

**CAPÍTULO I**  
**EL PROYECTO**

Mejora del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC.

## **CAPÍTULO I**

### **EL PROYECTO**

#### **1 PRESENTACIÓN DEL PROYECTO**

##### **1.1 Título**

Mejora del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC.

##### **1.2 Área del proyecto**

Programa Canasta Alimentaria

##### **1.3 Responsables del proyecto**

Carrera de Ingeniería Informática – Taller III – Grupo I

##### **1.4 Entidades asociadas**

Subgobernación de Cercado – Tarija en el programa Canasta Alimentaria

##### **1.5 Compromiso del director del proyecto**

###### ***Tabla 1***

###### ***Compromiso del director del proyecto***

Yo, Daniel Fabian Ortega Galean, director del proyecto, acepto las bases y condiciones del proyecto, así mismo asumo la responsabilidad de cumplir los compromisos de ejecución del proyecto titulado: Desarrollo de un sistema Informático Web basado en Inteligencia Artificial para disminuir la insatisfacción del adulto mayor en la canasta alimentaria entregada por la Subgobernación.	
Director	Firma

##### **1.6 Grupo responsable del proyecto**

Universitario Daniel Fabian Ortega Galean

### 1.7 Duración

La ejecución del proyecto será de ocho meses de acuerdo al cronograma establecido

### 1.8 Área/línea de investigación priorizado

“GES” Sistema de Gestión.

Reconocimiento facial - Inteligencia artificial.

### 1.9 Director responsable del proyecto

*Tabla 2*

*Datos del encargado del proyecto*

Apellido Paterno	Apellido Materno	Nombre	Cedula de Identidad
Ortega	Galean	Daniel Fabian	7217419
Profesión	Carrera	Facultad	
Estudiante	Ingeniería Informática	Ciencias y Tecnología	
Celular	Correo	Firma	
72994143	<a href="mailto:danielortega782@gmail.com">danielortega782@gmail.com</a>		

### 1.10 Actividades previstas para los integrantes del equipo de investigación

*Tabla 3*

*Integrantes del equipo de investigación*

Responsable	Actividades
<b>Jefe de proyecto</b> Daniel Fabian Ortega Galean	El jefe de proyecto es el encargado de supervisar el proyecto en todas las áreas del mismo, desde la programación hasta el cumplimiento en las fases de la metodología RUP en la parte analítica del sistema. <ul style="list-style-type: none"><li>✓ Control y planificación del cronograma del proyecto.</li><li>✓ Organización de un equipo de proyecto adecuado focalizando en los objetivos del presente.</li><li>✓ Seguimiento a cada etapa del proyecto.</li><li>✓ Control y supervisión del proyecto.</li></ul>

	<ul style="list-style-type: none"> <li>✓ Presentación final del sistema.</li> </ul>
<b>Analista de Sistemas</b> Daniel Fabian Ortega Galean	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. <ul style="list-style-type: none"> <li>✓ Realización y especificación de requerimientos.</li> <li>✓ Elaboración de análisis.</li> <li>✓ Elaboración de diseño.</li> <li>✓ Diseño de diagramas UML.</li> <li>✓ Construcción de la base de datos.</li> </ul>
<b>Programador</b> Daniel Fabian Ortega Galean	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario. <ul style="list-style-type: none"> <li>✓ La programación del código debe ir de acuerdo a las especificaciones que se maneja el analista del sistema.</li> </ul>
<b>Ingeniero de software</b> Daniel Fabian Ortega Galean	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.

## **1.11 Perfil del Proyecto**

### **1.11.1. Introducción.**

El trabajo de investigación presentado aborda la situación actual por la que atraviesan las personas adultas mayores de la Provincia Cercado para poder retirar la canasta alimentaria, desde la perspectiva de las dificultades que ocurren tanto al momento del registro para acceder a este beneficio por primera vez como a todos los problemas como ser largas filas, desorganización por falta de una correcta planificación y oportuna asignación de las canastas existente en stock; que se presentan las veces que les toca recoger la canasta asignada.

El interés de este trabajo viene dado por el incremento de personas mayores que año tras año por cumplir con los requisitos son nuevos beneficiarios de la canasta otorgada por la Subgobernación de la Provincia Cercado del Departamento de Tarija, los procesos manuales y lentos con los que actualmente se realizan el control de asignación del mencionado beneficio, se traducen en molestias para las personas mayores por el tiempo que demoran en hacer largas filas para poder ser registrados o filas para poder valorar si les toca retirar la canasta o no en determinada fecha. El tiempo y la desorganización influyen negativamente en la imagen de esta repartición de la Subgobernación.

Por tanto, se debe buscar la manera de agilizar los procesos que garanticen un adecuado control tanto del registro de los nuevos beneficiarios como de una adecuada planificación de asignación de la canasta a cada usuario en las fechas correctas, siempre teniendo información oportuna del stock con el que se cuenta.

Este trabajo es uno de los primeros en el medio local que pretende hacer uso de la Inteligencia Artificial para realizar reconocimiento facial de rasgos físicos de las personas adultas mayores para poder garantizar a través del uso de esta tecnología que no exista suplantación de identidad y que el tiempo tanto de registro como la valoración se reduzcan al mínimo.

## **1.12. Descripción del Proyecto**

### **1.12.1. Antecedentes.**

La Subgobernación de Cercado del Gobierno Autónomo Departamental de Tarija en cumplimiento a la Ley Departamental N° 72 “LEY DEPARTAMENTAL DE LA CANASTA ALIMENTARIA PARA LAS PERSONAS DE LA TERCERA EDAD POR UNA CALIDAD DE VIDA DIGNA”, del 15 de febrero del 2013, Decreto Departamental N° 13 del 25 de julio 2014, anexo al Decreto Departamental N° 13 “Reglamento de Canasta Alimentaria para las personas de la Tercera Edad por Una Calidad de Vida Digna” y el Decreto Departamental N° 07 del 08 de abril de 2015, lleva adelante El Programa Recurrente “Canasta Alimentaria para las Personas de la Tercera”, que beneficia directamente a toda la población del sector adulto mayor de nuestra Provincia Cercado, mediante la dotación de alimentos no perecederos, la misma que cuenta con los alimentos nutricionales necesarios que requiere toda persona logrando de esta manera la disminución de desnutrición que afectan a este sector vulnerable de la tercera edad.

Según Morcillo (2020), el reconocimiento facial ha existido durante décadas, pero ha evolucionado rápidamente con la incorporación de técnicas de Deep Learning y su uso se ha vuelto más notable y accesible en los últimos años, ya que ahora se utiliza con soluciones innovadoras, como aplicaciones de fotos personales, autenticación en dispositivos móviles, etc.

De igual manera Becerra y Calles (2019), enfatizan que en la actualidad la tecnología se ha convertido en una herramienta para el desarrollo de una sociedad, por tal motivo las diferentes instituciones tienen la necesidad de implementar Sistemas que permitan automatizar los procesos que se ejecutan manualmente. Se utilizó la Metodología de desarrollo UWE para la ejecución de la propuesta, la misma que permitió establecer los requerimientos de los usuarios y diseñar el sistema de acuerdo a sus necesidades. Para el desarrollo del sistema de reconocimiento facial se utilizó OpenCV 3.4. Finalmente se implementó el Sistema Informático de Reconocimiento Facial para el registro y control de asistencia de los socios de la Cooperativa de Taxis y Camionetas Puyo, logrando convertirse en una herramienta

para obtener datos confiables de los socios en las reuniones beneficiando y facilitando el trabajo manual al personal administrativo de la institución.

Finalmente, Yañez (2019), con el desarrollo de su investigación brindó una solución frente al problema que radicaba en el control de acceso de los estudiantes a los laboratorios de la FIIS-UNAC. El tipo de estudio usado fue aplicado, nivel experimental y de diseño preexperimental. La metodología empleada en el desarrollo del Sistema Informático fue RUP (Rational Unified Process) mediante la herramienta Rational Rose, el sistema fue desarrollado con el lenguaje de programación PHP y el gestor de base de datos MYSQL. La población fue de 93, tomando como muestra 75 alumnos de la FIIS (Facultad de Ingeniería Industrial y Sistemas) de la UNAC (Universidad Nacional del Callao), la implementación de un Sistema de reconocimiento facial para el control de acceso de estudiantes, optimiza dicho proceso, que a su vez permite el cálculo del tiempo de retraso, porcentaje de asistencias y nivel de accesos. Se concluyó que el uso de un Sistema de Reconocimiento Facial mejoró el control de acceso de los estudiantes a los laboratorios de la FIIS - UNAC.

No se puede afirmar la existencia a nivel local y nacional de una investigación reciente relacionado con el tema estudiado en el presente proyecto de grado.

### **1.12.2. Justificación del Proyecto.**

#### **1.12.2.1. Tecnológico.**

Al implementar un Sistema aplicando Inteligencia Artificial con la librería OpenCv para automatizar procesos del control de registro y entrega de la canasta alimentaria, estamos aprovechando los avances tecnológicos que automaticen procesos críticos, demostrando que muchas cosas son posibles con las herramientas adecuadas.

Con la investigación y conclusión del trabajo se dotará una herramienta informática a la Subgobernación de Cercado que permita un control efectivo de la asignación de canastas alimentarias.

#### **1.12.2.2. Económico.**

Al desarrollar un Sistema basado en Software Libre, evita la erogación de recursos económicos a la Subgobernación en pagos por licencias, de la misma forma automatiza procesos para así evitar un gasto innecesario de personal que ocupe esa función.

#### **1.12.2.3. Social.**

El presente trabajo de investigación beneficia de manera directa al adulto mayor de la Provincia Cercado quien al ser un pilar fundamental de nuestra sociedad merece un trato ágil, rápido, justo, bueno al momento de la entrega de su canasta alimentaria.

#### **1.12.2.4. Personal.**

El presente proyecto representa un reto personal para aplicar todos los conocimientos adquiridos durante los años de mi formación académica con los cuales solucionaremos un problema puntual de la vida real.

### **1.12.3. Planteamiento del Problema.**

¿De qué manera la Subgobernación puede disminuir la insatisfacción del adulto mayor en la entrega de la canasta alimentaria?

### **1.12.4. Objetivos.**

#### **1.12.4.1. Objetivo General.**

Desarrollar un Sistema Web aplicando las TIC, para el control de la entrega de la canasta alimentaria de la Subgobernación de Cercado.

#### **1.12.4.2. Objetivos Específicos.**

- Desarrollar un sistema para la Subgobernación.
- Realizar un manual de instalación y uso del sistema.
- Aplicar métodos para la realización de talleres de capacitación.
- Automatizar el proceso de registro de datos de beneficiarios.
- Desarrollar un módulo de Inventario para gestión del stock de la canasta alimentaria.



- Diseñar un módulo de Planificación para la entrega de la canasta que garantice una correcta calendarización.
- Implementar un módulo Detector de rasgos faciales basado en inteligencia artificial para el reconocimiento facial que permita confirmar la identidad del adulto mayor.
- Desarrollar un módulo de Dashboard para mostrar estadísticas informativas de la entrega de la canasta alimentaria.
- Implementar un módulo de Reportes con la librería llamado Weasyprint.

### **1.12.5. Metodología de Desarrollo de Proyecto.**

#### **1.12.5.1. Tipo de Metodología.**

Se utilizará la Metodología Descriptiva simple porque pretende entender el proceso de registro y entrega de la canasta alimentaria.

La Investigación Descriptiva se refiere al diseño de la investigación, creación de preguntas y análisis de datos que se llevarán a cabo sobre el tema. Se conoce como método de investigación observacional porque ninguna de las variables que forman parte del estudio está influenciada.

#### ***Metodología RUP.***

La metodología de desarrollo RUP por sus siglas en inglés o Proceso de Desarrollo Unificado es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

#### ***Características esenciales de RUP.***

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

Proceso dirigido por Casos de Uso. - Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario

y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema. Los Casos de Uso para el presente Proyecto ayudarán a verificar el comportamiento de las funcionalidades del Sistema, así también para guiar el diseño, la implementación y prueba.

Proceso centrado en la arquitectura. - La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios). La arquitectura está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además, la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo.

Para el proyecto la arquitectura determinará la forma que tendrá el sistema, cumpliendo el desarrollo de cada caso de uso durante las fases de RUP.

Proceso iterativo e incremental. - La estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, y se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión final del producto. Al concluir el proyecto, se entregará una iteración final el cual contendrá el desarrollo completado de los casos de uso y la arquitectura definida.

### ***Fases de RUP.***

**Inicio.** - El objetivo de esta fase es establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Es significativamente importante para el desarrollo de nuevo software, ya que ayudará a mejorar el funcionamiento de la organización.

**Elaboración.** - En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

**Construcción.** - El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizadas por los usuarios y se realizan las mejoras para el proyecto.

**Transición.** - El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

***Lenguaje Unificado de Modelado (UML).*** UML es un Lenguaje de Modelado Unificado basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un Sistema programado. El UML modela el sistema mediante el uso de objetos que forman parte de él, así como, las relaciones estáticas o dinámicas que existen entre ellos. UML puede ser utilizado por cualquier metodología de análisis y diseño orientada por objetos para expresar los diseños.

**Tipos de Diagramas UML.** - UML utiliza los diagramas gráficos para obtener estos distintos puntos de vista de un sistema como Diagramas de Despliegue, Diagramas de Componentes, Diagramas de Casos de uso, Diagramas de Clases, Diagramas de Actividad, Diagramas de Estado, Diagramas de Secuencia, Diagramas de Objetos, Diagramas de Paquetes.

### ***Tipos de Diagramas Utilizados.***

Diagrama de Casos de Uso. - Es una herramienta esencial para la captura de requerimientos y para la planificación y control de un proyecto interactivo. Un diagrama de Casos de Uso consta de los siguientes elementos: Actor, Caso de Uso, Relaciones de Uso (herencia y comunicación). Este diagrama permitirá mostrar las funcionalidades de la organización y los módulos con sus opciones para la Gestión del Sistema.

Diagrama de Secuencia. - Un diagrama de secuencias muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo. Este diagrama mostrará que objetos participantes durante la ejecución de un evento en el Sistema, y los mensajes que se intercambian en una secuencia en el tiempo.

Diagrama de Clases. - Son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. El diagrama de clases para el proyecto describirá la estructura del Sistema, mostrando sus clases, atributos y las relaciones entre ellos.

### **1.12.6. Resultados Esperados.**

**Sistema Informático Web para la Subgobernación de Cercado.** - El sistema terminado para la ejecución en la subgobernación de Cercado.

**Estrategias de Socialización de Dicho Sistema.** - La capacitación de dicho sistema para su uso en el programa de canasta alimentaria de la Subgobernación.

### 1.12.7. Beneficiarios.

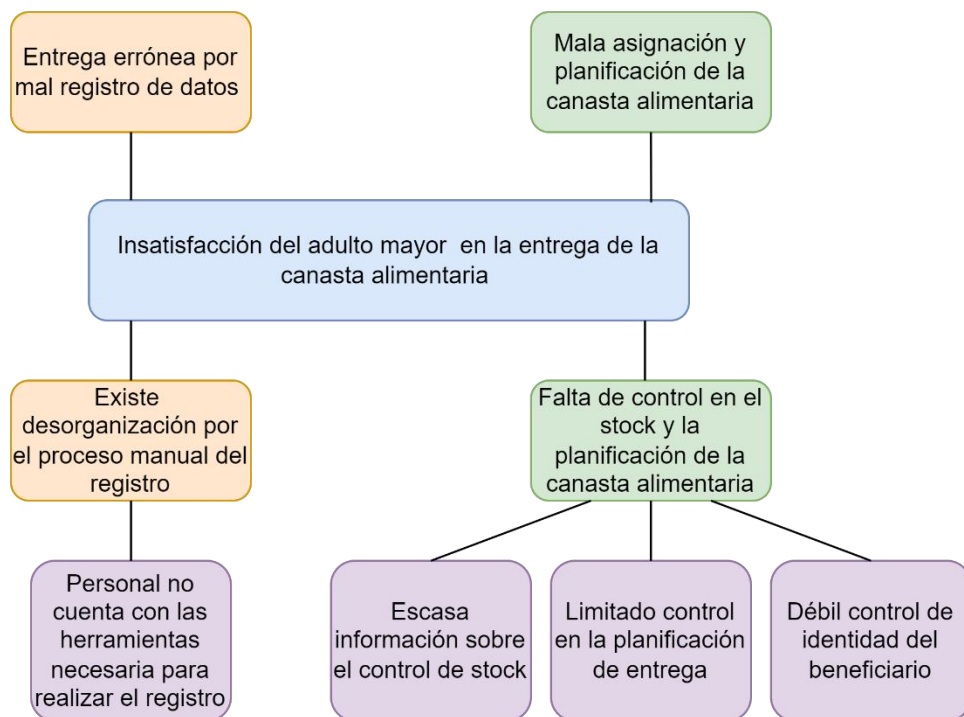
**Beneficiarios Directos.** - La población adulta mayor de la provincia Cercado Tarija se beneficia de manera directa con el presente proyecto.

**Beneficiarios Indirectos.** - La subgobernación de Cercado, personal administrativo y familiares del adulto mayor.

## 1.13 Sistematización del Problema y Abordaje de la Solución

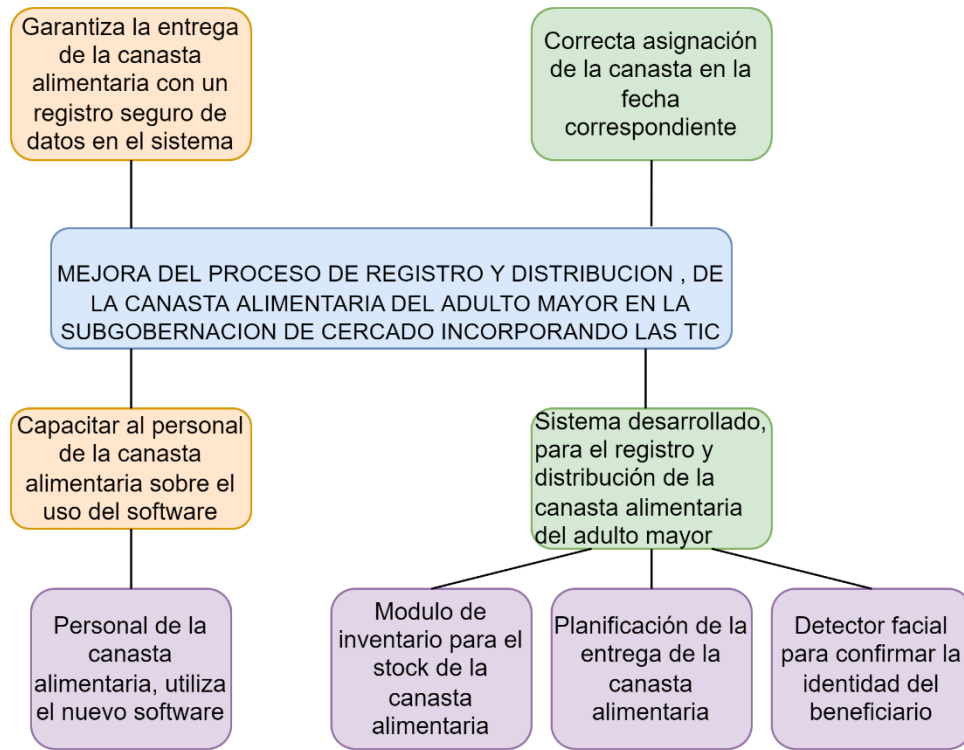
### Gráfica 1

#### 1.13.1 Árbol de Problemas



**Gráfica 2**

**1.13.2. Árbol de Objetivos**



**1.14 Situación Planteada con y sin Proyecto**

**Tabla 4**

**1.14.1. Situación planteada con y sin proyecto**

<b>Situación sin proyecto</b>	<b>Situación con proyecto</b>
<p>La situación del programa canasta alimentaria de la Subgobernación de Cercado actualmente es baja en el uso de las TIC, por lo que hay largas filas de los beneficiarios al momento de la entrega, tardanza en generar reportes para el programa.</p>	<p>La implementación de las TIC en el programa canasta alimentaria es un aporte de gran importancia ya que con este sistema podrán hacer el registro de la entrega más rápidamente como así también en la generación de reportes. Registro de beneficiarios con reconocimiento facial en el sistema Optimización de información en reportes para el programa canasta alimentaria</p>

### 1.15 Matriz de Marco Lógico

<b>Enunciado del Objetivo</b>	<b>Indicadores</b>	<b>Medios de Verificación</b>	<b>Supuestos</b>
<p><b>Fin (Objetivo de Desarrollo)</b></p> <p>Contribuir a mejorar la información en la canasta alimentaria.</p> <p>Disminuir el riesgo de pérdida de documentación.</p>	<p>Al primer año de ejecución del proyecto, en la canasta del adulto mayor entregada por la Subgobernación de Cercado Tarija, el 90% de los datos de la canasta son más confiables y exactos.</p>	<p>Estado de documentación del primer año de manera continua y registro de los nuevos beneficiarios de la primera gestión después de culminado el proyecto.</p>	<p>Se seguirá manteniendo la estructura de la empresa para el correcto funcionamiento del Sistema.</p>
<p><b>Propósito (Objetivo General)</b></p> <p>Mejora del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC.</p>	<p>Al concluir el proyecto y la ejecución del mismo, el tiempo de entrega y registro de la canasta alimentaria en la provincia cercado, se ha reducido en un 80%, cumpliendo satisfactoriamente el sistema con los</p>	<p>Reducción de tiempo y costos en la ejecución de los procesos del registro y entrega de la canasta alimentaria.</p> <p>Listado de entrega por día, en el año base y luego de dos años de concluido el proyecto.</p>	<p>La subgobernación de cercado debe apoyar con el Desarrollo del proyecto, brindando la información necesaria.</p>

Enunciado del Objetivo	Indicadores	Medios de Verificación	Supuestos
	requerimientos.		
<p>Componentes (Objetivos Específicos)</p> <ol style="list-style-type: none"> <li>1. Sistema desarrollado, para el registro y distribución de la canasta alimentaria del adulto mayor.</li> <li>2. Se realizará una capacitación de dicho sistema para el correcto uso del mismo</li> </ol>	<p>Al finalizar el proyecto se a desarrollado el sistema especificado de acuerdo a los requerimientos especificados a un 90%, bajo la norma IEEE830</p> <p>Al finalizar el proyecto, se han capacitado a 90% de los usuarios del sistema.</p>	<p>Informe del responsable del programa canasta alimentaria del adulto mayor, indicando que el sistema cumple con las expectativas definidas.</p> <p>Prueba escrita de conocimientos adquiridos del manejo del sistema.</p>	<p>El encargado del sistema ha dispuesto la ejecución de dicho sistema ya que cumple con los requerimientos.</p> <p>El personal encargado de la canasta alimentaria.</p>



<p><b>Actividades</b></p> <p><b>C1. Sistema Informático</b></p> <ol style="list-style-type: none"> <li>1. Especificación de requerimientos</li> <li>2. Elaboración de los diagramas UML que definen la arquitectura del sistema</li> <li>3. Diseño de la base de datos del sistema.</li> <li>4. Programación del sistema</li> </ol> <p><b>C2. Capacitación al personal</b></p> <ol style="list-style-type: none"> <li>1. Realizar la formación al personal en el uso de las utilidades del sistema</li> <li>2. Organización por niveles de capacitación.</li> </ol>	<p><b>Resumen presupuesto:</b></p> <p>Componente 1: Bs. 27.000</p> <p>Componente 2: Bs. 1.290</p> <p><b>Total, Proyecto: Bs. 28.290</b></p>	<p>Informe presupuestario del proyecto</p>	<p>La Subgobernación de Cercado Tarija, cuenta con el presupuesto para financiar el proyecto en general</p>
---	---	--	---



## **1.17 Marco Referencial**

### **1.17.1. Subgobernación de Cercado Tarija.**

La Subgobernación de la Provincia Cercado del Departamento de Tarija, es una entidad desconcentrada, administrativa, financiera, técnica y operativa del Órgano Ejecutivo Departamental, encargada de ejecutar políticas, programas, planes y proyectos en el marco de la política departamental de desarrollo en el territorio de su municipio, es parte integrante del Órgano Ejecutivo Departamental, se encuentra dentro del Nivel Ejecutivo de la Gobernación del Departamento de Tarija con dependencia lineal de la Máxima Autoridad Ejecutiva de la Gobernación del Departamento.

### **1.17.2. Canasta alimentaria-Subgobernación de Cercado Tarija.**

La Subgobernación de Cercado del Gobierno Autónomo Departamental de Tarija en cumplimiento a la Ley Departamental N.º 72 “LEY DEPARTAMENTAL DE LA CANASTA ALIMENTARIA PARA LAS PERSONAS DE LA TERCERA EDAD POR UNA CALIDAD DE VIDA DIGNA”, del 15 de febrero del 2013, Decreto Departamental N.º 13 del 25 de julio 2014, anexo al Decreto Departamental N.º 13 “Reglamento de Canasta Alimentaria Para las personas de la Tercera Edad por una Calidad de Vida Digna” y el Decreto Departamental N.º 07 del 08 de abril de 2015, lleva adelante El Programa Recurrente “Canasta Alimentaria para las Personas de la Tercera”, que beneficia directamente a toda la población del sector adulto mayor de nuestra Provincia Cercado, mediante la dotación de alimentos no perecederos, la misma que cuenta con los alimentos nutricionales necesarios que requiere toda persona logrando de esta manera la disminución de desnutrición que afectan a este sector vulnerable de la tercera edad.

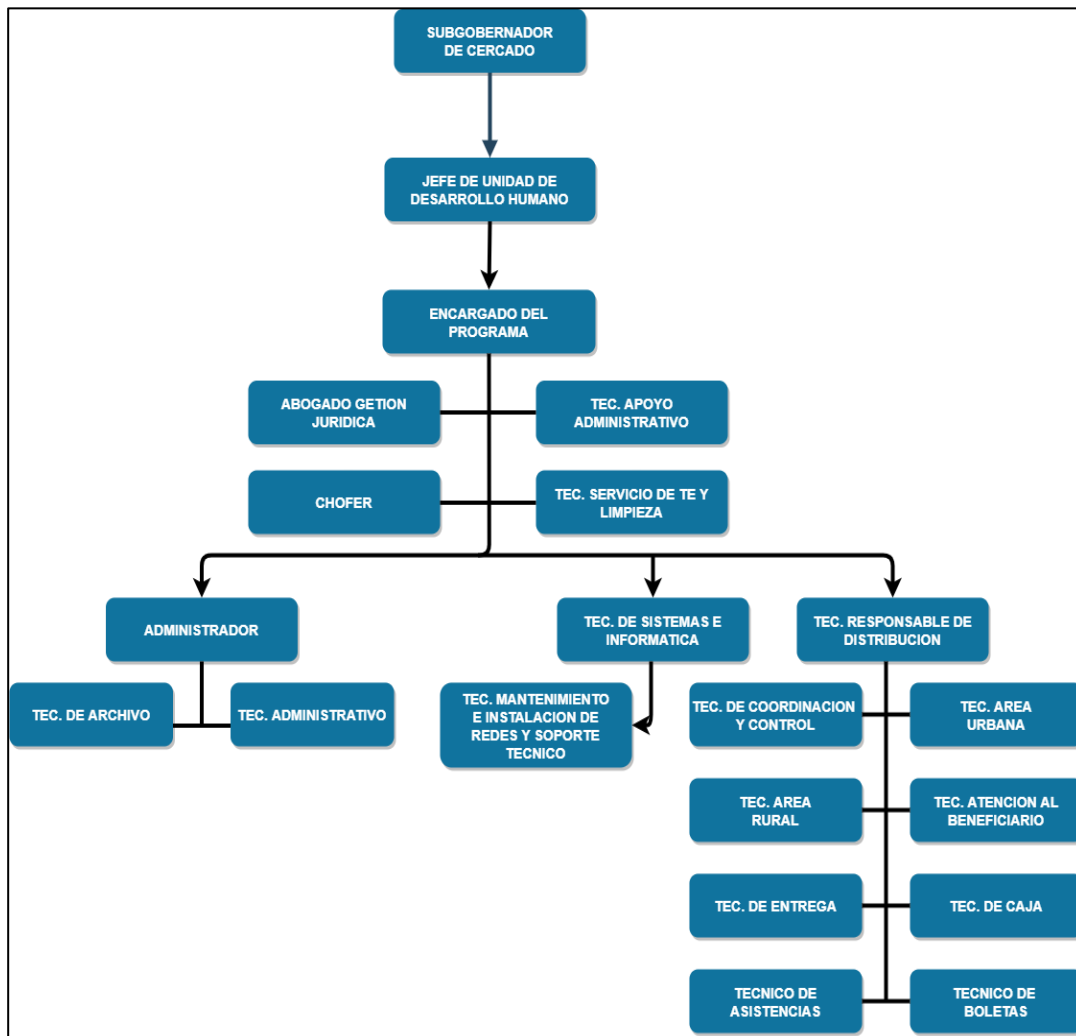
### **1.17.3. Misión.**

Mejorar la calidad de vida de la población del Adulto Mayor de la Provincia Cercado, generando una alimentación que reúna los requisitos nutricionales.

### 1.17.4. Visión.

El Programa Canasta Alimentaria de Sub - Gobernación Cercado, se constituye en un referente líder del departamento de Tarija, en cuanto a disminuir la desnutrición y la pobreza existente en la población comprendida desde los 60 años adelante. y de satisfacción de los beneficiarios a través de sus resultados obtenidos.

### 1.17.5. Organigrama.



Fuente: Subgobernación Cercado.

#### **1.17.6. Principios.**

Mejorar la calidad de vida en la población de la Provincia Cercado, impulsando su desarrollo integral de una manera eficaz y eficiente, apoyados en la sostenibilidad, transparencia y una planificación participativa, respetando las normas y reglamentos vigentes, coadyuvando así a la ejecución de los objetivos, políticas, planes y programas departamentales y nacionales a través de proyectos de inversión.

#### **1.17.7. Atención.**

La atención podría definirse como la capacidad de generar, seleccionar, dirigir y mantener un nivel de activación adecuado para procesar la información relevante. Dicho de otra forma, la atención es un proceso que tiene lugar a nivel cognitivo y que permite orientarnos hacia aquellos estímulos que son relevantes, ignorando los que no lo son para actuar en consecuencia.

#### **1.17.8. Adulto Mayor.**

Según el artículo 2 de la ley 369 de fecha de 1ro de mayo de 2013 (ley general de las personas mayores) son consideradas personas adultas mayores todos los bolivianos mayores de 60 años en todo el territorio nacional. Ley Departamental No 72 “LEY DEPARTAMENTAL DE LA CANASTA ALIMENTARIA PARA LAS PERSONAS DE LA TERCERA EDAD POR UNA CALIDAD DE VIDA DIGNA”, del 15 de febrero del 2013, Decreto Departamental N° 13 del 25 de julio 2014, anexo al Decreto Departamental No 13 “Reglamento de Canasta Alimentaria para las personas de la Tercera Edad Por Una Calidad de Vida Digna” y el Decreto Departamental No 07 del 08 de abril de 2015.

#### **1.17.9. Canasta Alimentaria.**

La “Canasta Alimentaria para la tercera edad por una calidad de vida digna”, tiene como finalidad garantizar al sector adulto mayor una alimentación que reúna los requisitos mínimos nutricionales, consecuentemente una calidad de vida digna, de urgente necesidad y con carácter de prioridad departamental.

## 1.18 Marco Tecnológico

### 1.18.1. Python.

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como la cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero; la sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C; la cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros; y, además, Python es gratuito, incluso para propósitos empresariales.

El creador del lenguaje es un europeo llamado Guido Van Rossum (2003), hace ya más de una década que diseñó Python ayudado y motivado por su experiencia en la creación de otro lenguaje llamado ABC. El objetivo de Guido era cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que sirviese para tratar diversas tareas dentro de la programación que habitualmente se hacía en Unix usando C (Álvarez, 2003, p.18).

Características de Python. - Un lenguaje sencillo, legible y elegante que atiende a un conjunto de reglas que hacen muy corta su curva de aprendizaje. Si ya tienes unas nociones de programación o vienes de programar en otros lenguajes como Java no te será difícil comenzar a leer y entender el código desarrollado en Python.

El siguiente paso es comenzar a programar, verás que con muy pocas líneas de código es posible programar algoritmos complejos. Esto hace de Python un lenguaje práctico que permite ahorrar mucho tiempo.

Futuro de Python: Uso en Data Science, Big Data e Inteligencia Artificial (IA): A lo largo de los últimos 30 años Python ha disfrutado de una evolución constante además del respaldo de la comunidad que le han hecho particularmente relevante en el desarrollo de aplicaciones en entorno servidor. Esto, junto con su sencillez, ha producido que pudiera situarse en la escena del Big Data y en especial del desarrollo de algoritmos de aprendizaje automatizado.

La generalización del Big Data en los últimos años, seguida de la explosión de la Inteligencia Artificial, Machine Learning, Deep Learning y la consolidación de la ciencia de datos como una nueva área de trabajo con especialistas propios, ha revolucionado el panorama. Al estar Python muy presente en el ámbito educativo siendo usada como lenguaje de referencia en escuelas y universidades su presencia en el campo de la investigación es más que justificado. Lo que ha producido que muchas de las herramientas que han surgido en este sector han sido desarrolladas en este lenguaje y explotadas por los ingenieros de datos y los científicos de datos. Algunos ejemplos son PySpark (Big Data) o Pandas, NumPy, Matplotlib o Jupyter (Data Science) (Robledano, 2019).

#### **1.18.1.1 Pycharm.**

Pycharm es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional, que incluye más características como el soporte a desarrollo web con varios precios. Las características que posee pycharm son las siguientes:

Asistencia y análisis de codificación, con finalización de código, sintaxis y resaltado de errores, integración delinter y arreglos rápidos; navegación de proyectos y códigos: vistas de proyectos especializados, vistas de estructura de archivos y saltos rápidos entre archivos, clases, métodos y

usos; PythonRefactoring que incluye renombrar, extraer método, introducir variable, introducir constante y otros; el soporte para frameworks web: Django, web2py y Flask; depurador integrado de Python; pruebas unitarias integradas, con cobertura line-by-line (línea por línea); desarrollo de Python de Google App Engine, finalmente integración de control de versiones: interfaz de usuario unificada para Mercurial, Git, Subversion, Perforce y CVS con listas de cambios y combinación (Gutiérrez Jáuregui, 2010, p.19).

### **1.18.2. PostgreSQL.**

PostgreSQL es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, siendo mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente. Esta comunidad es denominada PDGD (PostgreSQL Global Development Group, por sus siglas en inglés). Es reconocido actualmente como uno de los sistemas gestores de bases de datos relacionales más potentes del mercado. Presenta fácil accesibilidad, es multiplataforma y está disponible para su utilización en casi todos los sistemas operativos utilizados en la actualidad sin disminuir su rendimiento.

PostgreSQL es un gestor de bases de datos relacional que presenta varias características por las que destaca, siendo uno de los mejores y más utilizados motores de BD en la actualidad. A continuación, brindamos una lista de sus características y explicamos brevemente algunas de las más destacadas:

Presenta un sistema de alta concurrencia: Presenta un sistema denominado MVCC, el cual permite que mientras un proceso escribe una tabla, otros puedan acceder a la misma tabla sin necesidad de verse bloqueados, y cada usuario obtiene una visión consistente. Sistema "Hot Standby": Este proceso permite a los usuarios poder conectarse con el servidor y ejecutar



búsquedas en la bd mientras la misma está en modo de recuperación o "stand by". También se puede pasar de este modo a modo normal sin detener el flujo de búsquedas o consultas de los usuarios, manteniendo las conexiones abiertas. Esto es posible únicamente cuando la base de datos se encuentra en modo de solo-lectura (Arteaga Maraz, 2019, p.19).

Por otra parte, PostgreSQL es una de las opciones más interesantes en bases de datos relacionales open-source. Es gratuito y libre, además de que hoy nos ofrece una gran cantidad de opciones avanzadas. De hecho, es considerado el motor de base de datos más avanzado en la actualidad (Antón Dorantes, 2015, p.19)

PostgreSQL aporta mucha flexibilidad a nuestros proyectos. Por ejemplo, nos permite definir funciones personalizadas por medio de varios lenguajes. Algunos son PL/pgSQL, PL/Tcl, PL/Perl, PL/Python, PL/PHP, PL/Ruby y PL/Java.

#### **1.18.2.1. DBeaver.**

DBeaver es un gestor universal de BBDD multiplataforma, que ofrece soporte a las bases de datos más conocidas del mercado (MySQL, Oracle, DB2, SQL Server, PostgreSQL, etc.), así como algunas NoSQL (MongoDB, Cassandra). Como características más destacadas ofrece:

- Navegar por la estructura de BBDD
- Editar tablas, columnas, keys, etc.
- Permite ejecutar scripts SQL.
- Modificar columnas resultantes de una consulta.
- Resaltado de SQL (específico para cada tipo de motor)
- Autocompletado en el editor.
- Soporte para ver y editar archivos BLOB.
- Exportar tablas o resultados de queries.
- Generación de entidad-relación.

#### · Comparación de esquemas.

Dispone de dos versiones Community y Enterprise, las dos son gratuitas (licencia GPL 2), la única diferencia es que la Enterprise Edition incluye el soporte a base de datos NoSQL.

DBeaver tiene una interfaz de usuario bien diseñada, la plataforma basada en un marco de código abierto y permite escribir múltiples extensiones, así como también es compatible con cualquier base de datos. También incluye soporte para clientes nativos de MySQL y Oracle, administración de controladores, editor de SQL y formateo. DBeaver es una aplicación multiplataforma pues cuenta con soporte para las plataformas de MacOS, Windows y Linux (Pacheco Mengual, 2015, p.20)

### **1.18.3. Inteligencia Artificial**

La Inteligencia artificial es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes. En otras palabras, la IA es el concepto según el cual “las máquinas piensan como seres humanos. Normalmente, un sistema de IA es capaz de analizar datos en grandes cantidades (big data), identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión. Para nosotros, lo importante es que la IA permite que nuestras experiencias cotidianas sean más inteligentes. ¿Cómo? Al integrar análisis predictivos (hablaremos sobre esto más adelante) y otras técnicas de IA en aplicaciones que utilizamos diariamente.

Siri funciona como un asistente personal, ya que utiliza procesamiento de lenguaje natural. Facebook y Google Fotos sugieren el etiquetado y agrupamiento de fotos con base en el reconocimiento de imagen. Amazon ofrece recomendaciones de productos basadas en modelos de canasta de compra. Waze brinda información optimizada de tráfico y navegación en tiempo real.

Tipos de inteligencia artificial. - Los expertos en Ciencias de la computación Stuart Russell y Peter Norvig diferencian varios tipos de inteligencia artificial:

Sistemas que piensan como humanos: automatizan actividades como la toma de decisiones, la resolución de problemas y el aprendizaje. Un ejemplo son las redes neuronales artificiales. Sistemas que actúan como humanos: se trata de computadoras que realizan tareas de forma similar a como lo hacen las personas. Es el caso de los robots. Sistemas que piensan racionalmente: intentan emular el pensamiento lógico racional de los humanos, es decir, se investiga cómo lograr que las máquinas puedan percibir, razonar y actuar en consecuencia. Los sistemas expertos se engloban en este grupo. Sistemas que actúan racionalmente: idealmente, son aquellos que tratan de imitar de manera racional el comportamiento humano, como los agentes inteligentes (Martí, 2013, p.20).

#### **1.18.3.1. Reconocimiento Facial.**

Sistema de reconocimiento facial. Permite identificar a una persona analizando las características biométricas de su rostro. La extracción de la información relativa a estas características está actualmente ligada a sofisticados procesos matemáticos y algoritmos de coincidencia, lo que ha propulsado el avance a la tecnología de reconocimiento facial en diversos mercados. El objetivo de un sistema de reconocimiento facial es: dada una imagen de una cara "desconocida", o imagen de test, encontrar una imagen de la misma cara en un conjunto de imágenes "conocidas", o imágenes de entrenamiento. La gran dificultad añadida es la de conseguir que este proceso se pueda realizar en tiempo real. El sistema identificará las caras presentes en imágenes o videos automáticamente.

Funcionamiento del sistema. - El procedimiento de reconocimiento se realiza en las siguientes fases:

Una fase de detección: en la que se recoge la imagen del rostro del usuario a identificar a través del dispositivo elegido, ya sea una cámara fotográfica o una cámara de vídeo.

Preprocesado de la imagen: en esta fase se realizan tareas esenciales para la extracción de la información biométrica, tales como la alineación de la cara respecto a ciertas propiedades geométricas y para hacerla también independiente de la iluminación de la imagen capturada o de la gama de colores obtenida.

Fase de extracción de las características faciales: en la que se obtiene la información biométrica de los rasgos faciales, almacenándose esta en un patrón biométrico facial.

Fase de comparación: En esta fase se coteja la información biométrica obtenida con aquellas almacenadas en la base de datos, a través de una comparación 1:N. Los resultados obtenidos indican el porcentaje de similitud del usuario a identificar, con aquellos almacenados en la base de datos.

