del análisis de requerimientos, se definió que los actores del sistema serán los siguientes: Administrador, empleado, cliente, proveedor. A continuación, se detallan las características de cada actor.

### Administrador

Este perfil corresponde al administrador del sistema, el cual cuenta con acceso a toda la información del sistema. Podrá ingresar, modificar y eliminar datos del sistema. Este actor cuenta con la facultad de crear cuentas de usuario para los empleados y para otros administradores.

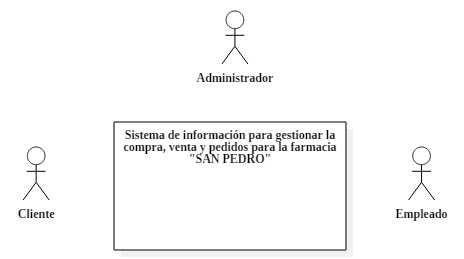
### Cliente

Este perfil corresponde a todos los clientes que se apersonan a las instalaciones de la empresa para comprar productos y también a aquellos que visitan la página web de la empresa para hacer el pedido de productos.

### Empleado

Este actor tiene los permisos para acceder a toda la información referente a los productos, pero este no podrá hacer cambios en la información de los mismos. Entre algunas de las funciones que tendrá, son las de registrar las ventas y las compras

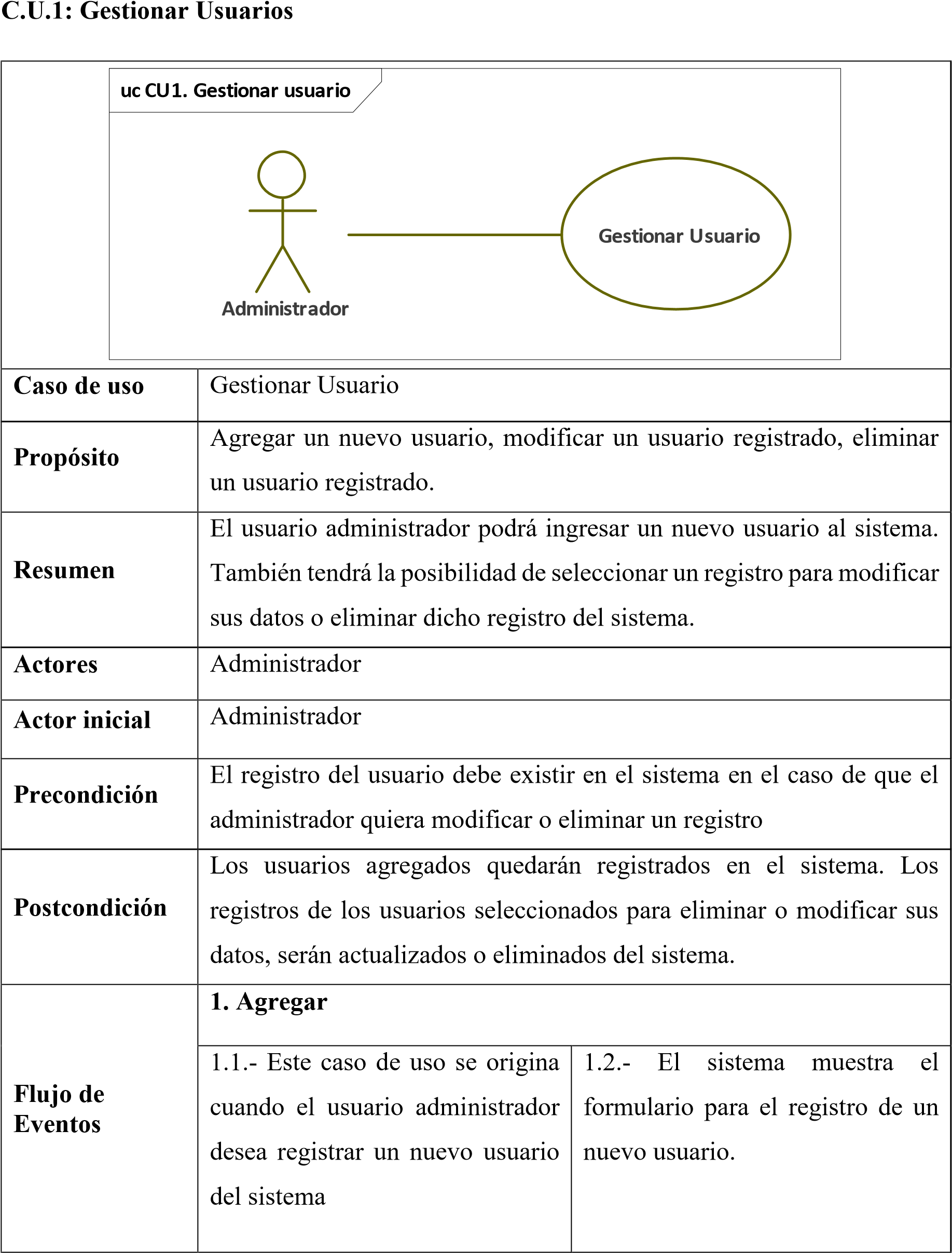
*Actores del sistema*



#### 2.2.3. Tabla de priorización de casos de uso

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Descripción** | **Prioridad** |
| C.U.1 | Gestionar Unidad Medida | Baja |
| C.U.2 | Gestionar Categoría | Baja |
| C.U.3 | Gestionar Laboratorio | Baja |
| C.U.4 | Gestionar Producto | Media |
| C.U.5 | Gestionar Empleado | Media |
| C.U.6 | Gestionar Cliente | Media |
| C.U.7 | Gestionar Proveedor | Media |
| C.U.8 | Gestionar Compra | Alta |
| C.U.9 | Gestionar Detalle Compra | Alta |
| C.U.10 | Gestionar venta | Alta |
| C.U.11 | Gestionar detalle de venta | Alta |
| C.U.12 | Gestionar pedido | Alta |
| C.U.13 | Gestionar detalle pedido | Alta |

### 2.3. CASOS DE USO



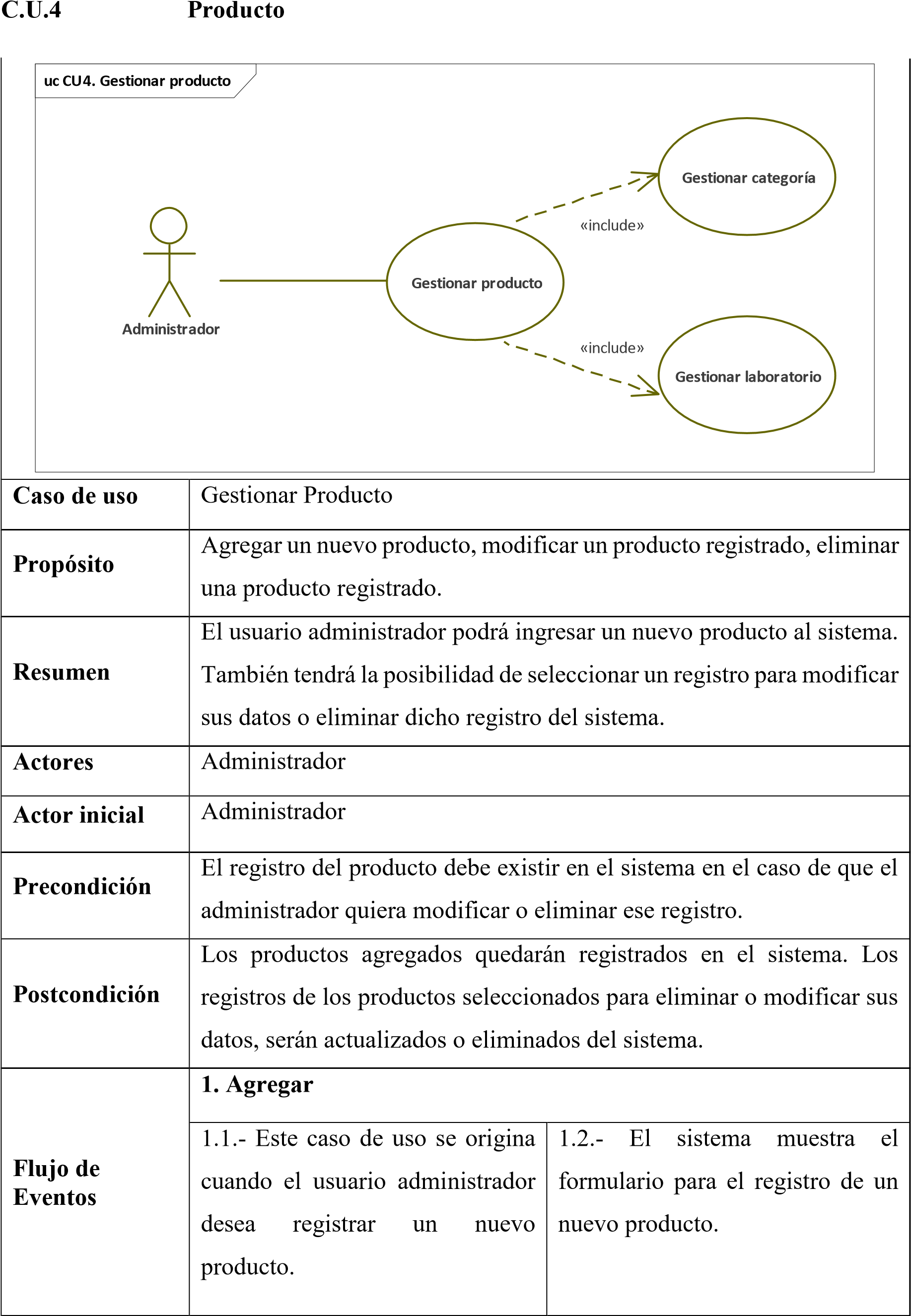
|  |  |  |
| --- | --- | --- |
|  | 1.3.- El administrador ingresa los datos requeridos: Nombre, tipo, y contraseña. | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador desea modificar el registro de algún usuario del sistema. | 2.2.- El sistema muestra una lista con los registros de todos los usuarios del sistema. |
| 2.3.- El administrador selecciona el registro del usuario que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del usuario seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 3.1.- Este caso de uso se da cuando el administrador desea eliminar el registro de algún usuario del sistema. | 3.2.- El sistema muestra una lista con los registros de todos los usuarios. |
| 3.3.- El administrador selecciona el registro del usuario que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del usuario seleccionado. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |

### C.U.2 Categoría

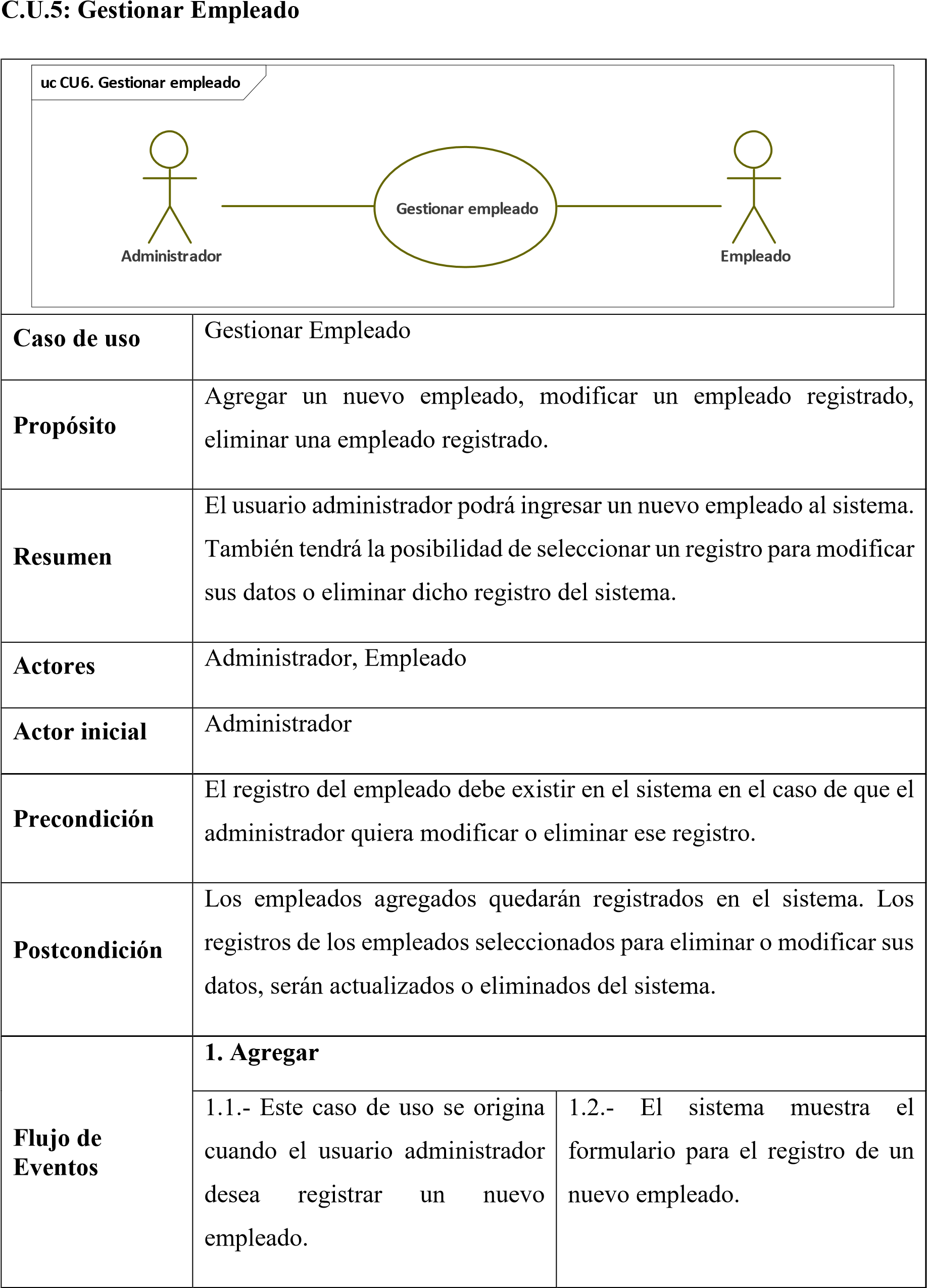
|  |  |  |
| --- | --- | --- |
| **uc CU2. Gestionar categoría**  **Administrador**  **Gestionar categoría** | | |
| **Caso de uso** | Gestionar Categoría | |
| **Propósito** | Agregar una nueva categoría, modificar una categoría registrada, eliminar una categoría registrada. | |
| **Resumen** | El usuario administrador podrá ingresar una nueva categoría al sistema. También tendrá la posibilidad de seleccionar un registro para modificar sus datos o eliminar dicho registro del sistema. | |
| **Actores** | Administrador | |
| **Actor inicial** | Administrador | |
| **Precondición** | El registro de la categoría debe existir en el sistema en el caso de que el administrador quiera modificar o eliminar ese registro. | |
| **Postcondición** | Las categorías agregadas quedarán registradas en el sistema. Los registros de las categorías seleccionadas para eliminar o modificar sus datos, serán actualizadas o eliminadas del sistema. | |
| **Flujo de Eventos** | **1. Agregar** | |
| 1.1.- Este caso de uso se origina cuando el usuario administrador desea registrar una nueva categoría. | 1.2.- El sistema muestra el formulario para el registro de una nueva categoría. |
|  | 1.3.- El administrador ingresa los datos requeridos: Nombre de la categoría | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador desea modificar el registro de una categoría. | 2.2.- El sistema muestra una lista con los registros de todas la categorías. |
| 2.3.- El administrador selecciona el registro de la categoría que desea modificar. | 2.4.- El sistema despliega un formulario con los datos de la categoría seleccionada en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 3.1.- Este caso de uso se da cuando el administrador desea eliminar el registro de alguna categoría que este registrada en el sistema. | 3.2.- El sistema muestra una lista con los registros de todas las categorías. |
| 3.3.- El administrador selecciona el registro de la categoría que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos de la categoría seleccionada. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |

### C.U.3 Laboratorio

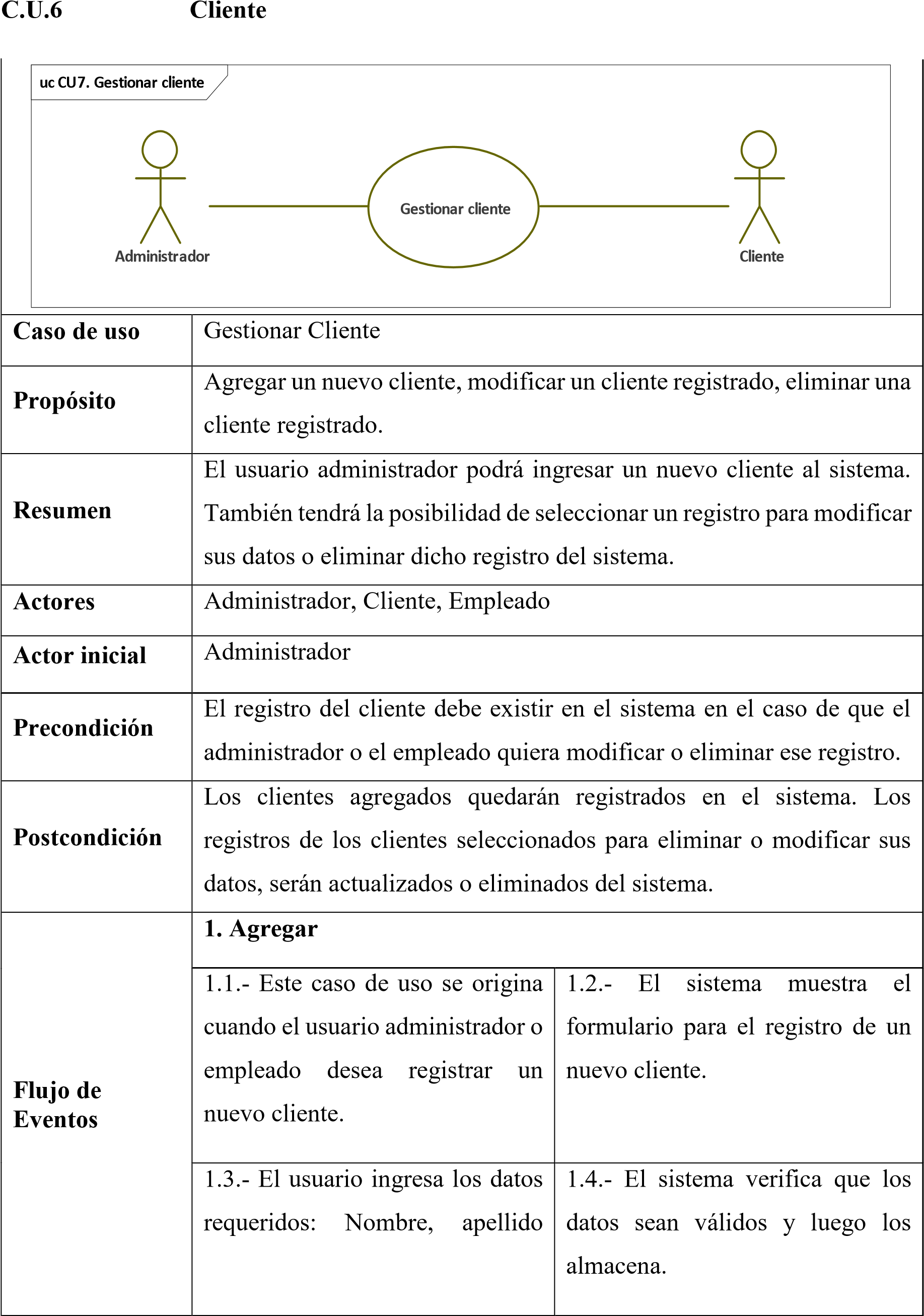
|  |  |  |
| --- | --- | --- |
| **uc CU3.Gestionar laboratorio**  **Administrador**  **Gestionar laboratorio** | | |
| **Caso de uso** | Gestionar Laboratorio | |
| **Propósito** | Agregar un nuevo laboratorio, modificar un laboratorio registrado, eliminar una laboratorio registrado. | |
| **Resumen** | El usuario administrador podrá ingresar un nuevo laboratorio al sistema. También tendrá la posibilidad de seleccionar un registro para modificar sus datos o eliminar dicho registro del sistema. | |
| **Actores** | Administrador | |
| **Actor inicial** | Administrador | |
| **Precondición** | El registro del laboratorio debe existir en el sistema en el caso de que el administrador quiera modificar o eliminar ese registro. | |
| **Postcondición** | Los laboratorios agregados quedarán registrados en el sistema. Los registros de los laboratorios seleccionados para eliminar o modificar sus datos, serán actualizados o eliminados del sistema. | |
| **Flujo de Eventos** | **1. Agregar** | |
| 1.1.- Este caso de uso se origina cuando el usuario administrador desea registrar un nuevo laboratorio. | 1.2.- El sistema muestra el formulario para el registro de un nuevo laboratorio. |
|  | 1.3.- El administrador ingresa los datos requeridos: Nombre del laboratorio. | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador desea modificar el registro de un laboratorio. | 2.2.- El sistema muestra una lista con los registros de todos los laboratorios. |
| 2.3.- El administrador selecciona el registro del laboratorio que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del laboratorio seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 1.- Este caso de uso se da cuando el administrador desea eliminar el registro de algún laboratorio que este registrado en el sistema. | 2.- El sistema muestra una lista con los registros de todos los laboratorios. |
| 3.- El administrador selecciona el registro del laboratorio que desea eliminar. | 4.- El sistema despliega un formulario con los datos del laboratorio seleccionado. |
| 5.- El administrador confirma la eliminación del registro. | 6.- El sistema procede a eliminar el registro. |



|  |  |  |
| --- | --- | --- |
|  | 1.3.- El administrador ingresa los datos requeridos: Nombre del producto, presentación, fecha de vencimiento, laboratorio y categoría. | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador desea modificar el registro de un producto. | 2.2.- El sistema muestra una lista con los registros de todos los productos. |
| 2.3.- El administrador selecciona el registro del producto que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 3.1.- Este caso de uso se da cuando el administrador desea eliminar el registro de algún producto que está registrado. | 3.2.- El sistema muestra una lista con los registros de todos los productos. |
| 3.3.- El administrador selecciona el registro del producto que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del producto seleccionado. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |



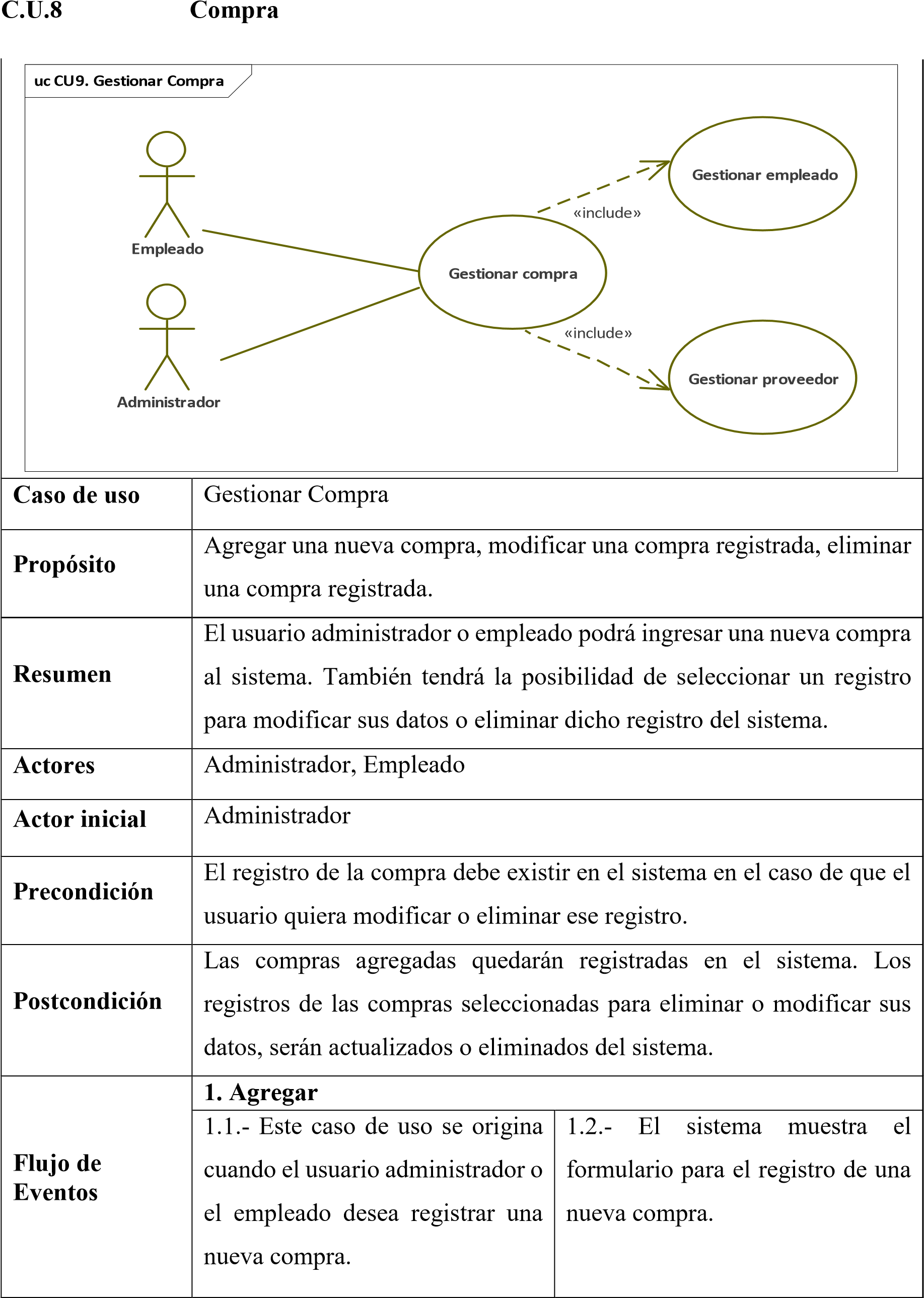
|  |  |  |
| --- | --- | --- |
|  | 1.3.- El administrador ingresa los datos requeridos: Nombre, apellido paterno, apellido materno, y teléfono. | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador desea modificar el registro de un empleado. | 2.2.- El sistema muestra una lista con los registros de todos los empleados. |
| 2.3.- El administrador selecciona el registro del empleado que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 3.1.- Este caso de uso se da cuando el administrador desea eliminar el registro de algún empleado que está registrado. | 3.2.- El sistema muestra una lista con los registros de todos los empleados. |
| 3.3.- El administrador selecciona el registro de del empleado que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |



|  |  |  |
| --- | --- | --- |
|  | paterno, apellido materno, y teléfono. |  |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador o el empleado desea modificar el registro de un cliente. | 2.2.- El sistema muestra una lista con los registros de todos los clientes. |
| 2.3.- El usuario selecciona el registro del cliente que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El usuario modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 1.- Este caso de uso se da cuando el administrador o el empleado desea eliminar el registro de algún cliente que se encuentre registrado. | 2.- El sistema muestra una lista con los registros de todos los clientes. |
| 3.- El usuario selecciona el registro del cliente que desea eliminar. | 4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 5.- El usuario confirma la eliminación del registro. | 6.- El sistema procede a eliminar el registro. |