Fase de toma de decisiones: en la que, utilizando la matriz de similitudes, se identifica al individuo como aquel que mayor porcentaje de similitud ha obtenido, siempre que se encuentre por encima de un umbral determinado.

Diferencia entre detección de rostro y reconocimiento facial: El reconocimiento facial y la detección de rostros son dos conceptos completamente distintos en su proceder, aunque pareciera que no fuera así. Detección de rostros: es un procedimiento analítico simple, que consiste en la captura de una imagen del rostro de cualquier persona, que aparezca en el ángulo de visión de la cámara asignada para ello. El algoritmo informático interactúa con la cámara mediante la funcionalidad “fase detección”, que debe estar presente en la cámara IP. Solo las cámaras IP pueden sostener esta funcionalidad, descartando de inmediato el uso de cámaras análogas para ejecutar esta función. Para capturar la imagen, el sistema detector de rostros identifica a las personas en el área de cobertura y procede automáticamente a extraer una foto de la cara, que es almacenada en el dispositivo de grabación, sea disco duro o memoria microSD. Reconocimiento facial: es diametralmente distinto y más complejo que la acción de detectar un rostro. Un algoritmo informático es el encargado de hacer el proceso de identificación facial de forma automática y en

tiempo real. Es un proceso más específico que la detección de rostros, requiere de mejor y mayor analítica de video para poder lograr saber si el rostro descubierto, está registrado en la base de datos.

Existen varias formas de algoritmos para la utilización de inteligencia artificial como, por ejemplo:

Holísticos. - Reconocen según toda la imagen facial. Son métodos basados en correlación. El esquema de clasificación más simple, donde se utilizan modelos de comparación para el reconocimiento, es el template matching.

Locales y geométricos. - Se comparan diferentes características geométricas de las caras. Existen dos divisiones, la basada en los vectores característicos extraídos del perfil, y la basada en los extraídos a partir de una vista frontal. Se utilizaba mucho anteriormente, pero sus resultados no son óptimos.

**Pingo:** Esta técnica utiliza sensores en 3D para captar información sobre la forma de la cara. Esta información se utiliza posteriormente para identificar rasgos característicos del rostro como por ejemplo la barbilla, el contorno de los ojos, la nariz o los pómulos, y reteniendo información espacial, a parte de la textura y la profundidad.

Una ventaja del reconocimiento facial en 3D es que no les afectan los cambios de iluminación, como pasa en el caso de otras técnicas. Además, otro punto a favor es que pueden reconocer una cara en diferentes ángulos, incluso de perfil (McCarthy, 2010, p.20)

#### **1.18.4. Dash.**

Dash es una poderosa biblioteca de código abierto que ayuda a crear paneles de controles interactivos y en vivo basados en la web usando Plotly, Flask y React. Dash permite a los equipos crear aplicaciones de ML y ciencia de datos que ponen el poder de Python, R y Julia en manos de los usuarios empresariales. Las aplicaciones de pila completa que normalmente requerirían un front-end, back-end y un equipo de DevOps ahora pueden ser construidas e implementadas en horas por científicos de datos con Dash.

Con Dash Open Source, las aplicaciones Dash se ejecutan en su computadora portátil o estación de trabajo local, pero otras personas de su organización no pueden acceder fácilmente a ellas. Dash es una biblioteca de interfaz de usuario para crear aplicaciones web analíticas. Aquellos que usan Python para análisis de datos, exploración de datos, visualización, modelado, control de instrumentos e informes encontrarán un uso inmediato para Dash (Agarwal, 2018).

Las aplicaciones Dash son servidores web que ejecutan Flask y comunican paquetes JSON a través de solicitudes HTTP. La interfaz de Dash procesa componentes usando React.js, la biblioteca de interfaz de usuario de Javascript escrita y mantenida por Facebook. El matraz es genial. Es ampliamente adoptado por la comunidad de Python y se implementa en entornos de producción en todas partes. La instancia subyacente de Flask y todas sus propiedades configurables son accesibles para los desarrolladores de aplicaciones Dash. Para los desarrolladores avanzados, las aplicaciones Dash también se pueden extender a través del amplio conjunto de complementos de Flask .

#### **1.18.5. Dashboard.**

Un dashboard es una herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los indicadores clave de desempeño (KPI), métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico.

Podemos pensar en el dashboard como una especie de "resumen" que recopila datos de diferentes fuentes en un solo sitio y los presenta de manera digerible para que lo más importante salte a la vista. Estas son algunas de las características que debe tener este centro de control.

Personalizado. - Un dashboard debe contener únicamente los KPI que sean relevantes para el departamento, campaña o proceso que nos ocupa. Para orientarlo, podemos pensar en las preguntas principales a las que queremos responder. Por ejemplo, cuáles son las principales fuentes de tráfico a nuestra web, cómo está funcionando nuestro embudo de ventas o cuáles son los cinco productos que nos generan más ingresos.

Visual. - La idea de un dashboard es que podamos obtener la información que buscamos a golpe de vista. Por ello, los datos se presentan en forma de gráficos y debemos contar con indicadores rápidos a través de claves de color, flechas hacia arriba o abajo o cifras destacadas, por ejemplo.

Práctico. - La función principal de un dashboard siempre debe ser orientar las acciones de nuestro equipo. Por tanto, debe facilitarnos la información necesaria para que podamos saber cuáles son los siguientes pasos a seguir para mejorar los resultados.

En tiempo real. - A día de hoy, las acciones de marketing digital evolucionan con gran rapidez y aprovechar el momento clave es esencial. Por eso, la información debería estar actualizada al momento en todas las fuentes y mostrarse en el dashboard en tiempo real (Ortiz, 2020).

#### **1.18.6. OpenCv.**

Es una biblioteca informática de código abierto desarrollado originalmente por la visión de Intel. Es gratuito para uso comercial y la investigación bajo una licencia BSD. La biblioteca es multiplataforma y funciona en Mac- OSX, Windows y Linux. Se centra principalmente hacia procesamiento imagen tiempo real como tal si encuentra Intel Integrated Performance Primitives sobre el sistema, utilizará estas rutinas optimizado comerciales para acelerarse. Sus funciones son las siguientes: Capturas a tiempo real, importación de archivos de video, el tratamiento básico de imágenes (brillo, contraste, umbral), detección de objetos (cara, cuerpo) y Blob detección.

Las aplicaciones en que se utilizan e utilizaron son las siguientes: OpenCV ha sido usado en el sistema de visión del vehículo no tripulado Stanley de la Universidad de Stanford, el ganador en el año 2005 del Gran desafío DARPA. OpenCV se usa en sistemas de vigilancia de vídeo. OpenCV es la clave en el programa Swistrack, una herramienta de seguimiento distribuida (Marín, 2015).

### **1.18.7. ReportLab.**

ReportLab es un *toolkit* de código abierto para crear documentos PDF desde Python. Se trata de una librería muy extensa y con muchas funcionalidades, desde pequeños textos y figuras geométricas a grandes gráficos e ilustraciones, todo ello puede ser incluido dentro de un PDF. En este artículo estaremos sopesando sus características generales y sus principales funciones para crear este tipo de documentos. ReportLab incluye una API de bajo nivel para generar documentos PDF directamente desde Python, y un lenguaje de plantillas de más alto nivel —similar a HTML y a los sistemas de plantilla que se emplean en el desarrollo web— llamado RML. Generalmente la segunda opción suele ser más conveniente para aquellos que deban hacer un uso exhaustivo de las capacidades de la librería al momento de generar documentos. Para el resto de los casos, será suficiente con la API de bajo nivel que describiremos en este artículo (Araya García, 2019).

### **1.18.8. Toolkit.**

Se refiere a una interfaz de programación de aplicaciones (API) para desarrollar aplicaciones accesibles para plataformas libres y de código abierto, como GNU/Linux o OpenBSD, desarrollada por el Proyecto GNOME. A partir de esta definición podemos nosotros decir que es un software o un conjunto de software que nos permite el desarrollo de un sistema informático de una manera más fácil y eficiente, esto nos va a impactar en la reducción de tiempos y de costo del mismo (Pedraza, 2003).



**CAPÍTULO II**  
**COMPONENTES DEL SISTEMA**

## 2. PLAN DE DESARROLLO DEL SOFTWARE RUP

### 2.1 Introducción

Este plan de desarrollo del software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas 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 fue desarrollado por el universitario Daniel Fabián Ortega Galeán basado en la metodología de Rational Unified Process (RUP) en la que únicamente se procederá a cumplir con las tres primeras fases las cuales marcan la metodología, por lo tanto, se trabajará con las fases de inicio, elaboración y construcción.

El presente documento es una especificación de requerimientos de software (ERS) para el sistema del programa canasta alimentaria de la subgobernación para posteriormente especificar los requerimientos según la norma IEEE-830.

**Inicio.** En esta fase, se establecen los requisitos del negocio que cubrirá el negocio, mediante el cual se obtendrá la especificación de requerimientos.

**Elaboración.** En esta fase, se analiza el problema desde el punto de vista de equipo de desarrollo. Al final de esta fase, se tiene definida la arquitectura y el modelo de requisitos del sistema empleando los diagramas de casos de uso especificados en el diagrama UML.

**Construcción.** En esta fase, se profundiza el diseño de componentes de dicho sistema, como ser las funcionalidades del sistema, pruebas, y permitiendo a la vez que dichos requerimientos puedan cambiarse. Al final de esta fase se obtiene un sistema completamente operativo y la documentación (diagrama de clases, de secuencia, modelo entidad-relación, modelo de dominio, manual de instalación, manual de usuario) para entregar a los usuarios. El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los entregables que serán generados. Este documento es a su vez uno de los artefactos de RUP.

### **2.1.1. Propósito**

El propósito del plan de desarrollo del software es proporcionar la información necesaria para controlar el proyecto. En el cual se describe el enfoque de desarrollo del software. Los usuarios del plan de desarrollo de software son todos los beneficiarios adultos mayores que lo usen para el registro y recojo correspondiente de la canasta alimentaria y todos los miembros del área de caja e informática que lo usen para registro y la entrega de la canasta alimentaria al beneficiario.

### **2.1.2. Alcance**

El plan de desarrollo del software describe el plan global titulado MEJORA DEL PROCESO DE REGISTRO Y DISTRIBUCIÓN, DE LA CANASTA ALIMENTARIA DEL ADULTO MAYOR EN LA SUBGOBERNACIÓN DE CERCADO, INCORPORANDO LAS TIC, el detalle de la documentación se presenta por separado para alcanzar las mejoras planteadas en la especificación de requerimientos. El Sistema Web para el programa canasta alimentaria de la Subgobernación de Cercado debe registrar información de un registro de usuarios, un registro de beneficiarios, un registro de productos, un registro de áreas de entrega, reportes de entrega, reportes de productos y reporte de beneficiarios.

### **2.1.3. Resumen**

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

## **2.2. Vista General del Proyecto.**

### **2.2.1. Propósito.**

Contribuir de manera eficiente y eficaz en el control y manejo adecuado de la información de la canasta alimentaria que permitirá hacer el registro y entrega al beneficiario más rápidamente.

### **2.2.2. Alcance.**

Realizar la automatización del registro de beneficiarios de la canasta alimentaria, entrega de la canasta alimentaria.

- Utilizará un gestor de base de datos de Postgresql para el almacenamiento de registros del programa canasta alimentaria, se usa este gestor por la libre licencia de uso y respuesta rápida del programa.
- Realizará registro y entrega de los beneficiarios mediante reconocimiento facial.
- Brindará un acceso a usuarios registrados mediante un usuario y contraseña registrados por el administrador del sistema.
- Automatizará los reportes, como así también manejará la información del beneficiario de forma eficaz.

## **2.3. Objetivos.**

### **2.3.1. Objetivo General.**

Mejorar el proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC.

### **2.3.2. Objetivos Específicos.**

- Diseñar una base de datos relacional para almacenar información de los beneficiarios y las entregas de las canastas mediante PostgreSQL.
- Crear un módulo de registro de beneficiarios para capturar información, obtener fotogramas y guardar rasgos faciales para la identificación y verificación del mismo.
- Implementar un módulo de reportes llamado Weasyprint para dar cumplimiento a los requerimientos con la librería.
- Aplicar el modelo del lenguaje unificado (UML).
- Aplicar la metodología RUP.
- Crear un módulo para la entrega y asistencia de los beneficiarios que permita hacer el uso de Reconocimiento facial para identificar a la persona registrada en el sistema.

### **2.3.3. Superposiciones y Restricciones.**

#### **2.3.3.1. Superposiciones.**

El personal involucrado en el proyecto apoyará de manera constante para el correcto desarrollo y evolución del sistema. También se supondrá que la sección de Informática perteneciente al programa canasta alimentaria contará con el equipamiento necesario para su implementación y posterior uso. El equipo de desarrollo cuenta con todas las herramientas necesarias para el desarrollo.

#### **2.3.3.2. Restricciones.**

El sistema será solamente en castellano. El sistema no contará con interfaces con otras aplicaciones. El sistema no será accesible a través del internet y el sistema no estará aplicado a ningún sistema operativo móvil, pero si a navegadores móviles conectados a la misma red.

#### **2.3.4. Entregables del Proyecto.**

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

1. Plan de desarrollo de software
2. Visión
3. Modelo de casos de uso del negocio
4. Modelo de objetos del negocio
5. Glosario
6. Modelos de caso de uso
7. Especificación de los casos de uso
8. Especificaciones adicionales
9. Prototipo interfaces del usuario
10. Modelo de datos
11. Modelo de despliegue
12. Casos de prueba

- 13. Manual de usuario
- 14. Material de apoyo al usuario final
- 15. Diagrama de secuencia
- 16. Diagrama de componentes

**2.3.4.1. Evolución del Plan de Desarrollo de Software.**

En la evolución del plan de desarrollo de software se toma como el presente documento.

**2.3.4.2. Visión.**

*Introducción.* Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades del programa Canasta alimentaria. Constituye una base de acuerdo a los requerimientos del sistema.

*Propósito.* Desarrollar un Sistema Web basado en Inteligencia Artificial para el control de la entrega de la canasta alimentaria en la provincia Cercado.

*Alcance.* Se realizará registros de usuarios, registros de beneficiarios y registro de stock de la canasta.

*Limitación.* Entre las limitaciones del producto, señalamos que el sistema no contará con planilla de pago a los proveedores. No se realiza un control del pago por los productos que componen la canasta alimentaria.

**2.3.4.3. Oportunidad de Negocio.**

Con este sistema se mejorará la calidad de información de manera rápida y confiable.

**2.3.4.3.1. Sentencias que define el Proyecto.**

**Tabla 5**

*Sentencias que definen el proyecto*

El problema de:	No existe un adecuado registro de beneficiarios y productos de stock de la canasta alimentaria
Afecta a:	Adultos mayores Encargado de proyecto

	Técnico de distribución
El impacto asociado es:	Mejorar la calidad de información en la empresa. Mejorar y realzar la imagen institucional. Mantener la información actualizada y confiable.
Una solución adecuada sería:	Automatizar los registros correspondientes a la asignación, distribución de la canasta alimentaria usando un sistema de basado en reconocimiento facial para agilizar el registro y entrega al beneficiario.

### 2.3.4.3.2. Sentencias que definen la posición del Proyecto.

**Tabla 6**

#### *Sentencias que definen la Posición del Proyecto*

Para	Subgobernación de Cercado, programa canasta alimentaria.
¿Quiénes?	Interactúan de manera directa e indirecta con el sistema.
El nombre del producto	Sistema de Asignación y entrega de la canasta alimentaria (SICAL).
¿Qué?	Mejorar procesos de registro y distribución de la canasta. Tener información oportuna, adecuada y disponible Brindar seguridad a la información.
¿Cómo?	El proceso que se maneja actualmente, es mediante hojas electrónicas.
Nuestro producto	Registra al adulto mayor. Asigna la canasta alimentaria. Mantiene la información actualizada del stock. Genera reportes instantáneamente.

### 2.3.4.3.3. Descripción de los participantes en el desarrollo del Sistema y Usuarios.

Para proveer una forma efectiva de productos y servicios que se ajusten a las necesidades de los usuarios, es necesario involucrar a todos los participantes en el desarrollo del sistema como parte del proceso de modelado de requerimientos.

#### **2.3.4.4. Descripción Global del Sistema.**

##### **2.3.4.4.1. Supuestos y Dependencias.**

Los supuestos y dependencias están mencionadas en la matriz de marco lógico del proyecto.

##### **2.3.4.5. Glosario.**

**Introducción.** Este documento recoge todos y cada uno de los términos manejados en el desarrollo del sistema, se trata de un diccionario informal de datos y de las definiciones de nomenclatura que se maneja, de modo que se maneja un estándar para todo el proyecto.

**Propósito.** El propósito es comprender con exactitud los términos y la tecnología que va desarrollando el proyecto, lo cual también sirve como guía de consulta para la aclaración de algunas palabras difíciles de entender.

**Alcance.** El alcance del presente se extiende a todo el sistema.

##### **2.3.4.5.1. Organización del Proyecto.**

El presente documento está organizado por definiciones de términos ordenados en forma ascendente según el alfabeto:

UML. Lenguaje Unificado de Modelos

ERS. Especificación de Requerimientos de Software

SICAL. Sistema Informático Canasta Alimentaria

Adulto Mayor.

Canasta Alimentaria.

Stock.

Reconocimiento Facial.

Usuario. Encargado de programa/Técnico responsable de distribución/técnico de área/técnico de entrega.

Encargado de Programa. Administra el programa, designa roles.

Técnico responsable de Distribución. Planifica la distribución de la canasta, supervisa cantidad de stock.

Técnico de Área. Encargado de puntos de distribución.

Técnico de Entrega. Distribuye la canasta alimentaria.



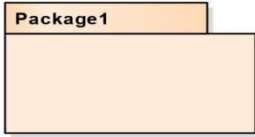

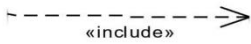
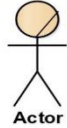

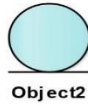


Técnico de Atención al Beneficiario. Registra al beneficiario.

## 2.4 Glosario de Diagramas

Tabla 7

### Glosario de Diagramas

Glosario de Diagramas	
<b>Objetos de Negocio</b>	
<b>Caso de Uso</b>	
<b>Componentes</b>	
<b>Relación de Extensión</b>	
<b>Relación de Inclusión</b>	
<b>Actor del Negocio</b>	 Actor
<b>Actor</b>	 Actor
<b>Representa una base de datos</b>	 Object2

## 2.5 Modelo de Casos de Uso del Negocio

**Introducción.** Este es un modelo de funciones del negocio visto desde la perspectiva desde agentes externos que permite situar al sistema en el contexto organizacional haciendo énfasis a los objetivos de dicho ámbito. La definición del conjunto de procesos del negocio es una de las tareas principales ya que define los límites

del proceso modelado, teniendo en cuenta los objetivos para la identificación de los procesos del negocio. Para lo cual se presentan modelos definidos en RUP (modelo de casos de uso del negocio y modelo de objetos).

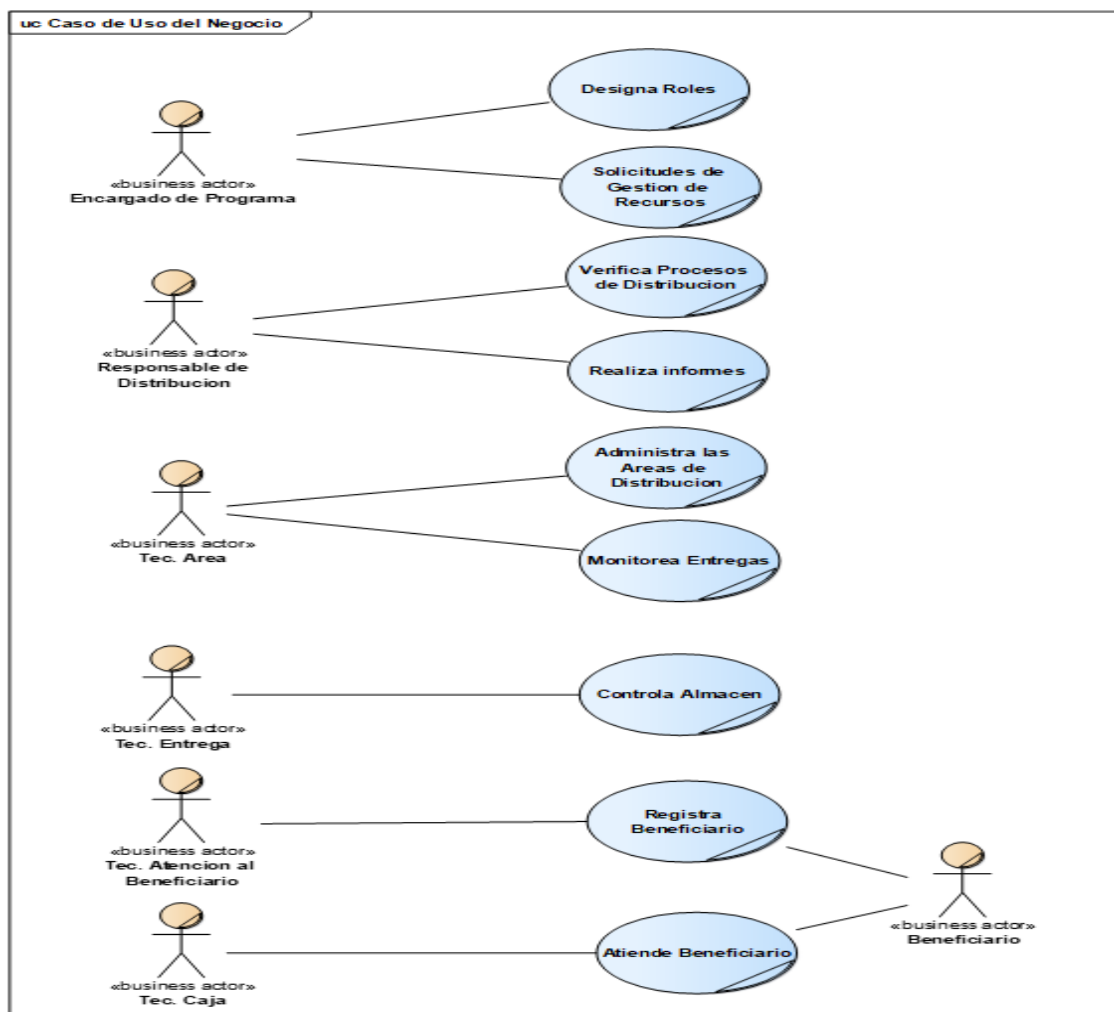
**Propósitos.** Comprender la estructura organizacional del programa canasta alimentaria, y los procesos que llevan a cabo e identificar los problemas actuales para llevar a cabo posibles mejoras.

**Alcances.** Describir los procesos del negocio y de sus trabajadores. Definir un caso de uso del negocio para cada proceso e identificar y definir los procesos del negocio según los objetivos del programa.

### 2.5.1. Diagrama de Caso de Uso General del Negocio.

*Figura 1*

*Diagrama de Caso de Uso General del Negocio*



## 2.6 Modelos de Caso de Uso.

**Introducción.** El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones, este describe lo que hace el sistema para cada tipo de usuario, los mismos que son fragmentos de la funcionalidad, estos especifican una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

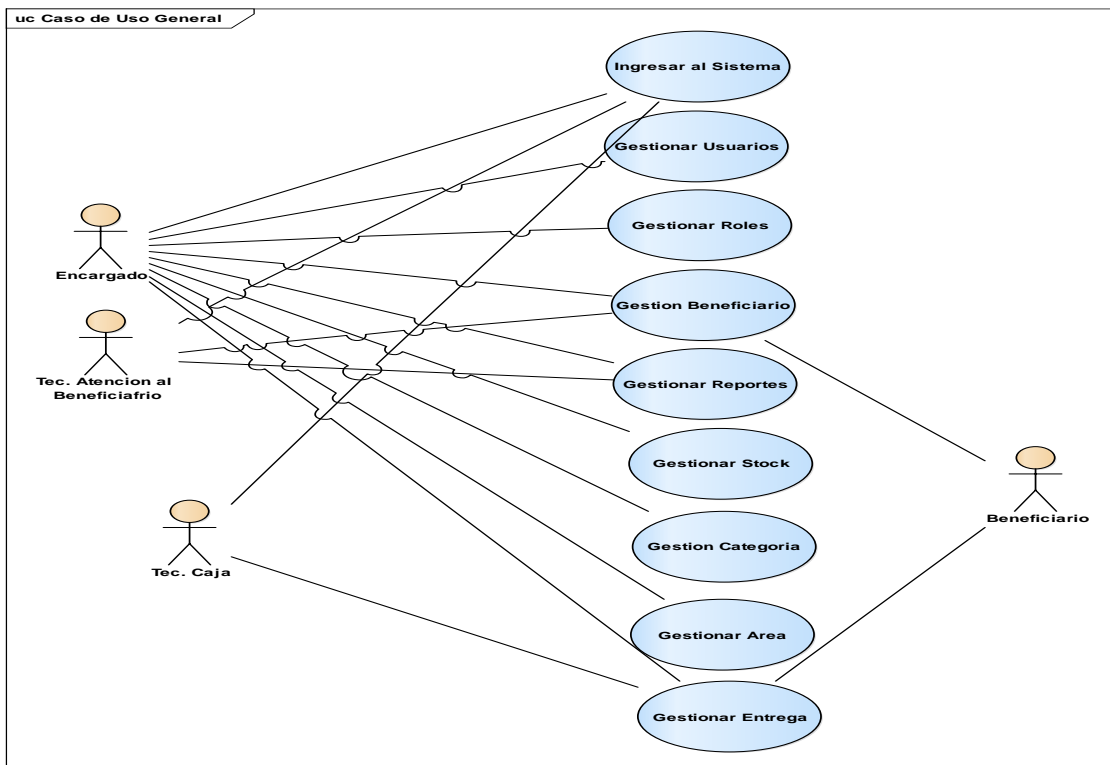
**Propósitos.** Comprender la estructura y la dinámica del sistema deseado para la organización e identificar las posibles mejoras para la empresa.

**Alcances.** Describir procesos del sistema. Identificar y definir los procesos del sistema según objetivos de la organización y definir un caso de uso para cada proceso del sistema.

### 2.6.1. Diagrama de Caso de Uso General.

*Figura 2*

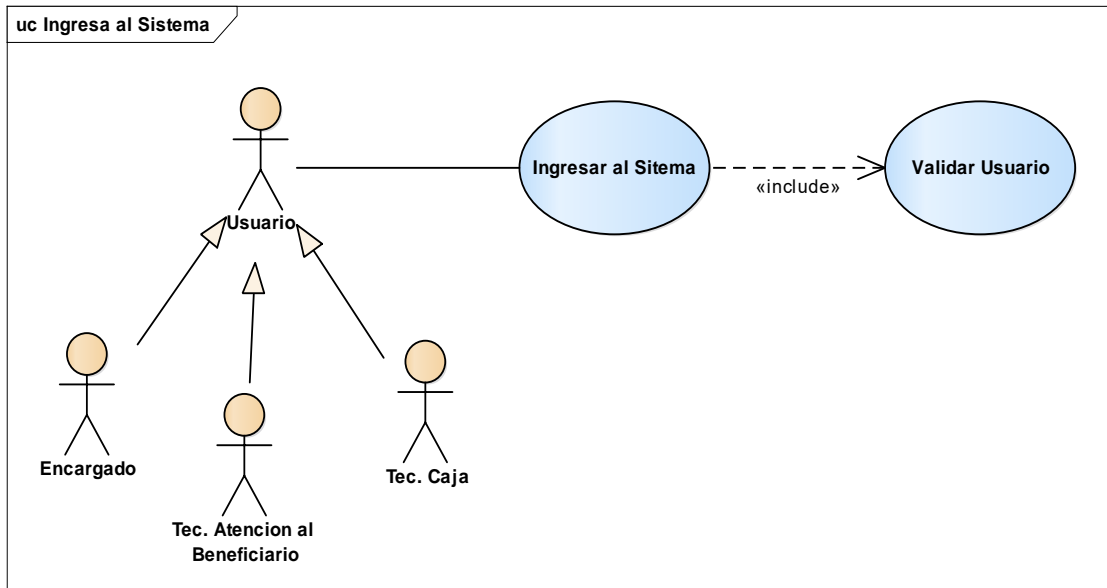
*Diagrama de Caso de Uso General*



### 2.6.2. Diagramas de Caso de Uso Específico.

Figura 3

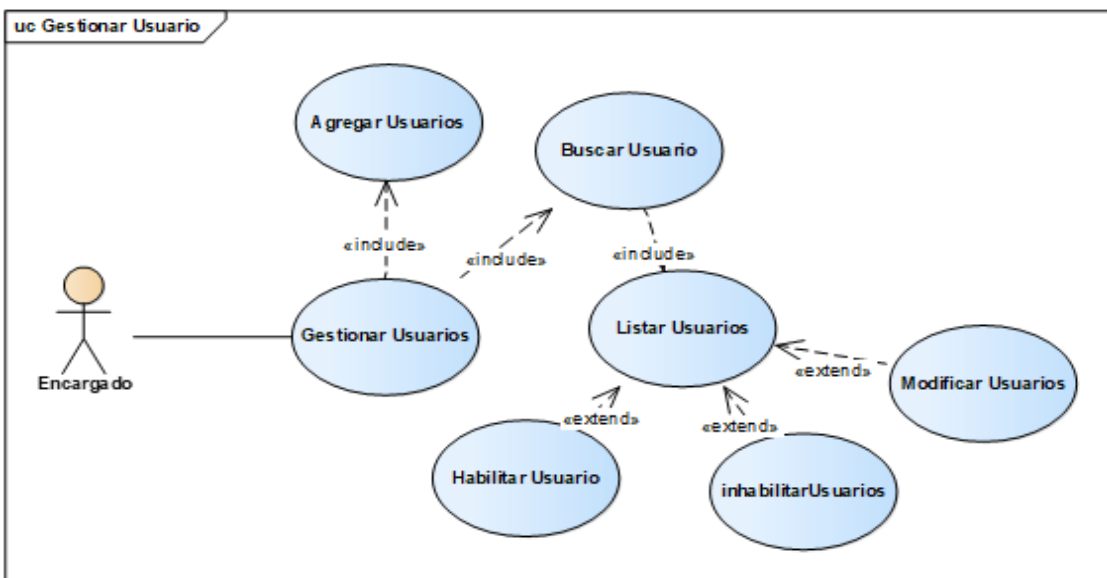
Diagrama de Caso de Uso: Ingreso al Sistema



### 2.6.3. Diagrama de Caso de Uso: Gestionar Usuario.

Figura 4

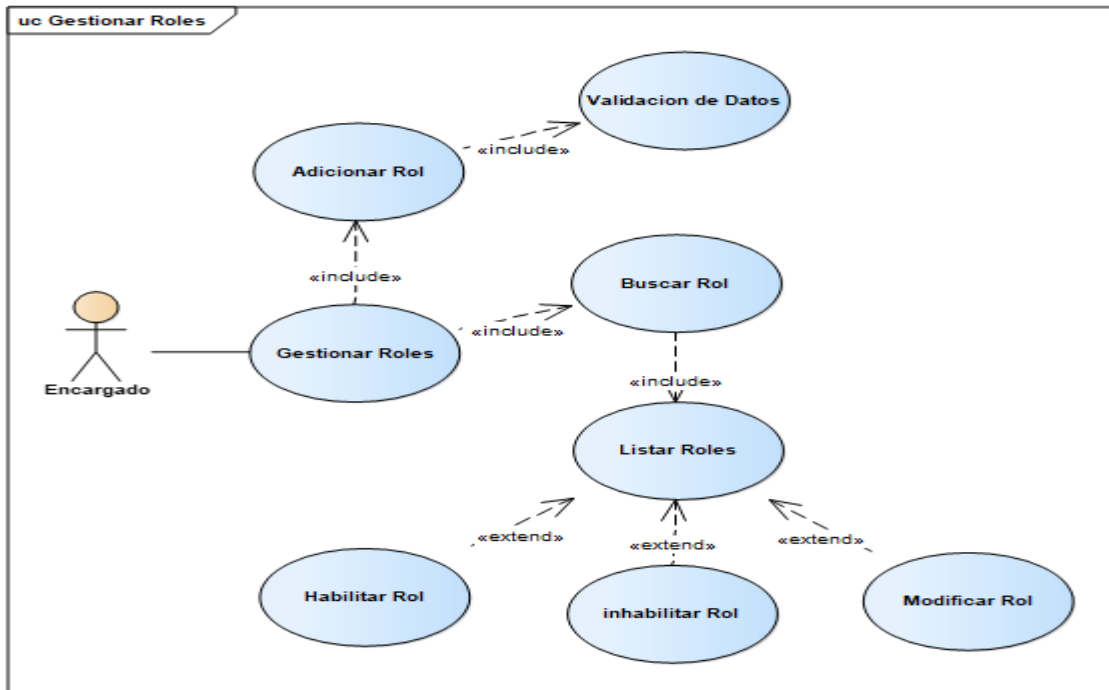
Diagrama de Caso de uso: Gestionar Usuario



## 2.6.4. Diagrama de Caso de Uso: Gestionar Roles.

Figura 5

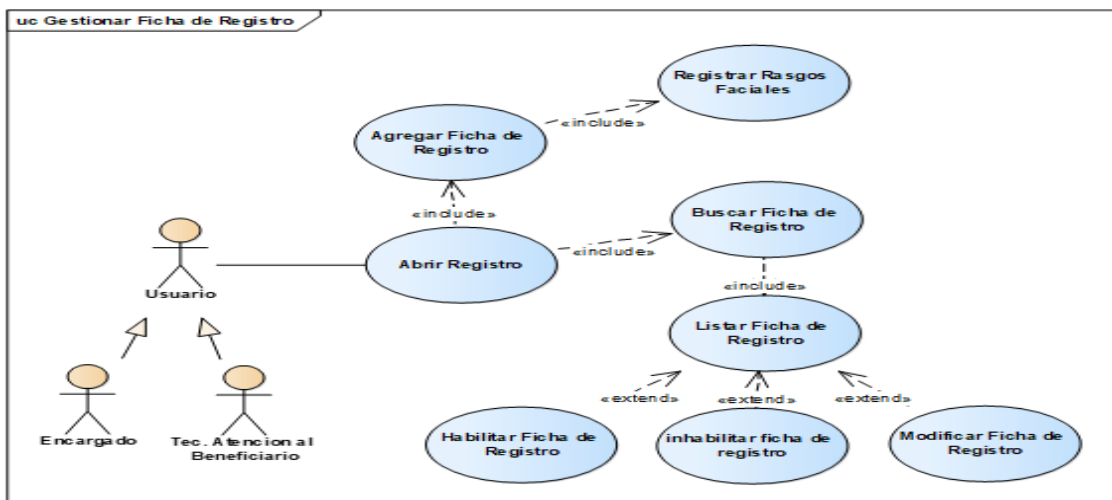
Diagrama de Caso de Uso: Gestionar Roles



## 2.6.5. Diagrama de caso de uso: Gestionar Ficha de Registro.

Figura 6

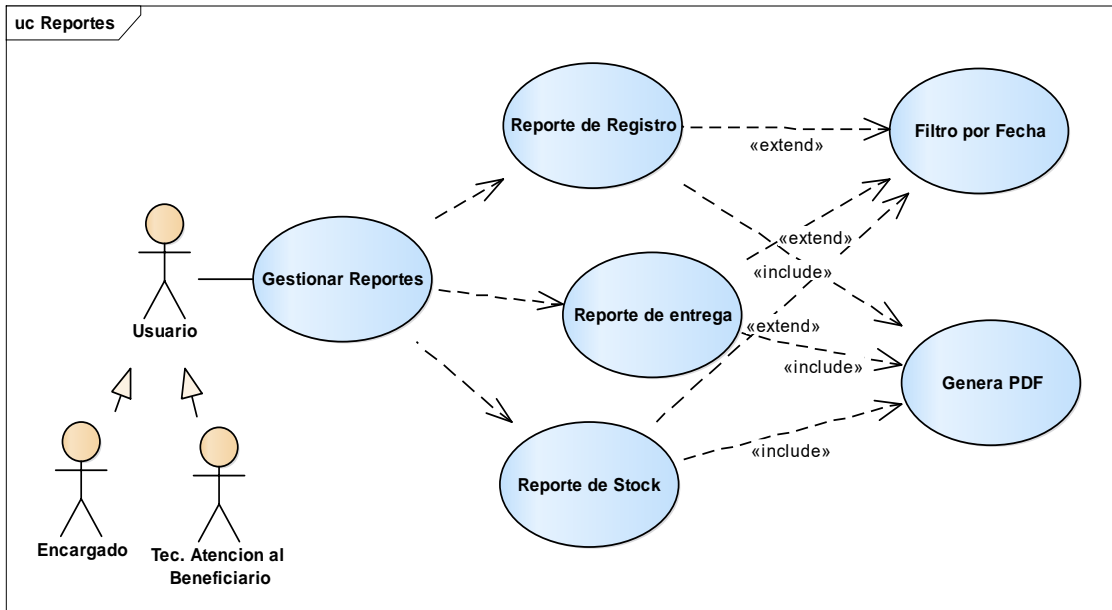
Diagrama de caso de uso: Gestionar Ficha de Registro



### 2.6.6. Diagrama de Caso de Uso: Gestionar Reportes.

Figura 7

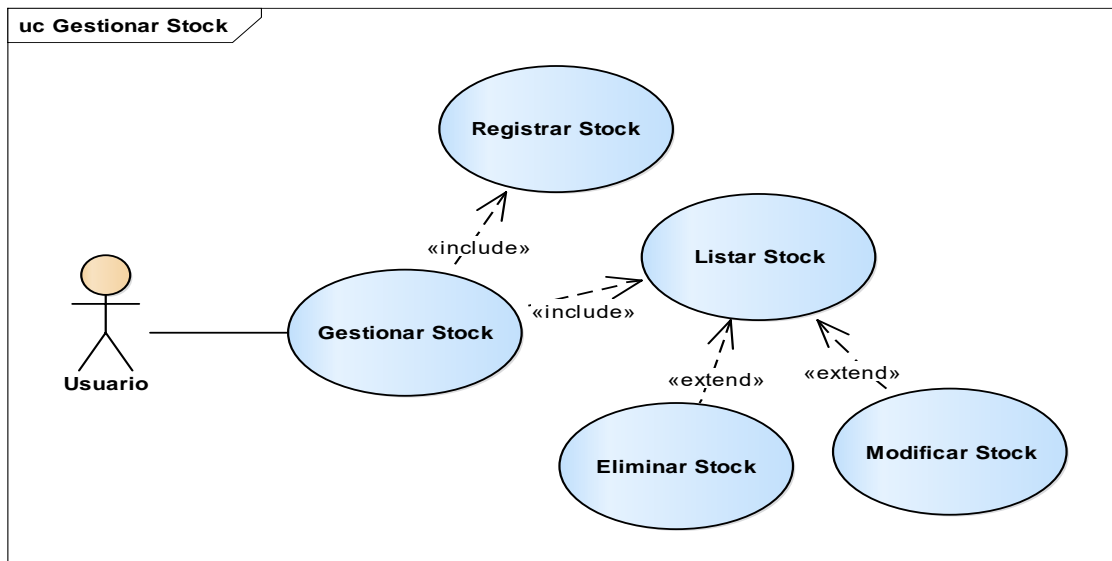
Diagrama de Caso de Uso: Gestionar Reportes



### 2.6.7. Diagrama de Caso de Uso: Gestionar Stock.

Figura 8

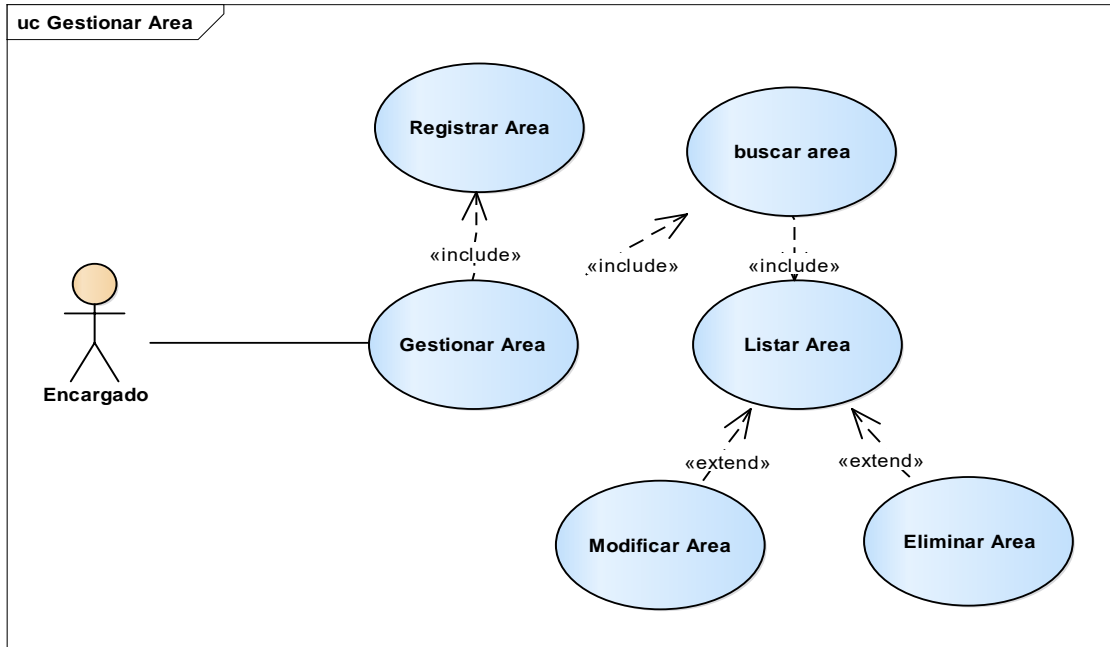
Diagrama de Caso de Uso: Gestionar Stock



## 2.6.8. Diagrama de Caso de Uso: Gestionar Área.

Figura 9

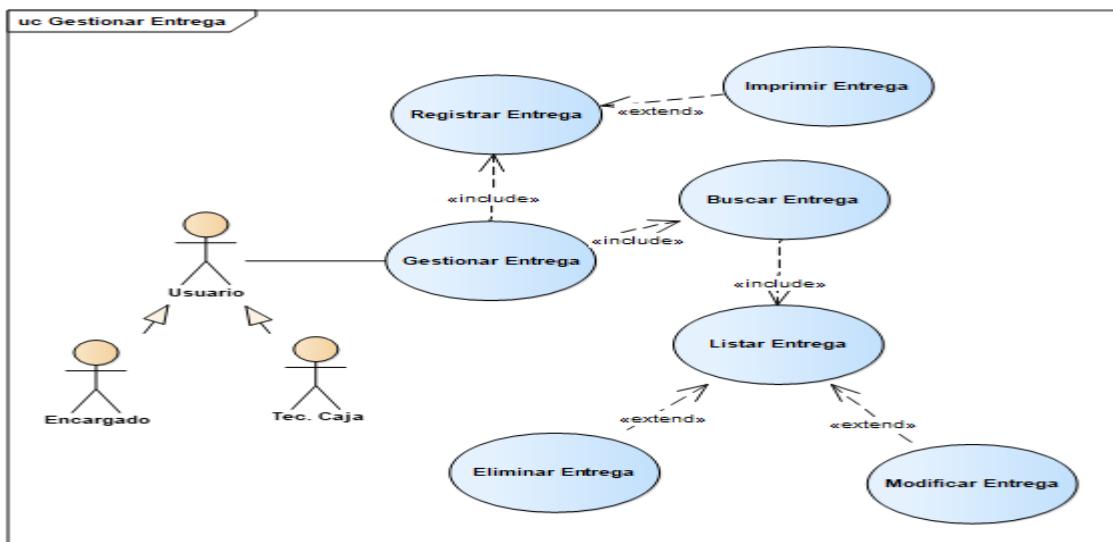
Diagrama de Caso de Uso: Gestionar Área



## 2.6.9. Diagrama de Caso de Uso: Gestionar Entrega.

Figura 10

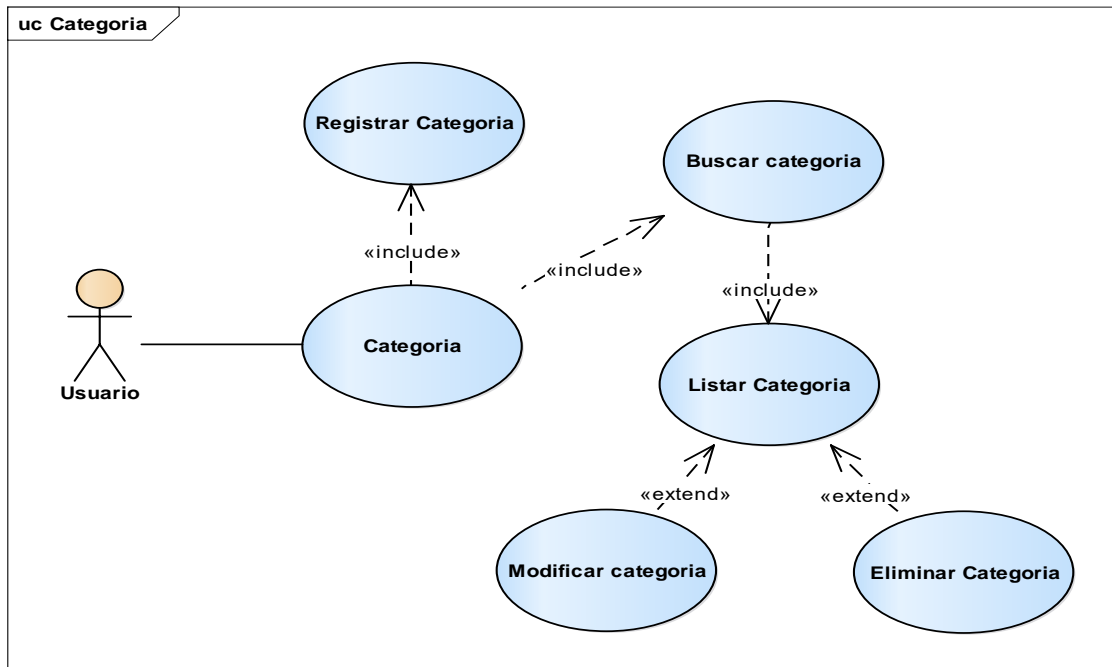
Diagrama de Caso de Uso: Gestionar Entrega



## 2.6.10. Diagrama de Caso de Uso: Categoría.

Figura 11

Diagrama de Caso de Uso: Categoría



## 2.7. Especificación de Casos de Usos.

**Introducción.** La especificación de los casos de uso es una descripción más detallada de los casos de uso del sistema. En las descripciones se hace énfasis a los conceptos que tiene cada caso de uso del sistema correspondiente.

**Propósitos.** Comprender los casos de uso del sistema y describir específicamente cada uno de los casos de uso del sistema.

**Alcances.** Describir los flujos de cada caso de uso según lo establecido por la organización y describir los procesos internos de los casos de uso.



### 2.7.1. Ingresar al sistema

**Tabla 8**

**Ingresar al sistema**

Nombre:	Ingresar al Sistema	
Actores:	Encargado, Técnico Atención al Beneficiario, Técnico Caja.	
Propósito:	Mostrar un formulario para ingresar usuario y password.	
Resumen:	Que el usuario ingrese sus datos para poder ingresar al sistema	
Tipo:	Básico	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del sistema</b>
	1-El Usuario elige acceder a la pantalla "Login".	2-El sistema mostrará la pantalla "Login".
	3-El Usuario introduce los datos de "Usuario" y "Password" y eligen la opción "Aceptar" de la pantalla "Login".	4-Muestra la pantalla "Principal" con los datos encontrados en las tablas "usuario" y "rol"
		5-Muestra una alerta "Bienvenido al Sistema"
Flujo Alternativo:		4.-Muestra un mensaje "Usuario o Clave incorrecta" "Ingresa login y clave nuevamente"
Precondición:	El usuario debe haber ingresado a la pantalla "Login" del Sistema	

## 2.7.2. Desplegar Menú

**Tabla 9**

### *Desplegar Menú*

Nombre:	Desplegar Menú	
Actores:	Encargado, Técnico Atención al Beneficiario, Técnico Caja.	
Propósito:	Desplegar menús del sistema.	
Resumen:	Permite desplegar menú y sus opciones en el sistema.	
Tipo:	Básico.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del sistema</b>
	1-El Usuario selecciona la opción “Cualquiera de los que están en el menú” según su rol y elige la opción “para desplegar” de la pantalla “Gestión Menús”.	2-Se abre opciones en la pantalla “Seleccionada”.
	3-El Usuario elige la opción del formulario “Menú”.	4-Se cierra “Menú”, muestra la página seleccionada.
Flujo alternativo:	3.-El Usuario no elige la opción.	4-No se muestra ninguna pantalla solo la principal.
Precondición:	Para que el administrador pueda ver un menú debe estar en la pantalla “home”.	

### 2.7.3. Gestión Usuario

**Tabla 10**

**Gestión Usuario**

Nombre:	Gestión usuarios	
Actores:	Encargado.	
Propósito:	Gestionar los usuarios del sistema	
Resumen:	Adicionar usuarios, modificar usuarios, eliminar usuarios, habilitar usuarios.	
Tipo:	Include.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del sistema</b>
	1-El usuario elige la opción “Gestión Usuarios” del menú “Sistema” de la pantalla “principal”.	2-Muestra las opciones Buscar Usuario y Agregar Usuario, estos son mostrados en la pantalla “Gestión Usuarios”.
	3-El usuario realiza la búsqueda por nombre y apellidos del usuario de la pantalla “Gestión Usuarios”.	4-El sistema busca en la tabla “Usuario” y lista a estas en la pantalla “Gestión Usuarios”.
	5-El usuario selecciona (Agregar Usuario), de la pantalla “Gestión Usuarios”.	6-Despliega el formulario “Adicionar Nuevo Usuario”.
	7-El usuario selecciona un registro y puede seleccionar (Modificar), (Eliminar), (Ver) de la pantalla “Gestión Usuarios”.	8.-Despliega los formularios respectivos de las ventanas “Modificar Usuario”, “Eliminar Usuario” y “Ver Usuario”
	9-El usuario elige la opción “Pág. Ant.” o “Pág. Sig.” de la pantalla “Gestión Usuarios”.	10-Se muestra los anteriores registros o siguientes registros de la tabla “persona” del sistema, en la pantalla “Gestión Usuarios”.
Flujo alternativo:		2-Si no existe persona registrada el sistema muestra una lista vacía.
Precondición:	El administrador debe haber ingresado logueado previamente al sistema como administrador.	

## 2.7.4. Agregar Nuevo Usuario

**Tabla 11**

### *Agregar Nuevo Usuario*

Nombre:	Agregar nuevo usuario	
Actores:	Encargado.	
Propósito:	Agregar nuevos usuarios al sistema.	
Resumen:	Permite agregar nuevos usuarios al sistema bajo ciertos requisitos.	
Tipo:	Include.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del sistema</b>
	1-El encargado elige la opción (Agregar) de la pantalla “Gestión Usuarios”.	2-Se abre el formulario “Agregar Nuevo Usuario” en la pantalla “Gestión Usuarios”.
	3-El encargado carga los datos en el formulario y elige la opción “Guardar” del formulario “Agregar Nuevo Usuario”.	4-Se abre el formulario “Confirmación Adicionar Persona”.
	5-El encargado elige la opción “Si”.	6-Se cierra el formulario “Agregar Nuevo Usuario”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Flujo alternativo:	5.-El encargado elige la opción “No”.	4- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	7.-El encargado elige la opción “Cancelar”	6-Se cierra el formulario “Agregar Nuevo Usuario”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Usuarios”.
		8.-Se cierra el formulario “Agregar Nuevo Usuario” y retorna a la pantalla “Gestión Usuarios”
Precondición:	Para que el encargado pueda Agregar un nuevo usuario debe estar en la pantalla “Gestión Usuarios”.	

## 2.7.5. Modificar Usuario

**Tabla 12**

### *Modificar Usuario*

Nombre:	Modificar Usuario.	
Actores:	Encargado.	
Propósito:	Modificar Usuario del sistema.	
Resumen:	Permite modificar usuarios del sistema bajo ciertos requisitos.	
Tipo:	Extend.	
	<b>Acción del Actor</b>	<b>Evento del Sistema</b>
Flujo principal:	1-El encargado elige la opción Buscar Usuario de la pantalla “Gestion usuarios”.	2-El sistema lista al usuario en la pantalla “Gestion Usuarios”.
	3-El encargado elige la opción (Modificar) de la pantalla “Gestión Usuarios”.	4-El sistema abre el formulario “Modificar Usuario” en la pantalla “Gestión Usuarios”.
	5-El encargado carga los datos en el formulario y elige la opción “Guardar” del formulario “Modificar Persona”.	6-Se abre el formulario “Confirmación Modificar Usuario”.
	7-El encargado elige la opción “Si”.	8-Se cierra el formulario “Modificar Usuario”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Flujo Alternativo:	7.-El Usuario elige la opción “No”.	6- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	9.-El Usuario elige la opción “Cancelar”	8-Se cierra el formulario “Modificar Usuario”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
		10.-Se cierra el formulario “Confirmación Modificar Usuario”, y retorna al formulario “Modificar Usuario”.
		11.-Se cierra el formulario “Modificar

Precondición:	Para que el encargado pueda Modificar una persona debe estar en la pantalla “Gestión Usuarios”.	Usuario” y retorna a la pantalla “Gestión Usuarios”.
---------------	---	--

## 2.7.6. Eliminar Usuario

**Tabla 13**

### *Eliminar Usuario*

Nombre:	Eliminar Usuario.	
Actores:	Encargado.	
Propósito:	Eliminar Usuario del sistema.	
Resumen:	Permite Eliminar usuario del sistema.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del Actor</b>	<b>Evento del Sistema</b>
	1-El encargado elige la opción Buscar Usuario de la pantalla “Gestión usuarios”.	2-El Sistema lista al usuario en la pantalla “Gestión Usuarios”.
	3-El encargado elige la opción (Eliminar) de la pantalla “Gestión Usuarios”.	4-Se abre un formulario “Eliminar Usuario” en la pantalla “Gestión Usuarios”.
	5-El Administrador elige la opción “Si” del formulario “Eliminar Persona”.	6-Se cierra el formulario “Eliminar Usuario”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Flujo alternativo:	5.-El Usuario elige la opción “No”	6-Se cierra el formulario “Eliminar Usuario”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Usuarios”.
Precondición:	Para que el encargado pueda eliminar una persona debe estar en la ventana “Gestión Usuarios”.	

## 2.7.7. Habilitar Usuarios

**Tabla 14**

### *Habilitar Usuarios*

Nombre:	Habilitar Usuarios.	
Actores:	Encargado.	
Propósito:	Habilitar Usuarios que se dieron de baja del sistema.	
Resumen:	Permite habilitar Imputados que se dieron de baja en el sistema.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del Actor</b>	<b>Evento del Sistema</b>
	1-El encargado elige la opción Buscar Usuario de la pantalla “gestión usuarios”.	2-El sistema lista al usuario en la pantalla “Gestión Usuarios”.
	3-El encargado selecciona la opción (Habilitar) de un Usuarios en la pantalla “Gestión Usuarios”.	4-El sistema abre un formulario “Habilitar Usuarios” en la ventana “Gestión Usuarios”.
	5-El encargado elige la opción “Si” del formulario “Habilitar Usuarios”.	6-Se cierra el formulario “Habilitar Usuarios”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Flujo alternativo:	5.-El encargado elige la opción “No”	6-Se cierra el formulario “Habilitar Usuarios”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Precondición:	Para que el usuario pueda habilitar un usuario debe estar en la pantalla “Gestión Usuarios”.	

## 2.7.8. Inhabilitar Usuarios

**Tabla 15**

### *Inhabilitar Usuarios*

Nombre:	Inhabilitar usuarios.	
Actores:	Encargado.	
Propósito:	Habilitar Usuarios que se dieron de baja del Sistema.	
Resumen:	Permite habilitar Imputados que se dieron de baja en el Sistema.	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del sistema</b>
	1-El encargado elige la opción Buscar Usuario de la pantalla “gestión usuarios”.	2-El sistema lista al usuario en la pantalla “Gestión Usuarios”.
	3-El encargado selecciona la opción (Inhabilitar) de un Usuarios en la pantalla “Gestión Usuarios”.	4-El sistema abre un formulario “Inhabilitar Usuarios” en la ventana “Gestión Usuarios”.
	5-El encargado elige la opción “Si” del formulario “Inhabilitar Usuarios”.	6-Se cierra el formulario “Inhabilitar Usuarios”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Flujo Alternativo:	5.-El encargado elige la opción “No”	6-Se cierra el formulario “Inhabilitar Usuarios”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Usuarios”.
Precondición:	Para que el usuario pueda Habilitar un Usuario debe estar en la pantalla “Gestión Usuarios”.	



## 2.7.9. Gestión Roles

**Tabla 16**

### **Gestión Roles**

Nombre:	Gestión Roles.	
Actores:	Encargado.	
Propósito:	Gestionar los usuarios del sistema.	
Resumen:	Que puedan acceder a la gestión de roles, el sistema contará con accesos a los roles (adicionar, modificar, eliminar).	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El Encargado abre la opción “gestión de roles”.  3-El usuario podrá ver las distintas opciones para lo que precise hacer en el sistema en relación a los roles.	2-El sistema entra a la opción gestión de roles mostrando la lista de roles registrados en el sistema, el sistema contará con distintos tipos para los registros del sistema (Adicionar, modificar, eliminar un rol).  4-El sistema mostrará las opciones de gestión de roles en la pantalla
Precondición:	El Administrador debe haber ingresado logueado previamente al Sistema como administrador	

## 2.7.10. Adicionar Rol

*Tabla 17*

### *Adicionar Rol*

Nombre:	Adicionar Rol.	
Actores:	Encargado.	
Propósito:	Agregar nuevos roles al sistema.	
Resumen:	Que puedan agregar roles a la lista de roles registrados.	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario entra a la vista “gestión de roles”	2-El sistema mostrará la pantalla de lista de roles
	3-El usuario presiona en el botón “agregar nuevo rol”.	4-El sistema envía la a la pantalla un formulario de registro para el llenado del nuevo rol requerido, tendrá dos botones (guardar, cancelar)
	5.-El usuario presiona el botón de “guardar”.	6-El sistema guarda los datos de registro y vuelve a la lista de roles.
Flujo Alternativo:	5.-El usuario presiona el botón de “cancelar”.	6-El sistema no guarda los datos de registro y vuelve a la lista de roles.
Precondición:	Para que el encargado pueda Agregar un nuevo rol debe estar en la pantalla “Gestión Rol”.	

## 2.7.11. Modificar Rol

**Tabla 18**

### **Modificar Rol**

Nombre:	Modificar Rol.	
Actores:	Encargado.	
Propósito:	Modificar rol del sistema.	
Resumen:	Que puedan modificar los datos de los roles registrados	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del sistema</b>
	1-El usuario presionara en el botón “gestión de roles” para acceder a la lista de roles.	2-El sistema enviará a la pantalla de “gestión de roles” mostrando los roles registrados.
	3-El usuario presiona en modificar los datos del rol.	4-El sistema accede a un formulario para la modificación de los registros en el rol, el formulario contará con dos botones: actualizar y cancelar.
	5.-El usuario modifica los datos y presiona en el botón “actualizar”.	6.-El sistema validará los datos y registra los datos en la tabla “rol”.
Flujo Alternativo:	5.-El usuario presiona el botón de “cancelar”.	6-El sistema no hará nada y volverá a la vista de lista de roles.
Precondición:	Para que el encargado pueda modificar el rol debe estar en la pantalla “Gestión Rol”.	

## 2.7.12. Eliminar Rol

**Tabla 19**

### ***Eliminar Rol***

Nombre:	Eliminar Rol.	
Actores:	Encargado.	
Propósito:	Eliminar rol del sistema.	
Resumen:	Que el usuario pueda eliminar un registro de los roles registrados.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario accederá a la vista de gestión de roles.	2-El sistema enviará a la pantalla de gestión de roles mostrando los roles registrados.
	3-El usuario presiona sobre la opción eliminar.	4-El sistema enviará una ventana modal preguntando si quiere eliminar el rol
	5.-El usuario presionara sobre el botón de “aceptar”.	el sistema tendrá dos botones: aceptar y cancelar.
		6.-El sistema verificará los registros y eliminará el rol seleccionado.
Flujo alternativo:	5.-El usuario presionara sobre el botón “cancelar”.	6-El sistema no hará nada y volverá a la pantalla de listas de roles.
Precondición:	Para que el encargado pueda eliminar el rol debe estar en la pantalla “Gestión Rol”.	

## 2.7.13. Gestión Beneficiario

**Tabla 20**

### **Gestión Beneficiario**

Nombre:	Gestión Ficha de Registro	
Actores:	Encargado, técnico atención al beneficiario	
Propósito:	Gestionar las Ficha de Registro	
Resumen:	Agregar, modificar y habilitar Ficha de Registro.	
Tipo:	Include.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción “Abrir Registro” del menú de la pantalla “principal”.	2-Muestra las opciones Buscar Registro y Agregar Ficha, estos son mostrados en la pantalla “Gestión Usuarios”.
	3-El usuario realiza la búsqueda por nombre del Beneficiario de la pantalla “Ficha de Registro”.	4-El sistema lista todas las fichas actualmente registradas en el sistema, estas son mostradas en la pantalla “Gestión Registro”.
	5-El usuario selecciona (Agregar Ficha), de la pantalla “Gestión Registro”.	6-Despliega el formulario de la ventana “Agregar Nueva Ficha”.
	7-El usuario selecciona un registro y puede seleccionar (Modificar), (Habilitar) de la pantalla “Ficha Registro”.	8.-Despliega los formularios respectivos “Modificar Ficha” y “Habilitar Ficha”.
	9-El usuario elige la opción “Pág. Ant.” o “Pág. Sig.” de la pantalla “Ficha Registro”.	10-El sistema muestra los anteriores registros o siguientes registros de la tabla “Fecha_registro” del sistema, en la pantalla “Ficha Registro”.
Flujo Alternativo:		2-Si no existe Fichas registradas el sistema muestra una lista vacía.
Precondición:	El usuario debe haber ingresado logueado previamente al sistema.	

## 2.7.14 Agregar Beneficiario

**Tabla 21**

### *Agregar Beneficiario*

Nombre:	Agregar Nueva Ficha de Registro.	
Actores:	Encargado, técnico atención al beneficiario	
Propósito:	Agregar nuevas Fichas de Registro al sistema.	
Resumen:	Permite agregar nueva Ficha de Registro al sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Nueva Ficha de Registro) de la pantalla “Gestión Registro”.	2-Se abre el formulario “Agregar Nueva Ficha de Registro” en la pantalla “Ficha Registro”.
	3-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Agregar Nueva Ficha de Registro”.	4-Se cierra el formulario “Agregar Nueva Ficha de Registro”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Ficha de Registro”.
Flujo alternativo:	5.-El usuario elige la opción “Cancelar”.	4- Si los datos no son válidos el formulario muestra autoayudas para que el usuario corrija los datos ingresados.  4-Se cierra el formulario “Agregar Ficha de Registro”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Ficha Registro”.  6.- Se cierra el formulario “Agregar Ficha de Registro” y vuelve a la pantalla “Ficha Registro”.
Precondición:	Para que el usuario pueda Agregar una nueva ficha de registro debe estar en la pantalla “Ficha Registro”.	

## 2.7.15. Modificar Beneficiario

**Tabla 22**

### **Modificar Beneficiario**

Nombre:	Modificar Ficha de Registro.	
Actores:	Encargado, técnico atención al beneficiario	
Propósito:	Modificar Ficha de Registro.	
Resumen:	Permite modificar fichas de registro del sistema bajo ciertos requisitos.	
Tipo:	Extend.	
	<b>Acción del actor</b>	<b>Evento del Sistema</b>
Flujo principal:	1-El usuario elige la opción Buscar Usuario de la pantalla “Gestión usuarios”.	2-El sistema lista al usuario en la pantalla “Gestión Usuarios”.
	3-El usuario elige la opción (Modificar) de la pantalla “gestión fichas de registro”.	4-El sistema abre el formulario “Modificar fichas de registro” en la pantalla “Gestión Fichas de Registro”.
	5-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Modificar Registro”.	6-Se abre el formulario “Confirmación Modificar Registro”.
Flujo alternativo:	7-El usuario elige la opción “Si”.	8-Se cierra el formulario “Modificar Registro”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Fichas de Registro”.
	7.-El Usuario elige la opción “No”. 5.-El Usuario elige la opción “Cancelar”.	6- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados. 8-Se cierra el formulario “Modificar Registro”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Fichas de Registro”.

Precondición: Para que el usuario pueda Modificar un Registro debe estar en la pantalla “Gestión Fichas de Registro”.

## 2.7.16. Habilitar Beneficiario

**Tabla 23**

### *Habilitar Beneficiario*

Nombre:	Habilitar Ficha de Registro.	
Actores:	Encargado, técnico atención al beneficiario	
Propósito:	Habilitar Registros que se dieron de baja del sistema.	
Resumen:	Permite habilitar beneficiarios que se dieron de baja en el sistema.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El encargado elige la opción Buscar Usuario de la pantalla “Gestión usuarios”.	2-El sistema lista al usuario en la pantalla “Gestión Usuarios”.
	3-El usuario selecciona la opción (Habilitar) de un Registro en la pantalla “Gestión Fichas de Registro”.	4-El sistema abre un formulario “Habilitar Registro” en la ventana “Gestión Fichas de Registro”.
	5-El usuario elige la opción “Si” del formulario “Habilitar Registros”.	6-Se cierra el formulario “Habilitar Registros”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Fichas de Registros”.
Flujo alternativo:	5.-El usuario elige la opción “No”.	6-Se cierra el formulario “Habilitar Registros”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Fichas de Registros”.
Precondición:	Para que el usuario pueda Habilitar un Registro debe estar en la pantalla “Gestión Fichas de Registros”.	



## 2.7.17. Inhabilitar Beneficiario

**Tabla 24**

### ***Inhabilitar Beneficiario***

Nombre:	Inhabilitar Ficha de Registro.	
Actores:	Encargado, técnico atención al beneficiario	
Propósito:	Inhabilitar Registros que se dieron de baja del sistema.	
Resumen:	Permite Inhabilitar beneficiarios en el sistema.	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción Buscar Usuario de la pantalla “Gestión usuarios”.	2-El sistema lista al usuario en la pantalla “Gestión Usuarios”.
	3-El usuario selecciona la opción (inhabilitar) de un Registro en la pantalla “Gestión Fichas de Registro”.	4-El sistema abre un formulario “Inhabilitar Registro” en la ventana “Gestión Fichas de Registro”.
	5-El usuario elige la opción “Si” del formulario “Inhabilitar Registros”.	6-Se cierra el formulario “Inhabilitar Registros”, muestra una alerta “se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Fichas de Registros”.
Flujo Alternativo:	5.-El usuario elige la opción “No”	6-Se cierra el formulario “Inhabilitar Registros”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Fichas de Registros”.
Precondición:	Para que el usuario pueda Inhabilitar un Registro debe estar en la pantalla “Gestión Fichas de Registros”.	

## 2.7.18. Agregar Fotogramas Beneficiario

*Tabla 25*

### *Agregar Fotogramas Beneficiario*

Nombre:	Agregar fotogramas del beneficiario.	
Actores:	Encargado, técnico atención al beneficiario	
Propósito:	Agregar fotogramas del beneficiario al sistema.	
Resumen:	Permite agregar fotogramas de los Beneficiarios al sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Guardar) de la pantalla “Gestión Fichas de Registro (nuevo Registro)”.	2-Se abre la cámara web “Agrega Fotogramas del imputado” en la pantalla “Gestión Ficha de Registro”.
		3-Se cierra el formulario “Agregar Nuevo Imputado”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Imputados”.
Flujo Alternativo:		4- Si los datos no son válidos el formulario muestra autoayudas para que el usuario corrija los datos ingresados.
Precondición:	Para que el usuario pueda Agregar fotogramas de un beneficiario debe estar en la pantalla “Gestión Fichas de Registro” y posteriormente hacer click en guardar.	

## 2.7.19. Gestionar Stock

**Tabla 26 Gestionar Stock**

Nombre:	Gestionar Stock	
Actores:	Encargado, técnico área.	
Propósito:	Gestionar el stock de productos.	
Resumen:	Adicionar stock, modificar stock, eliminar stock.	
Tipo:	Include.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción “Gestionar Stock” del menú “Sistema” de la pantalla “principal”.	2-Lista el stock disponible registrado en el sistema, estos son mostrados en la pantalla “Gestionar Stock”.
	3-El usuario realiza la búsqueda del stock disponible “Gestionar stock”.	4-El sistema busca en la tabla “Stock” y lista a estas en la pantalla “Gestión Stock”.
	5-El usuario selecciona (Agregar Stock), de la pantalla “Gestionar Stock”.	6-Despliega el formulario “Adicionar Nuevo Stock”.
	7-El usuario selecciona un registro y puede seleccionar (Modificar), (Eliminar), (Ver) de la pantalla “Gestionar Stock”.	8.-Despliega los formularios respectivos de las ventanas “Modificar Stock”, “Eliminar Stock” y “Ver Stock”.
	9-El usuario elige la opción “Pág. Ant.” o “Pág. Sig.” de la pantalla “Gestionar Stock”.	10-Se muestra los anteriores registros o siguientes registros de la tabla “Stock” del sistema, en la pantalla “Gestionar Stock”.
Flujo alternativo:		2-Si no existe Stock registrado el sistema muestra una lista vacía.
Precondición:	El usuario debe haber ingresado logueado previamente al sistema como Encargado o técnico área.	

## 2.7.20. Agregar Nuevo Stock

**Tabla 27**

### *Agregar Nuevo Stock*

Nombre:	Agregar Nuevo Stock	
Actores:	Encargado, técnico área.	
Propósito:	Agregar nuevo stock al sistema.	
Resumen:	Permite agregar nuevo stock disponible al sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Agregar) de la pantalla “Gestionar Stock”.	2-Se abre el formulario “Agregar Nuevo Stock” en la pantalla “Gestión Stock”.
	3-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Agregar Nuevo Stock”.	4-Se abre el formulario “Confirmación Adicionar Stock”.
	5-El usuario elige la opción “Si”.	6-Se cierra el formulario “Agregar Nuevo Stock”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
Flujo alternativo:	5.-El usuario elige la opción “No”.	4- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	7.-El usuario elige la opción “Cancelar”.	6-Se cierra el formulario “Agregar Nuevo Stock”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
		8.-Se cierra el formulario “Agregar Nuevo Stock” y retorna a la pantalla “Gestión Usuarios”
Precondición:	Para que el encargado pueda Agregar un nuevo stock debe estar en la pantalla “Gestión Stock”.	

## 2.7.21. Modificar Stock

**Tabla 28**

### *Modificar Stock*

Nombre:	Modificar Stock.	
Actores:	Encargado.	
Propósito:	Modificar Stock disponible del sistema.	
Resumen:	Permite modificar stock disponible del sistema bajo ciertos requisitos.	
Tipo:	Extend.	
	<b>Acción del Actor</b>	<b>Evento del Sistema</b>
Flujo Principal:	1-El usuario elige la opción (Modificar) de la pantalla “Gestión Stock”.	2-El sistema abre el formulario “Modificar Stock” en la pantalla “Gestión Stock”.
	3-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Modificar Stock”.	4-Se abre el formulario “Confirmación Modificar Stock”.
	5-El usuario elige la opción “Si”.	6-Se cierra el formulario “Modificar Stock”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
Flujo Alternativo:	5.-El Usuario elige la opción “No”.	4- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	7.-El Usuario elige la opción “Cancelar”	6-Se cierra el formulario “Modificar Stock”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
		6.-Se cierra el formulario “Confirmación Modificar Stock”, y retorna al formulario “Modificar Stock”.
		8.-Se cierra el formulario “Modificar Stock” y retorna a la pantalla “Gestión Stock”.

Precondición:	Para que el usuario pueda Modificar el stock debe estar en la pantalla “Gestión stock”.
---------------	---

### 2.7.22. Eliminar Stock

**Tabla 29**

#### *Eliminar Stock*

Nombre:	Eliminar Stock.	
Actores:	Encargado, técnico área.	
Propósito:	Eliminar Stock del sistema.	
Resumen:	Permite Eliminar stock disponible del sistema.	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Eliminar) de la pantalla “Gestión Stock”.	2-Se abre un formulario “Eliminar Stock” en la pantalla “Gestión Stock”.
	3-El usuario elige la opción “Si” del formulario “Eliminar stock”.	4-Se cierra el formulario “Eliminar stock”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
Flujo Alternativo:	3.-El Usuario elige la opción “No”.	4.-Se cierra el formulario “Eliminar Stock” y retorna a la pantalla “Gestión Stock”.
		4-Se cierra el formulario “Eliminar Stock”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Stock”.
Precondición:	Para que el usuario pueda Eliminar stock disponible debe estar en la ventana “Gestión Stock”.	

### 2.7.23. Modificar Stock

**Tabla 30**

#### **Modificar Stock**

Nombre:	Modificar Stock.	
Actores:	Encargado, técnico área.	
Propósito:	Modificar Stock del sistema.	
Resumen:	Permite Modificar stock disponible del sistema.	
Tipo:	Extend.	
Flujo Prncipal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Modificar) de la pantalla “Gestión Stock”.	2-Se abre un formulario “Modificar Stock” en la pantalla “Gestión Stock”.
	3-El usuario elige la opción “Si” del formulario “Modificar stock”.	4-Se cierra el formulario “Modificar stock”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
Flujo alternativo:	3.-El Usuario elige la opción “No”	4.-Se cierra el formulario “Modificar Stock” y retorna a la pantalla “Gestión Stock”.
		4-Se cierra el formulario “Modificar Stock”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Stock”.
Precondición:	Para que el usuario pueda Modificar stock disponible debe estar en la ventana “Gestión Stock”.	

## 2.7.24. Gestionar Área

**Tabla 31**

### *Gestionar Área*

Nombre:	Gestionar Área	
Actores:	Encargado, técnico área.	
Propósito:	Gestionar el stock de productos	
Resumen:	Adicionar stock, modificar stock, liminar stock.	
Tipo:	Include.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	<p>1-El usuario elige la opción “Gestionar Área” del menú “Sistema” de la pantalla “principal”.</p> <p>3-El usuario realiza la búsqueda del Área disponible “Gestionar Área”.</p> <p>5-El usuario selecciona (Agregar Área), de la pantalla “Gestionar Área”.</p> <p>7-El usuario selecciona un registro y puede seleccionar (Modificar), (Eliminar), (Ver) de la pantalla “Gestionar Área”.</p> <p>9-El usuario elige la opción “Pág. Ant.” o “Pág. Sig.” de la pantalla “Gestionar Área”.</p>	<p>2-Lista las áreas disponibles registrado en el sistema, estas son mostradas en la pantalla “Gestionar Área”.</p> <p>4-El sistema busca en la tabla “Área” y lista a estas en la pantalla “Gestión Área”.</p> <p>6-Despliega el formulario “Adicionar Nuevo Área”.</p> <p>8.-Despliega los formularios respectivos de las ventanas “Modificar Área”, “Eliminar Área” y “Ver Área”</p> <p>10-Se muestra los anteriores registros o siguientes registros de la tabla “Área” del sistema, en la pantalla “Gestionar Stock”.</p>
Flujo alternativo:		2-Si no existe Área registrada el sistema muestra una lista vacía.
Precondición:	El usuario debe haber ingresado logueado previamente al sistema como Encargado o técnico área.	



## 2.7.25. Agregar Nueva Área

**Tabla 32**

### *Agregar Nueva Área*

Nombre:	Agregar Nueva Área	
Actores:	Encargado, técnico área.	
Propósito:	Agregar nueva área al sistema.	
Resumen:	Permite agregar nueva área al sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo Principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Agregar) de la pantalla “Gestionar Área”.	2-Se abre el formulario “Agregar Nuevo Área” en la pantalla “Gestión Área”.
	3-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Agregar Nueva Área”.	4-Se abre el formulario “Confirmación Adicionar Área”.
	5-El usuario elige la opción “Si”.	6-Se cierra el formulario “Agregar Nuevo Área”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Área”.
Flujo Alternativo:	5.-El usuario elige la opción “No”.	4- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	7.-El usuario elige la opción “Cancelar”	6-Se cierra el formulario “Agregar Nuevo Área”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Área”.
		8.-Se cierra el formulario “Agregar Nuevo Área” y retorna a la pantalla “Gestión Área”.
Precondición:	Para que el usuario pueda Agregar una nueva área debe estar en la pantalla “Gestión Área”.	

## 2.7.26. Modificar Área

**Tabla 33**

### *Modificar Área*

Nombre:	Modificar Área.	
Actores:	Encargado, técnico área.	
Propósito:	Modificar Área del sistema.	
Resumen:	Permite modificar el Área del sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El encargado elige la opción Buscar Área de la pantalla “Gestión Aérea”.	2-El sistema lista el área en la pantalla “Gestión Aérea”.
	3-El usuario elige la opción (Modificar) de la pantalla “Gestión Área”.	4-El sistema abre el formulario “Modificar Área” en la pantalla “Gestión Área”.
	5-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Modificar Área”.	6-Se abre el formulario “Confirmación Modificar Área”.
	7-El usuario elige la opción “Si”.	8-Se cierra el formulario “Modificar Área”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Área”.
Flujo alternativo:	7.-El Usuario elige la opción “No”.	6- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	5.-El Usuario elige la opción “Cancelar”.	8-Se cierra el formulario “Modificar Área”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Gestión Área”.
Precondición:	Para que el usuario pueda Modificar el Área debe estar en la pantalla “Gestión Área”.	

## 2.7.27. Eliminar Área

**Tabla 34**

### **Eliminar Área**

Nombre:	Eliminar Área.	
Actores:	Encargado, técnico área.	
Propósito:	Eliminar Área del sistema.	
Resumen:	Permite Eliminar Área disponible del sistema.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El encargado elige la opción Buscar Área de la pantalla “Gestión Aérea”.	2-El sistema lista el área en la pantalla “Gestión Aérea”.
	3-El usuario elige la opción (Eliminar) de la pantalla “Gestión Área”.	4-Se abre un formulario “Eliminar Área” en la pantalla “Gestión Área”.
	5-El usuario elige la opción “Sí” del formulario “Eliminar Área”.	6-Se cierra el formulario “Eliminar Área”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Gestión Área”.
Flujo alternativo:	5.-El Usuario elige la opción “No”.	6.-Se cierra el formulario “Eliminar Área” y retorna a la pantalla “Gestión Área”.
Precondición:	Para que el usuario pueda Eliminar Área disponible debe estar en la ventana “Gestión Área”.	

## 2.7.28. Categoría

**Tabla 35**

### *Categoría*

Nombre:	Categoría	
Actores:	Encargado	
Propósito:	Gestionar categorías de productos	
Resumen:	Adicionar categoría, modificar categoría, eliminar categoría.	
Tipo:	Include.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción “Categoría” del menú “Sistema” de la pantalla “principal”.	2-Lista las categorías disponibles registrado en el sistema, estas son mostradas en la pantalla “Categoría”.
	3-El usuario realiza la búsqueda de la categoría disponible “Categoría”.	4-El sistema busca en la tabla “Categoría” y lista a estas en la pantalla “Categoría”.
	5-El usuario selecciona (Agregar Categoría), de la pantalla “Categoría”.	6-Despliega el formulario “Adicionar Nueva Categoría”.
	7-El usuario selecciona un registro y puede seleccionar (Modificar), (Eliminar), (Ver) de la pantalla “Categoría”.	8.-Despliega los formularios respectivos de las ventanas “Modificar Categoría”, “Eliminar Categoría” y “Ver Categoría”.
	9-El usuario elige la opción “Pág. Ant.” o “Pág. Sig.” de la pantalla “Categoría”.	10-Se muestra los anteriores registros o siguientes registros de la tabla “Categoría” del sistema, en la pantalla “Categoría”.
Flujo alternativo:		2-Si no existe Categoría registrada el sistema muestra una lista vacía.
Precondición:	El usuario debe haber ingresado logueado previamente al sistema como encargado.	

## 2.7.26. Agregar Nueva Categoría

**Tabla 36**

### *Agregar Nueva Categoría*

Nombre:	Agregar nueva categoría.	
Actores:	Encargado.	
Propósito:	Agregar nueva categoría de producto al sistema.	
Resumen:	Permite agregar nueva categoría al sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El usuario elige la opción (Agregar) de la pantalla “Categoría”.	2-Se abre el formulario “Agregar Nueva Categoría” en la pantalla “Categoría”.
	3-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Agregar Nueva Categoría”.	4-Se abre el formulario “Confirmación Adicionar Categoría”.
	5-El usuario elige la opción “Si”.	6-Se cierra el formulario “Agregar Nueva Categoría”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Categoría”.
Flujo alternativo:	5.-El usuario elige la opción “No”.	4- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.
	7.-El usuario elige la opción “Cancelar”.	6-Se cierra el formulario “Agregar Nueva Categoría”, muestra una alerta “No se realizó con éxito el proceso” y vuelve a la pantalla “Categoría”.
		8.-Se cierra el formulario “Agregar Nueva Categoría” y retorna a la pantalla “Categoría”.
Precondición:	Para que el usuario pueda Agregar una nueva Categoría debe estar en la pantalla “Categoría”.	