### C.U.7 Proveedor

|  |  |  |
| --- | --- | --- |
| **uc CU8. Gestionar proveedor**  **Administrador**  **Proveedor**  **Gestionar proveedor** | | |
| **Caso de uso** | Gestionar Proveedor | |
| **Propósito** | Agregar un nuevo proveedor, modificar un proveedor registrado, eliminar una proveedor registrado. | |
| **Resumen** | El usuario administrador podrá ingresar un nuevo proveedor al sistema. También tendrá la posibilidad de seleccionar un registro para modificar sus datos o eliminar dicho registro del sistema. | |
| **Actores** | Administrador, Proveedor | |
| **Actor inicial** | Administrador | |
| **Precondición** | El registro del proveedor debe existir en el sistema en el caso de que el administrador quiera modificar o eliminar ese registro. | |
| **Postcondición** | Los proveedores agregados quedarán registrados en el sistema. Los registros de los proveedores seleccionados para eliminar o modificar sus datos, serán actualizados o eliminados del sistema. | |
| **Flujo de Eventos** | **1. Agregar** | |
| 1.1.- Este caso de uso se origina cuando el usuario administrador desea registrar un nuevo proveedor. | 1.2.- El sistema muestra el formulario para el registro de un nuevo proveedor. |
| 1.3.- El administrador ingresa los datos requeridos: Nombre, | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
|  | dirección, teléfono, correo electrónico. |  |
| **2. Modificar** |  |
| 2.1.- Este caso de uso se da cuando el administrador desea modificar el registro de un proveedor. | 2.2.- El sistema muestra una lista con los registros de todos los proveedores. |
| 2.3.- El administrador selecciona el registro del proveedor que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** |  |
| 3.1.- Este caso de uso se da cuando el administrador desea eliminar el registro de algún proveedor que se encuentre registrado. | 3.2.- El sistema muestra una lista con los registros de todos los proveedores. |
| 3.3.- El administrador selecciona el registro de del proveedor que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |

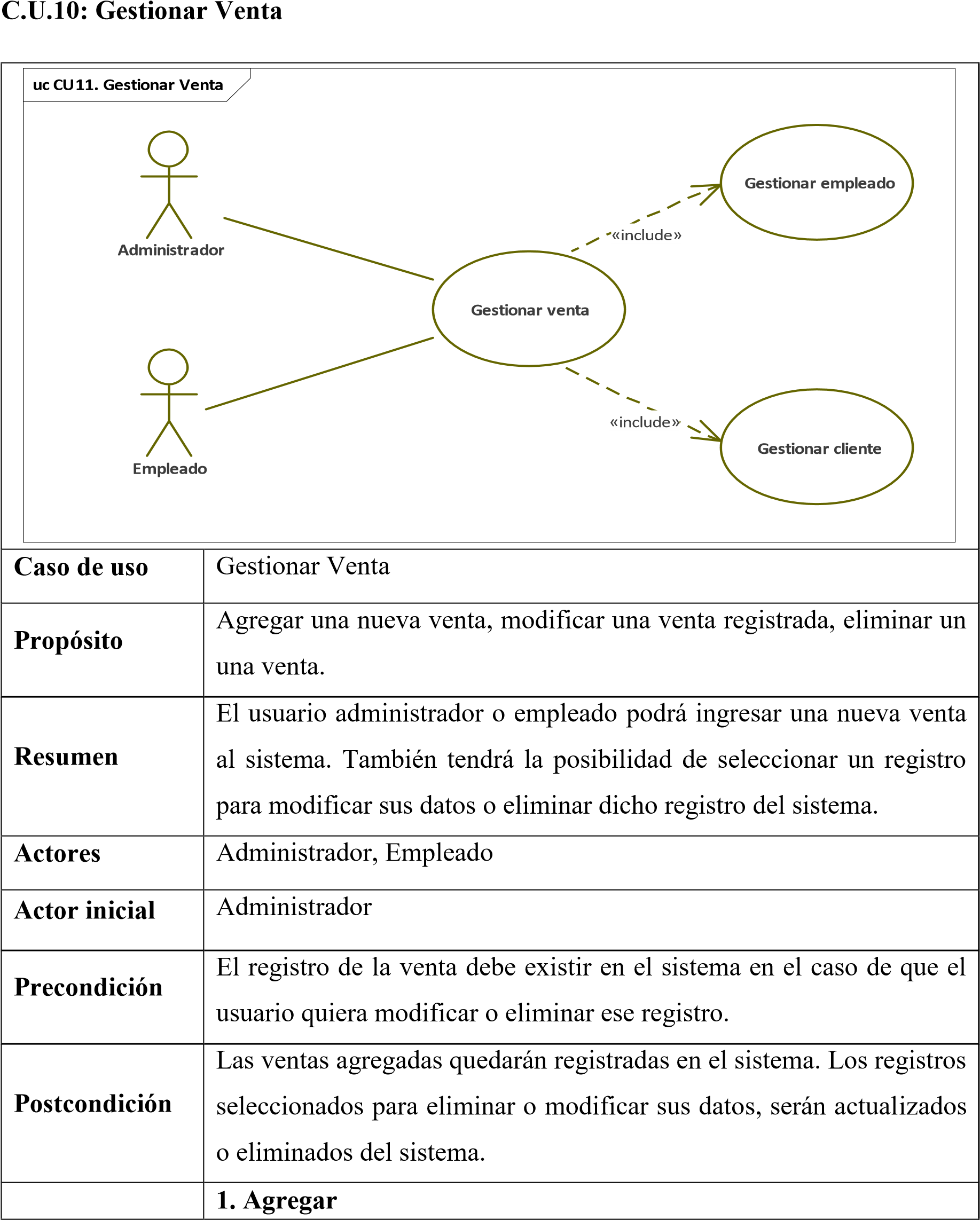


|  |  |  |
| --- | --- | --- |
|  | 1.3.- El administrador o el empleado ingresa los datos requeridos: producto a comprar, proveedor al que se le está comprando | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** | |
| 2.1.- Este caso de uso se da cuando el administrador o el empleado desea modificar el registro de una compra. | 2.2.- El sistema muestra una lista con los registros de todos las compras. |
| 2.3.- El administrador selecciona el registro de la compra que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** | |
| 3.1.- Este caso de uso se da cuando el administrador o el usuario empleado desea eliminar el registro de alguna compra ya registrada. | 3.2.- El sistema muestra una lista con los registros de todas las compras. |
| 3.3.- El administrador o el empleado selecciona el registro de la compra que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |

### C.U.9 Detalle Compra

|  |  |
| --- | --- |
|  | |
| **Caso de uso** | Gestionar Detalle Compra |
| **Propósito** | Agregar un nuevo detalle de compra, modificar un detalle de compra registrado, eliminar un detalle de compra. |
| **Resumen** | El usuario administrador o empleado podrá ingresar un nuevo detalle de compra al sistema. También tendrá la posibilidad de seleccionar un registro para modificar sus datos o eliminar dicho registro del sistema. |
| **Actores** | Administrador, Empleado |
| **Actor inicial** | Administrador |
| **Precondición** | El registro del detalle de compra debe existir en el sistema en el caso de que el usuario quiera modificar o eliminar ese registro. |
| **Postcondición** | Los detalles de compra agregados quedarán registrados en el sistema. Los registros seleccionados para eliminar o modificar sus datos, serán actualizados o eliminados del sistema. |
|  | **1. Agregar** |

|  |  |  |
| --- | --- | --- |
| **Flujo de Eventos** | 1.1.- Este caso de uso se origina cuando el usuario administrador o el empleado desea registrar un nuevo detalle de compra. | 1.2.- El sistema muestra el formulario para el registro de un nuevo detalle de compra |
| 1.3.- El administrador o el empleado ingresa los datos requeridos: producto a comprar, almacén en donde se alojará el producto, cantidad de producto a comprar, | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** | |
| 2.1.- Este caso de uso se da cuando el administrador o el empleado desea modificar el registro de un detalle de compra. | 2.2.- El sistema muestra una lista con los registros de todos los detalles de compras. |
| 2.3.- El usuario administrador o empleado selecciona el registro del detalle de compra que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** | |
| 3.1.- Este caso de uso se da cuando el administrador o el usuario empleado desea eliminar el registro de algún detalle de compra que se encuentre registrado. | 3.2.- El sistema muestra una lista con los registros de todos los detalles de compra. |
| 3.3.- El administrador o el empleado selecciona el registro de un detalle de compra que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
|  | 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |



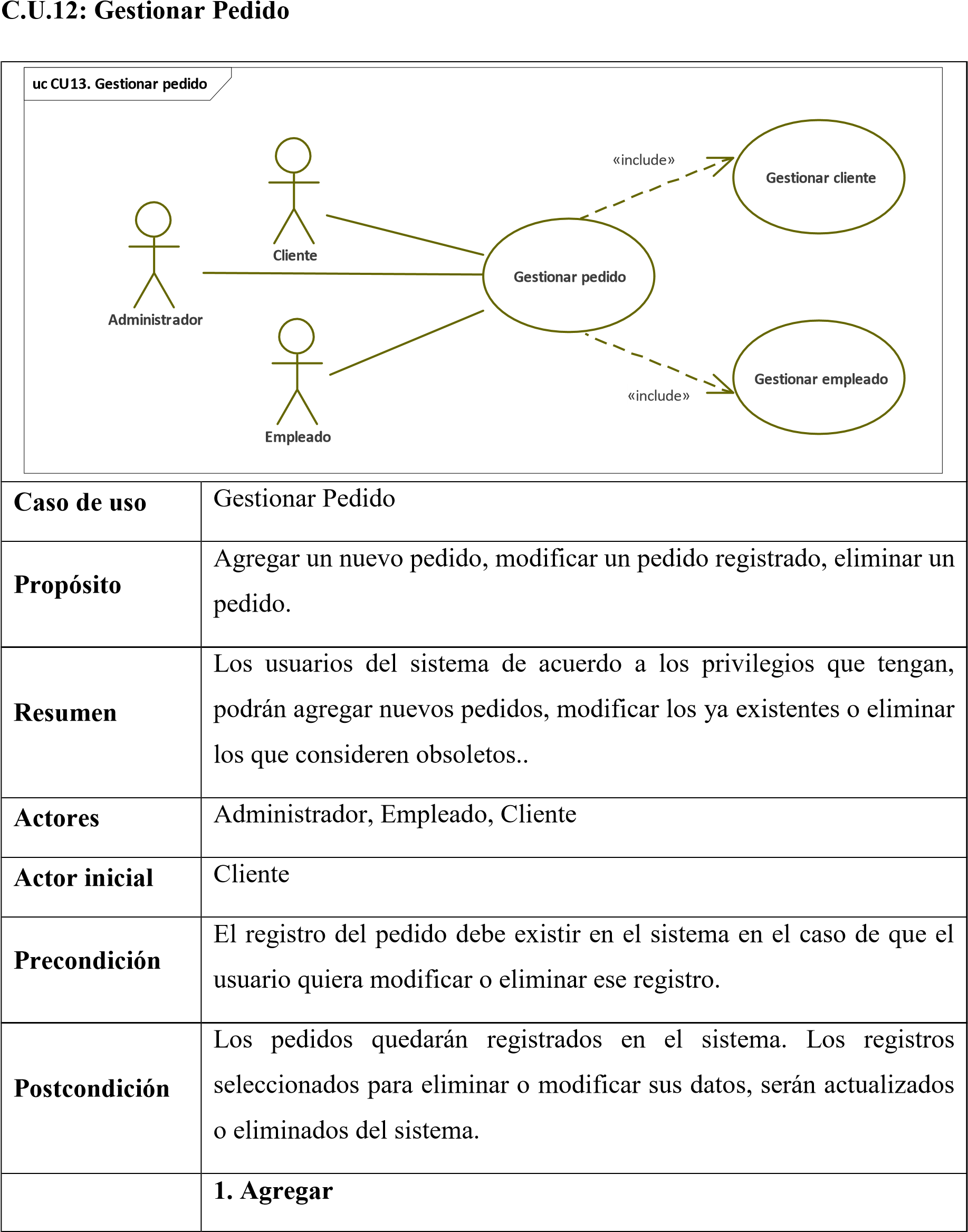
|  |  |  |
| --- | --- | --- |
| **Flujo de Eventos** | 1.1.- Este caso de uso se origina cuando el usuario administrador o el empleado desea registrar una nueva venta. | 1.2.- El sistema muestra el formulario para el registro de una nueva venta. |
| 1.3.- El administrador o el empleado ingresa los datos requeridos: producto y almacén de donde se están vendiendo los productos, el cliente a quien se realiza la venta. | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** | |
| 2.1.- Este caso de uso se da cuando el administrador o el empleado desea modificar el registro de una venta. | 2.2.- El sistema muestra una lista con los registros de todas las ventas. |
| 2.3.- El usuario administrador o empleado selecciona el registro de la venta que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** | |
| 3.1.- Este caso de uso se da cuando el administrador o el usuario empleado desea eliminar el registro de alguna venta que se encuentre registrada. | 3.2.- El sistema muestra una lista con los registros de todas las ventas |
| 3.3.- El administrador o el empleado selecciona el registro de una venta que desea eliminar. | 3.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
|  | 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |

### C.U.11: Gestionar Detalle Venta

|  |  |
| --- | --- |
|  | |
| **Caso de uso** | Gestionar Detalle Venta |
| **Propósito** | Agregar un nuevo detalle de venta, modificar un detalle de venta registrado, eliminar un detalle de venta. |
| **Resumen** | El usuario administrador o empleado podrá ingresar un nuevo detalle de venta al sistema. También tendrá la posibilidad de seleccionar un registro para modificar sus datos o eliminar dicho registro del sistema. |
| **Actores** | Administrador, Empleado |
| **Actor inicial** | Administrador |
| **Precondición** | El registro del detalle de venta debe existir en el sistema en el caso de que el usuario quiera modificar o eliminar ese registro. |
| **Postcondición** | Los detalles de venta agregados quedarán registrados en el sistema. Los registros seleccionados para eliminar o modificar sus datos, serán actualizados o eliminados del sistema. |

|  |  |  |
| --- | --- | --- |
| **Flujo de Eventos** | **1. Agregar** | |
| 1.1.- Este caso de uso se origina cuando el usuario administrador o el empleado desea registrar un nuevo detalle de venta. | 1.2.- El sistema muestra el formulario para el registro de un nuevo detalle de venta |
| 1.3.- El administrador o el empleado ingresa los datos requeridos: producto a vender, almacén de donde de donde está saliendo el producto, cantidad de producto a vender. | 1.4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** | |
| 2.1.- Este caso de uso se da cuando el administrador o el empleado desea modificar el registro de un detalle de venta. | 2.2.- El sistema muestra una lista con los registros de todos los detalles de ventas. |
| 2.3.- El usuario administrador o empleado selecciona el registro del detalle de venta que desea modificar. | 2.4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 2.5.- El administrador modifica el dato o los datos que desea. | 2.6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** | |
| 3.1.- Este caso de uso se da cuando el administrador o el usuario empleado desea eliminar el registro de algún detalle de venta que se encuentre registrado. | 3.2.- El sistema muestra una lista con los registros de todos los detalles de venta. |
| 3.3.- El administrador o el empleado selecciona el registro | 3.4.- El sistema despliega un formulario con los datos del |

|  |  |  |
| --- | --- | --- |
|  | de un detalle de venta que desea eliminar. | registro seleccionado en forma editable. |
| 3.5.- El administrador confirma la eliminación del registro. | 3.6.- El sistema procede a eliminar el registro. |



|  |  |  |  |
| --- | --- | --- | --- |
| **Flujo de Eventos** | 1.- Este caso de uso se origina cuando el usuario, desea registrar una nuevo pedido. | | 2.- El sistema muestra el formulario para el registro de un pedido. |
| 3.- El usuario ingresa los datos requeridos: Producto que desea solicitar. | | 4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** | |  |
| 1.- Este caso de uso se da cuando el usuario desea modificar el registro de un pedido. | | 2.- El sistema muestra una lista con los registros de todos los pedidos. |
| 3.- El usuario selecciona el registro de un pedido que desea modificar. | | 4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 5.- El usuario modifica el dato o los datos que desea. | | 6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** | |  |
| 1.- Este caso de uso se da cuando el usuario desea eliminar el registro de algún pedido que se encuentre registrado. | | 2.- El sistema muestra una lista con los registros de todos los pedidos. |
| 3.- El usuario selecciona el registro de un pedido que desea eliminar. | | 4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
|  | 5.- El usuario confirma eliminación del registro. | la | 6.- El sistema procede a eliminar el registro. |

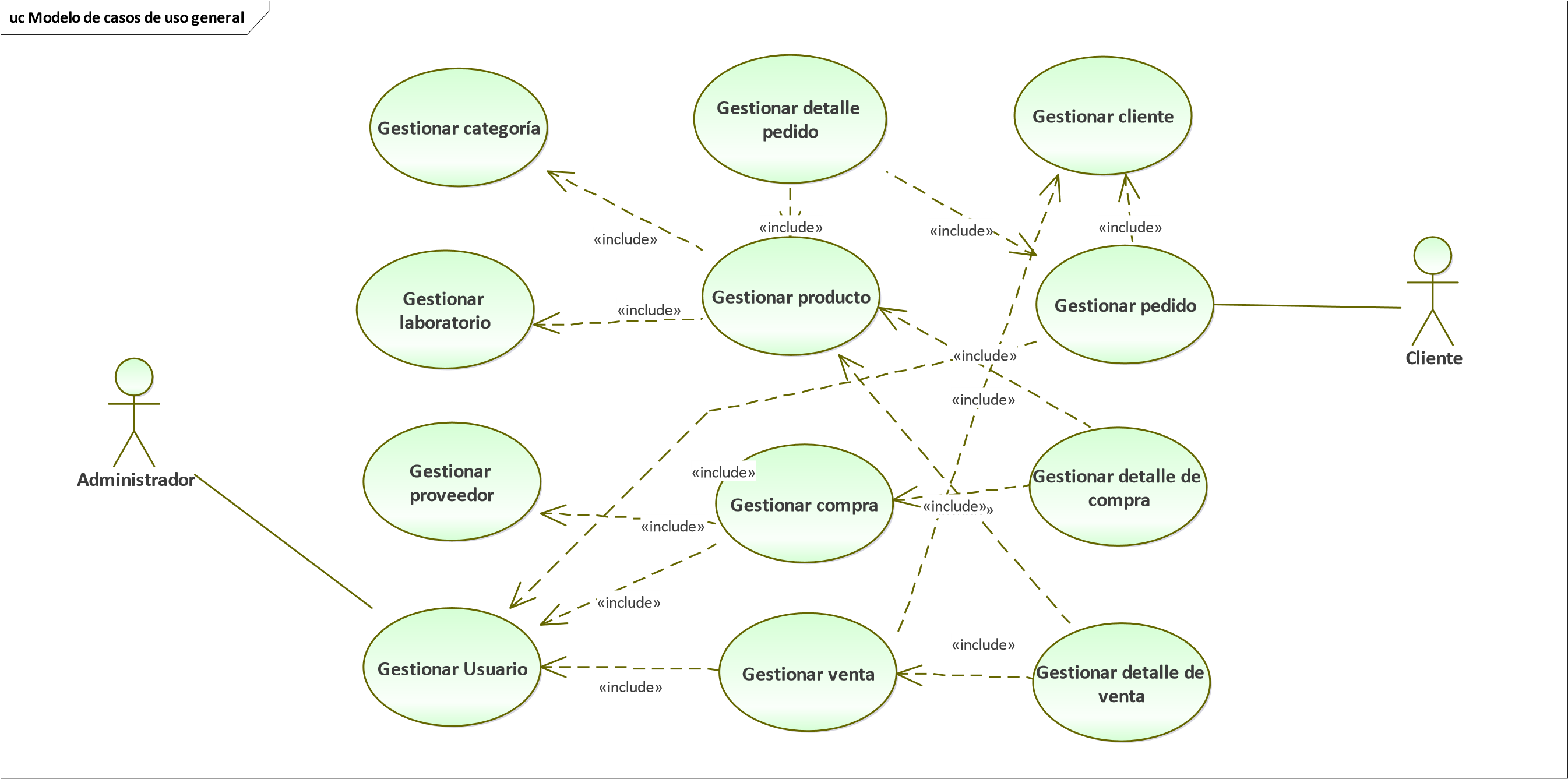
### C.U.13: Gestionar Detalle Pedido

|  |  |
| --- | --- |
|  | |
| **Caso de uso** | Gestionar Detalle Pedido |
| **Propósito** | Agregar un nuevo detalle del pedido, modificar un detalle del pedido registrado, eliminar un detalle pedido. |
| **Resumen** | Los usuarios del sistema de acuerdo a los privilegios que tengan, podrán agregar nuevos detalles de pedidos, modificar los ya existentes o eliminar los que consideren obsoletos.. |
| **Actores** | Administrador, Empleado, Cliente |
| **Actor inicial** | Cliente |
| **Precondición** | El registro del detalle del pedido debe existir en el sistema en el caso de que el usuario quiera modificar o eliminar ese registro. |

|  |  |  |
| --- | --- | --- |
| **Postcondición** | Los detalles de pedidos quedarán registrados en el sistema. Los registros seleccionados para eliminar o modificar sus datos, serán actualizados o eliminados del sistema. | |
| **Flujo de Eventos** | **1. Agregar** | |
| 1.- Este caso de uso se origina cuando el usuario desea registrar un nuevo detalle del pedido. | 2.- El sistema muestra el formulario para el registro de un nuevo detalle del pedido. |
| 3.- El usuario ingresa los datos requeridos: cantidad del producto que solicitará. | 4.- El sistema verifica que los datos sean válidos y luego los almacena. |
| **2. Modificar** | |
| 1.- Este caso de uso se da cuando el usuario desea modificar el registro de un detalle del pedido. | 2.- El sistema muestra una lista con los registros de todos los detalles del pedido. |
| 3.- El usuario selecciona el registro del detalle del pedido que desea modificar. | 4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 5.- El usuario modifica el dato o los datos que desea. | 6.- El sistema verifica los datos y si son válidos los actualiza. |
| **3. Eliminar** | |
| 1.- Este caso de uso se da cuando el usuario desea eliminar el registro de algún detalle del pedido que haya registrado. | 2.- El sistema muestra una lista con los registros de todos los detalles del pedido. |

|  |  |  |
| --- | --- | --- |
|  | 3.- El usuario selecciona el registro de un detalle del pedido que desea eliminar. | 4.- El sistema despliega un formulario con los datos del registro seleccionado en forma editable. |
| 5.- El usuario confirma la eliminación del registro. | 6.- El sistema procede a eliminar el registro. |

#### 2.1.2.5. Modelo general de casos de uso



43

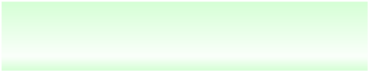
## CAPITULO 3: ANALISIS DEL SISTEMA

### 3.1. INTRODUCCIÓN

En este capítulo se muestra el modelo de análisis del sistema, el cual se deriva del análisis de los requerimientos realizados previamente.

### 3.2. IDENTIFICACIÓN DE PAQUETES DE ANÁLISIS

**class Diagrama de paquetes de casos de uso**



**Módulo Ventas**



C.U.11. Gestionar Venta

+



C.U.12. Gestionar Detalle Venta

+



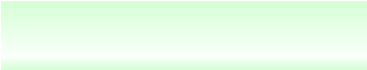
+

C.U.6. Gestionar Empleado



+

C.U.7. Gestionar Cliente



**Módulo Compras**



+

C.U.10. Gestionar Detalle Compra



C.U.6. Gestionar Empleado

+



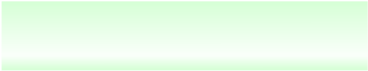
C.U.8. Gestionar Proveedor

+



C.U.9. Gestionar Compra

+



**Módulo Pedidos**



C.U.13. Gestionar Pedido

+



C.U.14. Gestionar Detalle Pedido

+



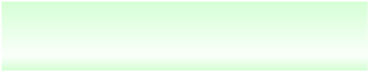
+

C.U.6. Gestionar Empleado



C.U.7. Gestionar Cliente

+



**Módulo Almacén**



C.U.2. Gestionar Categoría

+



C.U.3. Gestionar Laboratorio

+



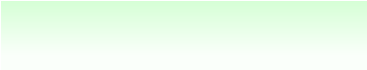
C.U.4. Gestionar Producto

+



+

C.U.5. Gestionar Almacén



**Módulo Usuarios**



C.U.1.Gestionar Usuario

+

**3.3. REALIZACIÓN DE CASOS DE USO (DIAG. DE COLABORACIÓN)**

### Diagrama colaboración: C.U.1: Gestionar Usuario

frmUsuario

clsUsuario

tblUsuario

frmBuscarUsuario

()

: Guardar

1

2

()

: GuardarUsuario

: BuscarUsuario

()

3

4

()

: BuscarUsuario

5

()

: DevolverDatos

()

6

: MostrarUsuario

()

7

: SeleccionarUsuario

8

: Modificar

()

()

: ModificarUsuario

9

: Eliminar

()

10

11

: EliminarUsuario

()

### Diagrama de colaboración: C.U.2: Gestionar Categoría

frmCategoria

clsCategoria

tblCategoria

frmBuscarCategoria

()

: Guardar

1

2

()

: GuardarCategoria

()

: BuscarCategoria

3

()

4

: BuscarCategoria

5

: DevolverDatos

()

()

: MostrarCategoria

6

: SeleccionarCategoria

()

7

8

: Modificar

()

9

: ModificarCategoria

()

10

: Eliminar

()

()

11

: EliminarCategoria

### Diagrama de colaboración: C.U.3: Gestionar Laboratorio

frmLaboratorio

clsLaboratorio

tblLaboratorio

frmBuscarLaboratorio

()

1

: Guardar

()

2

: GuardarLaboratorio

: BuscarLaboratorio

3

()

()

4

: BuscarLaboratorio

()

: DevolverDatos

5

6

: MostrarLaboratorio

()

: SeleccionarLaboratorio

()

7

: Modificar

8

()

()

: ModificarLaboratorio

9

10

()

: Eliminar

11

: EliminarLaboratorio

()

### Diagrama de colaboración: C.U.4: Gestionar Producto

frmProducto

clsProducto

tblProducto

frmBuscarCategoria

frmBuscarLaboratorio

frmBuscarUnidadMedida

clsCategoria

clsLaboratorio

clsUnidadMedida

tblLaboratorio

tblUnidadMedida

tblCategoria

()

1

: BuscarLaboratorio

()

: BuscarLaboratorio

2

()

: DevolverDatos

3

4

()

: MostrarLaboratorio

()

5

: SeleccionarLaboratorio

6

: BuscarUnidadMedida

()

: BuscarUnidadMedida

7

()

()

: DevolverDatos

8

9

()

: MostrarUnidadMedida

()

10

: SeleccionarUnidadMedida

: BuscarCategoria

()

11

12

: BuscarCategoria

()

13

: DevolverDatos

()

14

: MostrarCategoria

()

15

: SeleccionarCategoria

()

: Guardar

16

()

()

: GuardarCategoria

17

### Diagrama de colaboración: C.U.6: Gestionar Empleado

frmEmpleado

clsEmpleado

tblEmpleado

frmBuscarEmpleado

()

: Guardar

1

()

: GuardarEmpleado

2

: BuscarEmpleado

()

3

()

4

: BuscarEmpleado

()

: DevolverDatos

5

6

: MostrarEmpleado

()

7

: SeleccionarEmpleado

()

8

: Modificar

()

9

: ModificarEmpleado

()

: Eliminar

10

()

11

: EliminarEmpleado

()

### Diagrama de colaboración: C.U.7: Gestionar Cliente

frmCliente

clsCliente

tblCliente

frmBuscarCliente

()

: Guardar

1

: GuardarCliente

()

2

: BuscarCiente

3

()

()

: BuscarCliente

4

5

()

: DevolverDatos

6

: MostrarCliente

()

7

: SeleccionarCliente

()

8

: Modificar

()

()