## 2.7.27 Modificar Categoría

*Tabla 37*

### *Modificar Categoría*

Nombre:	Modificar Categoría.	
Actores:	Encargado.	
Propósito:	Modificar categoría de producto del sistema.	
Resumen:	Permite modificar la categoría del sistema bajo ciertos requisitos.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	<p>1-El encargado elige la opción Buscar Categoría de la pantalla “Categoría”.</p> <p>3-El usuario elige la opción (Modificar) de la pantalla “Categoría”.</p> <p>5-El usuario carga los datos en el formulario y elige la opción “Guardar” del formulario “Modificar Categoría”.</p> <p>7-El usuario elige la opción “Si”.</p>	<p>2-El sistema lista la Categoría en la pantalla “Categoría”.</p> <p>4-El sistema abre el formulario “Modificar Categoría” en la pantalla “Categoría”.</p> <p>6-Se abre el formulario “Confirmación Modificar Categoría”.</p> <p>8-Se cierra el formulario “Modificar Categoría”, muestra una alerta “Se realizó con éxito el proceso”, y vuelve a la pantalla “Categoría”.</p>
Flujo alternativo:	<p>7.-El Usuario elige la opción “No”.</p> <p>5.-El Usuario elige la opción “Cancelar”.</p>	<p>6- Si los datos no son válidos el formulario muestra autoayudas para que usuario corrija los datos ingresados.</p> <p>8-Se cierra el formulario “Modificar Categoría”, muestra una alerta “No se realizó con éxito el proceso”, y vuelve a la pantalla “Categoría”.</p>
Precondición:	Para que el usuario pueda Modificar la Categoría debe estar en la pantalla “Categoría”.	

## 2.7.28. Eliminar Categoría

**Tabla 38**

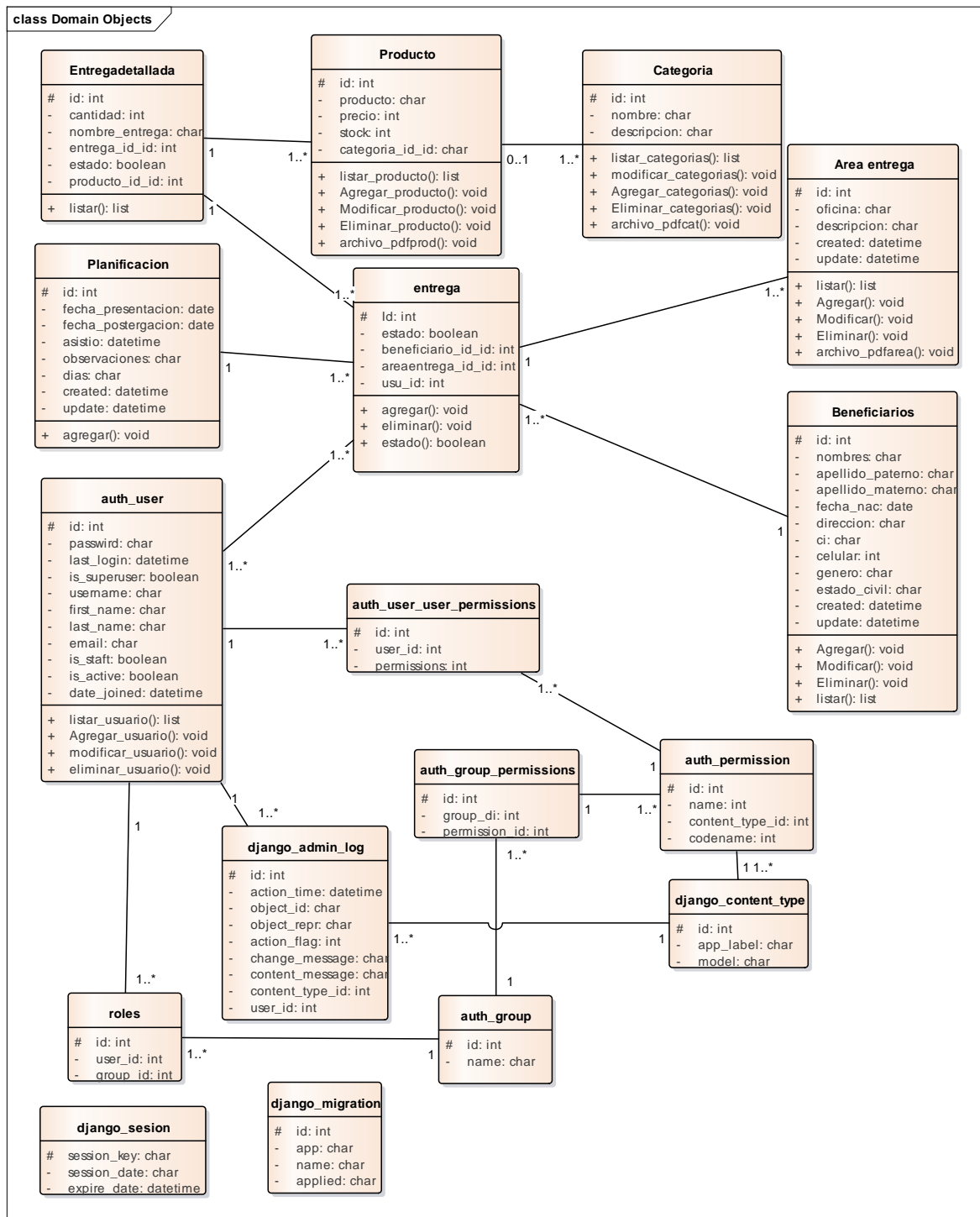
### *Eliminar Categoría*

Nombre:	Eliminar categoría.	
Actores:	Encargado.	
Propósito:	Eliminar categoría de producto del sistema.	
Resumen:	Permite eliminar área disponible del sistema.	
Tipo:	Extend.	
Flujo principal:	<b>Acción del actor</b>	<b>Evento del Sistema</b>
	1-El encargado elige la opción Buscar Categoría de la pantalla “Categoría”.	2-El sistema lista la Categoría en la pantalla “Categoría”.
	3-El usuario elige la opción (Eliminar) de la pantalla “Categoría”.	4-Se abre un formulario “Eliminar Categoría” en la pantalla “Categoría”.
	5-El usuario elige la opción “Si” del formulario “Eliminar Categoría”.	6-Se cierra el formulario “Eliminar Categoría”, muestra una alerta “Se realizó con éxito el proceso” y vuelve a la pantalla “Categoría”.
Flujo alternativo:	5.-El Usuario elige la opción “No”.	6.-Se cierra el formulario “Eliminar Categoría” y retorna a la pantalla “Categoría”.
Precondición:	Para que el usuario pueda Eliminar Categoría disponible debe estar en la ventana “Categoría”.	

## 2.8. Diagrama de Clases

Figura 12

Diagrama de Clases







### 2.9.1. Script de la Base de Datos.

```
DROP TABLE IF EXISTS "administraciones_areaentrega";
DROP SEQUENCE IF EXISTS administraciones_areaentrega_id_seq;
CREATE SEQUENCE administraciones_areaentrega_id_seq INCREMENT 1
MINVALUE 1 MAXVALUE 9223372036854775807 START 1 CACHE 1;
CREATE TABLE "public"."administraciones_areaentrega" ("id" bigint
DEFAULT nextval('administraciones_areaentrega_id_seq') NOT NULL,
"oficina" character varying (40) NOT NULL,
"descripcion" character varying (100) NOT NULL,
"created" timestamptz NOT NULL,
"updated" timestamptz NOT NULL,
CONSTRAINT "administraciones_areaentrega_pkey" PRIMARY KEY ("id")
WITH (oids = false);

DROP TABLE IF EXISTS "administraciones_categoria";
DROP SEQUENCE IF EXISTS administraciones_categoria_id_seq;
CREATE SEQUENCE administraciones_categoria_id_seq INCREMENT 1
MINVALUE 1 MAXVALUE 9223372036854775807 START 1 CACHE 1;
CREATE TABLE "public"."administraciones_categoria" ("id" bigint
DEFAULT nextval('administraciones_categoria_id_seq') NOT NULL,
"nombre" character varying (40) NOT NULL,
"descripcion" character varying (100) NOT NULL,
"created" timestamptz NOT NULL,
"updated" timestamptz NOT NULL,
CONSTRAINT "administraciones_categoria_pkey" PRIMARY KEY ("id")
WITH (oids = false);

DROP TABLE IF EXISTS "administraciones_producto";
DROP SEQUENCE IF EXISTS administraciones_producto_id_seq;
CREATE SEQUENCE administraciones_producto_id_seq INCREMENT 1
MINVALUE 1 MAXVALUE 9223372036854775807 START 1 CACHE 1;
```

```

CREATE TABLE "public"."administraciones_producto" ("id" bigint
DEFAULT nextval('administraciones_producto_id_seq') NOT NULL,
"producto" character varying (40) NOT NULL,
"precio" integer NOT NULL,
"stock" integer NOT NULL,
"created" timestamptz NOT NULL,
"updated" timestamptz NOT NULL,
"categoria_id_id" bigint NOT NULL,
CONSTRAINT "administraciones_producto_pkey" PRIMARY KEY ("id"),
CONSTRAINT "administraciones_pro_categoria_id_id_fdeeb145_fk_administr"
FOREIGN KEY (categoria_id_id) REFERENCES administraciones_categoria(id)
DEFERRABLE INITIALLY DEFERRED DEFERRABLE
WITH (oids = false);
CREATE INDEX "administraciones_producto_categoria_id_id_fdeeb145" ON
"public"."administraciones_producto" USING btree ("categoria_id_id");
DROP TABLE IF EXISTS "auth_group";
DROP SEQUENCE IF EXISTS auth_group_id_seq;
CREATE SEQUENCE auth_group_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;
CREATE TABLE "public"."auth_group" ("id" integer DEFAULT
nextval('auth_group_id_seq') NOT NULL,
"name" character varying (150) NOT NULL,
CONSTRAINT "auth_group_name_key" UNIQUE ("name"),
CONSTRAINT "auth_group_pkey" PRIMARY KEY ("id")
WITH (oids = false);
CREATE INDEX "auth_group_name_a6ea08ec_like" ON
"public"."auth_group" USING btree ("name");
DROP TABLE IF EXISTS "auth_group_permissions";
DROP SEQUENCE IF EXISTS auth_group_permissions_id_seq;
CREATE SEQUENCE auth_group_permissions_id_seq INCREMENT 1 MINVALUE
1 MAXVALUE 2147483647 START 1 CACHE 1;

```

```

CREATE TABLE "public"."auth_group_permissions" ("id" integer DEFAULT
nextval('auth_group_permissions_id_seq') NOT NULL,
"group_id" integer NOT NULL,
"permission_id" integer NOT NULL,
CONSTRAINT "auth_group_permissions_group_id_permission_id_0cd325b0_uniq"
UNIQUE ("group_id", "permission_id"),
CONSTRAINT "auth_group_permissions_pkey" PRIMARY KEY ("id"),
CONSTRAINT "auth_group_permissio_permission_id_84c5c92e_fk_auth_perm"
FOREIGN KEY (permission_id) REFERENCES auth_permission(id) DEFERRABLE
INITIALLY DEFERRED DEFERRABLE,
CONSTRAINT "auth_group_permissions_group_id_b120cbf9_fk_auth_group_id"
FOREIGN KEY (group_id) REFERENCES auth_group(id) DEFERRABLE
INITIALLY DEFERRED DEFERRABLE
WITH (oids = false);

CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON
"public"."auth_group_permissions" USING btree ("group_id");

CREATE INDEX "auth_group_permissions_permission_id_84c5c92e" ON
"public"."auth_group_permissions" USING btree ("permission_id");

DROP TABLE IF EXISTS "auth_permission";
DROP SEQUENCE IF EXISTS auth_permission_id_seq;
CREATE SEQUENCE auth_permission_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;

CREATE TABLE "public"."auth_permission" (
"id" integer DEFAULT nextval('auth_permission_id_seq') NOT NULL,
"name" character varying (255) NOT NULL,
"content_type_id" integer NOT NULL,
"codename" character varying (100) NOT NULL,
CONSTRAINT
"auth_permission_content_type_id_codename_01ab375a_uniq" UNIQUE
("content_type_id", "codename"),
CONSTRAINT "auth_permission_pkey" PRIMARY KEY ("id"),

```

```

CONSTRAINT "auth_permission_content_type_id_2f476e4b_fk_django_co"
FOREIGN KEY (content_type_id) REFERENCES django_content_type(id)
DEFERRABLE INITIALLY DEFERRED DEFERRABLE
) WITH (oids = false);

CREATE INDEX "auth_permission_content_type_id_2f476e4b" ON
"public"."auth_permission" USING btree ("content_type_id");

DROP TABLE IF EXISTS "auth_user";

DROP SEQUENCE IF EXISTS auth_user_id_seq;

CREATE SEQUENCE auth_user_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;

CREATE TABLE "public"."auth_user" (
"id" integer DEFAULT nextval('auth_user_id_seq') NOT NULL,
"password" character varying (128) NOT NULL,
"last_login" timestamptz,
"is_superuser" boolean NOT NULL,
"username" character varying (150) NOT NULL,
"first_name" character varying (150) NOT NULL,
"last_name" character varying (150) NOT NULL,
"email" character varying (254) NOT NULL,
"is_staff" boolean NOT NULL,
"is_active" boolean NOT NULL,
"date_joined" timestamptz NOT NULL,
CONSTRAINT "auth_user_pkey" PRIMARY KEY ("id"),
CONSTRAINT "auth_user_username_key" UNIQUE ("username")
) WITH (oids = false);

CREATE INDEX "auth_user_username_6821ab7c_like" ON
"public"."auth_user" USING btree ("username");

DROP TABLE IF EXISTS "auth_user_groups";

DROP SEQUENCE IF EXISTS auth_user_groups_id_seq;

CREATE SEQUENCE auth_user_groups_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;

```

```

CREATE TABLE "public"."auth_user_groups" (
" id" integer DEFAULT nextval('auth_user_groups_id_seq') NOT NULL,
"user_id" integer NOT NULL,
"group_id" integer NOT NULL,
CONSTRAINT "auth_user_groups_pkey" PRIMARY KEY ("id"),
CONSTRAINT "auth_user_groups_user_id_group_id_94350c0c_uniq" UNIQUE
("user_id", "group_id"),
CONSTRAINT "auth_user_groups_group_id_97559544_fk_auth_group_id"
FOREIGN KEY (group_id) REFERENCES auth_group(id) DEFERRABLE
INITIALLY DEFERRED DEFERRABLE,
CONSTRAINT "auth_user_groups_user_id_6a12ed8b_fk_auth_user_id" FOREIGN
KEY (user_id) REFERENCES auth_user(id) DEFERRABLE INITIALLY DEFERRED
DEFERRABLE
) WITH (oids = false);

CREATE INDEX "auth_user_groups_group_id_97559544" ON
"public"."auth_user_groups" USING btree ("group_id");

CREATE INDEX "auth_user_groups_user_id_6a12ed8b" ON
"public"."auth_user_groups" USING btree ("user_id");

DROP TABLE IF EXISTS "auth_user_user_permissions";
DROP SEQUENCE IF EXISTS auth_user_user_permissions_id_seq;
CREATE SEQUENCE auth_user_user_permissions_id_seq INCREMENT 1
MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1;

CREATE TABLE "public"."auth_user_user_permissions" (
" id" integer DEFAULT nextval('auth_user_user_permissions_id_seq') NOT NULL,
"user_id" integer NOT NULL,
"permission_id" integer NOT NULL,
CONSTRAINT "auth_user_user_permissions_pkey" PRIMARY KEY ("id"),
CONSTRAINT "auth_user_user_permissions_user_id_permission_id_14a6b632_uniq"
UNIQUE ("user_id", "permission_id"),

```

```

CONSTRAINT "auth_user_user_permi_permission_id_1fbb5f2c_fk_auth_perm"
FOREIGN KEY (permission_id) REFERENCES auth_permission(id) DEFERRABLE
INITIALLY DEFERRED DEFERRABLE,
CONSTRAINT "auth_user_user_permissions_user_id_a95ead1b_fk_auth_user_id"
FOREIGN KEY (user_id) REFERENCES auth_user(id) DEFERRABLE INITIALLY
DEFERRED DEFERRABLE
) WITH (oids = false);

CREATE INDEX "auth_user_user_permissions_permission_id_1fbb5f2c" ON
"public"."auth_user_user_permissions" USING btree ("permission_id");

CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON
"public"."auth_user_user_permissions" USING btree ("user_id");

DROP TABLE IF EXISTS "django_admin_log";
DROP SEQUENCE IF EXISTS django_admin_log_id_seq;
CREATE SEQUENCE django_admin_log_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;

CREATE TABLE "public"."django_admin_log" (
    "id" integer DEFAULT nextval('django_admin_log_id_seq') NOT NULL,
    "action_time" timestamptz NOT NULL,
    "object_id" text,
    "object_repr" character varying (200) NOT NULL,
    "action_flag" smallint NOT NULL,
    "change_message" text NOT NULL,
    "content_type_id" integer,
    "user_id" integer NOT NULL,
CONSTRAINT "django_admin_log_pkey" PRIMARY KEY ("id"),
CONSTRAINT "django_admin_log_content_type_id_c4bce8eb_fk_django_co"
FOREIGN KEY (content_type_id) REFERENCES django_content_type(id)
DEFERRABLE INITIALLY DEFERRED DEFERRABLE,
CONSTRAINT "django_admin_log_user_id_c564eba6_fk_auth_user_id" FOREIGN
KEY (user_id) REFERENCES auth_user(id) DEFERRABLE INITIALLY DEFERRED
DEFERRABLE

```

```

) WITH (oids = false);
    CREATE INDEX "django_admin_log_content_type_id_c4bce8eb" ON
"public"."django_admin_log" USING btree ("content_type_id");
    CREATE INDEX "django_admin_log_user_id_c564eba6" ON
"public"."django_admin_log" USING btree ("user_id");
    DROP TABLE IF EXISTS "django_content_type";
DROP SEQUENCE IF EXISTS django_content_type_id_seq;
CREATE SEQUENCE django_content_type_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;
    CREATE TABLE "public"."django_content_type" (
    "id" integer DEFAULT nextval('django_content_type_id_seq') NOT NULL,
    "app_label" character varying (100) NOT NULL,
    "model" character varying (100) NOT NULL,
    CONSTRAINT "django_content_type_app_label_model_76bd3d3b_uniq" UNIQUE
("app_label", "model"),
    CONSTRAINT "django_content_type_pkey" PRIMARY KEY ("id")
) WITH (oids = false);
    DROP TABLE IF EXISTS "django_migrations";
DROP SEQUENCE IF EXISTS django_migrations_id_seq;
CREATE SEQUENCE django_migrations_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 2147483647 START 1 CACHE 1;
    CREATE TABLE "public"."django_migrations" (
    "id" integer DEFAULT nextval('django_migrations_id_seq') NOT NULL,
    "app" character varying (255) NOT NULL,
    "name" character varying (255) NOT NULL,
    "applied" timestamptz NOT NULL,
    CONSTRAINT "django_migrations_pkey" PRIMARY KEY ("id")
) WITH (oids = false);
    DROP TABLE IF EXISTS "django_session";
CREATE TABLE "public"."django_session" (
    "session_key" character varying (40) NOT NULL,

```



```

"session_data" text NOT NULL,
"expire_date" timestamptz NOT NULL,
CONSTRAINT "django_session_pkey" PRIMARY KEY ("session_key")
) WITH (oids = false);
        CREATE      INDEX      "django_session_expire_date_a5c62663"      ON
"public"."django_session" USING btree ("expire_date");
        CREATE      INDEX      "django_session_session_key_c0390e0f_like"      ON
"public"."django_session" USING btree ("session_key");
        DROP TABLE IF EXISTS "registros_beneficiario";
DROP SEQUENCE IF EXISTS registros_beneficiario_id_seq;
CREATE SEQUENCE registros_beneficiario_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1 CACHE 1;
        CREATE TABLE "public"."registros_beneficiario" (
        "id" bigint DEFAULT nextval('registros_beneficiario_id_seq') NOT NULL,
        "nombres" character varying (50) NOT NULL,
        "apellido_paterno" character varying (20),
        "apellido_materno" character varying (20),
        "direccion" character varying (50) NOT NULL,
        "ci" character varying (20) NOT NULL,
        "fecha_nac" date NOT NULL,
        "celular" integer NOT NULL,
        "genero" character varying (20) NOT NULL,
        "estado_civil" character varying (20) NOT NULL,
        "created" timestamptz NOT NULL,
        "updated" timestamptz NOT NULL,
        CONSTRAINT "registros_beneficiario_pkey" PRIMARY KEY ("id")
) WITH (oids = false);
        DROP TABLE IF EXISTS "registros_entrega";
DROP SEQUENCE IF EXISTS registros_entrega_id_seq;
CREATE SEQUENCE registros_entrega_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1 CACHE 1;

```

```

CREATE TABLE "public"."registros_entrega" (
  "id" bigint DEFAULT nextval('registros_entrega_id_seq') NOT NULL,
  "estado" boolean NOT NULL,
  "created" timestamptz NOT NULL,
  "updated" timestamptz NOT NULL,
  "beneficiario_id_id" bigint,
  "areaentrega_id_id" bigint,
  "usu_id" integer,
  CONSTRAINT "registros_entrega_pkey" PRIMARY KEY ("id"),
  CONSTRAINT "registros_entrega_areaentrega_id_id_175ca5d4_fk_administr"
  FOREIGN KEY (areaentrega_id_id) REFERENCES administraciones_areaentrega(id)
  DEFERRABLE INITIALLY DEFERRED DEFERRABLE,
  CONSTRAINT "registros_entrega_beneficiario_id_id_9b38ab07_fk_registros"
  FOREIGN KEY (beneficiario_id_id) REFERENCES registros_beneficiario(id)
  DEFERRABLE INITIALLY DEFERRED DEFERRABLE,
  CONSTRAINT "registros_entrega_usu_id_a243e199_fk_auth_user_id" FOREIGN
  KEY (usu_id) REFERENCES auth_user(id) DEFERRABLE INITIALLY DEFERRED
  DEFERRABLE
) WITH (oids = false);

CREATE INDEX "registros_entrega_areaentrega_id_id_175ca5d4" ON
"public"."registros_entrega" USING btree ("areaentrega_id_id");
CREATE INDEX "registros_entrega_beneficiario_id_id_9b38ab07" ON
"public"."registros_entrega" USING btree ("beneficiario_id_id");
CREATE INDEX "registros_entrega_usu_id_a243e199" ON
"public"."registros_entrega" USING btree ("usu_id");

DROP TABLE IF EXISTS "registros_entregadetallada";
DROP SEQUENCE IF EXISTS registros_entregadetallada_id_seq;
CREATE SEQUENCE registros_entregadetallada_id_seq INCREMENT 1
MINVALUE 1 MAXVALUE 9223372036854775807 START 1 CACHE 1;
CREATE TABLE "public"."registros_entregadetallada" (
  "id" bigint DEFAULT nextval('registros_entregadetallada_id_seq') NOT NULL,

```

```

"nombre_entrega" character varying (40) NOT NULL,
"cantidad" integer NOT NULL,
"estado" boolean NOT NULL,
"created" timestamptz NOT NULL,
"updated" timestamptz NOT NULL,
"entrega_id_id" bigint NOT NULL,
"producto_id_id" bigint NOT NULL,
CONSTRAINT "registros_entregadetallada_pkey" PRIMARY KEY ("id"),
CONSTRAINT "registros_entregadet_entrega_id_id_cb7fc585_fk_registros"
FOREIGN KEY (entrega_id_id) REFERENCES registros_entrega(id) DEFERRABLE
INITIALLY DEFERRED DEFERRABLE,
CONSTRAINT "registros_entregadet_producto_id_id_56200d1b_fk_administr"
FOREIGN KEY (producto_id_id) REFERENCES administraciones_producto(id)
DEFERRABLE INITIALLY DEFERRED DEFERRABLE
) WITH (oids = false);

CREATE INDEX "registros_entregadetallada_entrega_id_id_cb7fc585" ON
"public"."registros_entregadetallada" USING btree ("entrega_id_id");

CREATE INDEX "registros_entregadetallada_producto_id_id_56200d1b" ON
"public"."registros_entregadetallada" USING btree ("producto_id_id");

DROP TABLE IF EXISTS "registros_planificacion";
DROP SEQUENCE IF EXISTS registros_planificacion_id_seq;
CREATE SEQUENCE registros_planificacion_id_seq INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1 CACHE 1;

CREATE TABLE "public"."registros_planificacion" (
"id" bigint DEFAULT nextval('registros_planificacion_id_seq') NOT NULL,
"fecha_presentacion" date NOT NULL,
"fecha_postergacion" date,
"asistio" timestamptz,
"observacion" character varying (50),
"dias" character varying (100) NOT NULL,
"created" timestamptz NOT NULL,

```

```

"updated" timestamptz NOT NULL,
"entrega_id_id" bigint NOT NULL,
CONSTRAINT "registros_planificacion_pkey" PRIMARY KEY ("id"),
CONSTRAINT "registros_planificac_entrega_id_id_3a5a7aa9_fk_registros"
FOREIGN KEY (entrega_id_id) REFERENCES registros_entrega(id) DEFERRABLE
INITIALLY DEFERRED DEFERRABLE
) WITH (oids = false);
CREATE INDEX "registros_planificacion_entrega_id_id_3a5a7aa9" ON
"public"."registros_planificacion" USING btree ("entrega_id_id");

```

## 2.9.2. Diccionario de Datos

**Tabla 39**

*User.*

Entidad: auth_user	
Atributo llave reserva	
Nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

Columna nombre	Llave Primaria	Dato Tipo	Not NULL	Descripción
id	Yes	INTEGER	Yes	id del usuario
username	No	CHARACTER VARYING(50)	Yes	Nombre de login del usuario
first_name	No	CHARACTER VARYING(100)	Yes	Nombres del usuario
last_name	No	CHARACTER VARYING(100)	Yes	Apellidos del usuario
email	No	CHARACTER VARYING(50)	Yes	Correo del usuario
is_superuser	No	BOOLEAN	No	Superusuario del usuario activo=true,

				inactivo=false
is_staff	No	BOOLEAN	No	Si está conectado el usuario activo=true, inactivo=false
Is_active	No	BOOLEAN	No	estado del usuario activo=true, inactivo=false
Last_login	No	DATETIME	No	Ultima conexión del usuario
Date_joined	No	DATETIME	No	Fecha de conexión del usuario

**Tabla 40**  
**Registro Beneficiarios.**

<b>Entidad: registros_beneficiario</b>	
Atributo llave	
reserva nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

<b>Columna nombre</b>	<b>Llave primaria</b>	<b>Dato tipo</b>	<b>Not NULL</b>	<b>Descripción</b>
id	Yes	INTEGER	Yes	id del beneficiario
nombres	No	CHARACTER VARYING(50)	Yes	Nombres del beneficiario
Apellido_pat	No	CHARACTER VARYING(50)	No	Apellido paterno del beneficiario
Apellido_mat	No	CHARACTER VARYING(50)	No	Apellido materno del beneficiario
dirección	No	CHARACTER VARYING(100)	Yes	Dirección del beneficiario
ci	No	CHARACTER VARYING(20)	Yes	Documento de identidad del beneficiario

Fecha_nac	No	DATE	Yes	Fecha de nacimiento del beneficiario
género	No	CHARACTER VARYING(20)	Yes	Género del beneficiario
Estado civil	No	CHARACTER VARYING(20)	Yes	Estado civil del beneficiario
celular	No	INTEGER	Yes	Número de celular del beneficiario
Created	No	DATETIME	Yes	Fecha y hora de creación
updated	No	DATETIME	Yes	Fecha y hora de modificación

**Tabla 41**

**Registros de Planificación.**

<b>Entidad: registros_planificacion</b>	
Atributo llave reserva nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

Columna nombre	Llave primaria	Dato tipo	Not NULL	Descripción
id	Yes	INTEGER	Yes	id de planificación
fecha_presentación	No	DATE	Yes	Fecha de presentación de planificación
fecha_postergación	No	DATE	Yes	Fecha de postergación de planificación
asistio	No	DATETIME	Yes	Fecha de asistencia de planificación
observación	No	CHARACTER VARYING(150)	Yes	Descripción de fecha de postergación de planificación
dias	No	INTEGER	Yes	Número de días de presentarse de planificación

Created	No	DATETIME	Yes	Fecha y hora de creación de planificación
updated	No	DATETIME	Yes	Fecha y hora de modificación de ficha de planificación
Entrega_id_id	No	INTEGER	Yes	Id de ficha de entrega

Restricciones:

Nombre	Tipo	Level	Constraint
	LLave primaria	Table constraint	Llave primaria (Id)
	Not Null	Columna Constraint	Not Null (id)
	Foreign Key	Table constraint	Foreing Key (registro_id) References ficha_registro(id)
	Not Null	Columna constraint	Not Null(ficha_registro_id)

**Tabla 42**

**Registro de Entrega.**

Entidad: registros_entrega	
Atributo llave reserva	
nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

Columna nombre	Llave primaria	Dato Tipo	Not NULL	Descripción
id	Yes	INTEGER	Yes	id de registro
Estado	No	Booleano	Yes	Estado de la entrega
Created	No	DATETIME	Yes	Fecha y hora de creación de la entrega
updated	No	DATETIME	Yes	Fecha y hora de modificación de la entrega
Beneficiario_id_id	No	INTEGER	Yes	Id del beneficiario
Areaentrega_id	No	INTEGER	Yes	Id del área de entrega

d_id				
usu_id	No	INTEGER	Yes	Id del usuario

Restricciones:

Nombre	Tipo	Level	Constraint
	Llave Primaria	Table constraint	Llave primaria (Id)
	Not Null	Columna Constraint	Not Null (id)
	Foreign Key	Table constraint	Foreing Key(beneficiario_id) References registro beneficiario(id)

**Tabla 43**

**Registros Entrega Detallada**

Entidad: registros_entregadetallada	
Atributo llave reserva	
nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

Columna Nombre	Llave primaria	Dato Tipo	Not NULL	Descripción
id	Yes	INTEGER	Yes	id detalle
Nombre	No	CHARACTER VARYING(50)	Yes	Nombre de la entrega
cantidad	No	CHARACTER VARYING(20)	Yes	Cantidad de productos
Estado	No	Booleano	Yes	Estado de la entrega
Created	No	DATETIME	Yes	Fecha y hora de creación de entrega
updated	No	DATETIME	Yes	Fecha y hora de modificación de ficha de entrega
Entega_id_id	No	INTEGER	Yes	Id de la entrega
Producto_id_id	No	INTEGER	Yes	Id del producto



Restricciones:

Nombre	Tipo	Level	Constraint
	Llave primaria	Table constraint	Llave primaria (Id)
	Not Null	Columna constraint	Not Null (id)
	Foreign Key	Table constraint	Foreing Key (Entrega_id_id) References Registro de entrega(id)
	Foreign Key	Table constraint	Foreing Key (Producto_id_id) REFERENCES administraciones producto(id)

**Tabla 44**

**Administraciones Producto.**

<b>Entidad: administraciones_producto</b>	
Atributo llave reserva	
nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

Columna nombre	Llave primaria	Dato Tipo	Not NULL	Descripción
id	Yes	INTEGER	Yes	id producto
Producto	No	CHARACTER VARYING(50)	Yes	Nombre del producto
precio	No	INTEGER	Yes	Precio de los productos
Stock	No	INTEGER	Yes	Cantidad disponible
Created	No	DATETIME	Yes	Fecha y hora de creación
updated	No	DATETIME	Yes	Fecha y hora de modificación
categoria_id_id	No	INTEGER	Yes	Id de la categoria

Restricciones:

Nombre	Tipo	Level	Constraint
	Llave primaria	Table constraint	Llave primaria (Id)
	Not Null	Columna constraint	Not Null (id)

	Foreign Key	Table constraint	Foreign Key (Entrega_id_id) References Registro de entrega(id)
	Foreign Key	Table constraint	Foreign Key (categoria_id_id) References administraciones administraciones_categoria(id)

**Tabla 45**

**Administraciones Categoría.**

<b>Entidad: Administraciones_categoría</b>	
Atributo llave reserva nombre	
Descripción	
Table options	WITH (OIDS=FALSE)

Atributos:

<b>Columna nombre</b>	<b>Llave primaria</b>	<b>Dato tipo</b>	<b>Not NULL</b>	<b>Descripción</b>
id	Yes	INTEGER	Yes	id categoría
Nombre	No	CHARACTER VARYING(50)	Yes	Nombre de la categoría
Created	No	DATETIME	Yes	Fecha y hora de creación
updated	No	DATETIME	Yes	Fecha y hora de modificación

Restricciones:

<b>Nombre</b>	<b>Tipo</b>	<b>Level</b>	<b>Constraint</b>
	Llave primaria	Table constraint	Llave primaria (Id)
	Not Null	Columna constraint	Not Null (id)

**Tabla 46**

**Administraciones Área Entrega.**

<b>Entidad: Administraciones_areaentrega</b>	
Atributo llave reserva nombre Descripción Table options	WITH (OIDS=FALSE)

Atributos:

<b>Columna nombre</b>	<b>Llave primaria</b>	<b>Dato Tipo</b>	<b>Not NULL</b>	<b>Descripción</b>
id	Yes	INTEGER	Yes	id producto
Oficina	No	CHARACTER VARYING(50)	Yes	Lugar de entrega
Descripción	No	CHARACTER VARYING(50)	Yes	Descripción del lugar
Created	No	DATETIME	Yes	Fecha y hora de creación
updated	No	DATETIME	Yes	Fecha y hora de modificación

Restricciones:

<b>Nombre</b>	<b>Tipo</b>	<b>Level</b>	<b>Constraint</b>
	Llave primaria	Table constraint	Llave primaria (Id)
	Not Null	Columna constraint	Not Null (id)

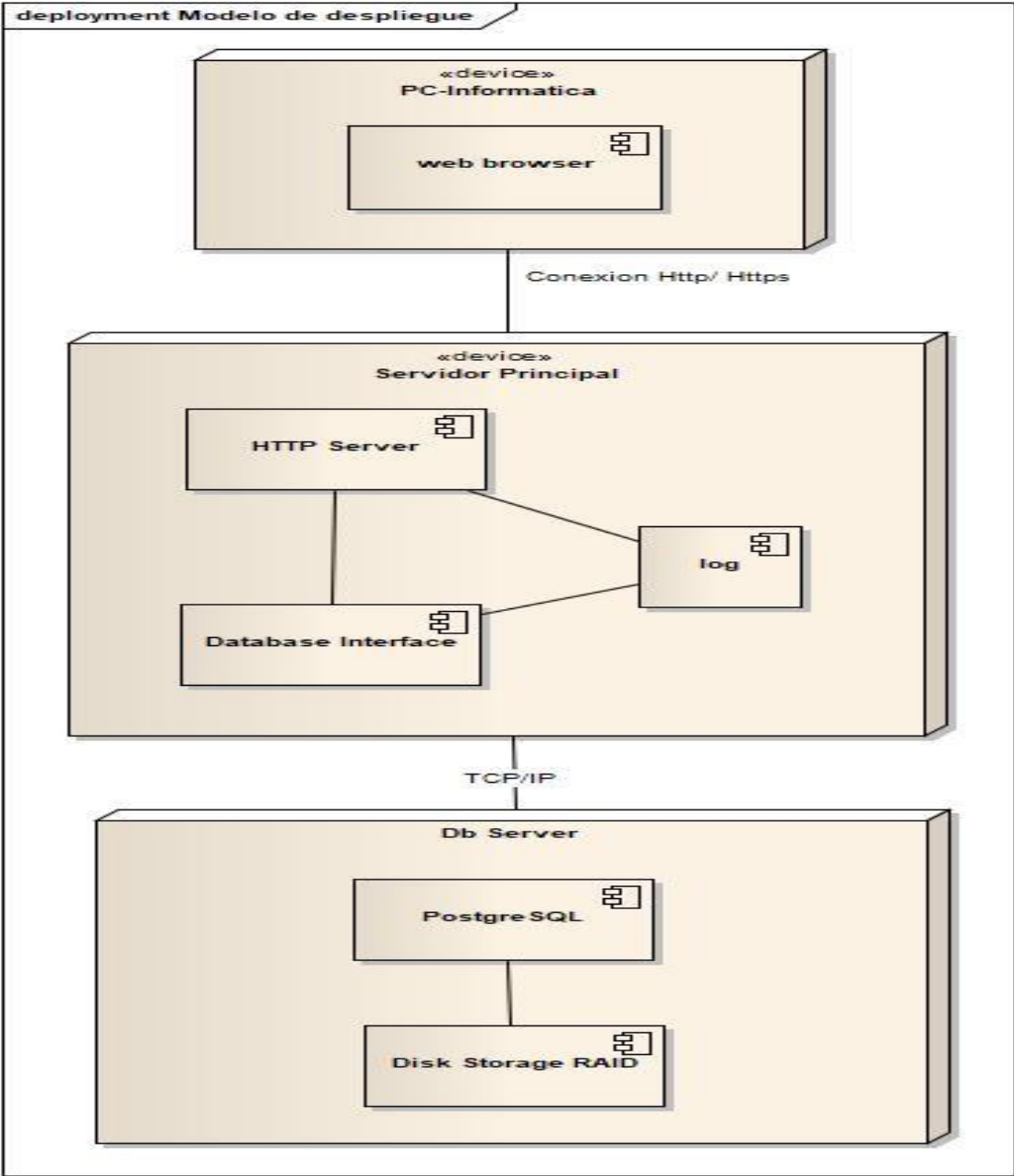
### **2.10. Modelo de Despliegue**

Se trata de visualizar los procesadores/ nodos/dispositivos de hardware de un Sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware.

2.10.1. Diagrama de Despliegue.

Figura 14

Diagrama de Despliegue

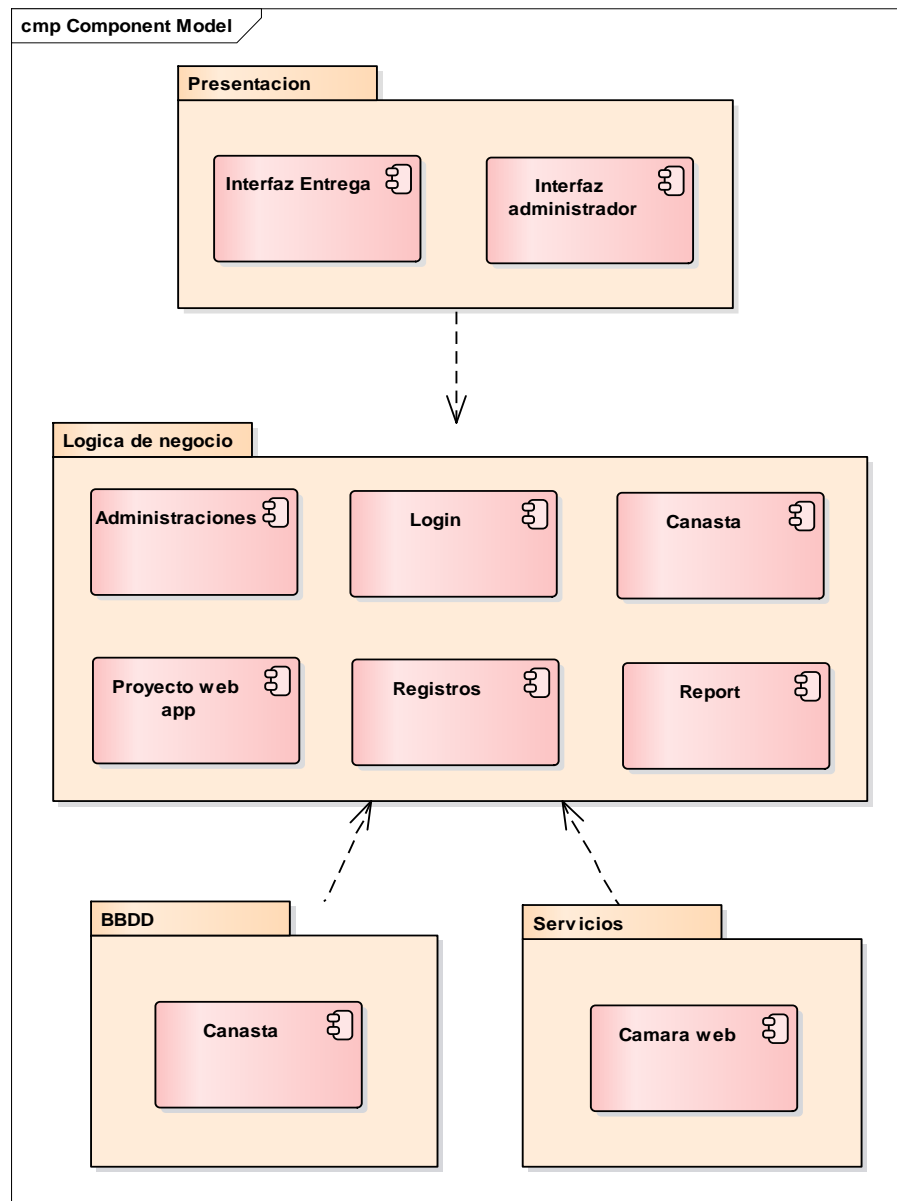


## 2.11. Modelo de Componentes

Se trata de proporcionar una vista de alto nivel de los componentes dentro de un sistema. Los componentes pueden ser un componente de *software*, como una base de datos o una interfaz de usuario; o un componente de *hardware* como un circuito, microchip o dispositivo; o una unidad de negocio como un proveedor, nómina o envío.