9

: ModificarCliente

: Eliminar

()

10

()

: EliminarCliente

11

### Diagrama de colaboración: C.U.8: Gestionar Proveedor

frmProveedor

clsProveedor

tblProveedor

frmProveedor

()

: Guardar

1

()

: GuardarProveedor

2

()

3

: BuscarProveedor

()

4

: BuscarProveedor

: DevolverDatos

5

()

6

: MostrarProveedor

()

7

: SeleccionarProveedor

()

8

: Modificar

()

9

: ModificarProveedor

()

: Eliminar

()

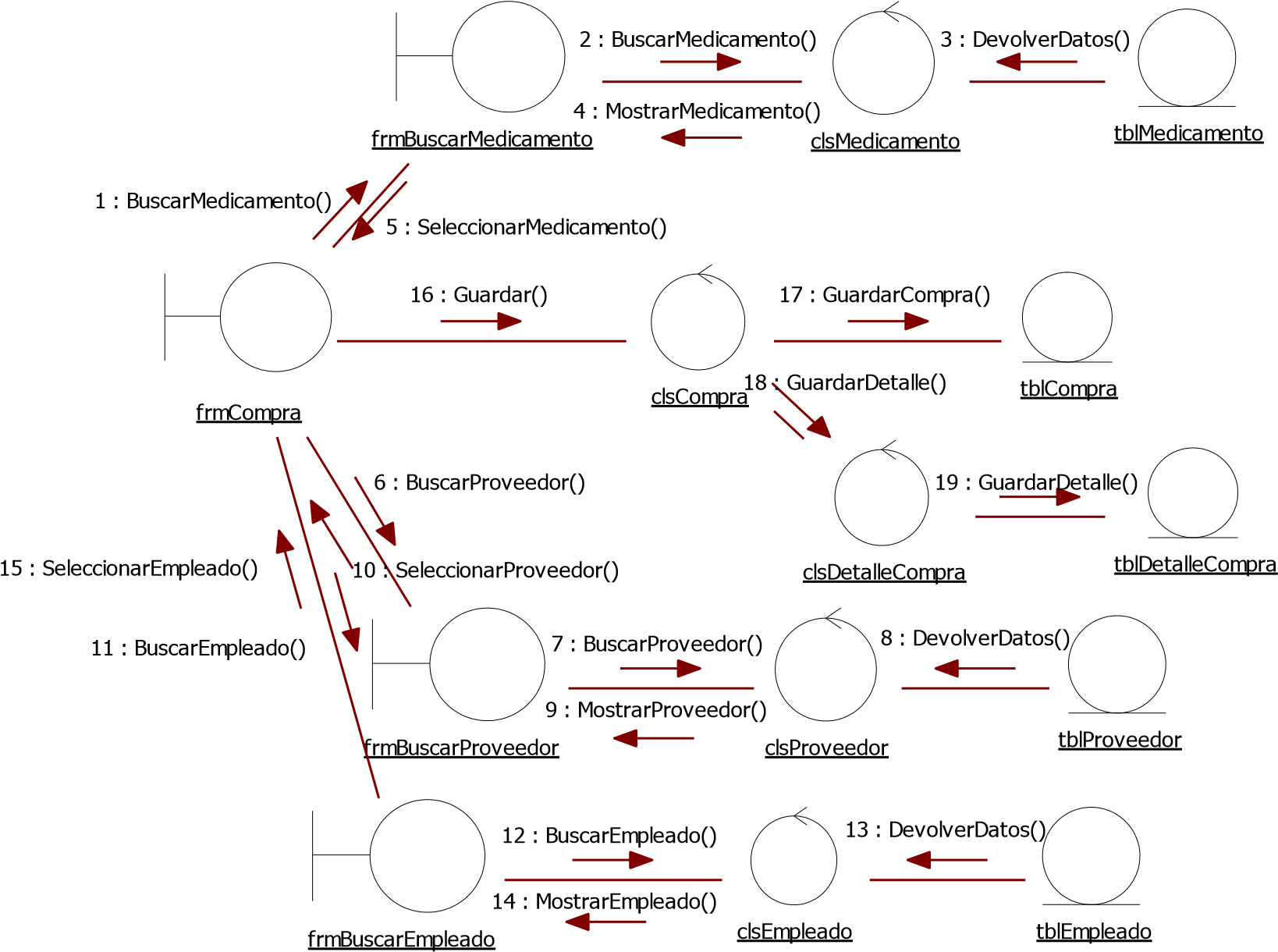
10

11

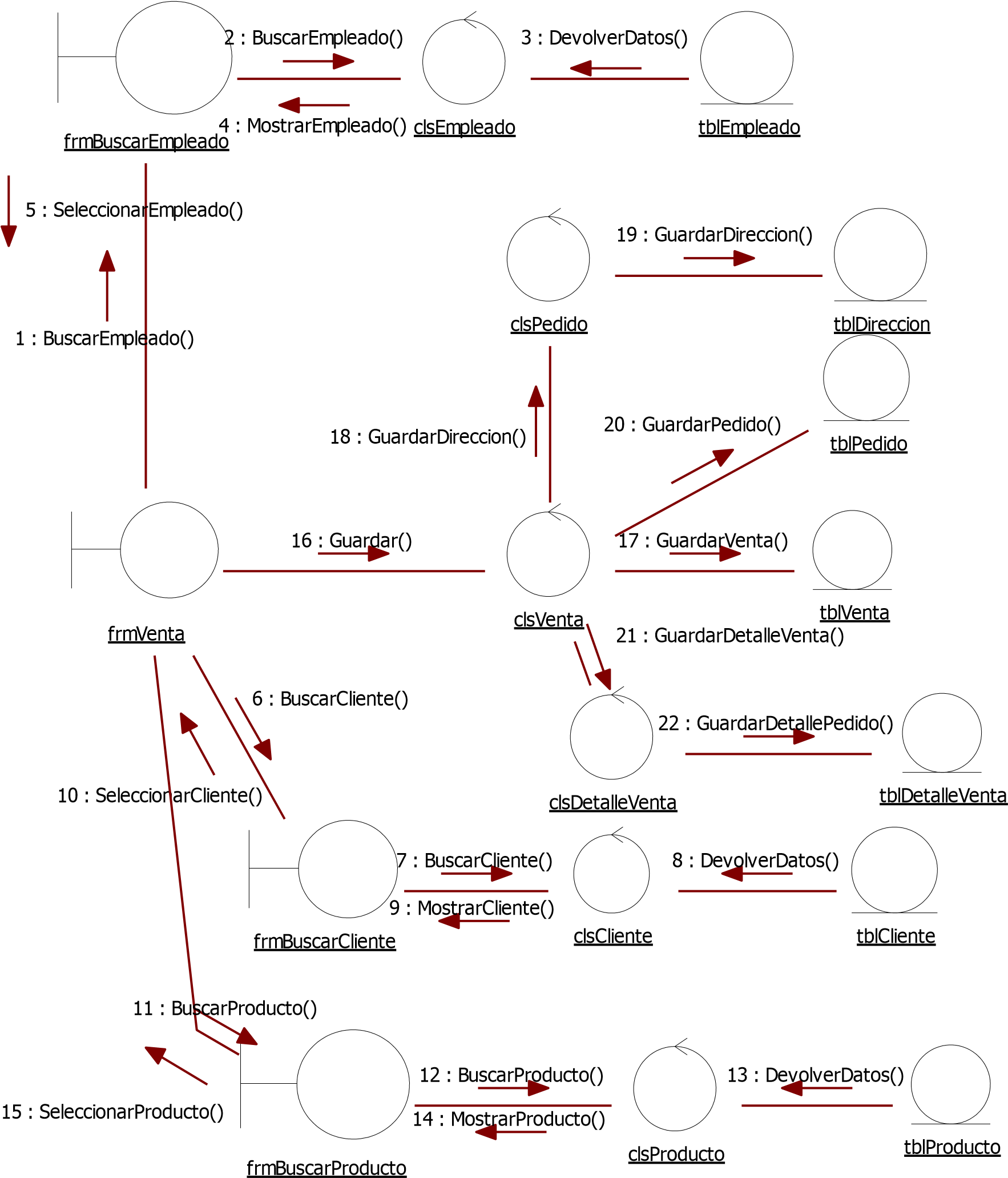
: EliminarProveedor

()

### Diagrama de colaboración: C.U.9: Gestionar Compra



### Diagrama de colaboración: C.U.11: Gestionar Venta



## CAPITULO 4: DISEÑO DEL SISTEMA

### 4.1. INTRODUCCIÓN

En este capítulo se muestra el modelo de diseño del sistema el cual se deriva del modelo de análisis realizado previamente.

El modelo de diseño se realizó utilizando el Proceso Unificado de Desarrollo de Software (PUDS).

### 4.2. ARQUITECTURA DEL SISTEMA (DIAG. DE PAQUETES E INTERFACES)

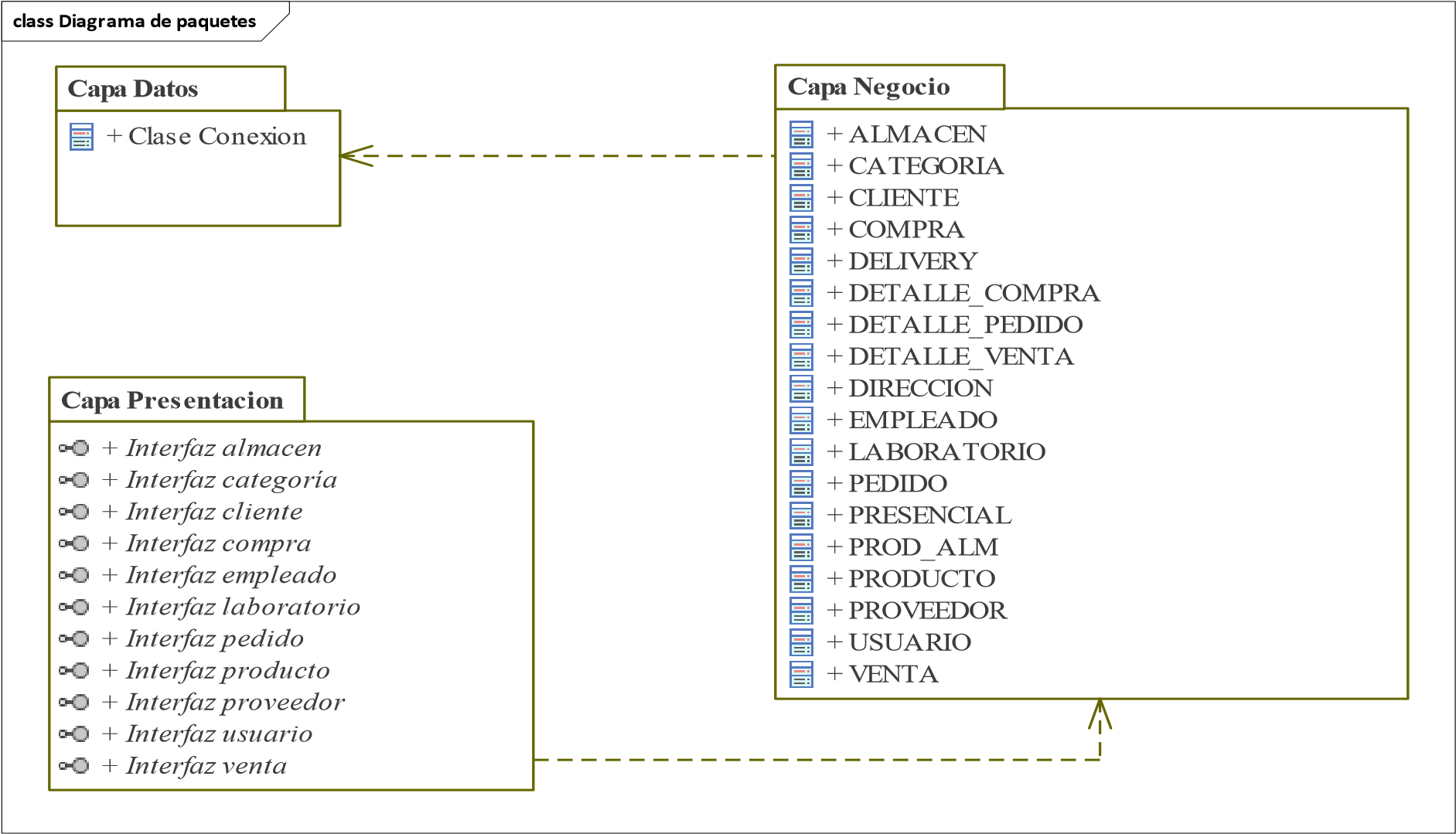
**4.2.1. Diagrama de paquetes organizado en capas.**

Las capas que se muestra en el diagrama son los siguientes:

**La capa de datos:** Esta capa se encarga de guardar los datos, en esta capa se gestiona la conexión a la base de datos, así como también la creación, edición y eliminación de los datos de la misma.

**La capa negocio:** Esta capa se encarga de gestionar la lógica del sistema, Esta capa está conectada con la capa de datos para realizar sus funciones.

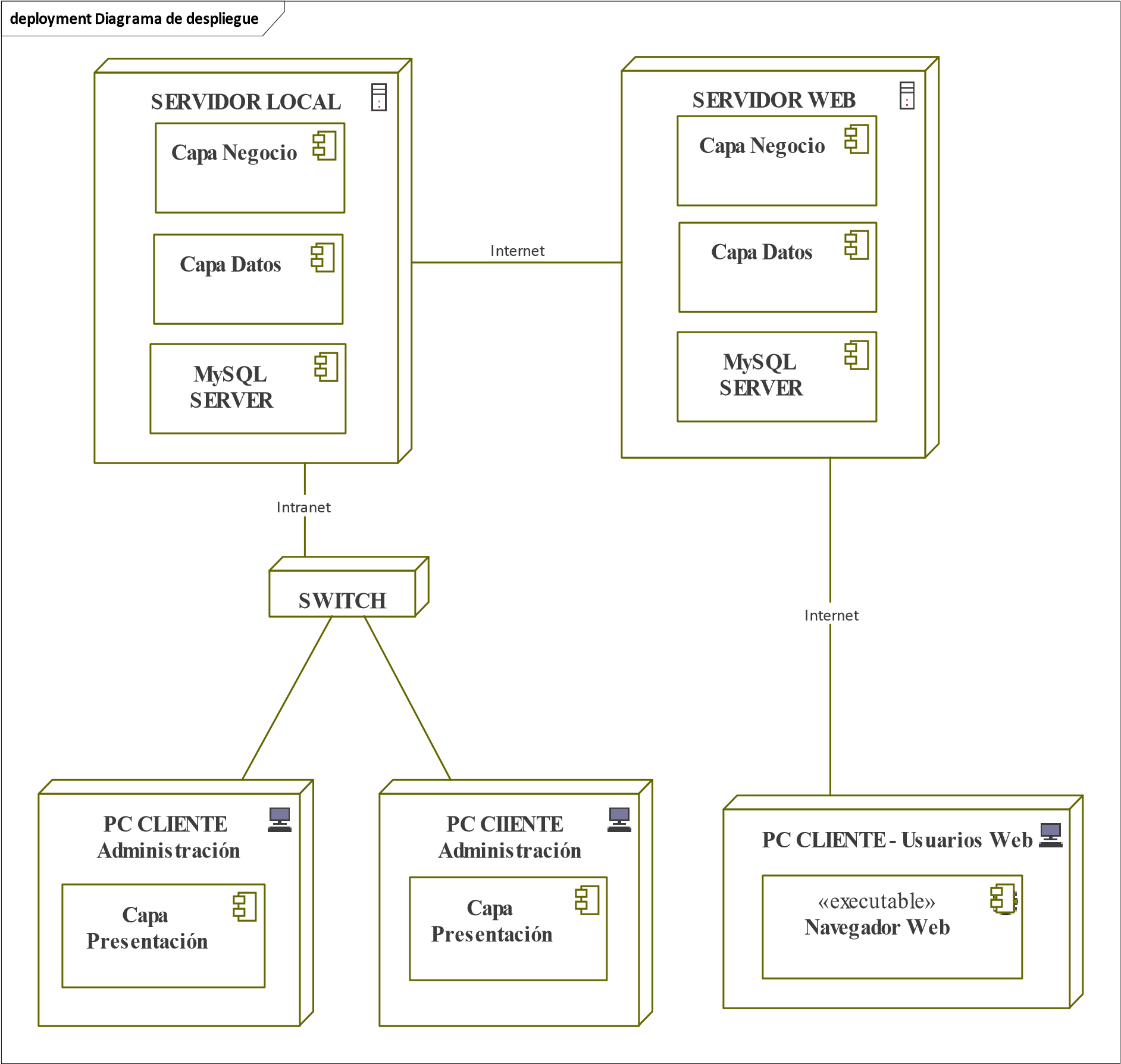
**La capa presentación:** En esta capa se encuentra las interfaces de usuario del sistema. Su función es pasarle las acciones que realice el usuario a la capa negocio.



**4.2.2. Interfaces**

### 4.3. DIAGRAMA DE DESPLIEGUE (HARDWARE)

El modelo de despliegue que podemos observar en la siguiente figura muestra los nodos que componen la configuración de la red en la que será ejecutado el sistema.



### 4.4. DISEÑO DE LOS CASOS DE USO

**4.4.1. Diagramas de secuencia para cada caso de uso**

### Diagrama de secuencia: C.U.1: Gestionar usuario

frmUsuario

clsUsuario

tblUsuario

frmBuscarUsuario

: Guardar

()

1

()

: GuardarUsuario

2

3

()

: BuscarUsuario

()

: BuscarUsuario

4

5

: DevolverDatos

()

6

: MostrarUsuario

()

7

: SeleccionarUsuario

()

8

: Modificar

()

9

: ModificarUsuario

()

10

: Eliminar

()

11

: EliminarUsuario

()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 1 | **Caso de Uso:** | Gestionar Usuario |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el cliente crea un usuario web y este se añade a la base de datos | | | |

### Diagrama de secuencia: C.U.2: Gestionar categoría

frmCategoria

clsCategoria

tblCategoria

frmBuscarCategoria

: Guardar

()

1

()

: GuardarCategoria

2

3

()

: BuscarCategoria

()

: BuscarCategoria

4

5

: DevolverDatos

()

6

: MostrarCategoria

()

7

: SeleccionarCategoria

()

8

: Modificar

()

9

: ModificarCategoria

()

10

: Eliminar

()

11

: EliminarCategoria

()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 2 | **Caso de Uso:** | Gestionar Categoría |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el usuario crea una nueva categoría y este se añade a la base de datos | | | |

### Diagrama de secuencia: C.U.3: laboratorio

frmLaboratorio

clsLaboratorio

tblLaboratorio

frmBuscarLaboratorio

: Guardar

()

1

()

: GuardarLaboratorio

2

3

()

: BuscarLaboratorio

()

: BuscarLaboratorio

4

5

: DevolverDatos

()

6

: MostrarLaboratorio

()

7

: SeleccionarLaboratorio

()

8

: Modificar

()

9

: ModificarLaboratorio

()

10

: Eliminar

()

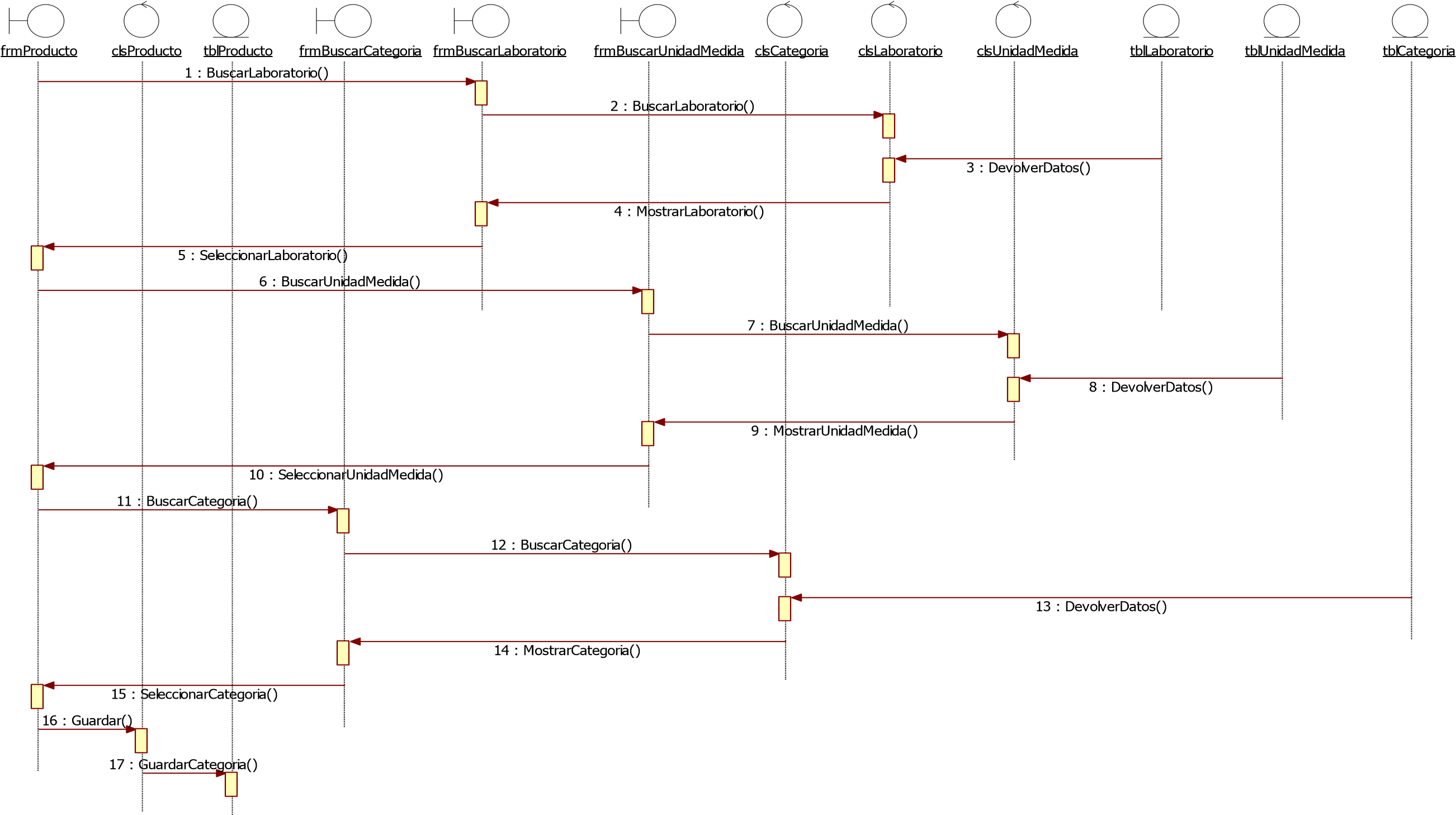
11

: EliminarLaboratorio

()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 3 | **Caso de Uso:** | Gestionar Laboratorio |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el empleado crea un laboratorio y este se añade a la base de datos. | | | |

### Diagrama de secuencia: C.U.4: Gestionar producto



|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 4 | **Caso de Uso:** | Gestionar Producto |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el empleado añade un producto a la base de datos. | | | |

55

### Diagrama de secuencia: C.U.5: empleado

frmEmpleado

clsEmpleado

tblEmpleado

frmBuscarEmpleado

: Guardar

()

1

()

: GuardarEmpleado

2

3

()

: BuscarEmpleado

()

: BuscarEmpleado

4

5

: DevolverDatos

()

6

: MostrarEmpleado

()

7

: SeleccionarEmpleado

()

8

: Modificar

()

9

: ModificarEmpleado

()

10

: Eliminar

()

11

: EliminarEmpleado

()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 5 | **Caso de Uso:** | Gestionar Empleado |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el usuario crea un empleado y este se añade a la base de datos | | | |

### Diagrama de secuencia: C.U.6: cliente

frmCliente

clsCliente

tblCliente

frmBuscarCliente

: Guardar

()

1

()

: GuardarCliente

2

3

()

: BuscarCiente

()

: BuscarCliente

4

5

: DevolverDatos

()

6

: MostrarCliente

()

7

: SeleccionarCliente

()

8

: Modificar

()

9

: ModificarCliente

()

10

: Eliminar

()

11

: EliminarCliente

()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 6 | **Caso de Uso:** | Gestionar Cliente |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el usuario crea un nuevo cliente y este se añade a la base de datos | | | |

### Diagrama de secuencia: C.U.7: proveedor

frmProveedor

clsProveedor

tblProveedor

frmProveedor

: Guardar

()

1

()

: GuardarProveedor

2

3

()

: BuscarProveedor

()

: BuscarProveedor

4

5

: DevolverDatos

()

6

: MostrarProveedor

()

7

: SeleccionarProveedor

()

8

: Modificar

()

9

: ModificarProveedor

()

10

: Eliminar

()

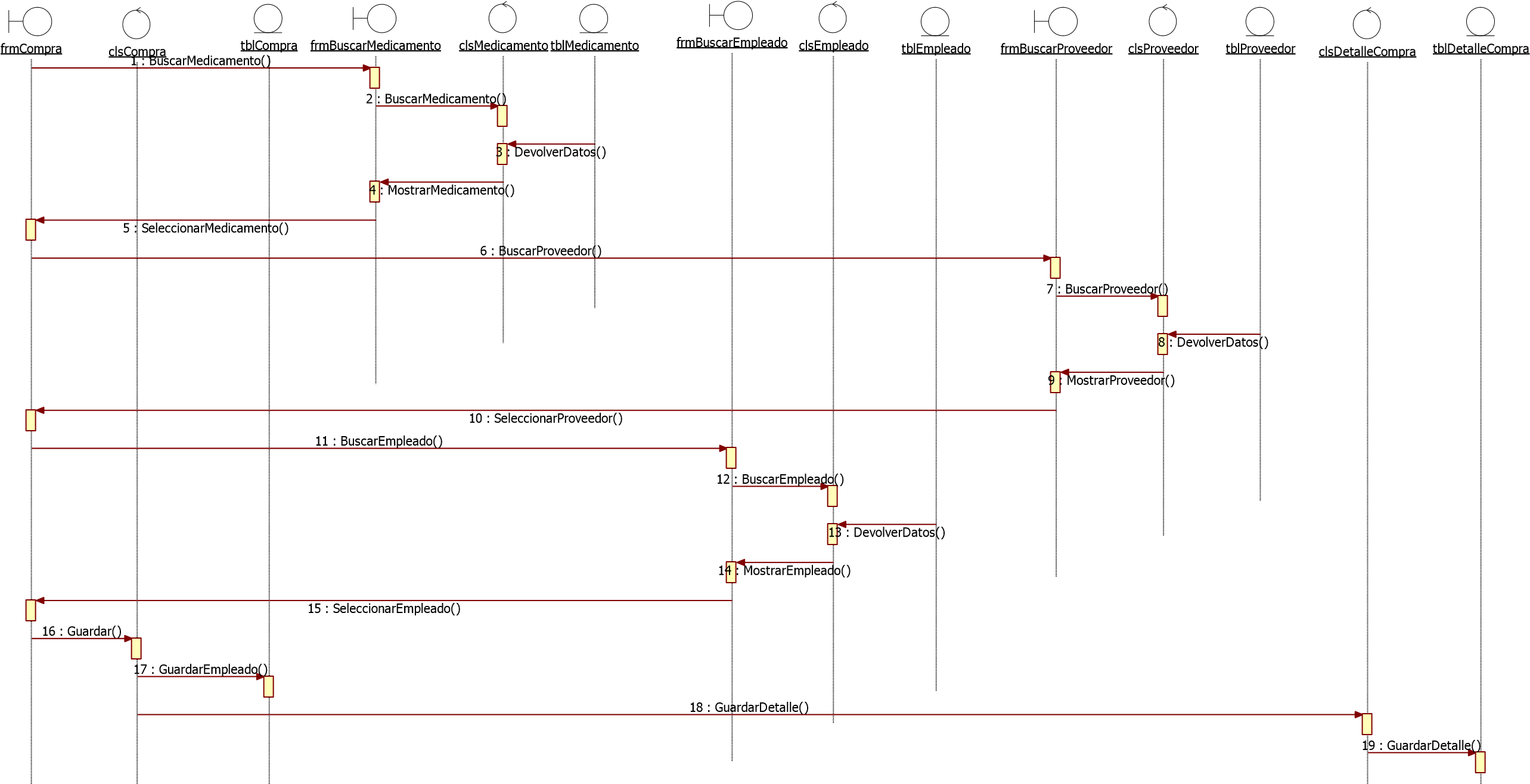
11

: EliminarProveedor

()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 7 | **Caso de Uso:** | Gestionar Proveedor |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el usuario crea un Proveedor y este se añade a la base de datos | | | |