*Figura 15*

*Diagrama de Componentes*



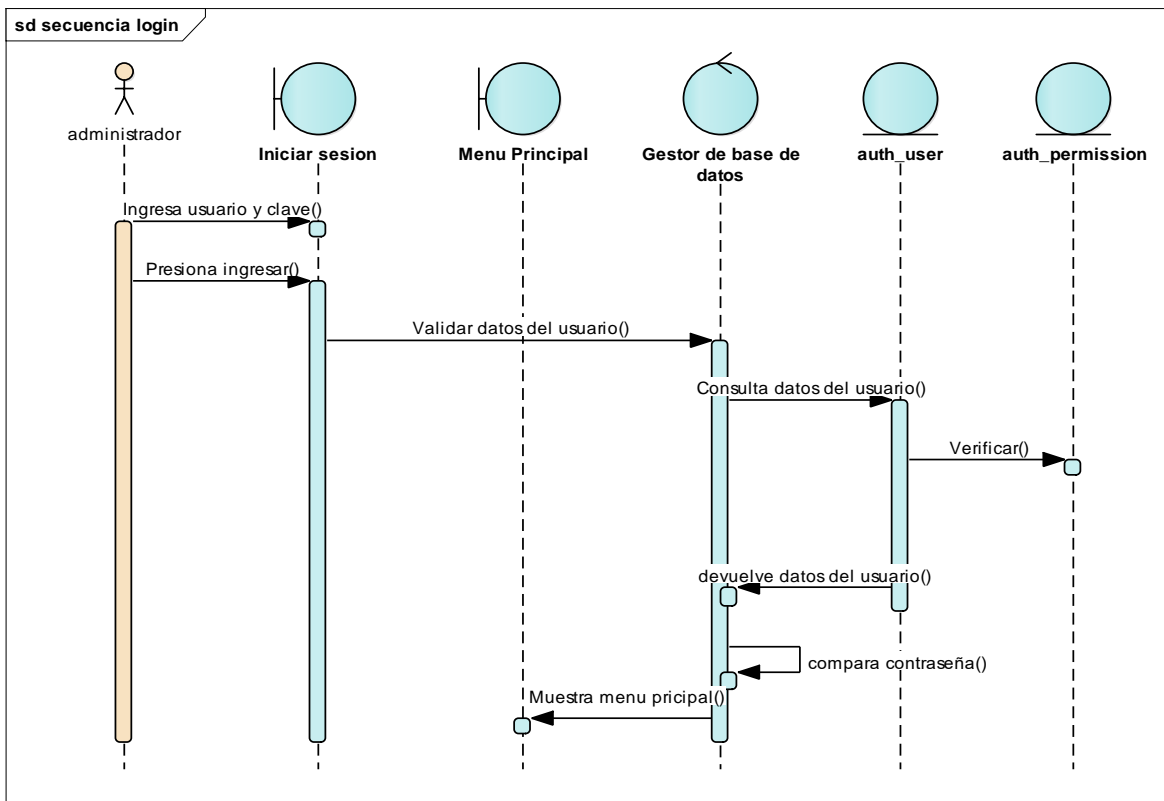
## 2.12. Diagramas de Secuencia.

En un diagrama de secuencias muestra una iteración ordenada según la secuencia temporal de eventos en particular muestra los objetos participantes en la iteración y los mensajes (llamadas a métodos) que intercambian según su secuencia en el tiempo. Frecuentemente, estos diagramas se ubican bajo los casos de uso o componentes en el modelo para ilustrar un escenario, un conjunto de pasos comunes que siguen en respuesta a un evento externo y que generalmente un resultado. El modelo incluye, que inicia la actividad en el sistema, que procesamientos y cambios ocurren internamente y que salidas se generan. Muchas veces las instancias de los objetos se representan usando iconos especialmente estereotipo; existen iconos para objetos de interfaz, controladores, entidades persistentes, etc.