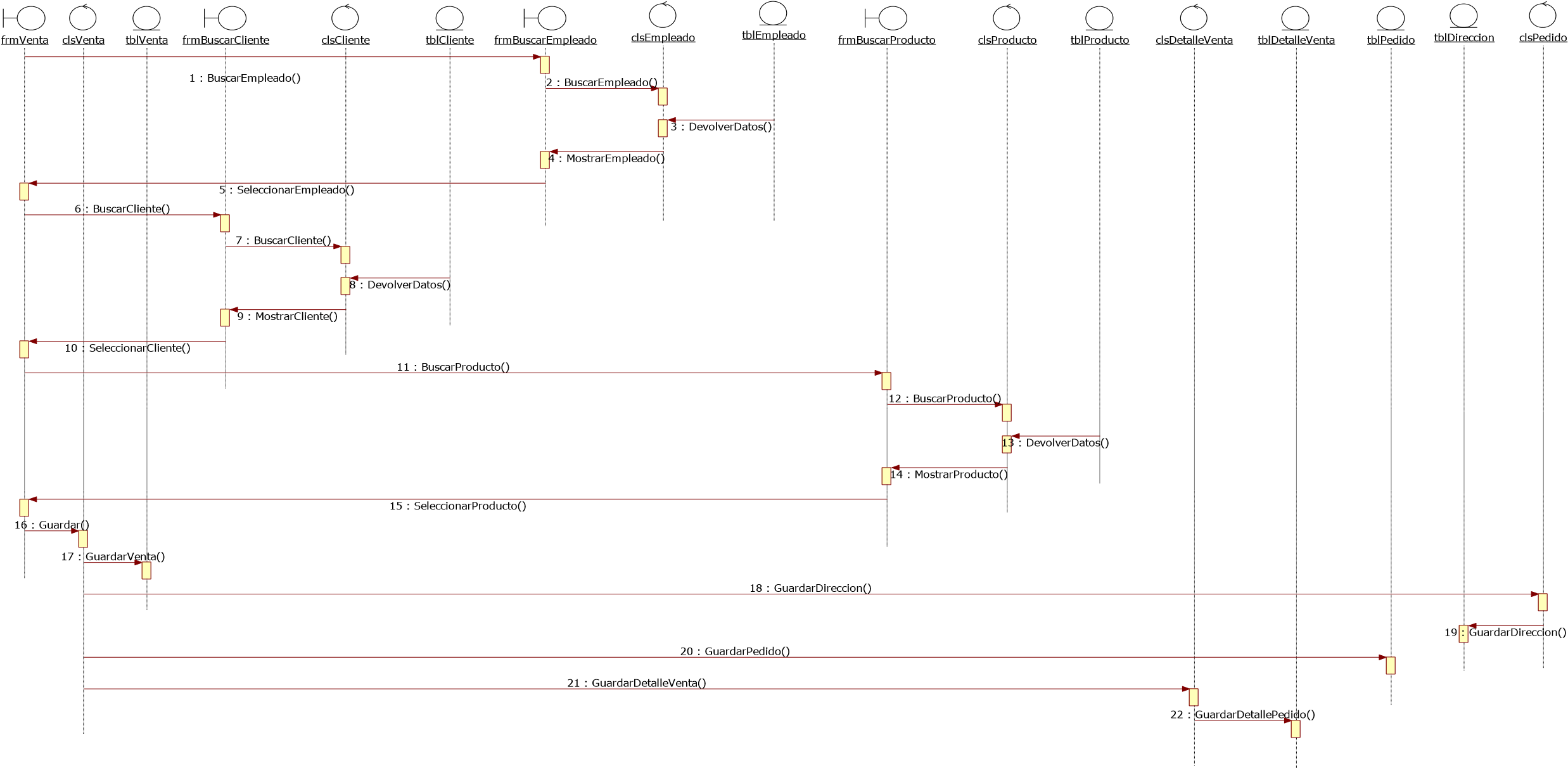
### Diagrama de secuencia: C.U.8: Gestionar compra



|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 8 y 9 | **Casos de Usos:** | Gestionar compra |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el usuario crea una compra y esta se añade a la base de datos | | | |

59

### Diagrama de secuencia: C.U.10: Gestionar venta



|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre:** | Diagrama de Secuencia 10 y 11 | **Caso de Uso:** | Gestionar venta y detalle venta |
| **Descripción** | | | |
| El diagrama muestra la secuencia de pasos que suceden cuando el cliente crea una venta y esta se añade a la base de datos. | | | |

60

### 4.5 DIAGRAMA DE CLASES

**class Diagrama de clases**

**CATEGORIA**

-

id\_Categoria: int

+

nombre: varchar

**LABORATORIO**

-

id\_laboratorio: int

+

nombre: varchar

**PRODUCTO**

-

id\_producto: int

+

nombre: varchar

+

precio: float

+

fecha\_vencimiento\_producto: date

+

stock: int

+

imagen: longblob

**COMPRA**

-

id\_compra: int

+

total: float

+

fecha: date

**DETALLE\_COMPRA**

+

cantidad: int

+

precio: float

**PROVEEDOR**

-

id\_proveedor: int

+

nombre: vachar

+

direccion: varchar

+

telefono: int

+

mail: varchar

**EMPLEADO**

-

id\_empleado: bigint

+

nombre: varchar

+

user: varchar

+

apellidos: varchar

+

telefono: varchar

+

pwd: varchar

+

cargo: varchar

+

stado: varchar

**VENTA**

-

id\_venta: int

+

total: float

+

fecha: date

+

tipo: varchar

**DETALLE\_VENTA**

+

cantidad: int

+

precio: float

**CLIENTE**

-

id\_cliente: bigint

+

nombre: varchar

+

apellido: varchar

+

telefono: varchar

+

user\_name: varchar

+

passwdr: varchar

+

std\_cliente: varchar

**PEDIDO**

+

estado: varchar

+

fecha\_entega: date

**DIRECCION**

-

id\_ubicacion: bigint

+

latitud: double

+

longitud: doble

+

lin\_ubi: text

+

referencia\_ubi: text

**UNIDADMEDIDA**

-

id\_unidadmedida: bigint

+

nombre: varchar

1

1..\*

1

0..\*

1

0..\*

1..\*

1

1..\*

1..\*

1..\*

1

1..\*

1

1

1..\*

1

1..\*

1..\*

1..\*

### 4.6. DISEÑO LÓGICO DE LA BASE DE DATOS (CAMPOS)

cliente

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id\_cliente FK | nombre | apellido | telefono | user | pwd | estado |

laboratorio

|  |  |
| --- | --- |
| id\_laboratorio FK | nombre |

unidadmedida

|  |  |  |
| --- | --- | --- |
| id\_unidadmedida FK | abreviatura | nombre |

Categoría

|  |  |  |
| --- | --- | --- |
| id\_categoria FK | nombre | descripcion |

Producto

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id\_producto FK | nombre | descripcion | precio | stock | Fecha\_vencimiento | Laboratorio FK | Unidadmedida FK | Categoría FK | imagen |

Empleado

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id\_empleado FK | nombre | apellidos | direccion | user | pwd | cargo | estado |

Nota\_venta

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id\_venta FK | fecha | total | tipo | Empleado FK | Cliente FK |

Compra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_compra FK | fecha | total | Empleado FK | Proveedor FK |

|  |  |
| --- | --- |
| id\_venta FK FK | nit |

Presencial

Ubicación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_ubicacion FK | latitud | longitud | link\_ubi | referencia\_ubi |

Pedido

|  |  |  |  |
| --- | --- | --- | --- |
| id\_venta FK | fecha\_entrega | estado | ubicación FK |

DetalleVenta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_detalle FK | id\_venta FK | cantidad | precioV | Producto FK |

DetalleCompra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_detalle FK | id\_compra FK | cantidad | precioC | Producto FK |

### 4.7. DISEÑO FÍSICO DE LA BASE DE DATOS (TABLAS Y CAMPOS)

**DROP DATABASE IF EXISTS** farmacia\_san\_pedro; **CREATE DATABASE** farmacia\_san\_pedro;

**use** farmacia\_san\_pedro;

**DROP TABLE IF EXISTS** cliente; **CREATE TABLE** cliente(

id\_cliente bigint **PRIMARY** **KEY** **auto\_increment**, nombre\_cliente **varchar**(200), apellidos\_cliente **varchar**(200), telefono\_cliente **varchar**(8), user\_cliente **varchar**(30) not null, pwd\_cliente **varchar**(250) not null,

std\_cliente **varchar**(10) DEFAULT 'ACTIVO'

);

**DROP TABLE IF EXISTS** laboratorio; **CREATE TABLE** laboratorio(

id\_laboratorio **bigint** **PRIMARY** **KEY** **auto\_increment**, nombre **varchar**(200)

);

**DROP TABLE IF EXISTS** unidadMedida; **CREATE TABLE** unidadMedida(

id\_unidadMedida **bigint PRIMARY KEY** **auto\_increment**,

abreviatura **varchar**(200),

nombre **varchar**(200)

);

**DROP TABLE IF EXISTS** categoria; **CREATE TABLE** categoria(

id\_categoria **bigint PRIMARY KEY** **auto\_increment**, nombre **varchar**(200),

descripcion **varchar**(200)

);

**DROP TABLE IF EXISTS** proveedor; **CREATE TABLE** provedor(

id\_proveedor **bigint PRIMARY KEY** **auto\_increment**, nombre\_proveedor **varchar**(200), telefono\_proveedor **varchar**(500), descripcion\_provedor **varchar**(200), mail\_proveedor **varchar**(900)

);

**DROP TABLE IF EXISTS** producto; **CREATE TABLE** producto(

id\_producto **bigint PRIMARY KEY** **auto\_increment**, nombre\_producto **varchar**(200), descripcion\_producto **varchar**(200), precio\_producto **float**, stock\_producto **int**, fecha\_vencimiento\_producto **date**,

laboratorio **bigint**, unidadMedida **bigint**, categoria **bigint**, imagen **longblob**,

**FOREIGN KEY** (laboratorio) **REFERENCES** laboratorio(id\_laboratorio),

**FOREIGN KEY** (unidadMedida) **REFERENCES** unidadMedida(id\_unidadMedida),

**FOREIGN KEY** (categoria) **REFERENCES** categoria(id\_categoria)

);

**DROP TABLE IF EXISTS** empleado; **CREATE TABLE** empleado(

id\_empleado **bigint PRIMARY KEY** **auto\_increment**, nombre\_empleado **varchar**(200), apellidos\_empleado **varchar**(200), direccion\_empleado **varchar**(200), telefono\_empleado **varchar**(8), user\_empleado **varchar**(30) unique, pwd\_empleado **varchar**(250) not null,

cargo\_empleado **varchar**(250) not null comment 'ADMIN-PRODUCCION-DELIVERY', std\_empleado **varchar**(10) **DEFAULT** 'ACTIVO'

);

**DROP TABLE IF EXISTS** notaVenta; **CREATE TABLE** notaVenta(

id\_venta **bigint PRIMARY KEY** **auto\_increment**, fecha **timestamp**, total **float** **default** 0, tipo **varchar**(20), empleado **bigint default** null,

cliente **bigint**,

**FOREIGN** **KEY** (empleado) **REFERENCES** empleado(id\_empleado),

**FOREIGN** **KEY** (cliente) **REFERENCES** cliente(id\_cliente)

);

**DROP TABLE IF EXISTS** compra; **CREATE** **TABLE** compra(

id\_compra **bigint PRIMARY KEY** **auto\_increment**, fecha **timestamp**, total **float default** 0, empleado **bigint default** null,

proveedor **bigint**,

**FOREIGN KEY** (empleado) **REFERENCES** empleado(id\_empleado),

**FOREIGN KEY** (proveedor) **REFERENCES** provedor(id\_proveedor)

);

**DROP TABLE IF EXISTS** presencial; **CREATE TABLE** presencial( id\_venta bigint, NIT varchar(50),

**PRIMARY KEY** (id\_venta),

**FOREIGN KEY**(id\_venta) **REFERENCES** notaVenta(id\_venta)

);

**DROP TABLE IF EXISTS** ubicacion; **CREATE TABLE** ubicacion(

id\_ubicacion **bigint** **PRIMARY KEY** **auto\_increment**, latitud **double DEFAULT** 0, longitud **double DEFAULT** 0,

link\_ubi **text**,

referencia\_ubi **text**

);

**DROP TABLE IF EXISTS** pedido; **CREATE TABLE** pedido( id\_venta **bigint** , fecha\_entrega **timestamp**,

estado **varchar**(50) **DEFAULT** 'PENDIENTE' comment 'PENDIENTE/ENTREGADO/CANCELADO', ubicacion **bigint**,

**PRIMARY KEY** (id\_venta),

**FOREIGN KEY**(id\_venta) **REFERENCES** notaVenta(id\_venta),

**FOREIGN KEY**(ubicacion) **REFERENCES** ubicacion(id\_ubicacion)

);