### 2.12.1. Ingresar al Sistema.

*Figura 16*

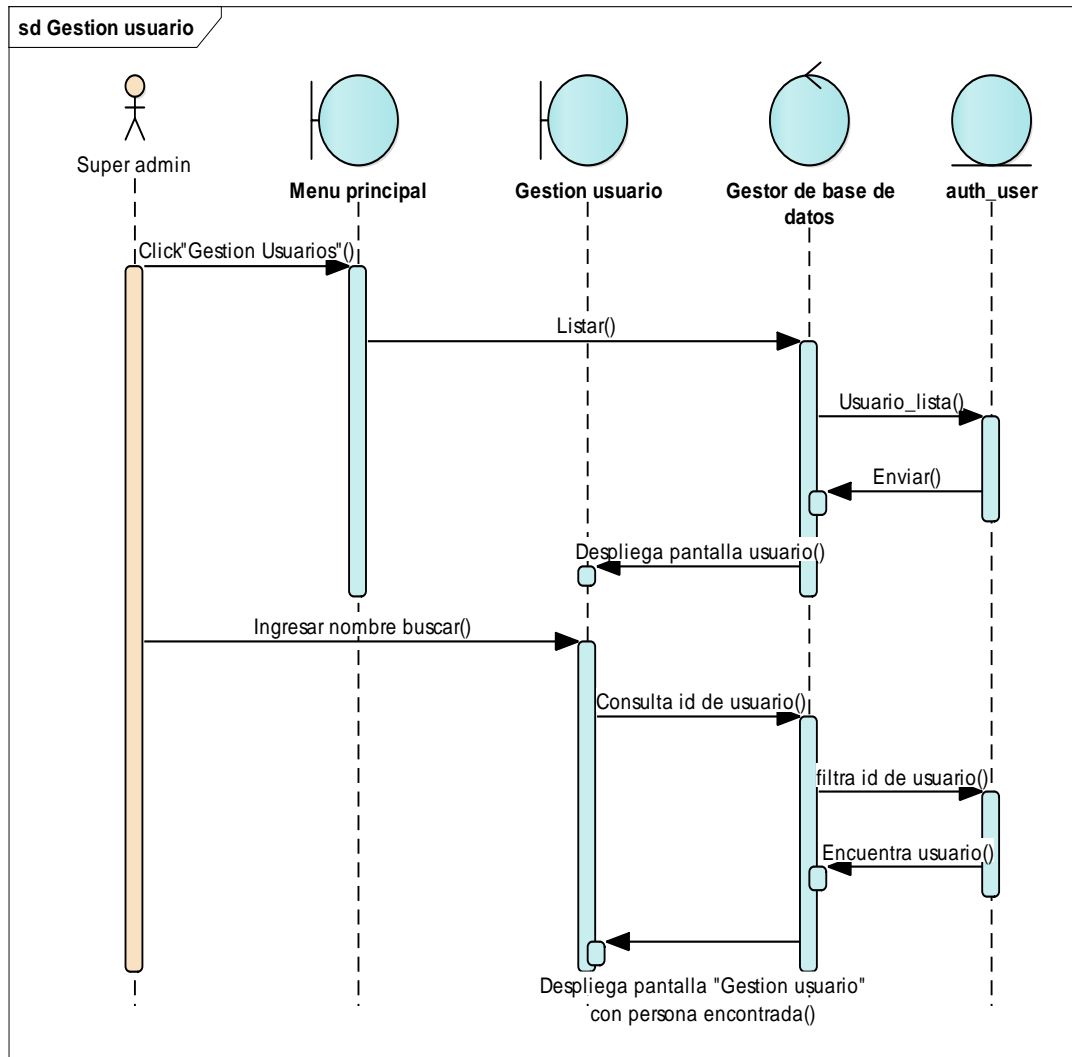
#### *Ingresar al Sistema*



## 2.12.2. Gestión Usuarios.

Figura 17

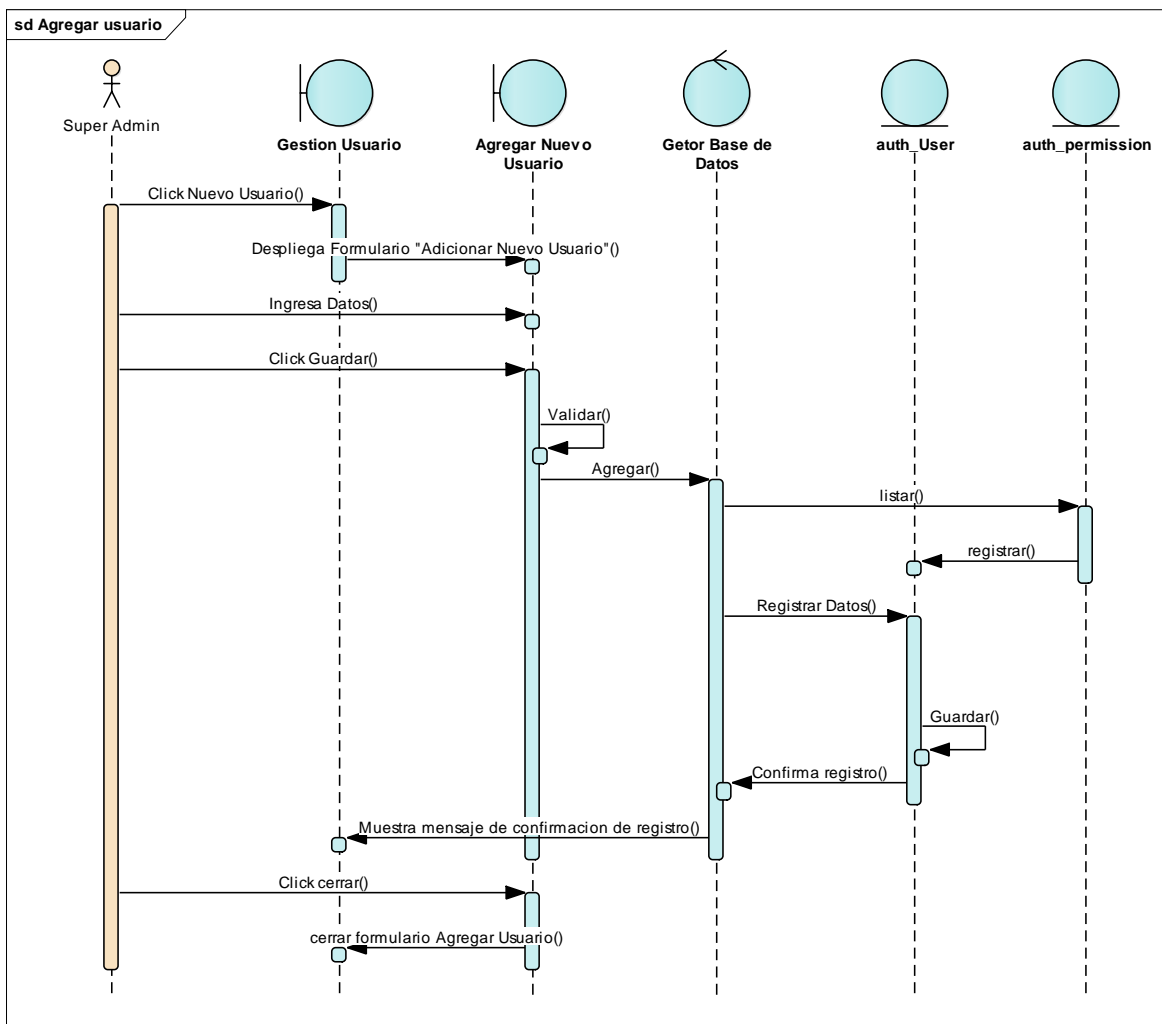
### Gestión Usuarios



### 2.12.3. Agregar Usuarios.

**Figura 18**

#### **Agregar Usuarios**

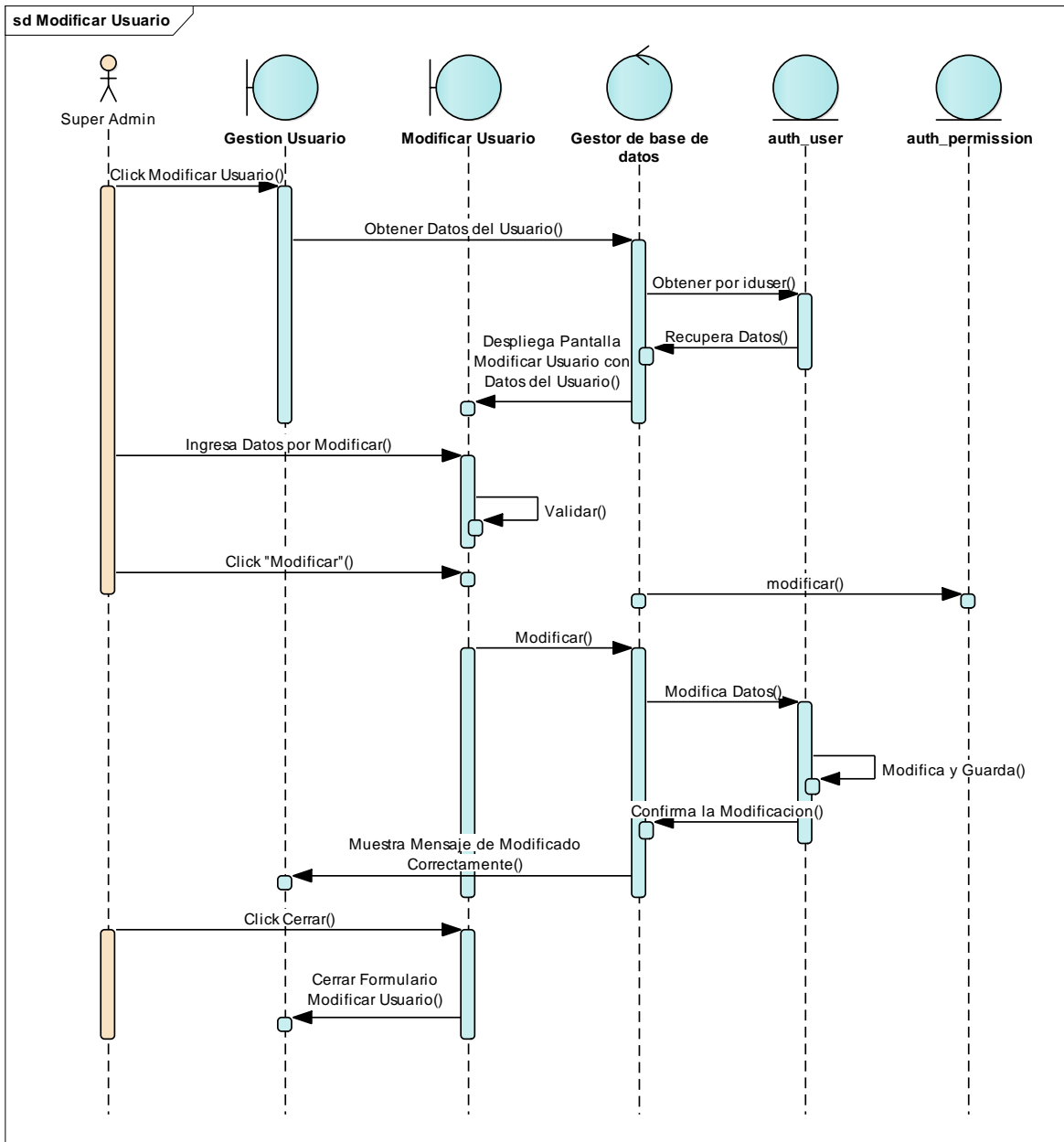




## 2.12.4. Modificar Usuario.

Figura 19

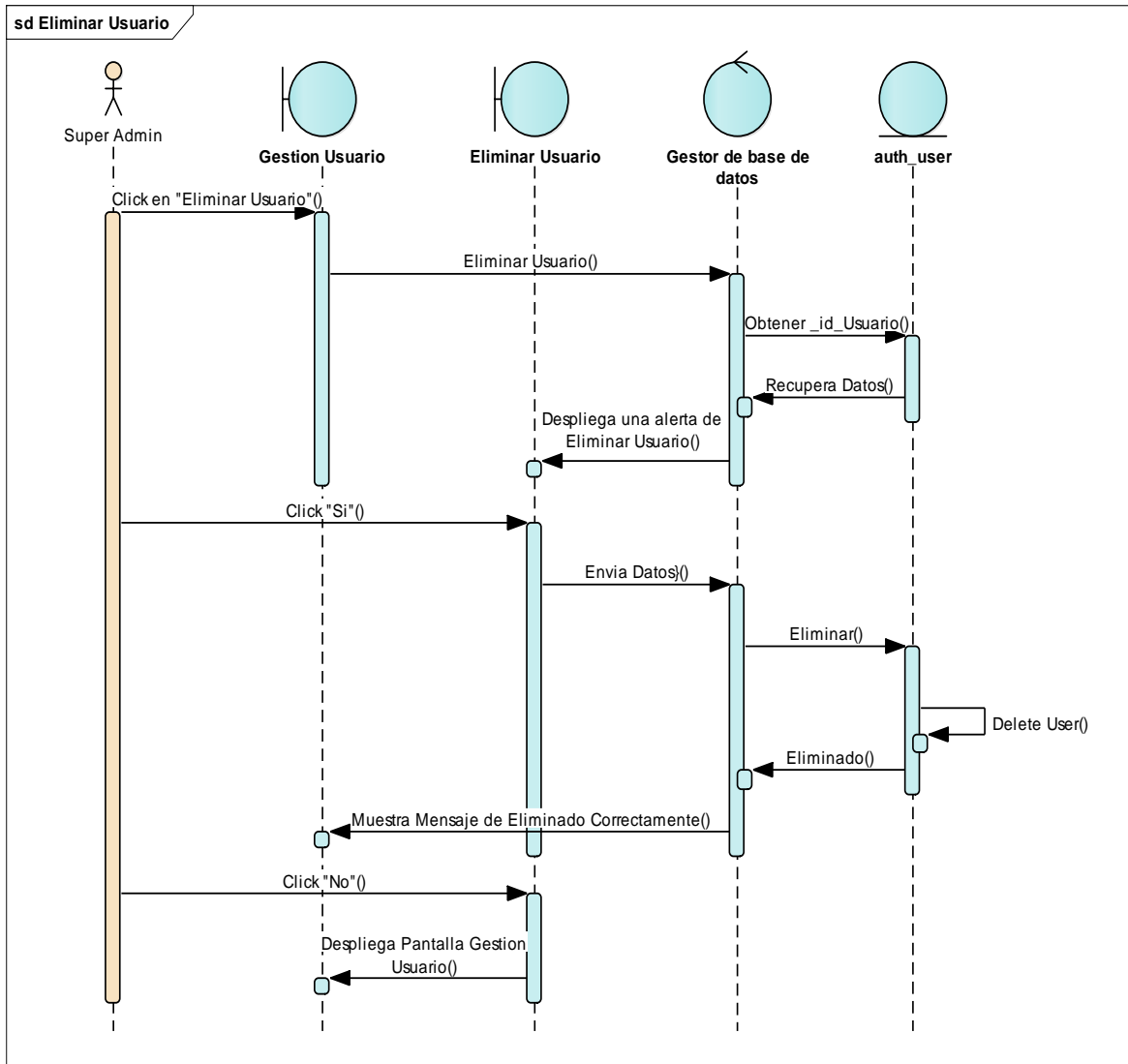
### Modificar Usuario



## 2.12.5. Eliminar Usuario.

Figura 20

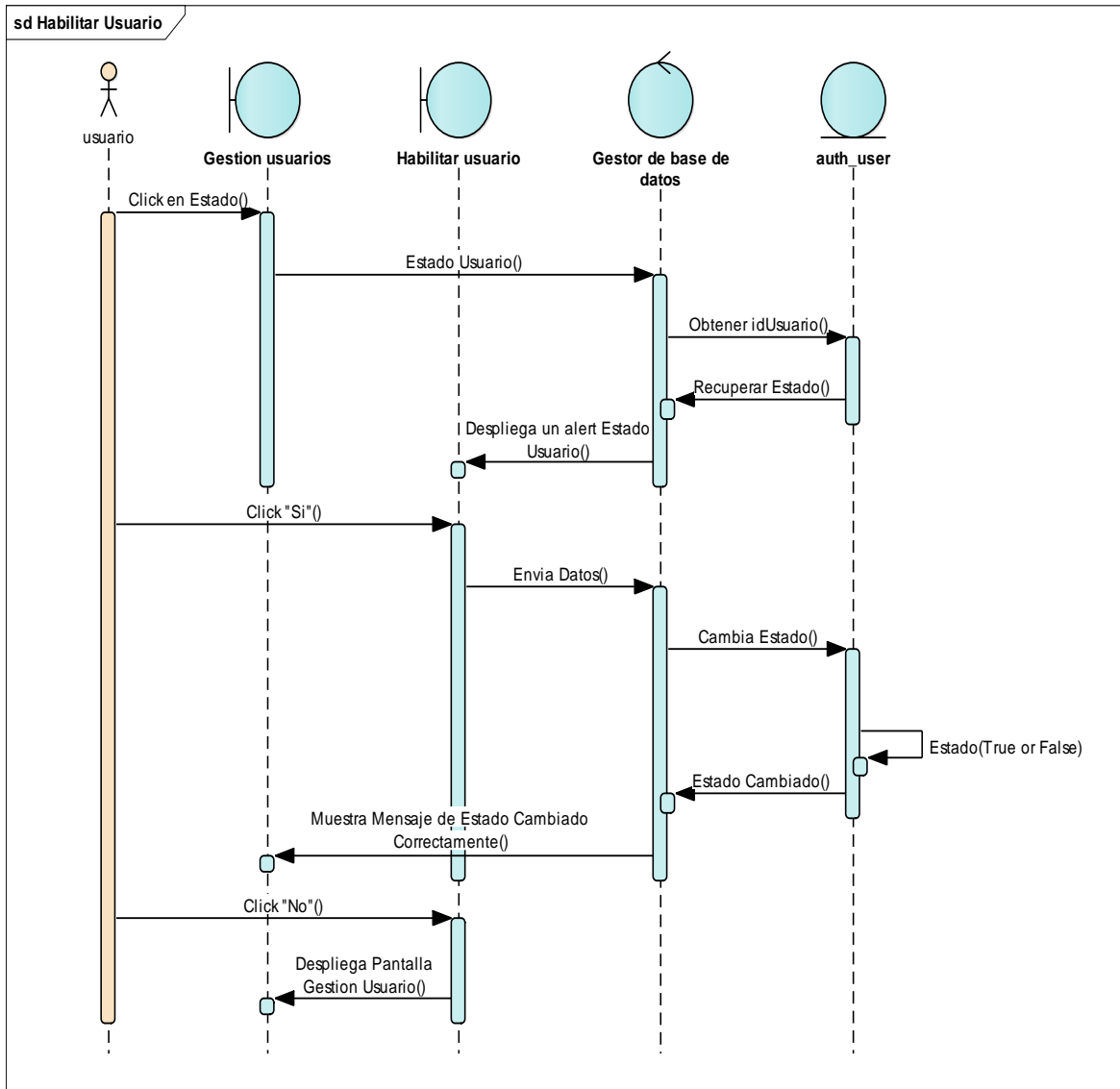
### Eliminar Usuario



## 2.12.6. Habilitar Usuario.

Figura 21

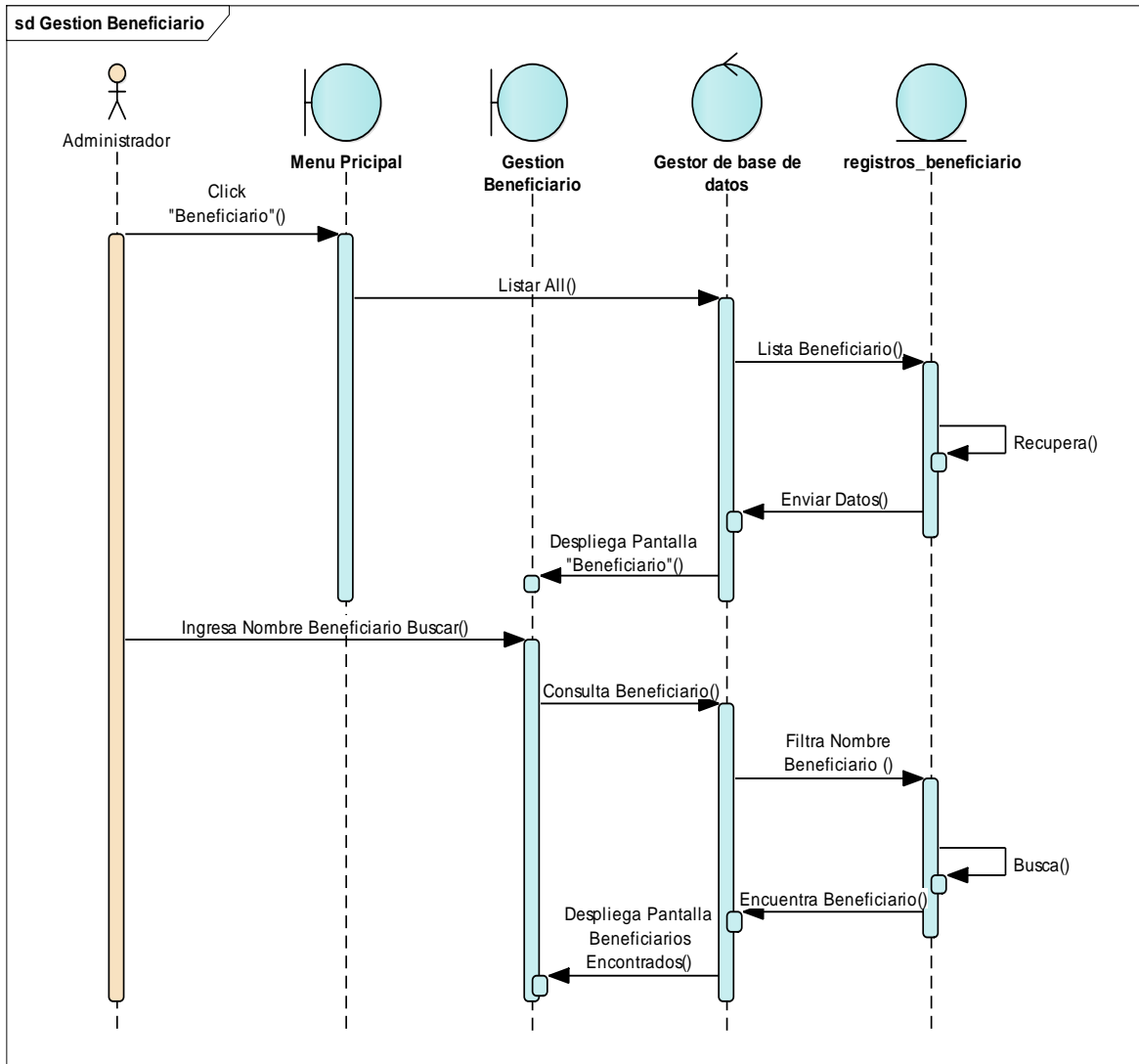
Habilitar Usuario



## 2.12.7. Gestión Beneficiario.

Figura 22

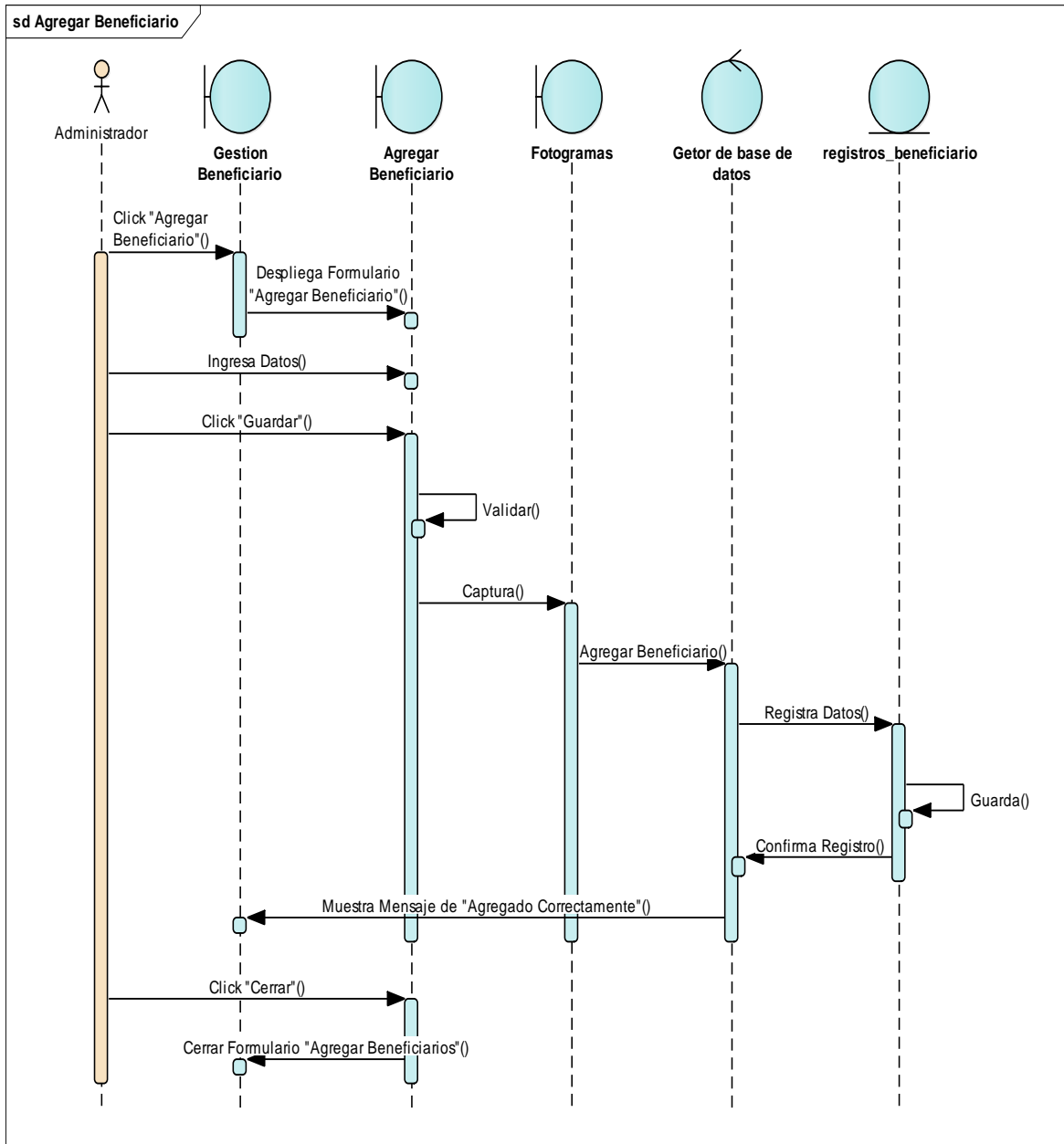
### Gestión Beneficiario



## 2.12.8. Agregar Beneficiario.

Figura 23

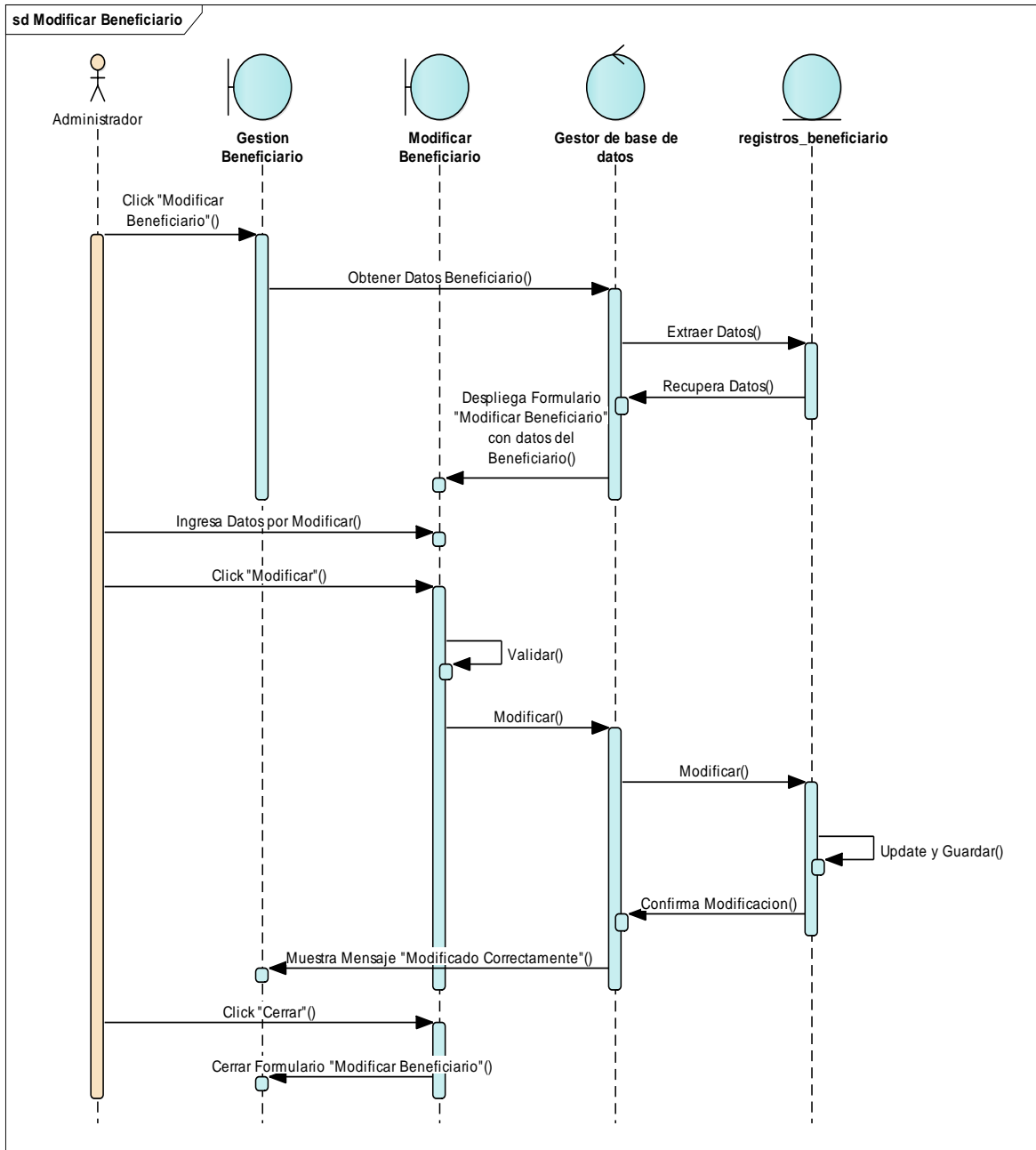
### Agregar Beneficiario



## 2.12.9. Modificar Beneficiario.

Figura 24

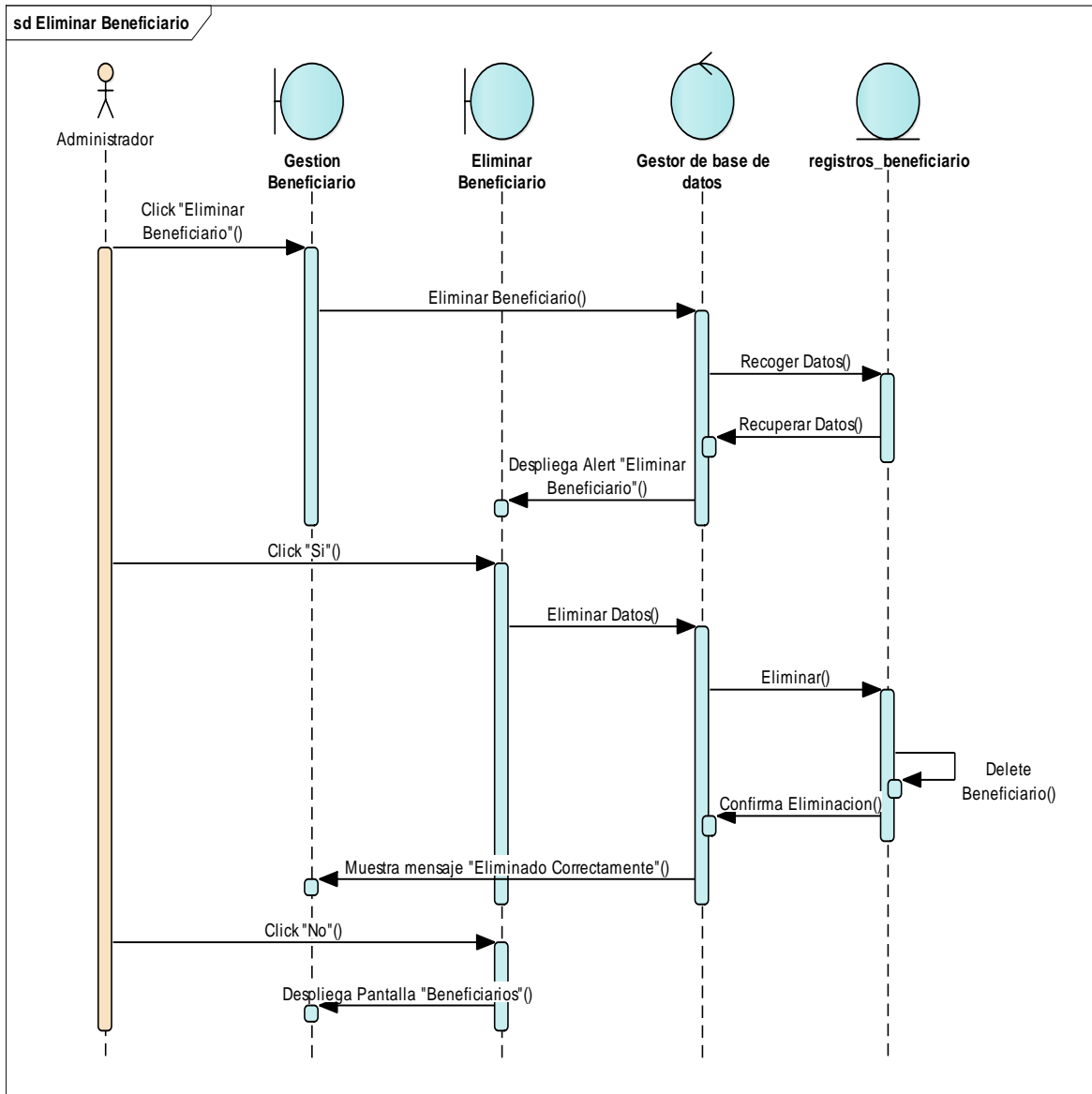
### Modificar Beneficiario



## 2.12.10. Eliminar Beneficiario.

Figura 25

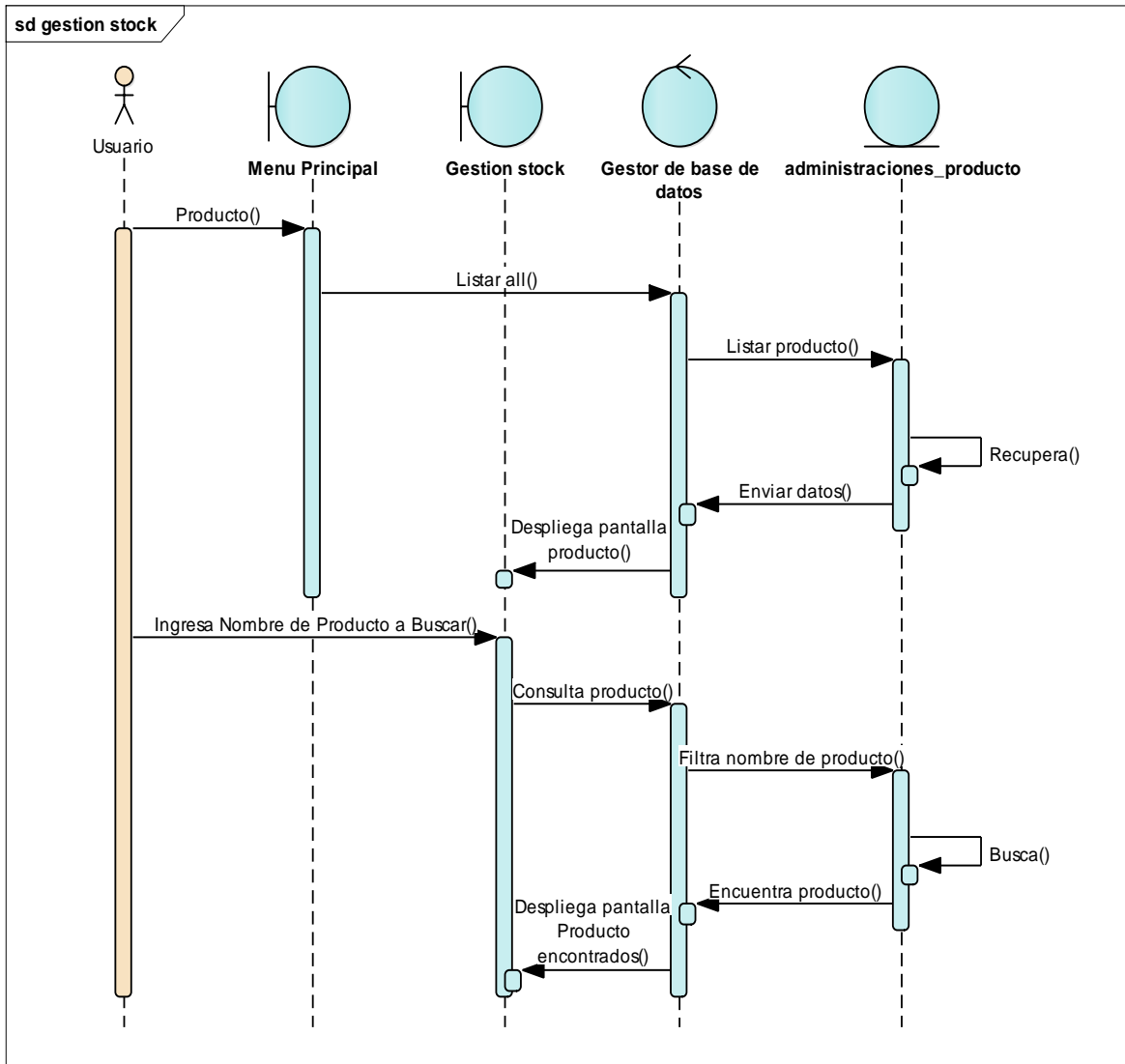
### Eliminar Beneficiario



## 2.12.11. Gestión Stock.

Figura 26

Gestión Stock

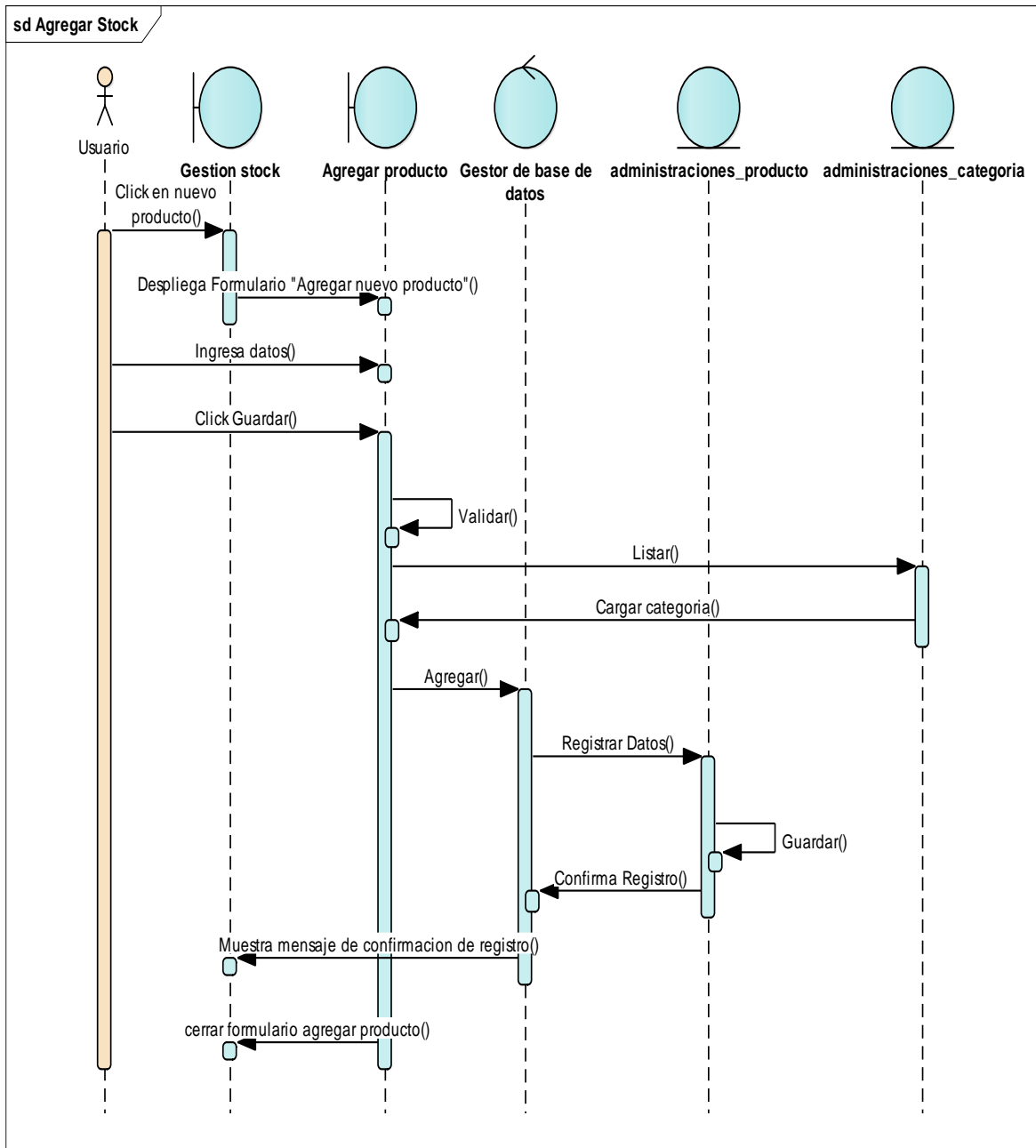




## 2.12.12. Agregar Stock.

Figura 27

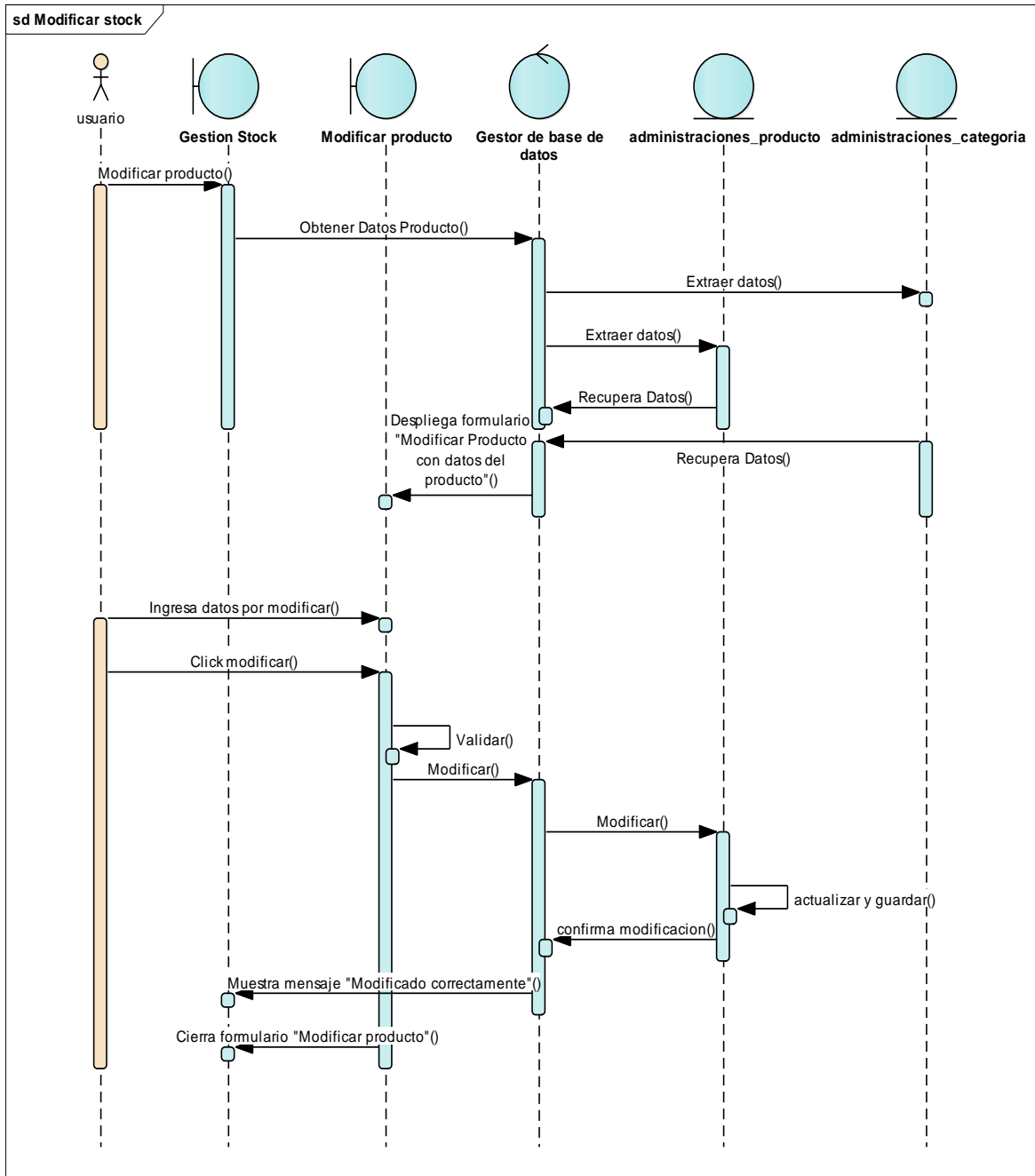
### Agregar Stock



## 2.12.13. Modificar Stock.

Figura 28

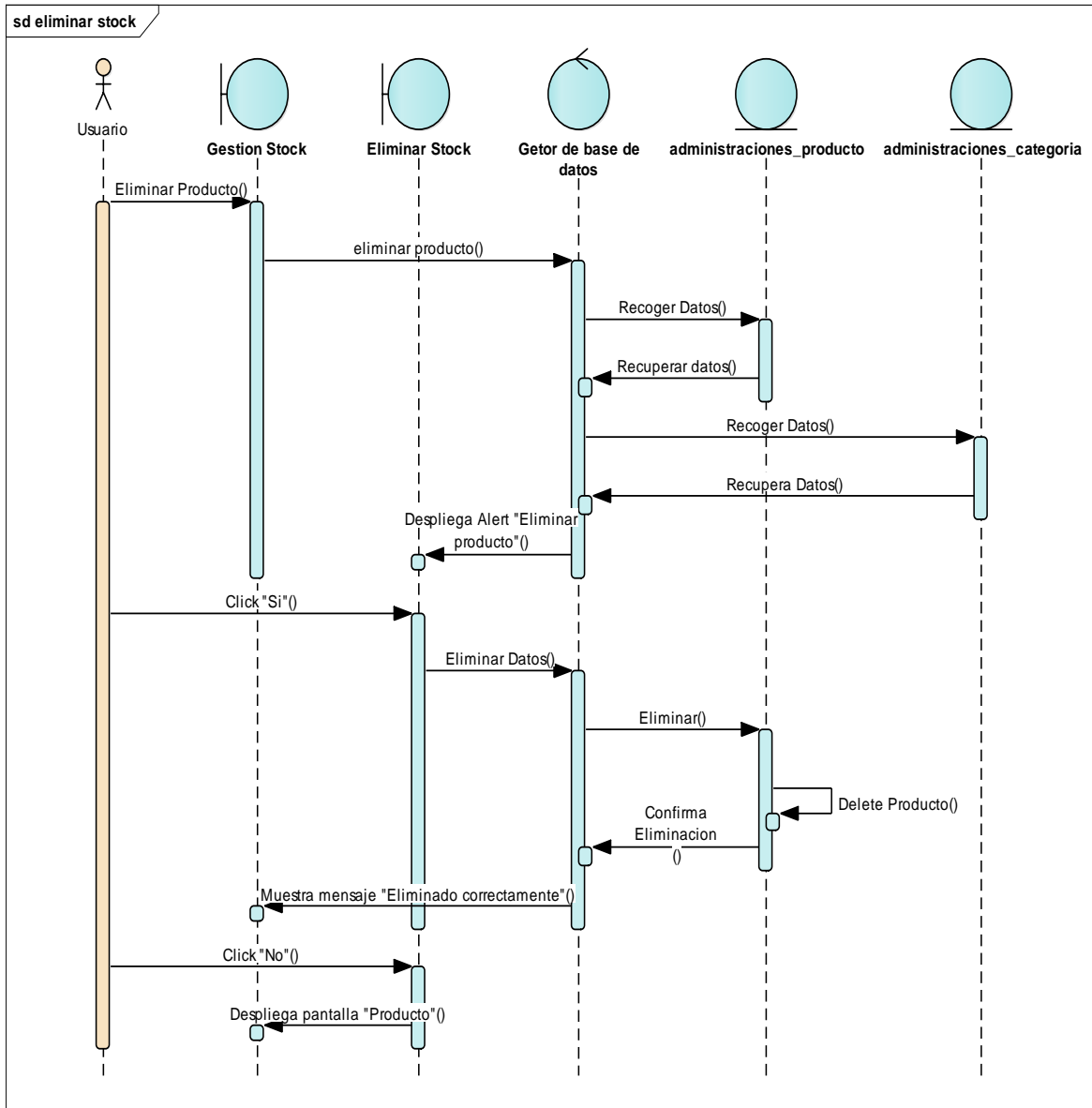
### Modificar Stock



## 2.12.14. Eliminar Stock.

Figura 29

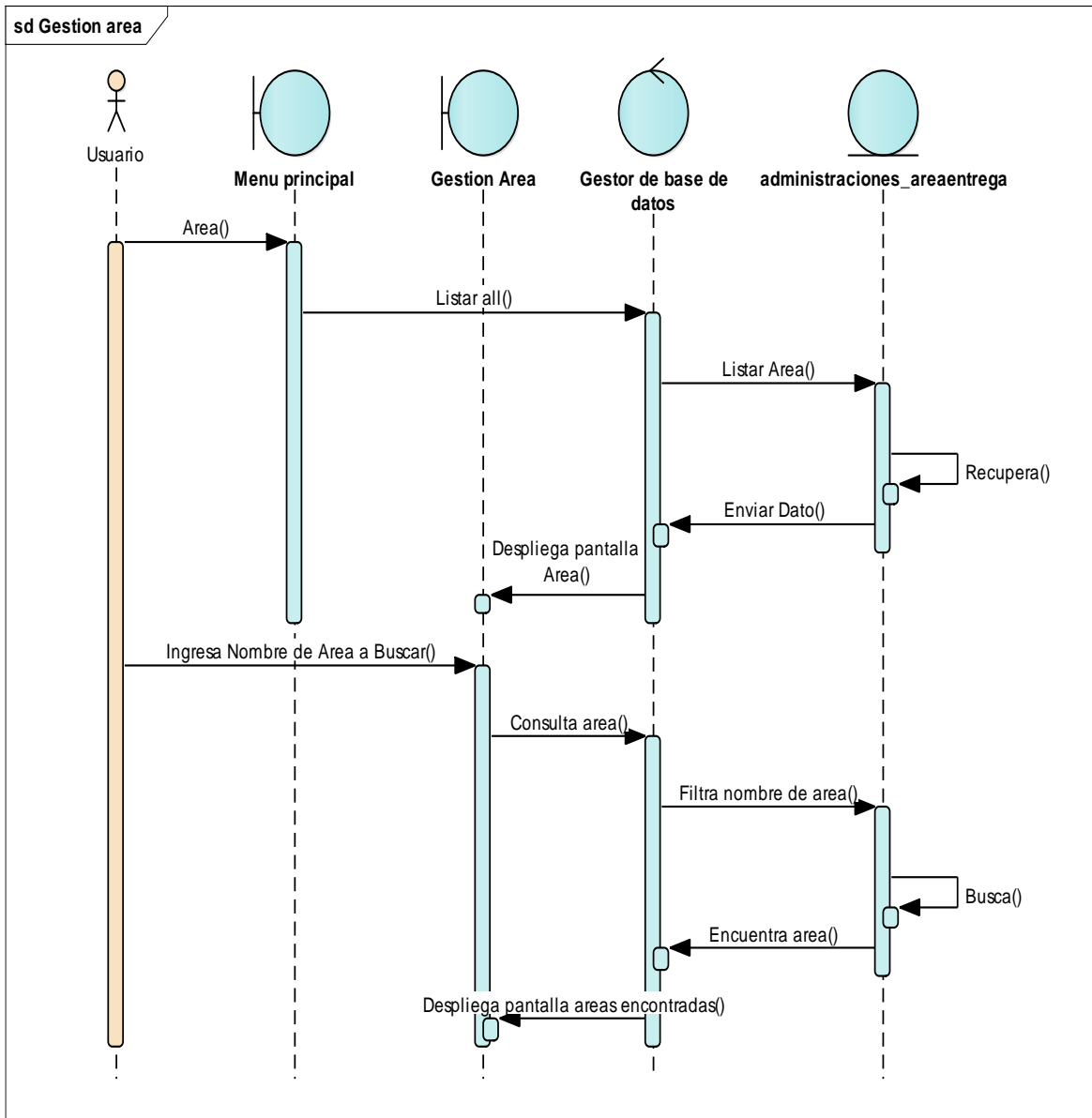
### Eliminar Stock



## 2.12.15. Gestión Área.

Figura 30

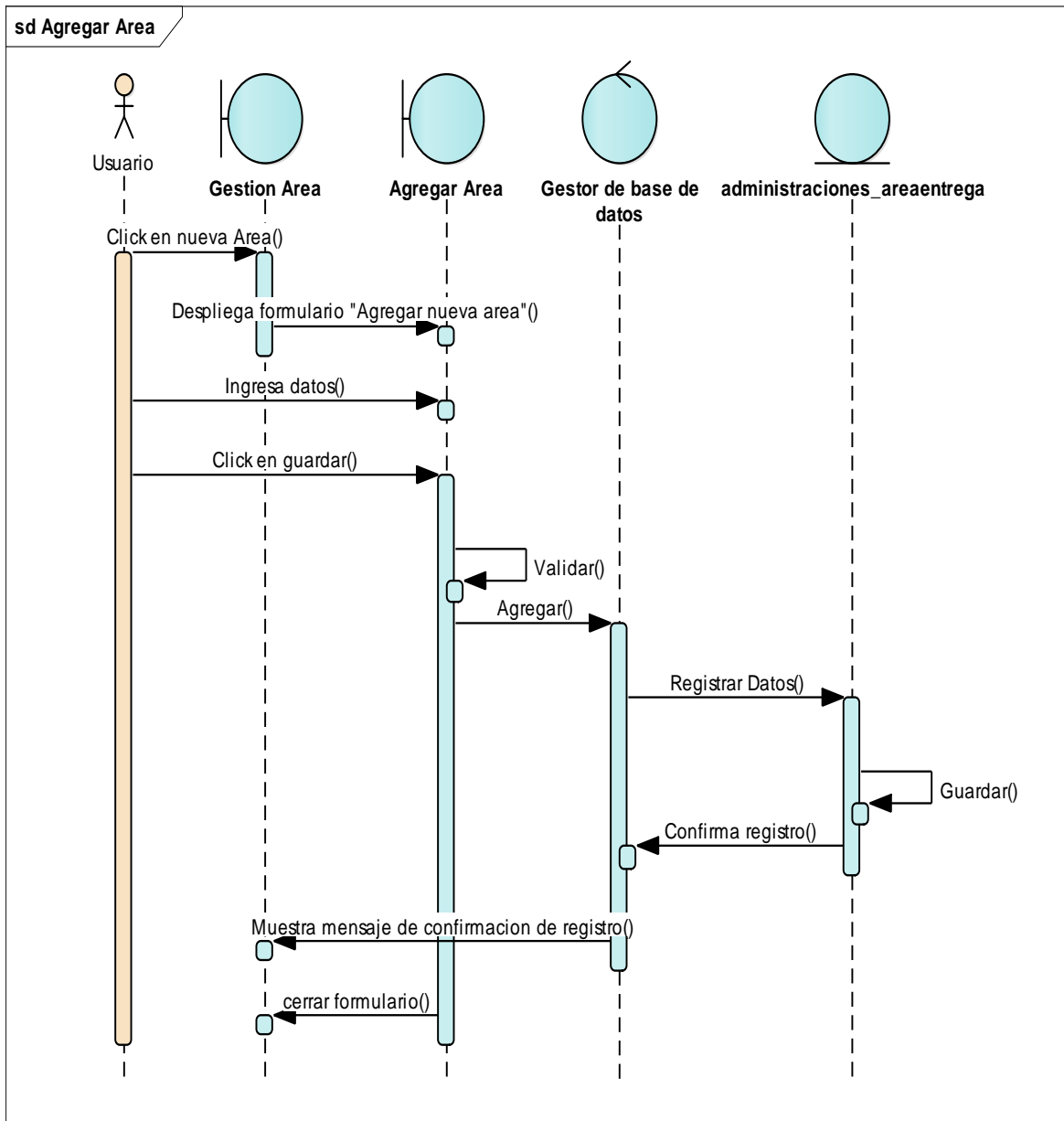
### Gestión Área



## 2.12.16. Agregar Área.

Figura 31

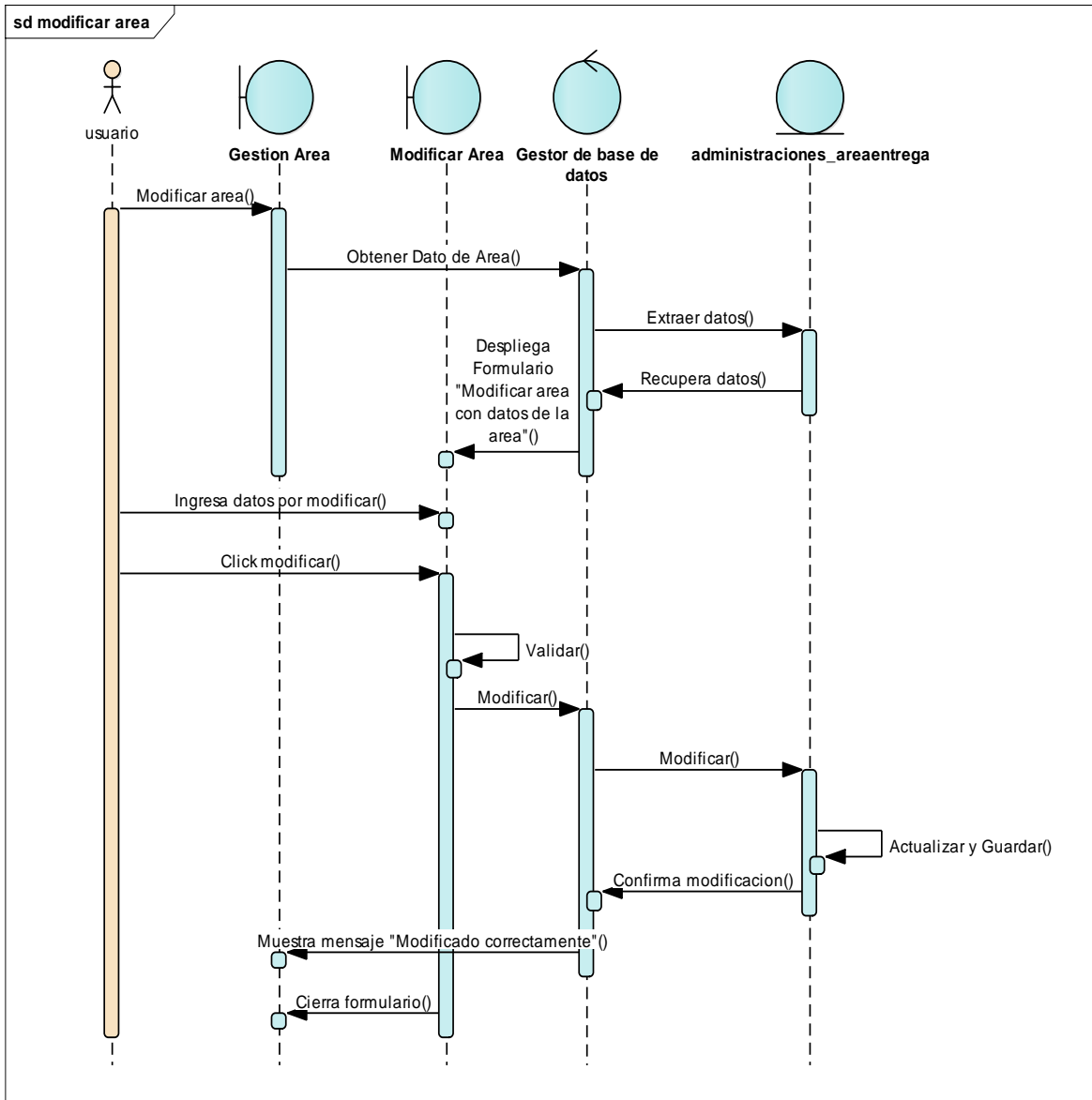
### Agregar Área



## 2.12.17. Modificar Área.

Figura 32

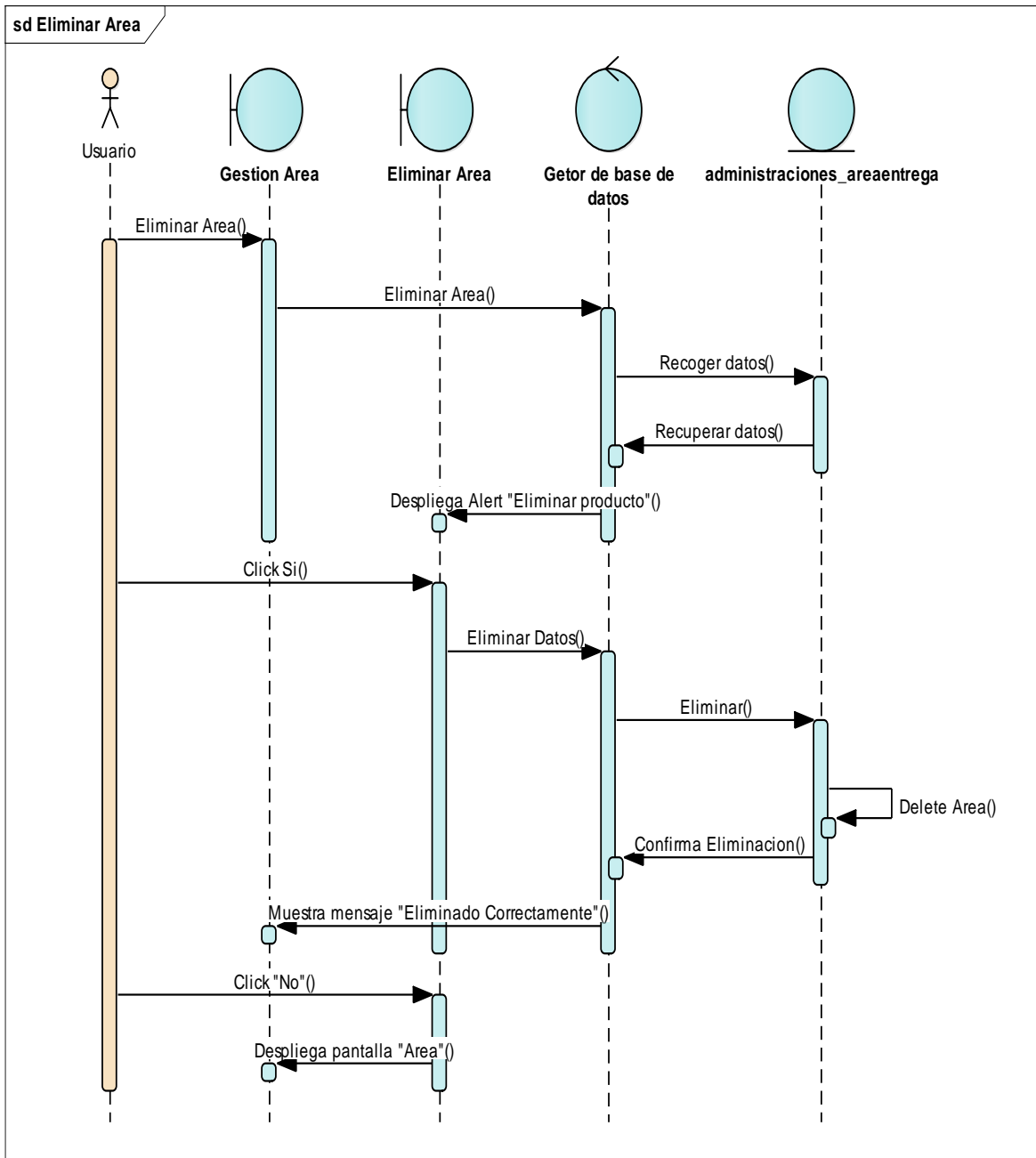
### Modificar Área



## 2.12.18. Eliminar Área.

Figura 33

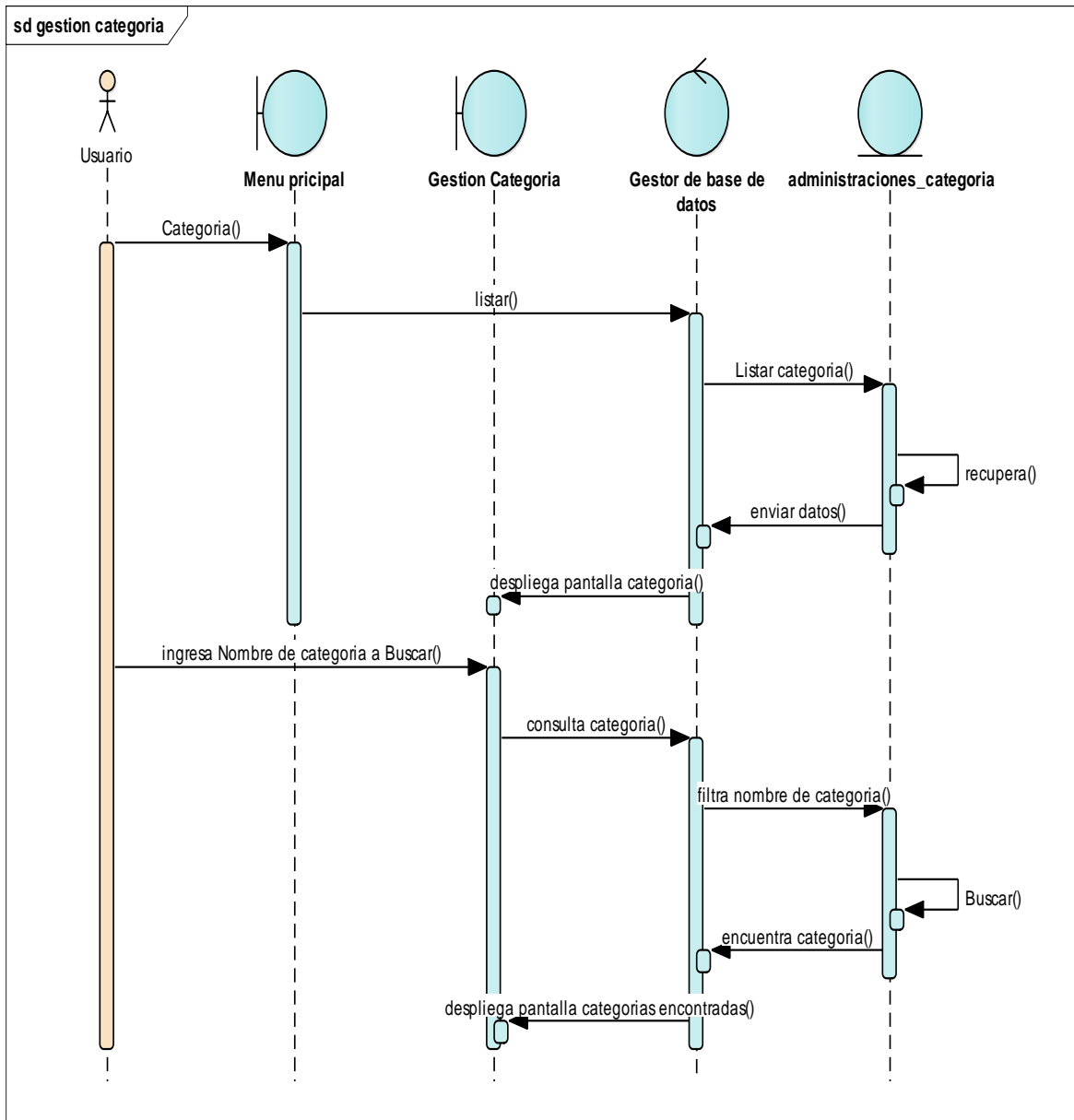
### Eliminar Área



## 2.12.19. Gestión Categoría.

Figura 34

Gestión Categoría

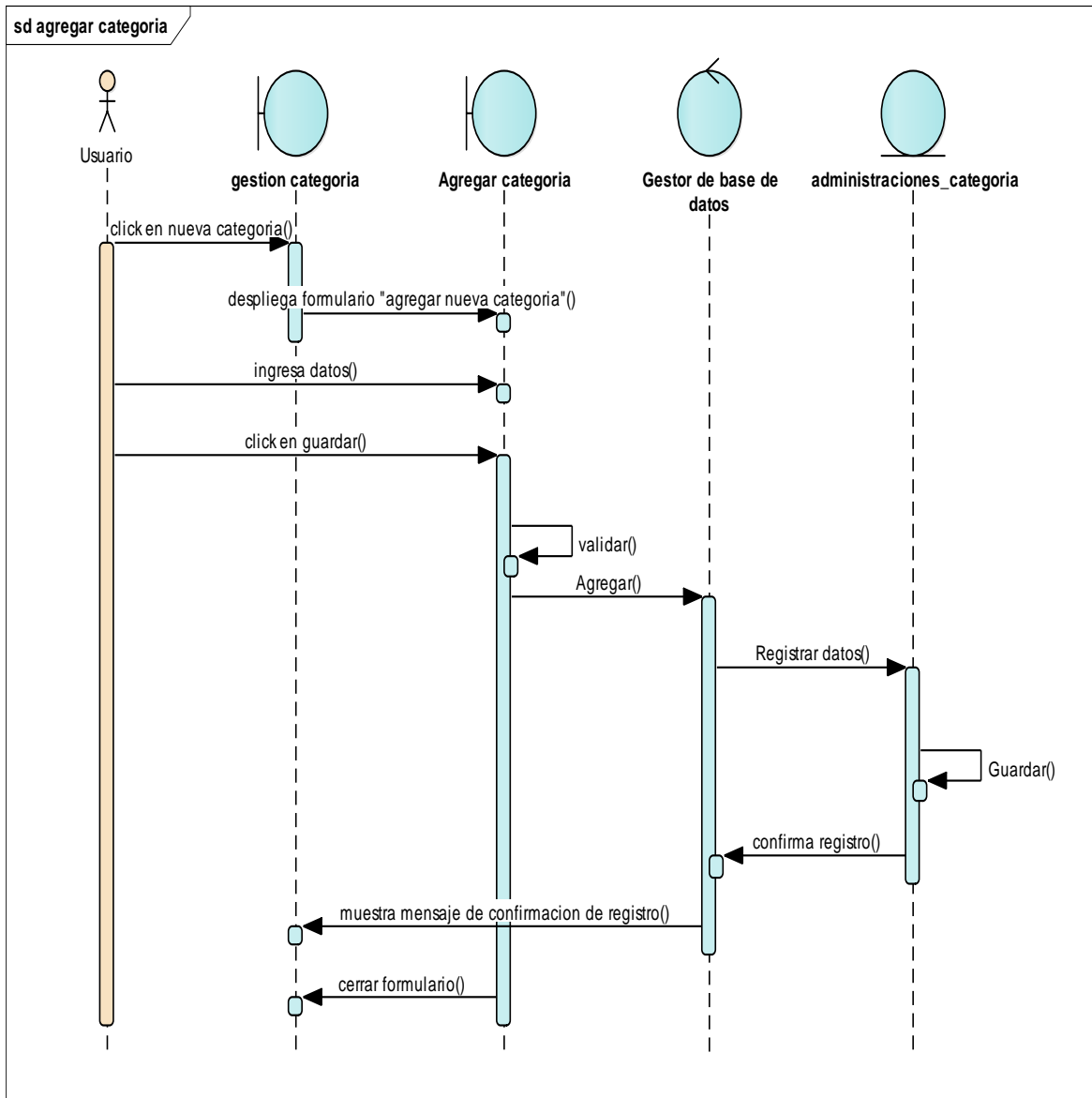




## 2.12.20. Agregar Categoría.

Figura 35

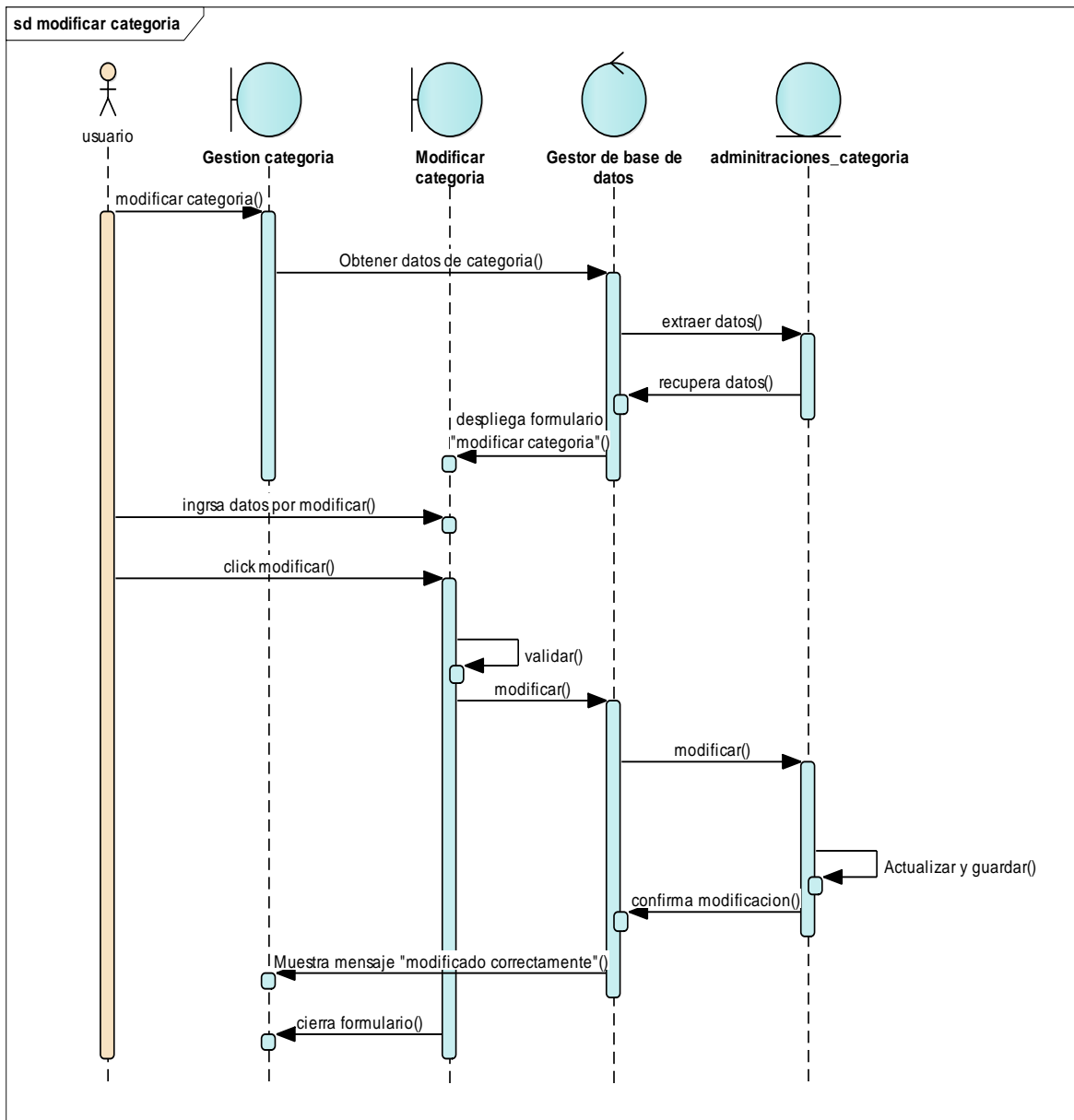
### Agregar Categoría



## 2.12.21. Modificar Categoría.

Figura 36

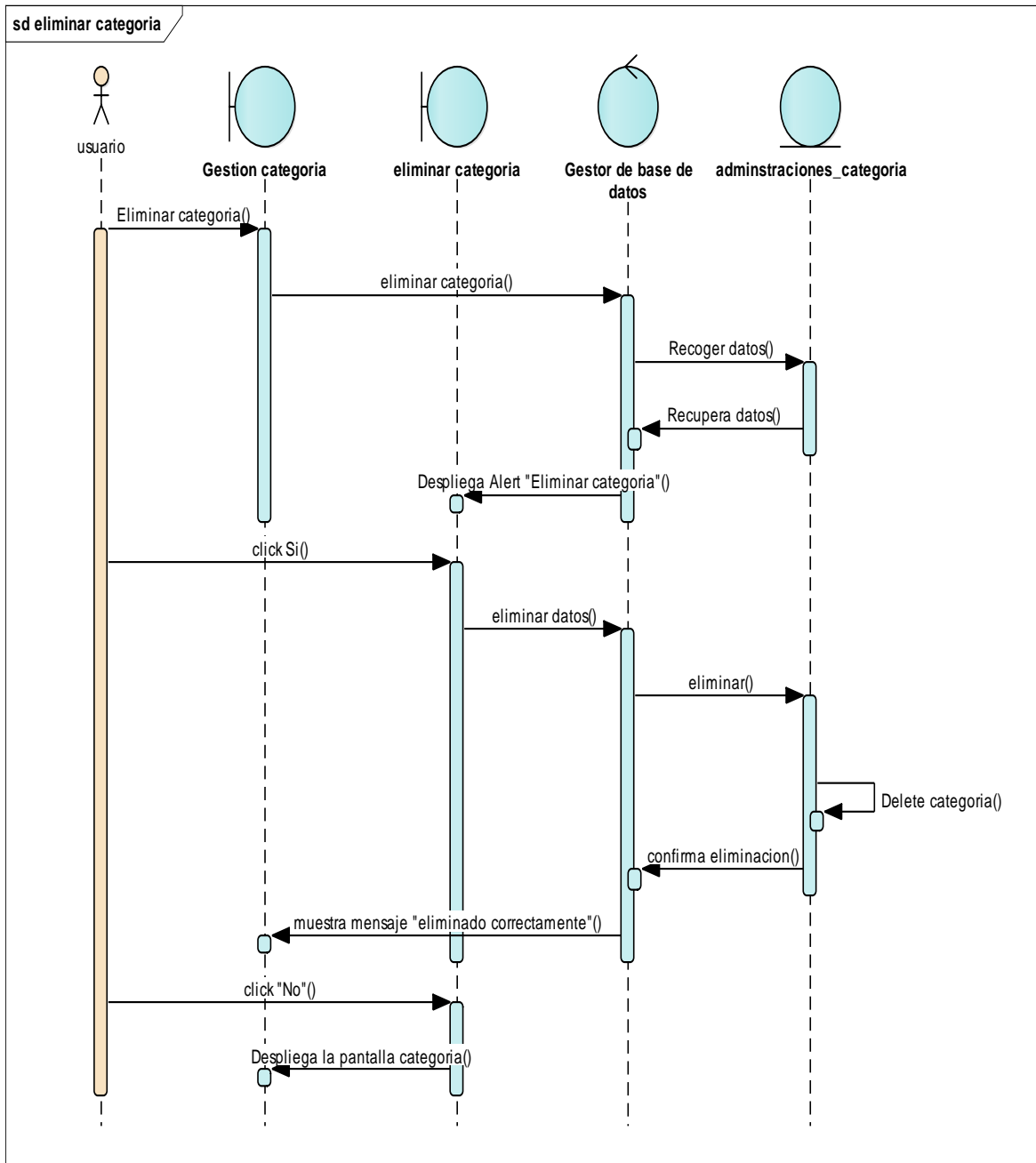
### Modificar Categoría



## 2.12.22. Eliminar Categoría.

Figura 37

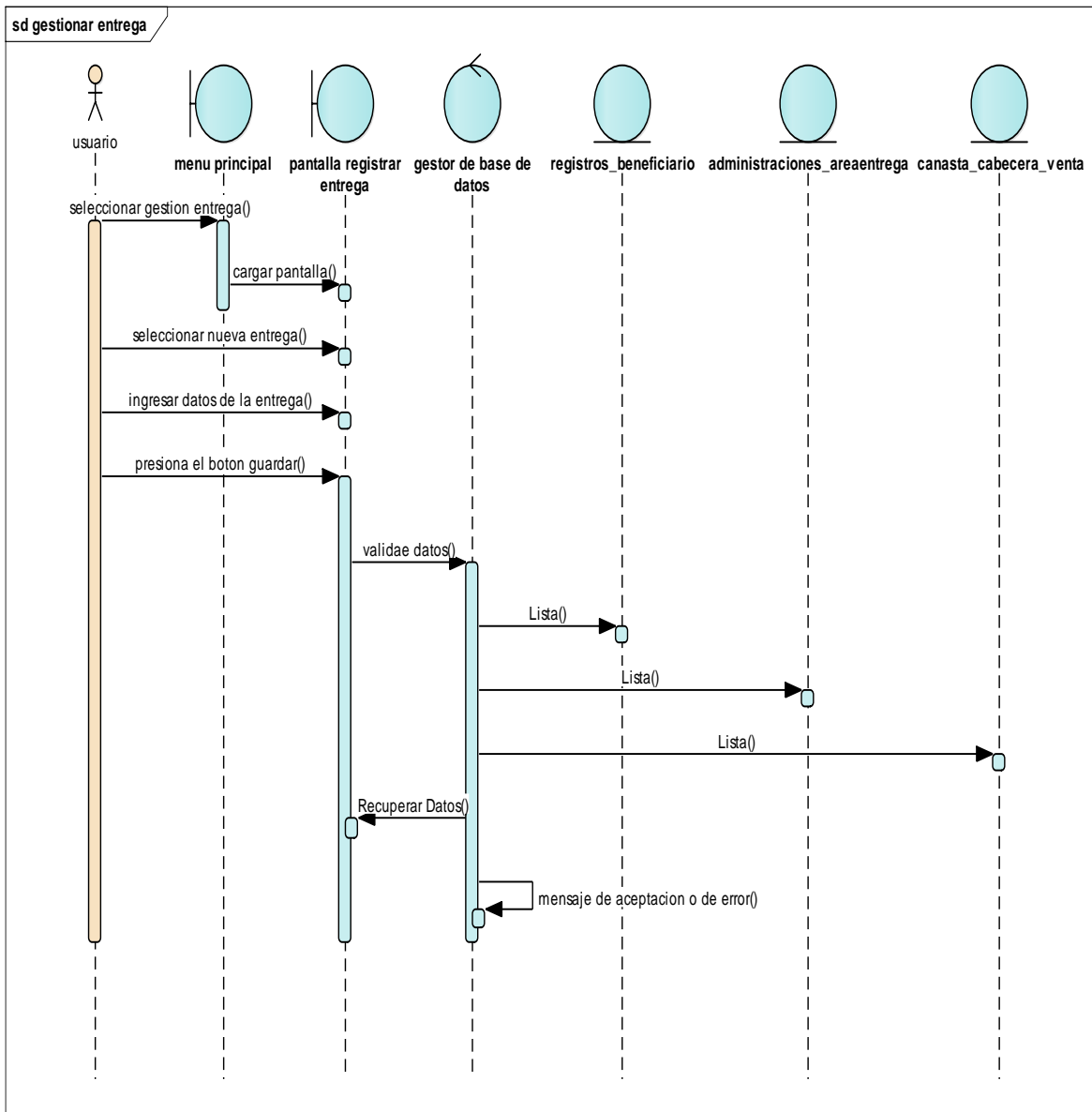
### Eliminar Categoría



## 2.12.23. Registrar Entrega.

Figura 38

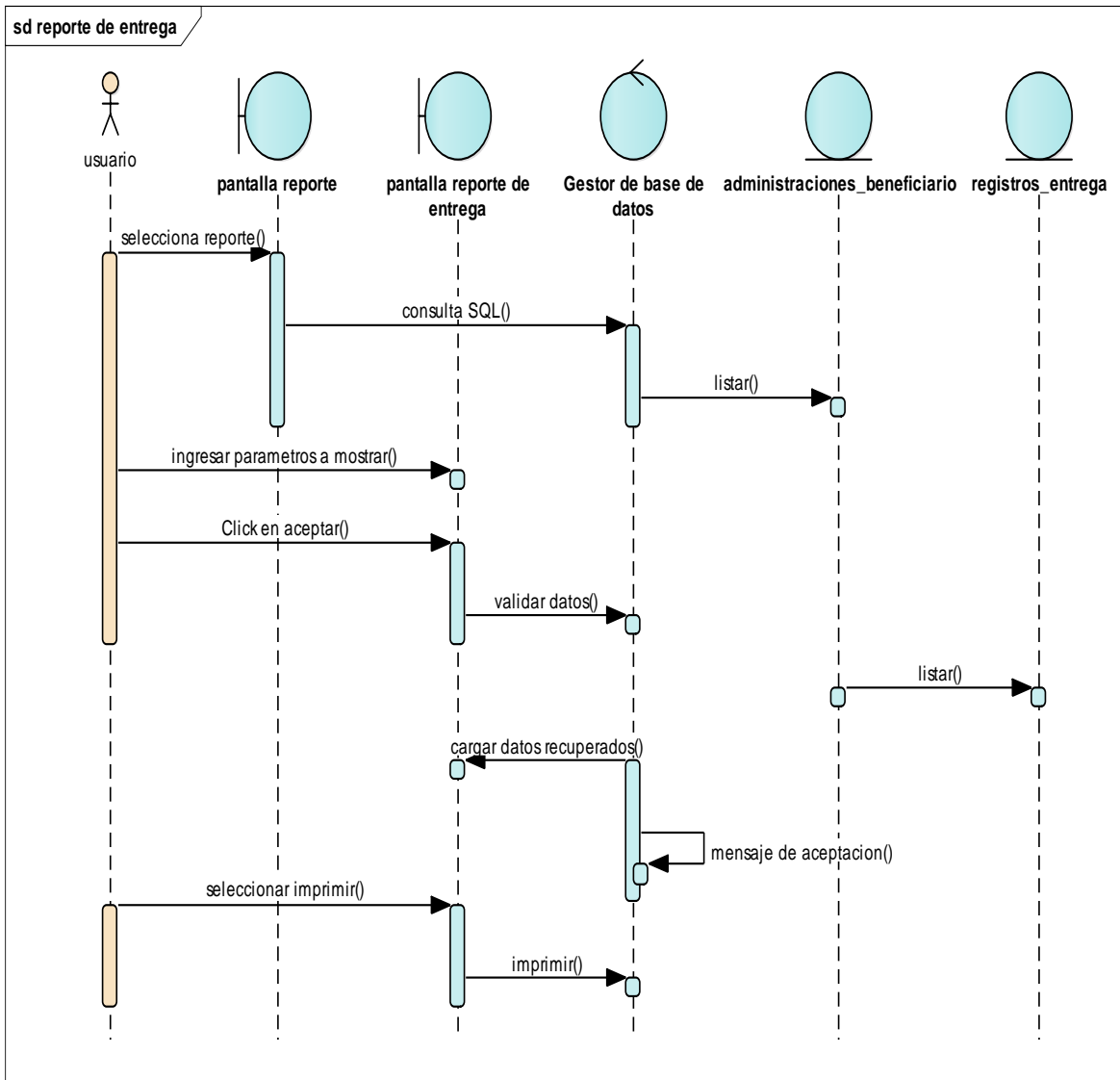
### Registrar Entrega



## 2.12.24. Reporte de Entrega.

Figura 39

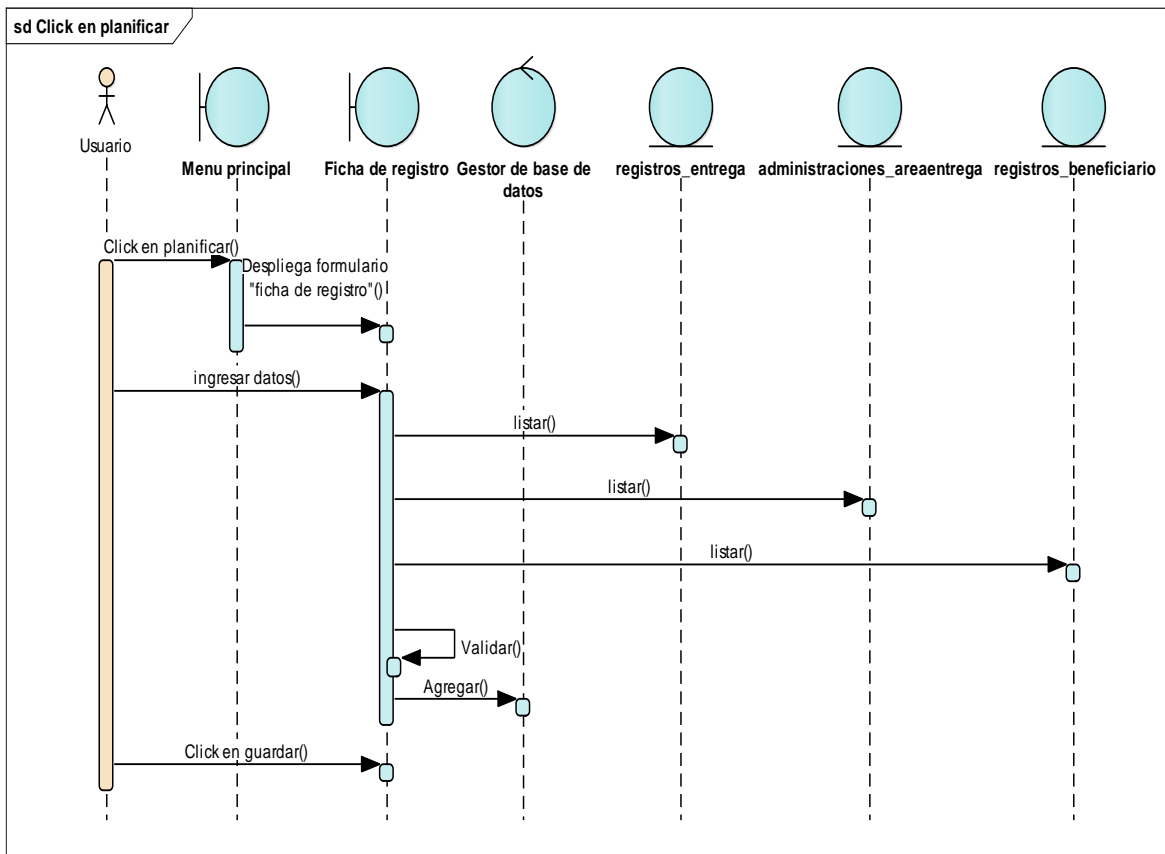
Reporte de Entrega



## 2.12.25. Ficha de registro.

Figura 40

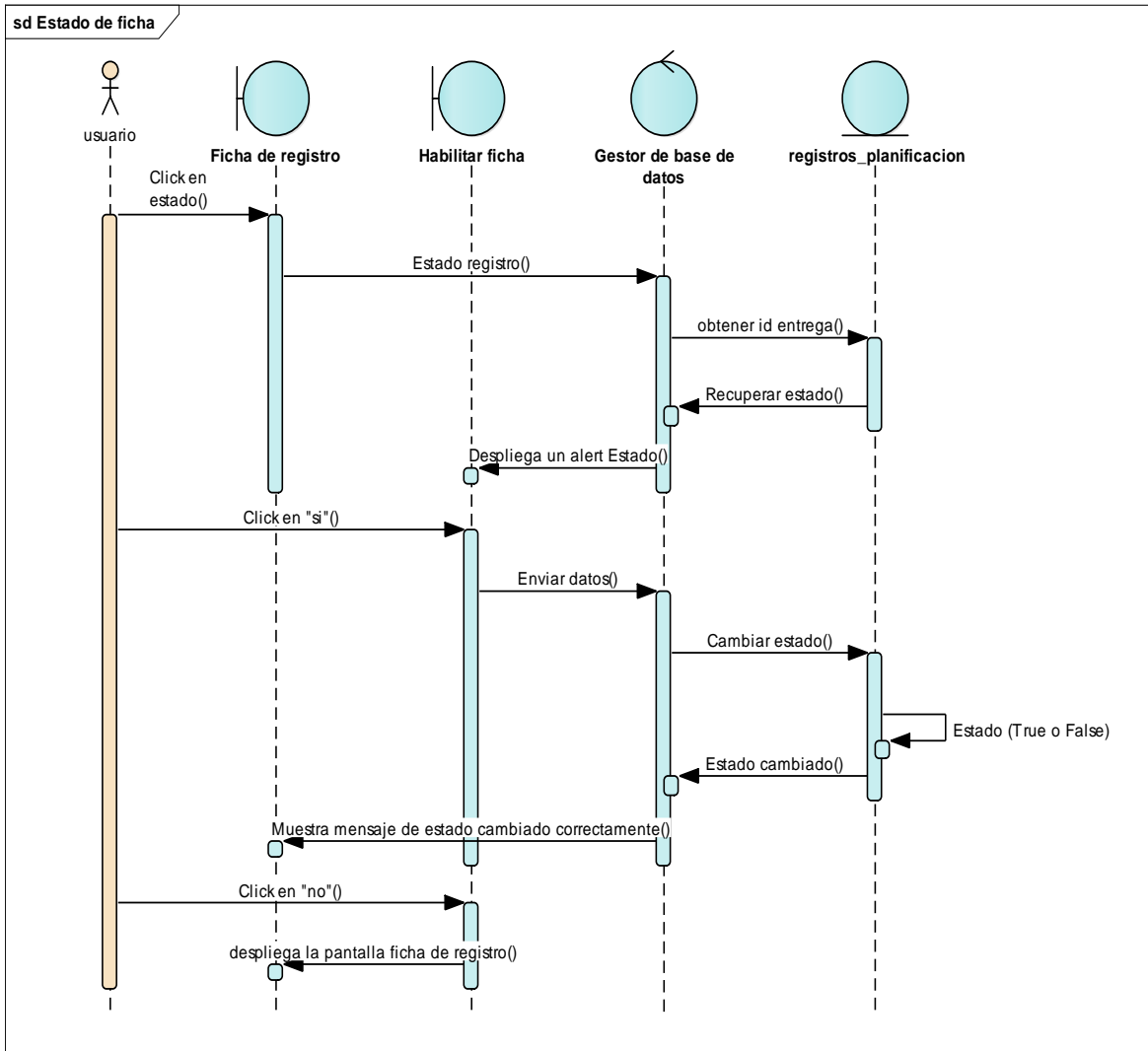
Ficha de registro



## 2.12.26. Estado Ficha de registro.

Figura 41

Estado Ficha de registro



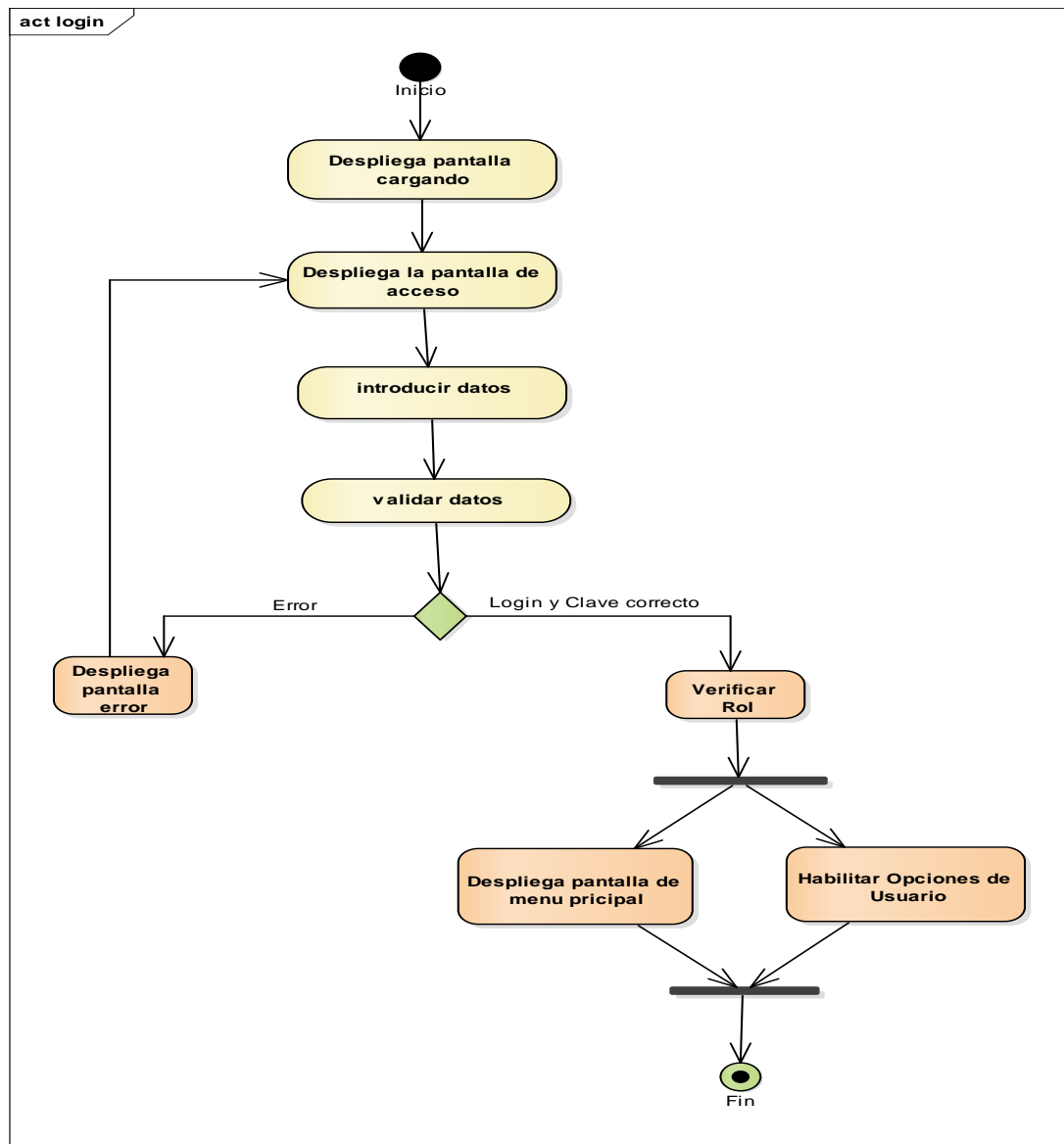
### 2.13. Diagramas de Actividad.

En un diagrama de actividades muestra la iteración de un conjunto de objetos en una aplicación a través del tiempo, nos permite mostrar el flujo de los datos que pasan de una acción a otra, en estos diagramas no se muestra ni se describe la estructura de los datos.