**DROP TABLE IF EXISTS** detalleVenta; **CREATE TABLE** detalleVenta( id\_detalle **bigint auto\_increment**, id\_venta **bigint**, cantidad **int**, precioV **float**, producto **bigint**,

**PRIMARY KEY** (id\_detalle,id\_venta),

**FOREIGN KEY** (id\_venta) **REFERENCES** notaVenta(id\_venta),

**FOREIGN KEY** (producto) **REFERENCES** producto(id\_producto)

);

**DROP TABLE IF EXISTS** detalleCompra; **CREATE TABLE** detalleCompra( id\_detalle **bigint auto\_increment**, id\_compra **bigint**, cantidad **int**, precioV **float**, producto **bigint**,

**PRIMARY KEY** (id\_detalle,id\_compra),

**FOREIGN KEY** (id\_compra) **REFERENCES** compra(id\_compra),

**FOREIGN KEY** (producto) **REFERENCES** producto(id\_producto)

);

/\*LABORATORIO\*/

INSERT INTO laboratorio (`nombre`) VALUES ('arcor');

/\*CATEGORIA\*/

INSERT INTO categoria (`nombre`,`descripcion`) VALUES ('antibioticos','bajo receta medica');

/\*UNIDAD MEDIDA\*/

INSERT INTO unidadmedida (`abreviatura`,`nombre`) VALUES ('gr','gramos');

/\*PRODUCTO\*/

INSERT INTO `producto` (`id\_producto`, `nombre\_producto`, `descripcion\_producto`,

`precio\_producto`, `stock\_producto`, `fecha\_vencimiento\_producto`, `laboratorio`, `unidadMedida`,

`categoria`) VALUES (NULL, 'Parcetamol', 'cada 8 horas', '20', '50', '2019-01-30', '1', '1', '1');

/\*lista todo los productos

SELECT id\_producto,nombre\_producto,descripcion\_producto,precio\_producto, stock\_producto, fecha\_vencimiento\_producto,laboratorio.nombre laboratorio, categoria.nombre categoria, unidadmedida.abreviatura UMedida

FROM producto,laboratorio,categoria,unidadMedida where producto.laboratorio = laboratorio.id\_laboratorio and producto.categoria = categoria.id\_categoria and producto.unidadMedida = unidadmedida.id\_unidadMedida;

\*/

/\* VISTA PARA MOSTRAR LOS PEDIDOS \*/

DROP VIEW IF EXISTS nota;

CREATE VIEW nota AS

SELECT p.id\_venta id\_pedido,n.fecha fecha\_pedido,p.fecha\_entrega, n.total,p.estado std\_pedido, u.referencia\_ubi referencia,u.link\_ubi link, concat(c.nombre\_cliente,' ',c.apellidos\_cliente) cliente,c.telefono\_cliente FROM pedido p,notaventa n,ubicacion u, cliente c WHERE

p.id\_venta=n.id\_venta and n.cliente = c.id\_cliente and n.tipo='PEDIDO' and p.ubicacion =

u.id\_ubicacion GROUP BY n.id\_venta;

/\* EMPLEADO ADMINISTRADOR \*/

INSERT INTO `empleado` (`id\_empleado`, `nombre\_empleado`, `apellidos\_empleado`,

`direccion\_empleado`, `telefono\_empleado`, `user\_empleado`, `pwd\_empleado`, `cargo\_empleado`,

`std\_empleado`) VALUES (NULL, 'admin', 'admin', NULL, '12334567', 'admin', '123', 'ADMIN', 'ACTIVO');

DELIMITER $$

CREATE TRIGGER `edit\_stock\_prod` AFTER INSERT ON `detalleVenta` FOR EACH ROW BEGIN

UPDATE producto SET stock\_producto=stock\_producto-new.cantidad WHERE new.producto=id\_producto;

END

$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER `edit\_stock\_prod` AFTER INSERT ON `detalleCompra` FOR EACH ROW BEGIN UPDATE producto SET stock\_producto=stock\_producto+new.cantidad WHERE new.producto=id\_producto;

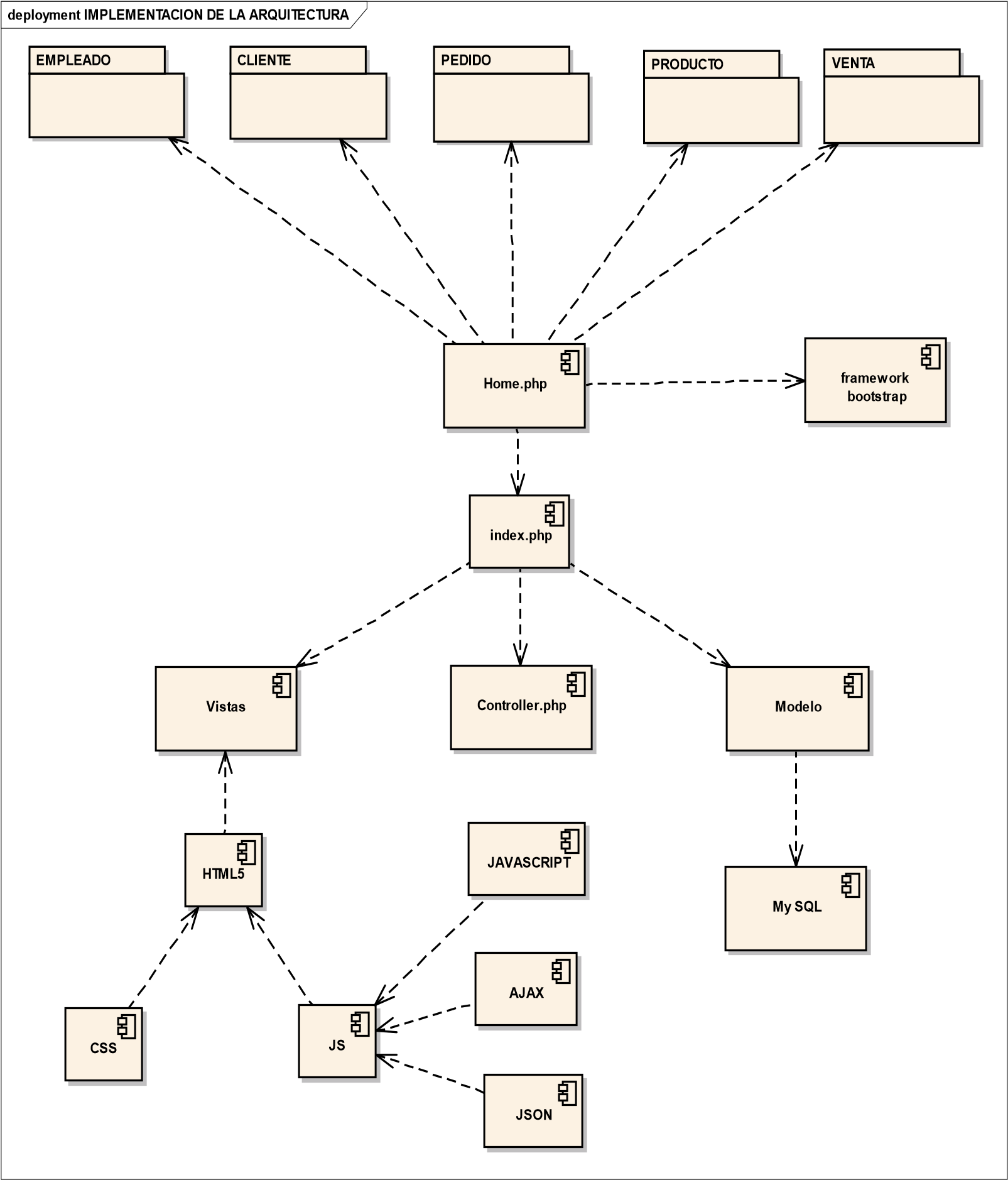
END

$$

DELIMITER ;

### 5. IMPLEMENTACION

El modelo de implementación describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como dicheros de código fuente, ejecutables, etc. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entono de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros. **Implementación de la Arquitectura**



#### Selección del entorno de desarrollo para la implementación

**XAMPP**

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas.

#### Características y requisitos

* XAMPP solamente requiere descargar y ejecutar un archivo zip, tar, o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará.
* XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.
* XAMPP es utilizado actualmente para servidor de sitios Web y, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes. Es básico para un programador web.

#### MySQL

Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuidos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresa u organismos administrativos. MySQL incluye todos los elementos necesarios para instalar el programa, prepara diferentes niveles de acceso de usuario, administrara el sistema, proteger y hacer volcados de datos.

Utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases relacionales. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos. **Editor de base de datos**

**phpMyAdmin:** Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar, y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. Se encuentra disponible bajo la licencia GPL.

#### Framework Bootstrap

Es una biblioteca multiplataforma o conjunto de herramientas de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en [HTML](https://es.wikipedia.org/wiki/HTML) y [CSS,](https://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada) así como extensiones de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo [front-end.](https://en.wikipedia.org/wiki/Front-end_web_development)

#### PHP

Es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

#### JAVA SCRIP

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora afectos como texto que parece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

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

#### JQUERY

Es una biblioteca de Java Script, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Al igual de otras bibliotecas ofrece una serie de funcionalidades basada en java script 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.

#### HTML5 (HyperText Markup Language)

Es la quinta revisión del lenguaje de marcado estándar que se emplea para la web. Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es bastante obvia: gracias a HTML5 podemos crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5.

#### CSS (Cascading Style Sheets)

Las hojas de estilo en cascada, CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en html o xml (y por extensión en XHTML), el W3 (Word wide web Consortium) es el encargado de formular las especificaciones de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

#### PRUEBA

En el flujo de trabajo de prueba se verifican los resultados de la implementación (ejecutable), probando cada construcción, poniendo a prueba cada una de las construcciones internas, intermedias y versiones finales, que serán entregas a los usuarios.

#### Plan de pruebas aplicadas al sistema

La estrategia de pruebas que se planificó en el presente sistema de información, para garantizar la calidad del software con la aplicación del proceso unificado son básicamente los casos de uso como casos de prueba, partiendo por la prueba de unidad.

Las pruebas de integración serán realizadas dentro de cada caso de uso, a la cual se irán integrando las clases independientes con las clases independientes con las clases que dependen de ellas, hasta obtener el sistema completo.

Las pruebas de validación serán dirigidas por los casos de uso, validando las entradas y analizando las salidas o resultados que se obtengan del sistema.

#### Modelo de Prueba

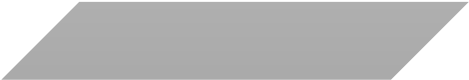
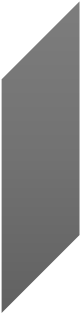
En el modelo de pruebas se detallan algunos casos de prueba del sistema de información, así como los procedimientos de prueba que especifican como realizar los mismos.

Los casos de prueba especifican la forma de probar el sistema, incluyendo entradas y el resultado a probarse. En los modelos de prueba de caja blanca, caja negra y también las pruebas basadas en fallos y las pruebas de escenario para verifica<r que el sistema se adapte a los requisitos del cliente.

#### Pruebas de Unidad

**Caja Negra:**

Es el elemento que es estudiado desde el punto de vista de las entradas que recibe las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesara su forma de interactuar con el medio que le rodea (en ocasiones, otros elementos que también podrían ser cajas negras) entendiendo que es lo que hace, pero sin dar importancia a como lo hace.



**FUNCIONES**

**ENTRADA**



**SALIDA**

**Caja Blanca:**

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles, procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El testeador escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados.

#### 5.1. CONCLUSIONES

El sistema web implementado es una innovadora y funcional forma de controlar las ventas y pedidos de la farmacia San Pedro, puesto que solucionara muchos de los problemas en la gestión de las ventas y pedidos realizados por numerosos clientes que posee, aprovechando la tecnología que permite la gestión, administración y comunicación de los datos de forma más provechosa, eficiente y ágil a como se venía realizando hasta ahora.

Además, el sistema ha logrado capturar una mayor cantidad de información, en comparación con el sistema actual, que es un manejo manual de la información, el nuevo sistema permite un control más real y objetivo en cuanto al manejo más eficiente de la información de la empresa, junto con ayudar en la toma de decisiones, ya que esta se maneja en el sistema en forma actualizada.

La solución propuesta, por tratar se una aplicación desarrollada con tecnología PHP, y con herramientas de carácter libre como lo son MySQL, proporciona un ahorro considerable en el costo de licencias y hosting.

El diseño de la aplicación permite una fácil utilización, ya sea por parte de los clientes al momento de realizar un pedido a través del sitio Web, como también, por parte de los vendedores y administradores que utilizaran el sistema.

La metodología de desarrollo facilito la incorporación de nuevas funcionalidades a la aplicación por su carácter iterativa-incremental y permitió obtener un producto funcional al término de cada incremento y retroalimentación por parte del cliente. Para la farmacia, esta aplicación le permitirá ampliar el mercado y obtener ventajas sobre sus competidores, utilizando el uso de las nuevas tecnologías que en la actualidad masifican el acceso practico y eficiente de los clientes con la empresa en este caso Farmacia Belén.

#### 5.2. RECOMENDACIONES

* A partir de este trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.
* Se debe realizar copias de seguridad del Gestor de base de datos
* Realizar continuo mantenimiento del sistema para un mejor desempeño.
* Con respecto al análisis y diseño del sistema, cuando se requiera la ampliación y creación de nuevos módulos, ser recomienda primero revisar la documentación para poder tomar una buena decisión, ya que el sistema presenta elementos reutilizables que podrían ser utilizados en los módulos nuevos.
* Se recomienda para trabajos fututos con características similares, utilizar algún framework, que facilite el desarrollo del producto.
* Se recomienda implementar, utilizar y administrar el sistema de acuerdo las instrucciones brindadas.
* La revisión periódica por cierto periodo de tiempo es recomendables para eficiencia y un funcionamiento adecuado del sistema.