#### 2.13.1. Ingresar al Sistema.

*Figura 42*

*Ingresar al Sistema*

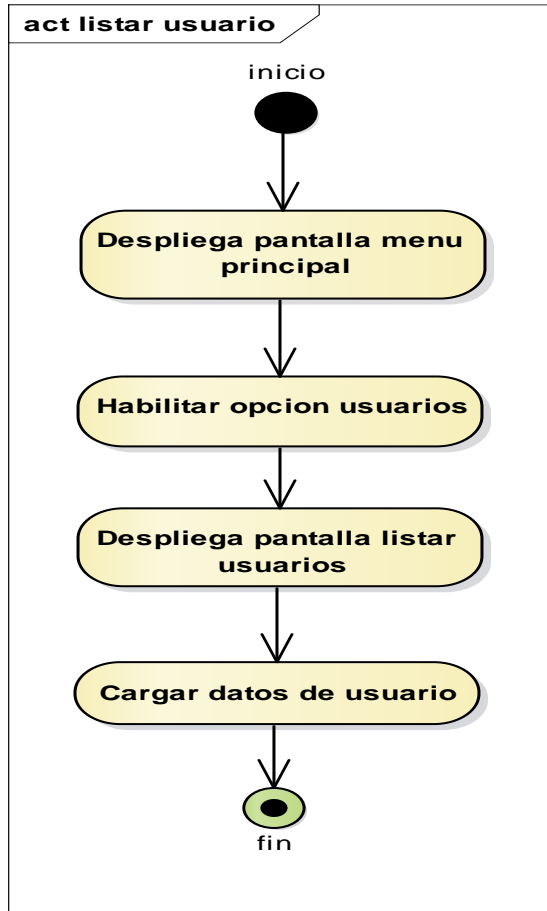




### 2.13.2. Listar Usuario.

*Figura 43*

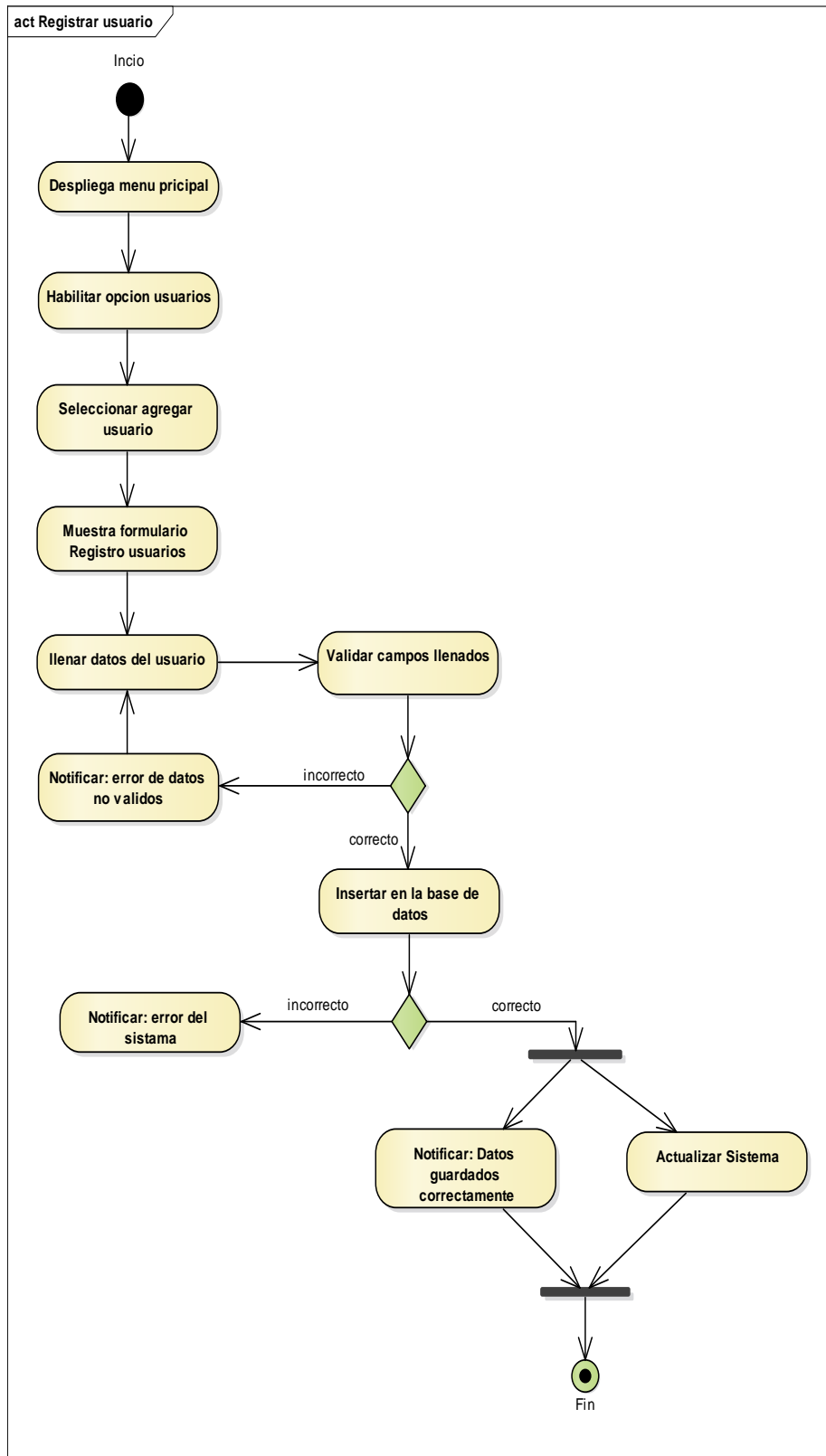
*Listar Usuario.*



### 2.13.3. Registrar Usuario.

*Figura 44*

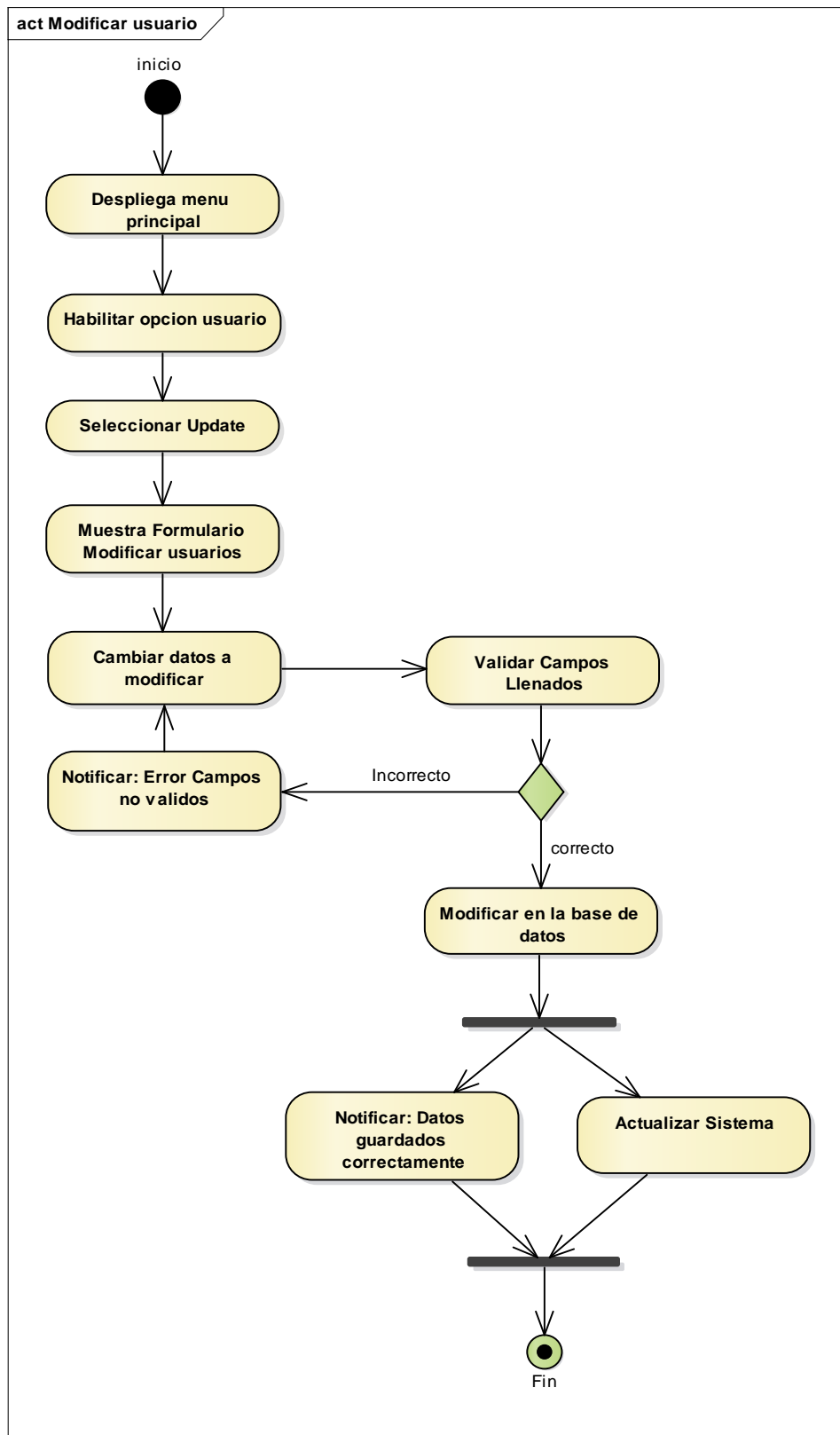
*Registrar Usuario.*



## 2.13.4. Modificar Usuario.

*Figura 45*

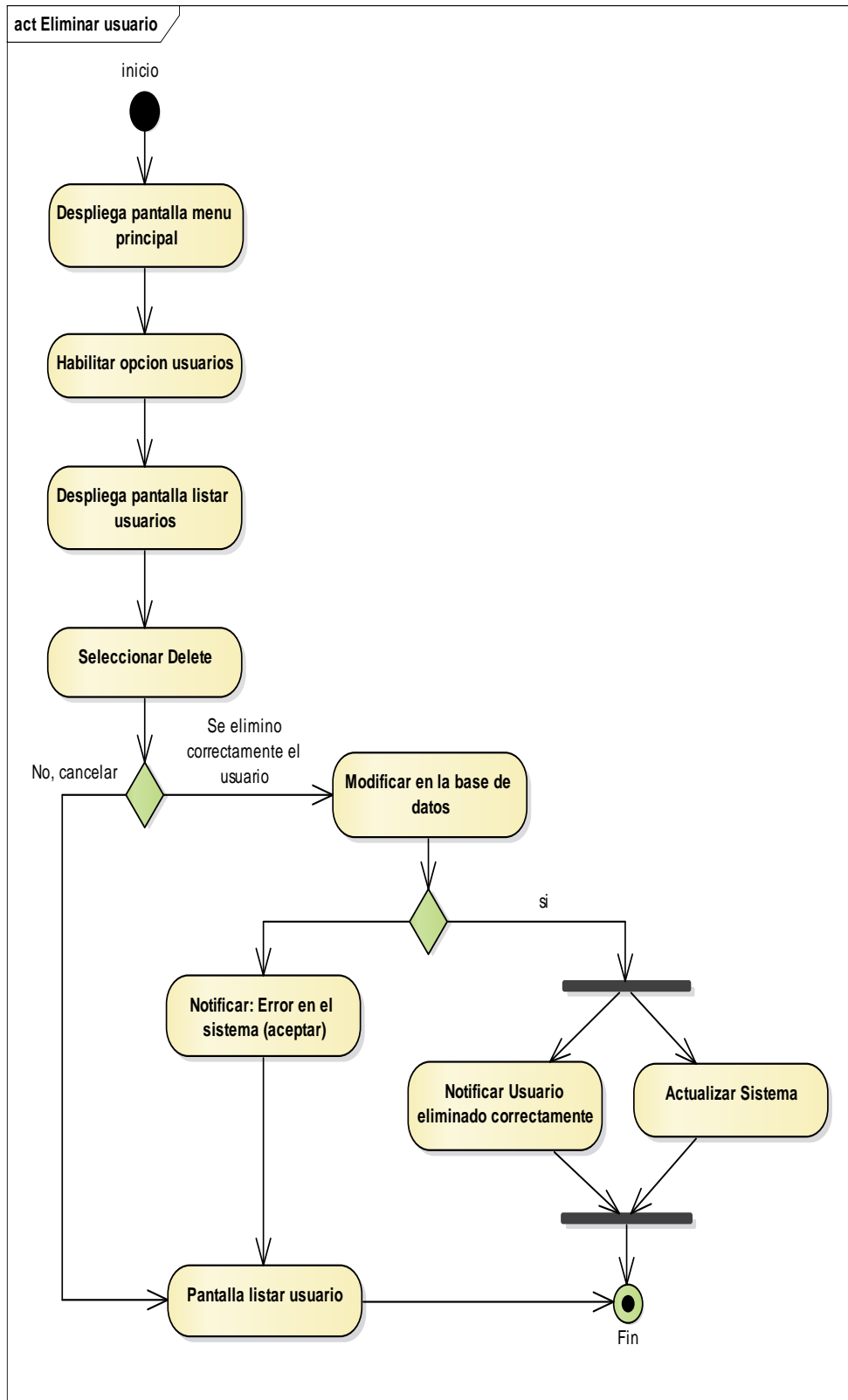
*Modificar Usuario.*



### 2.13.5. Eliminar Usuario.

*Figura 46*

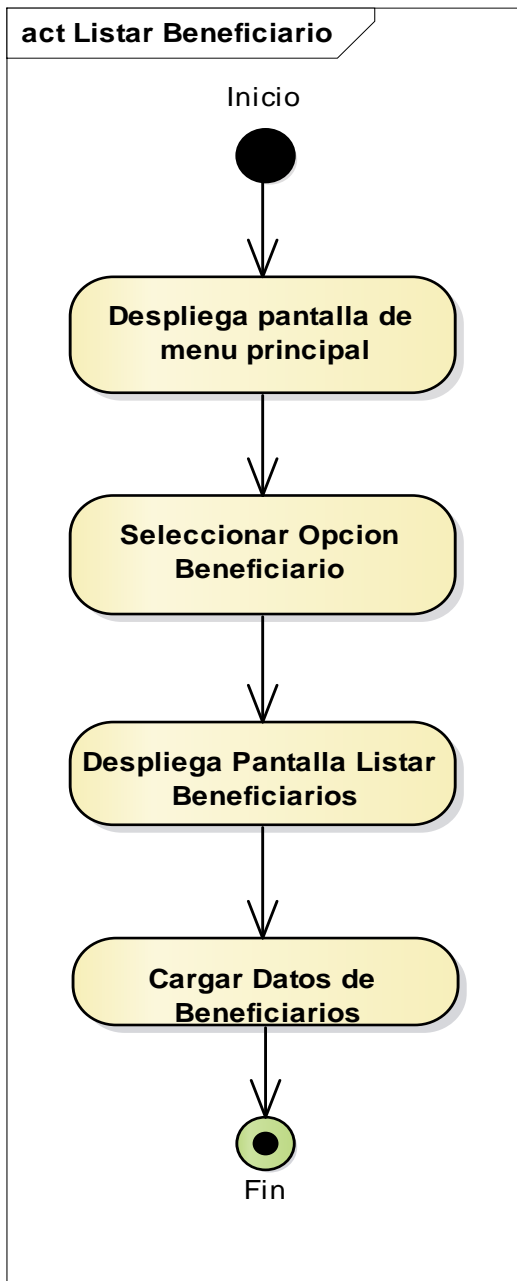
*Eliminar Usuario.*



### 2.13.6. Listar Beneficiario.

*Figura 47*

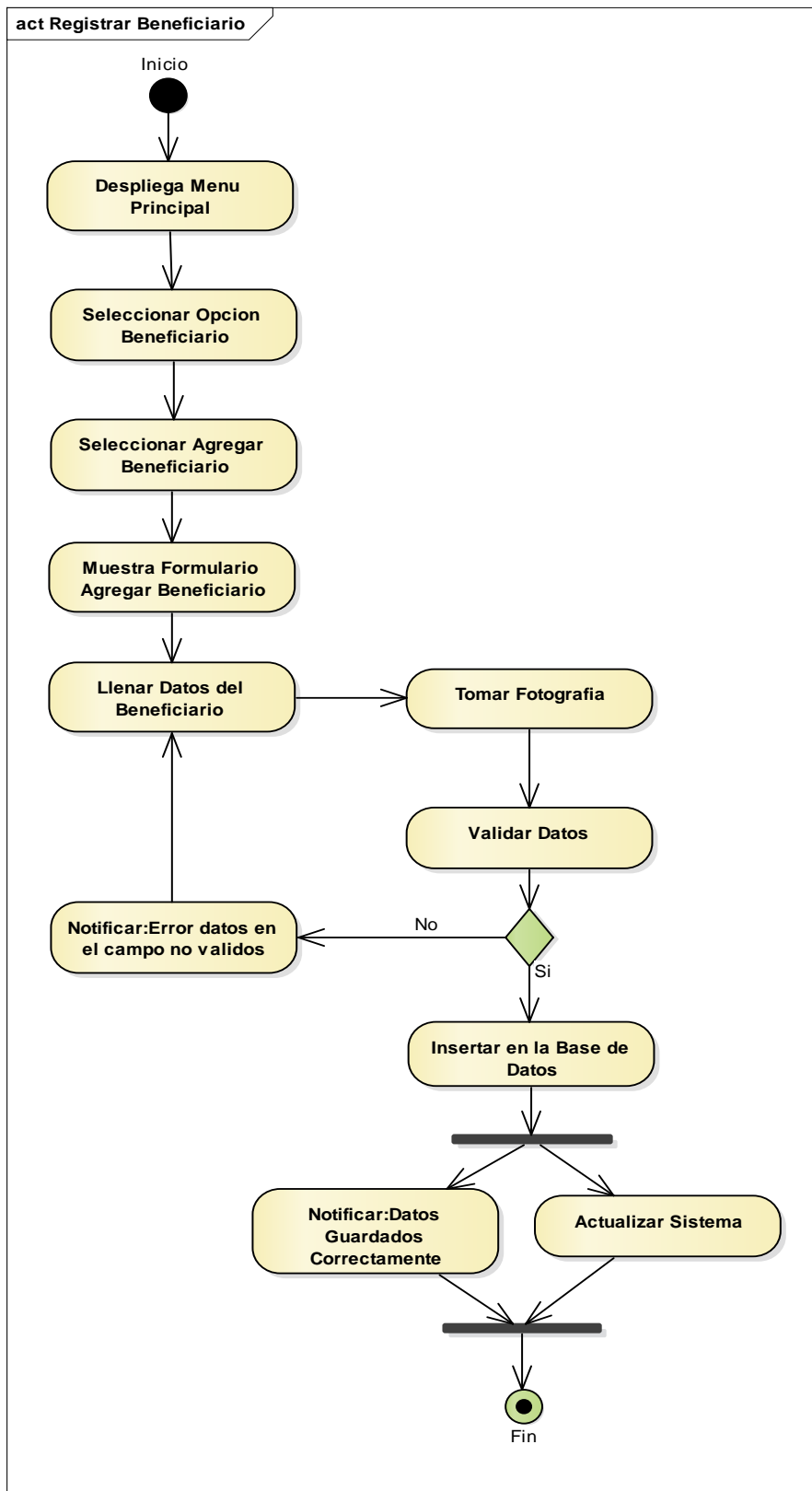
*Listar Beneficiario.*



### 2.13.7. Registrar Beneficiario.

*Figura 48*

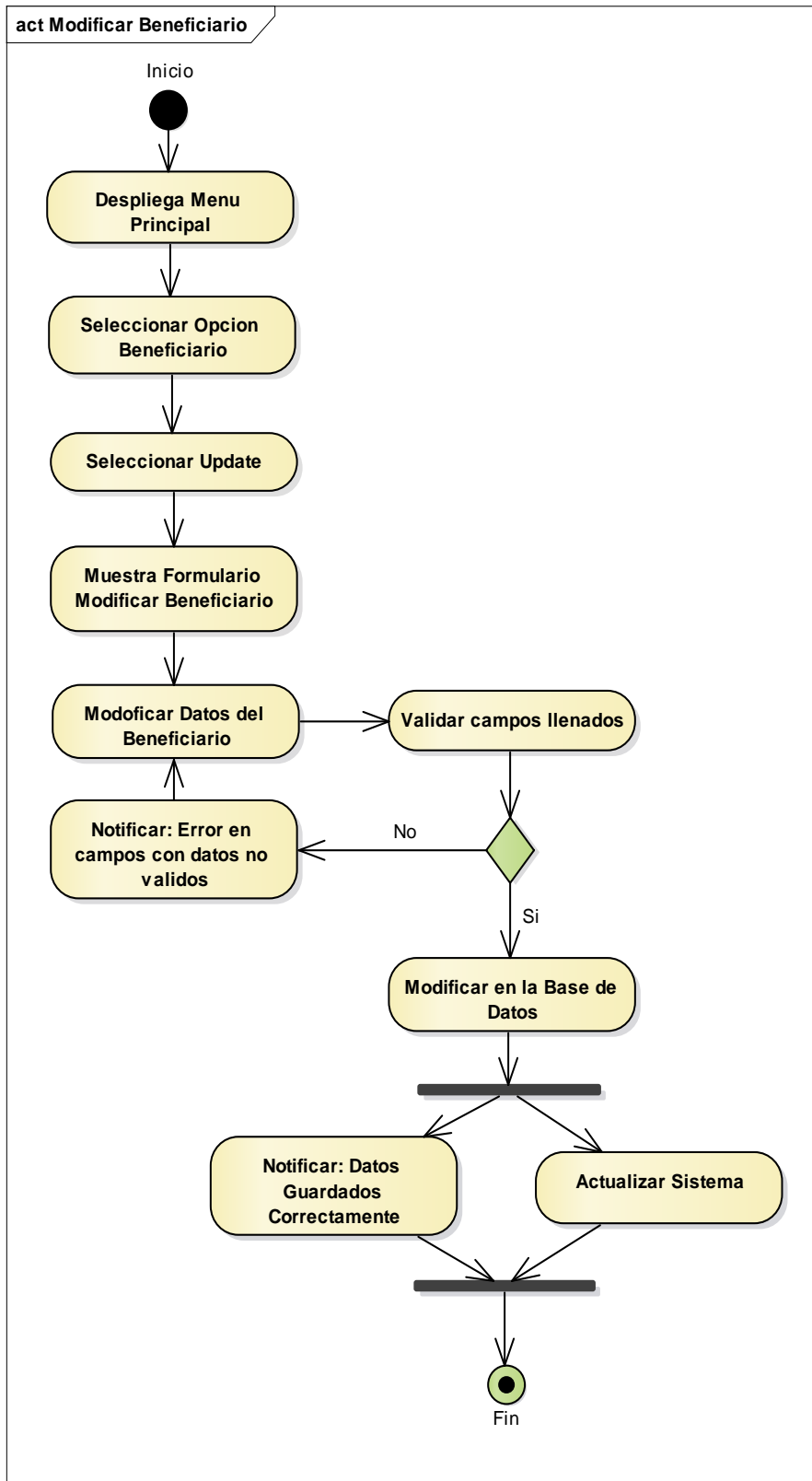
*Registrar Beneficiario.*



### 2.13.8. Modificar Beneficiario

Figura 49

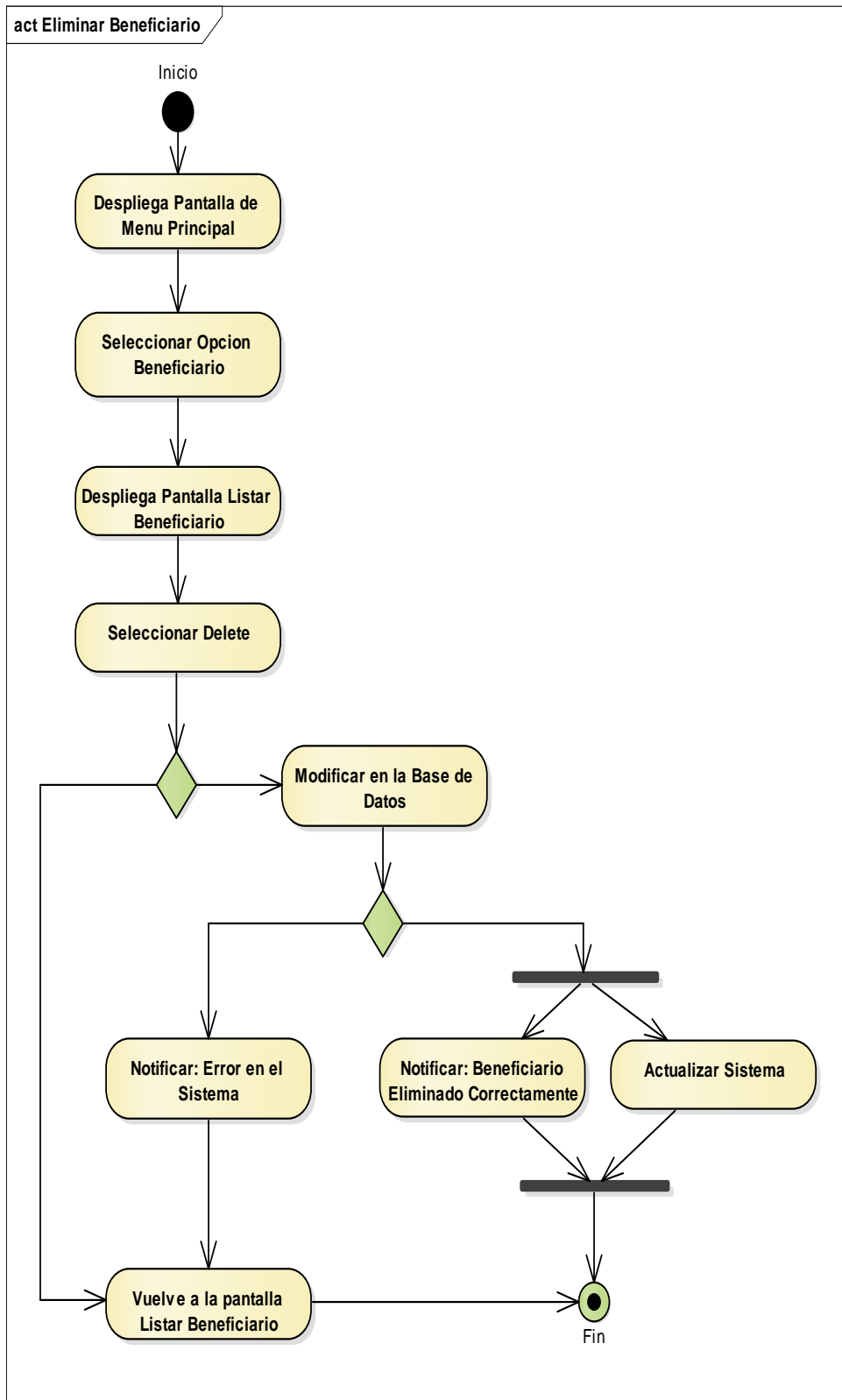
Modificar Beneficiario.



### 2.13.9. Eliminar Beneficiario

Figura 50

Eliminar Beneficiario

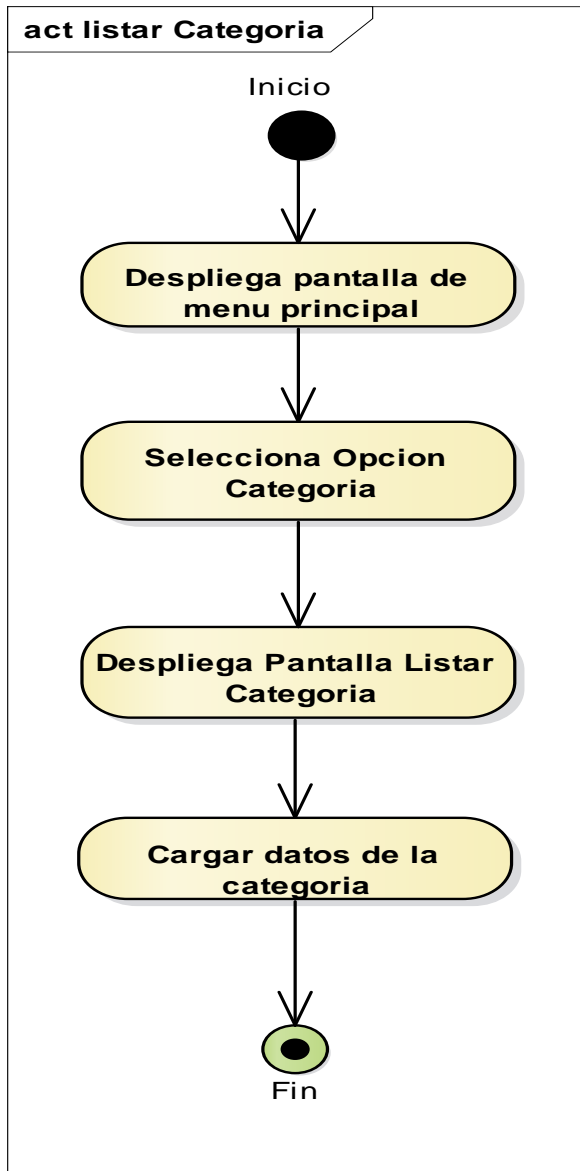




### 2.13.10. Listar Beneficiario.

*Figura 51*

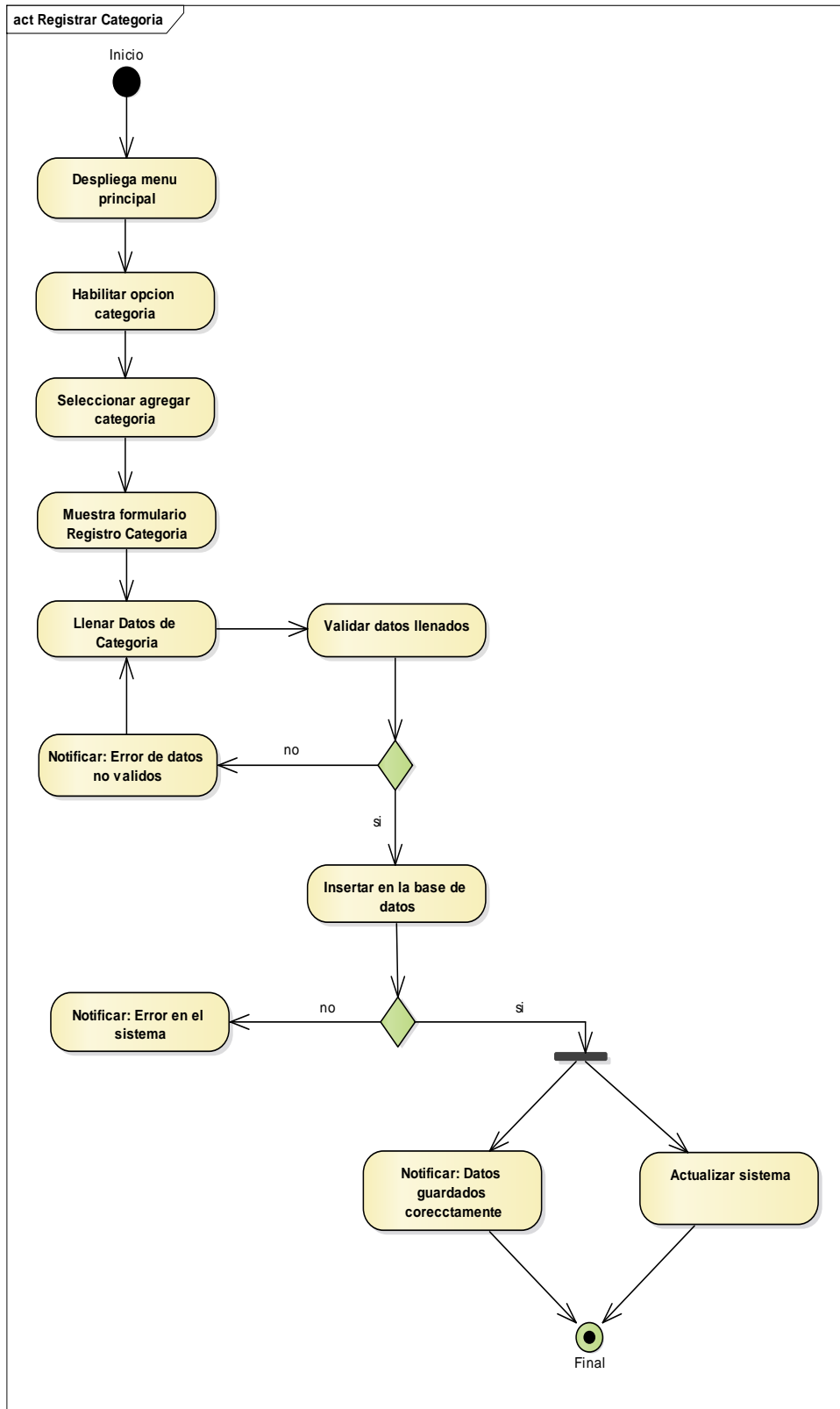
*Listar Beneficiario.*



### 2.13.11. Registrar Categoría

*Figura 52*

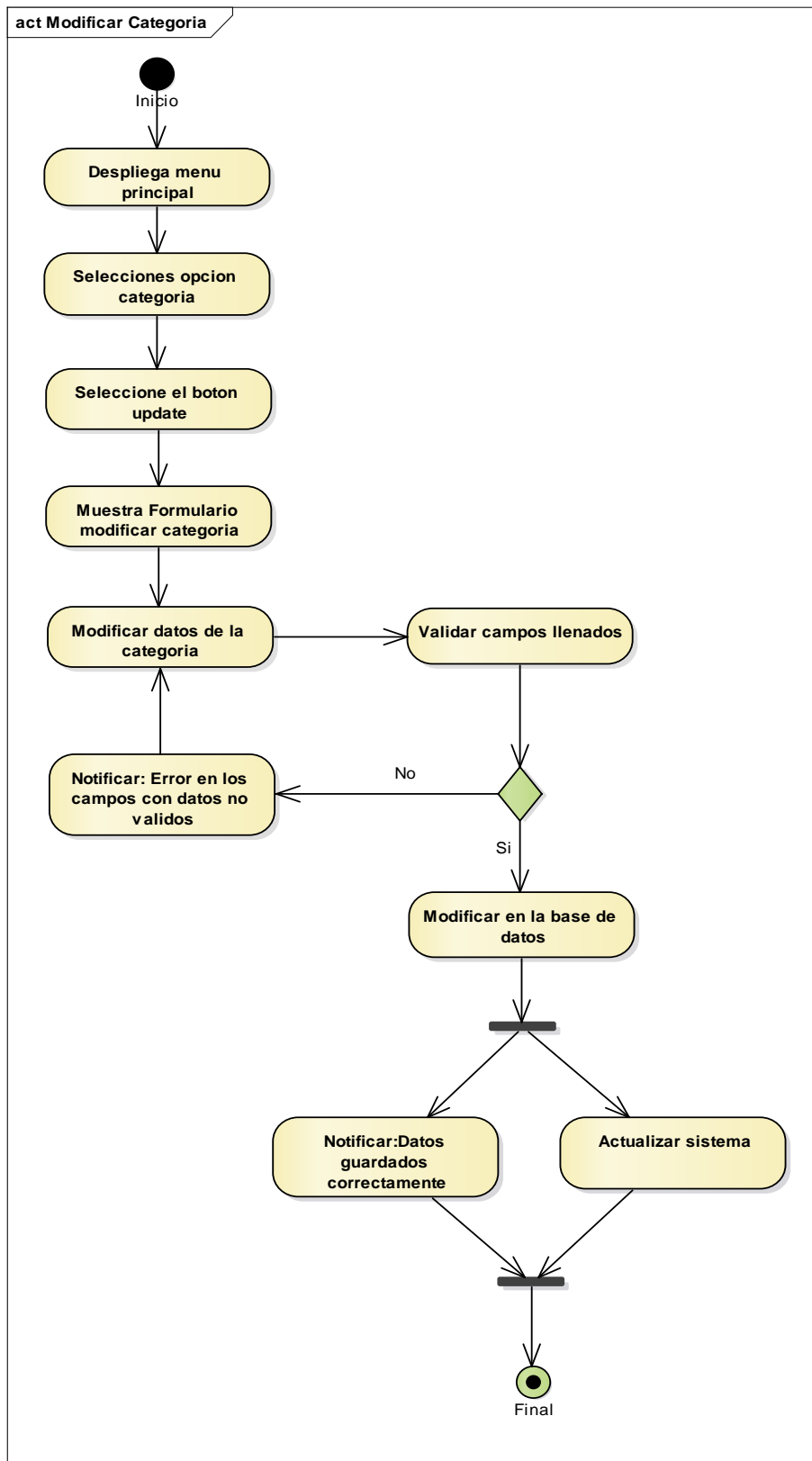
*Registrar Categoría.*



## 2.13.12. Modificar Categoría.

Figura 53

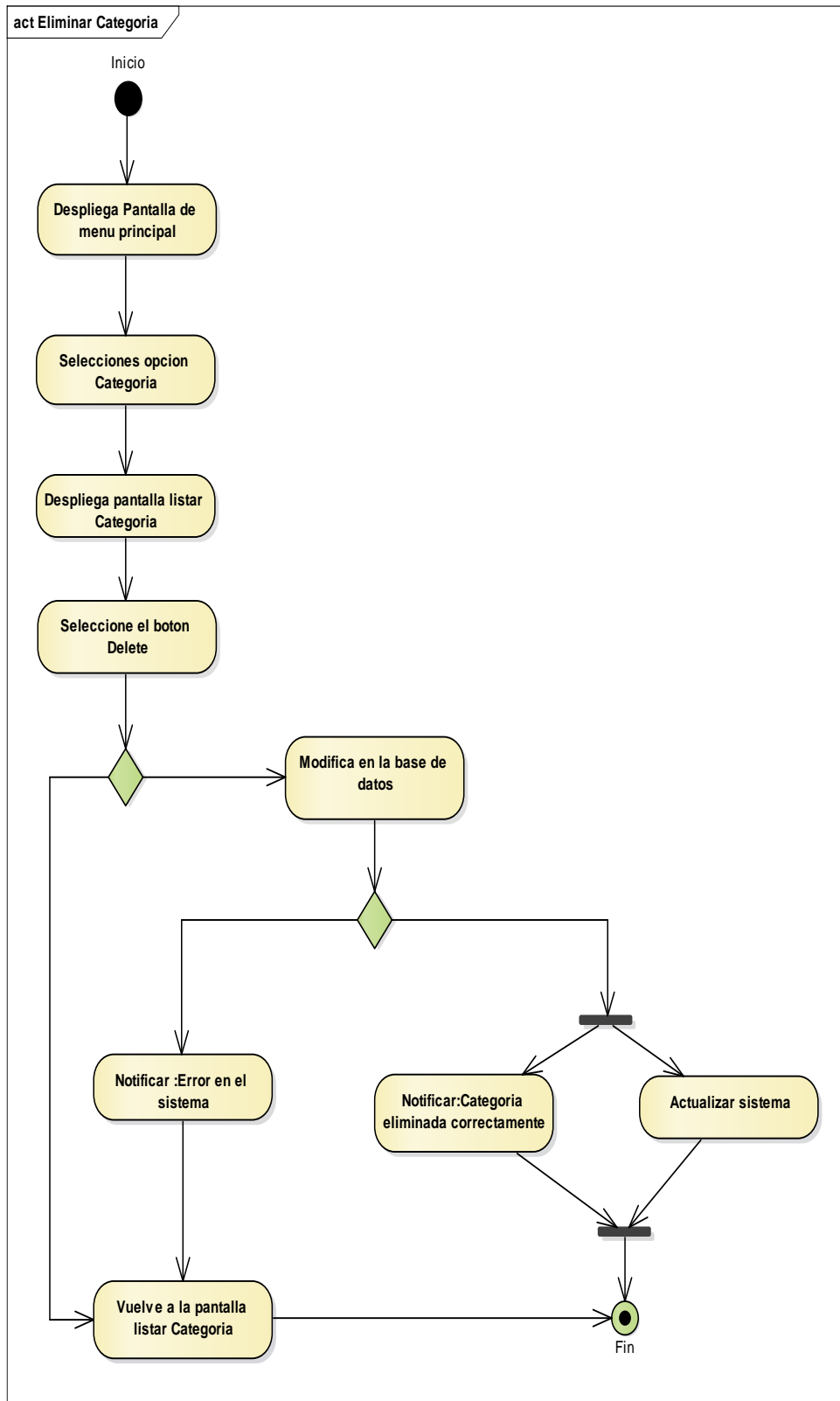
Modificar Categoría.



### 2.13.13. Eliminar Categoría.

*Figura 54*

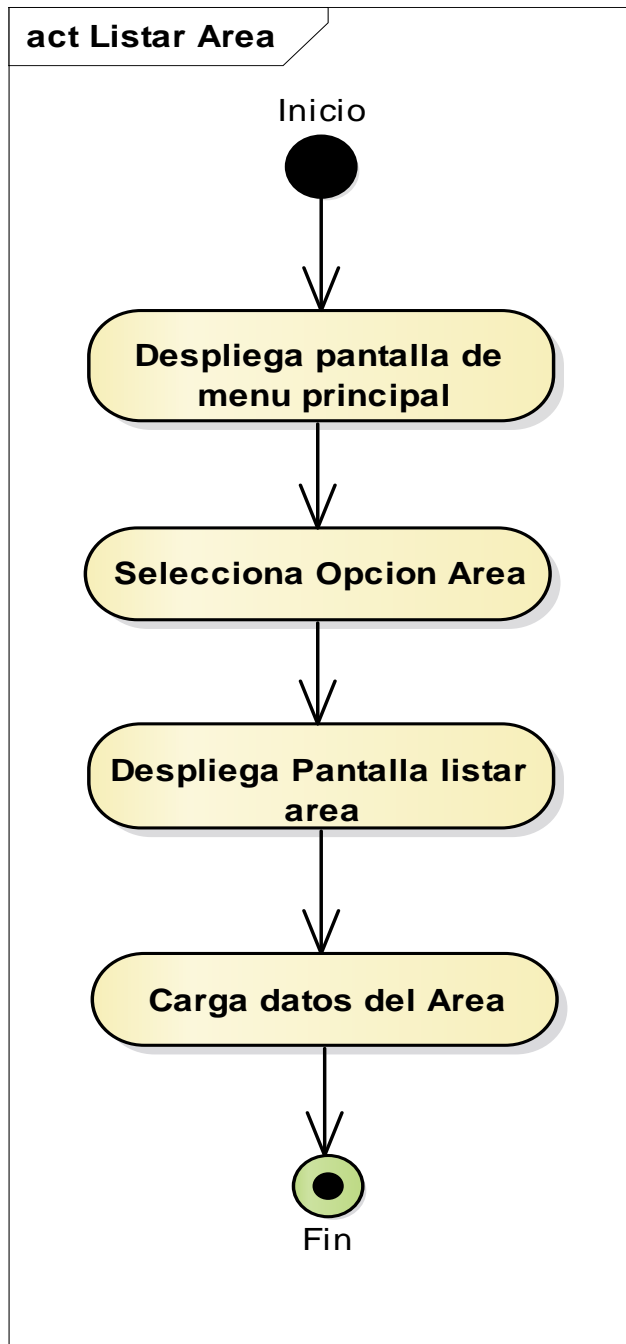
*Eliminar Categoría.*



### 2.13.14. Listar Área.

*Figura 55*

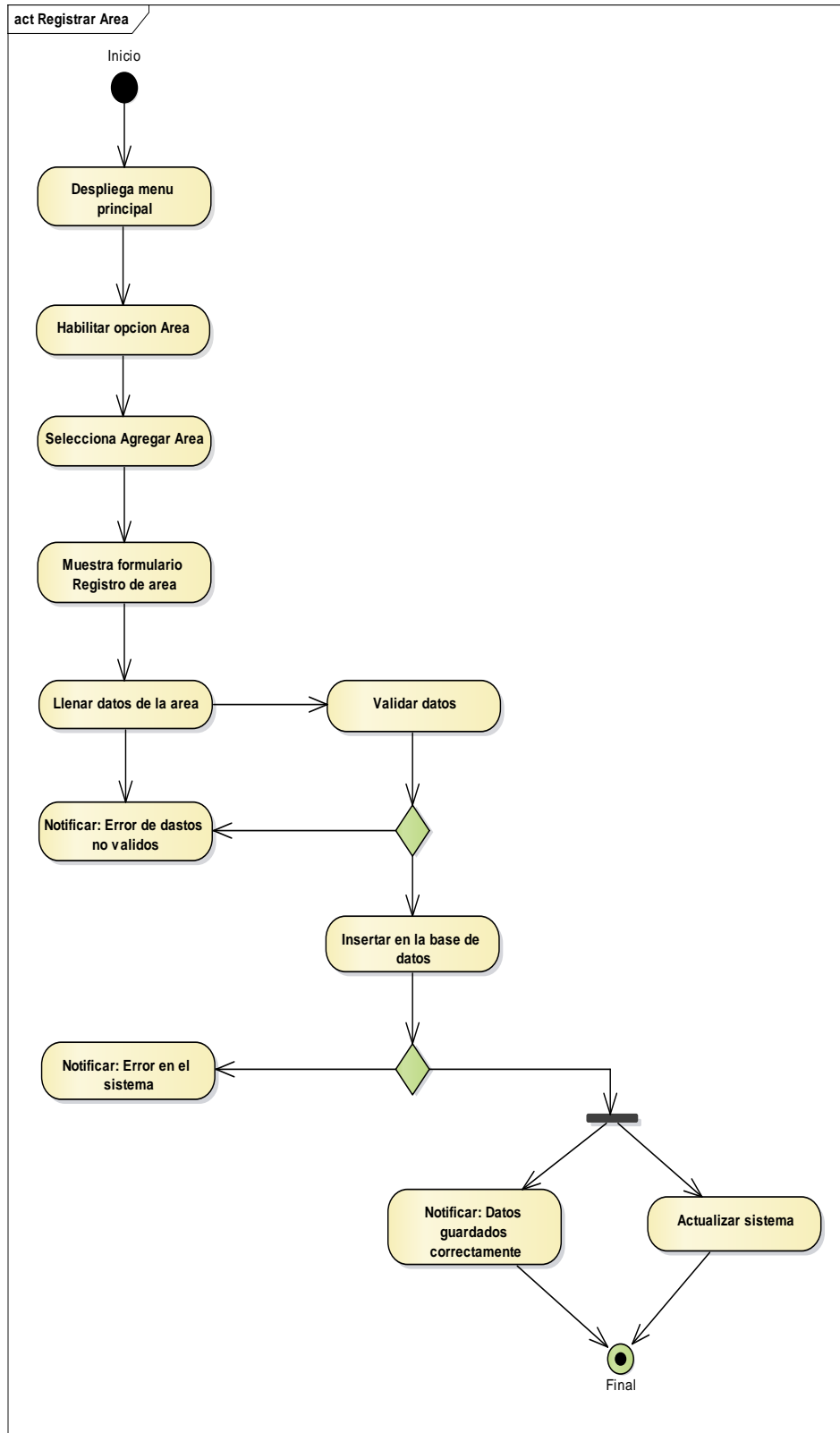
*Listar Área.*



### 2.13.15. Registrar Área.

*Figura 56*

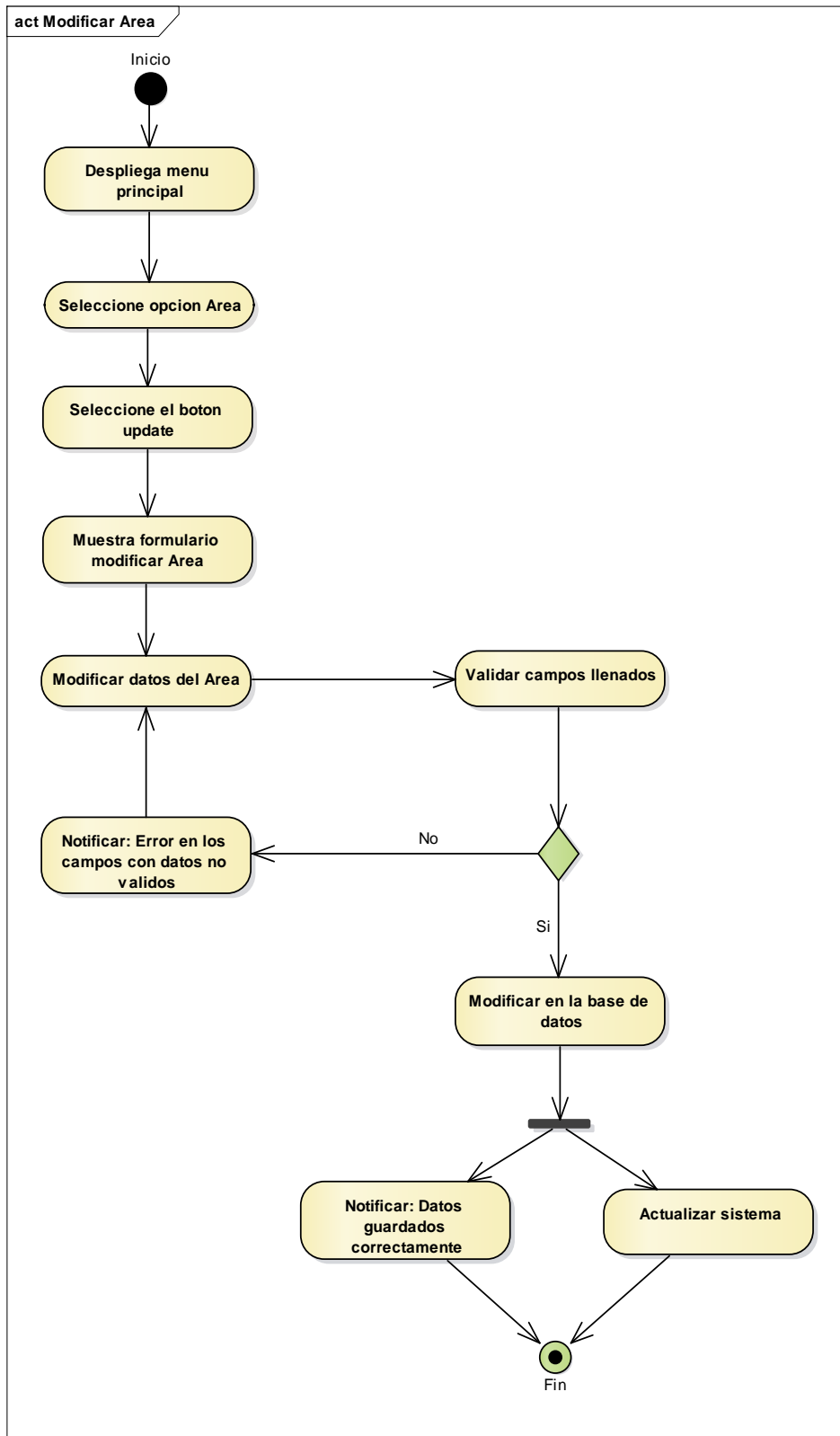
*Registrar Área.*



## 2.13.16. Modificar Área.

*Figura 57*

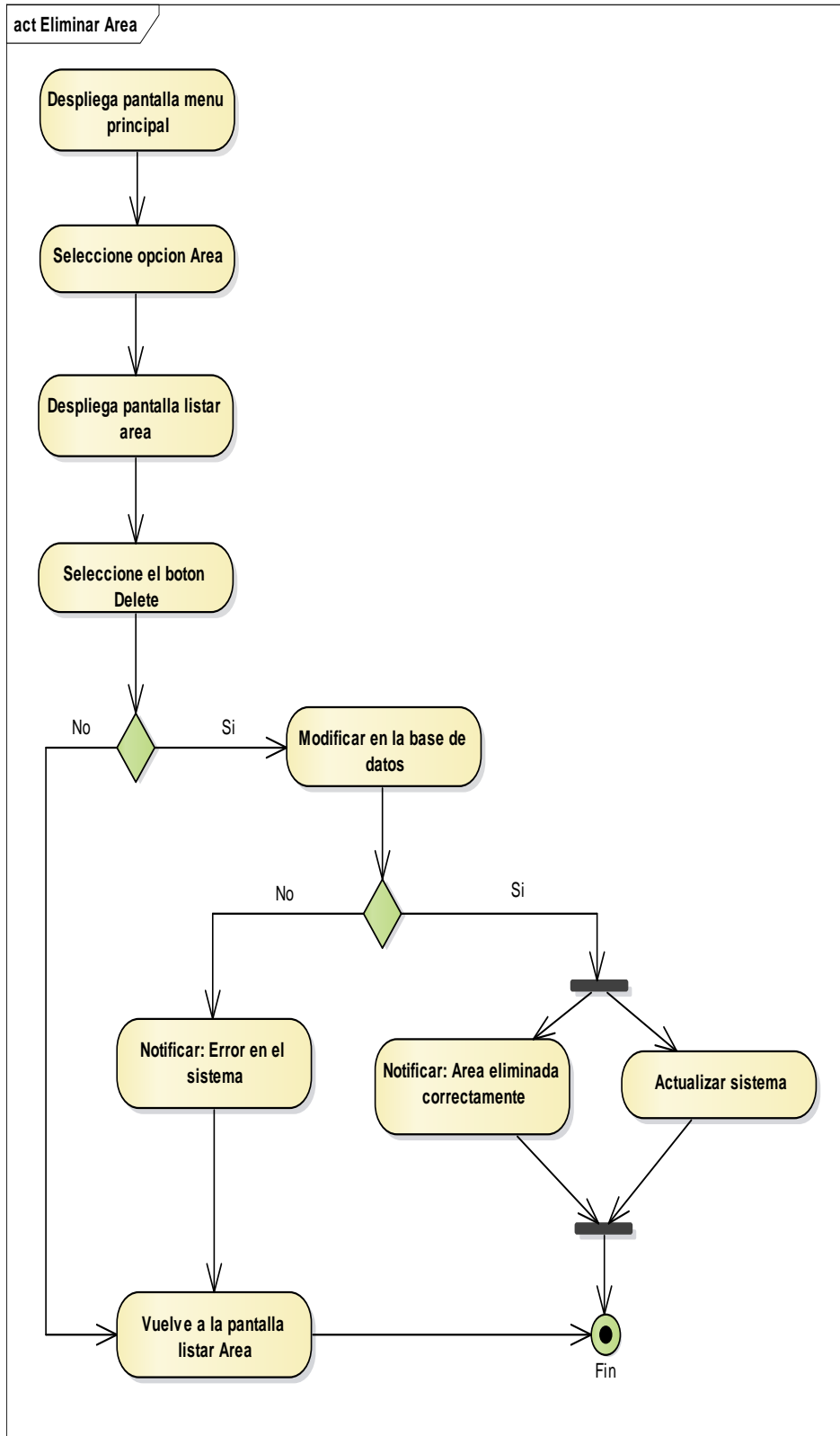
*Modificar Área.*



### 2.13.17. Eliminar Área.

Figura 58

Eliminar Área.

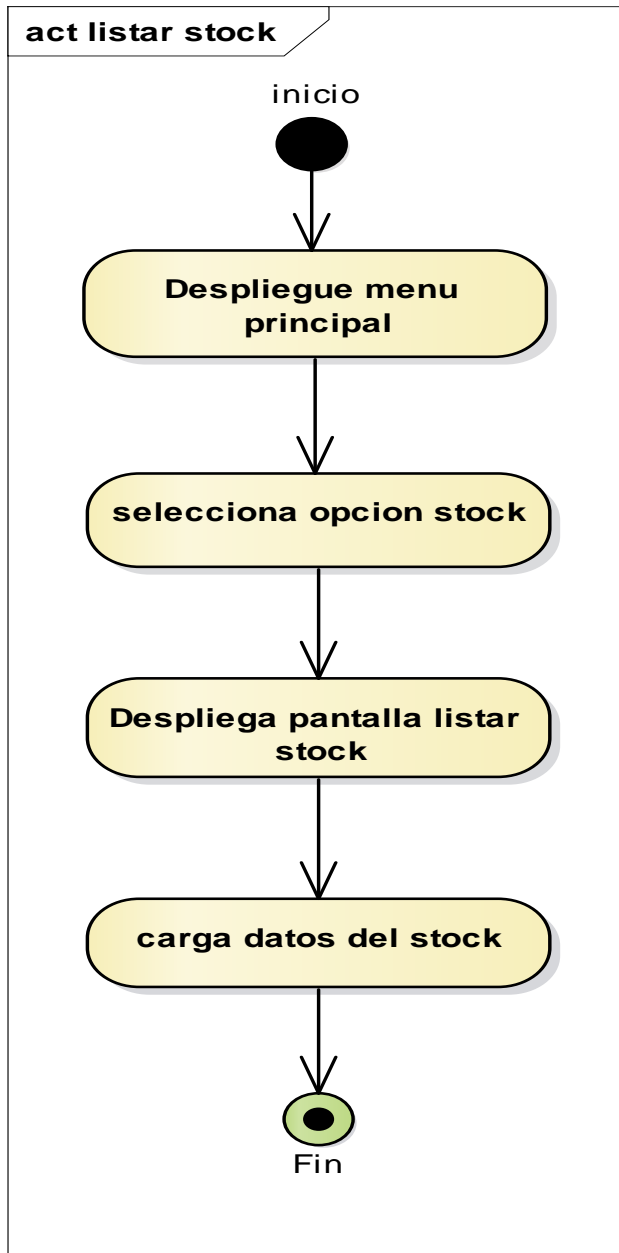




### 2.13.18. Listar Stock.

*Figura 59*

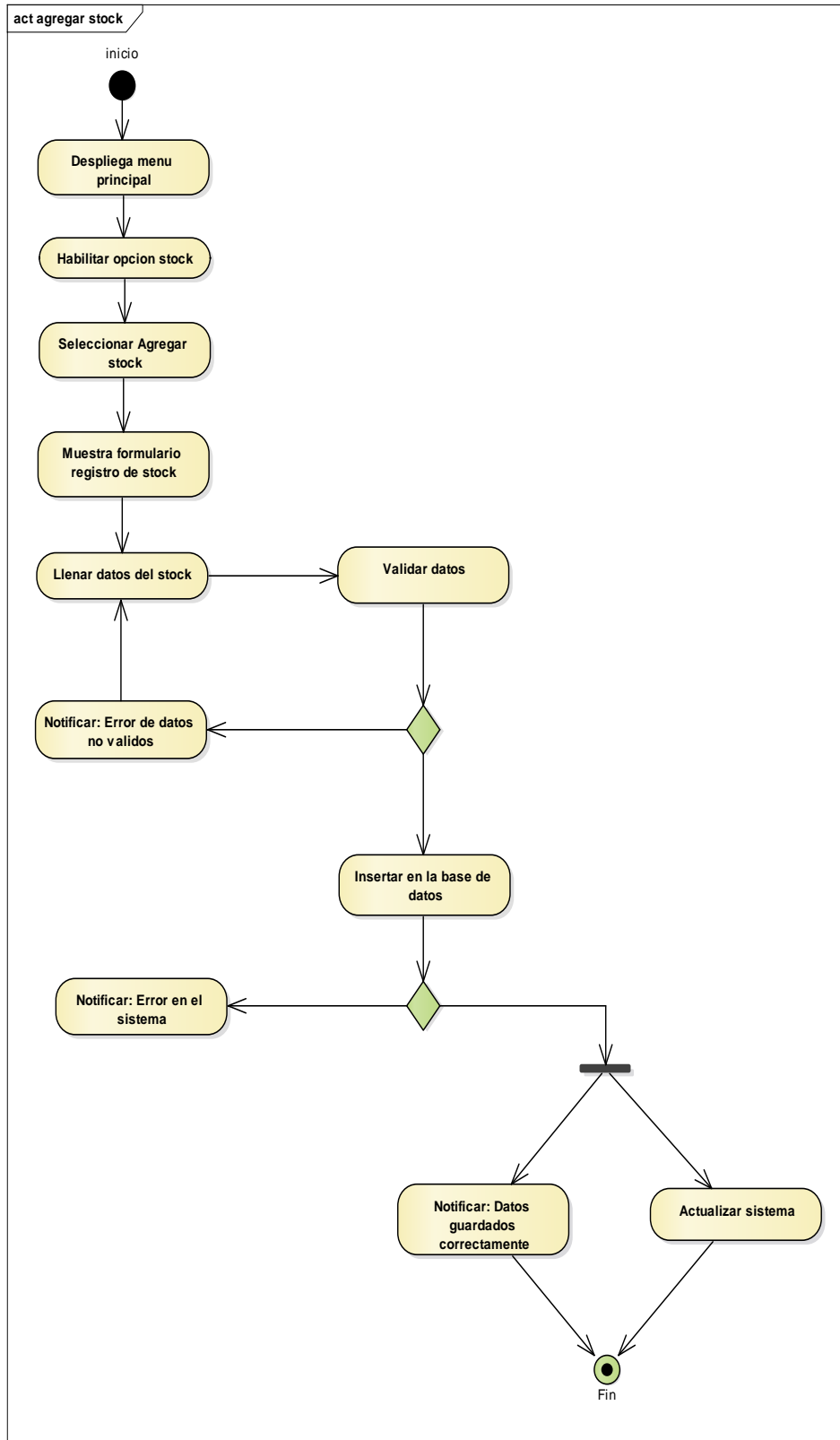
*Listar Stock.*



### 2.13.19. Agregar Stock.

*Figura 60*

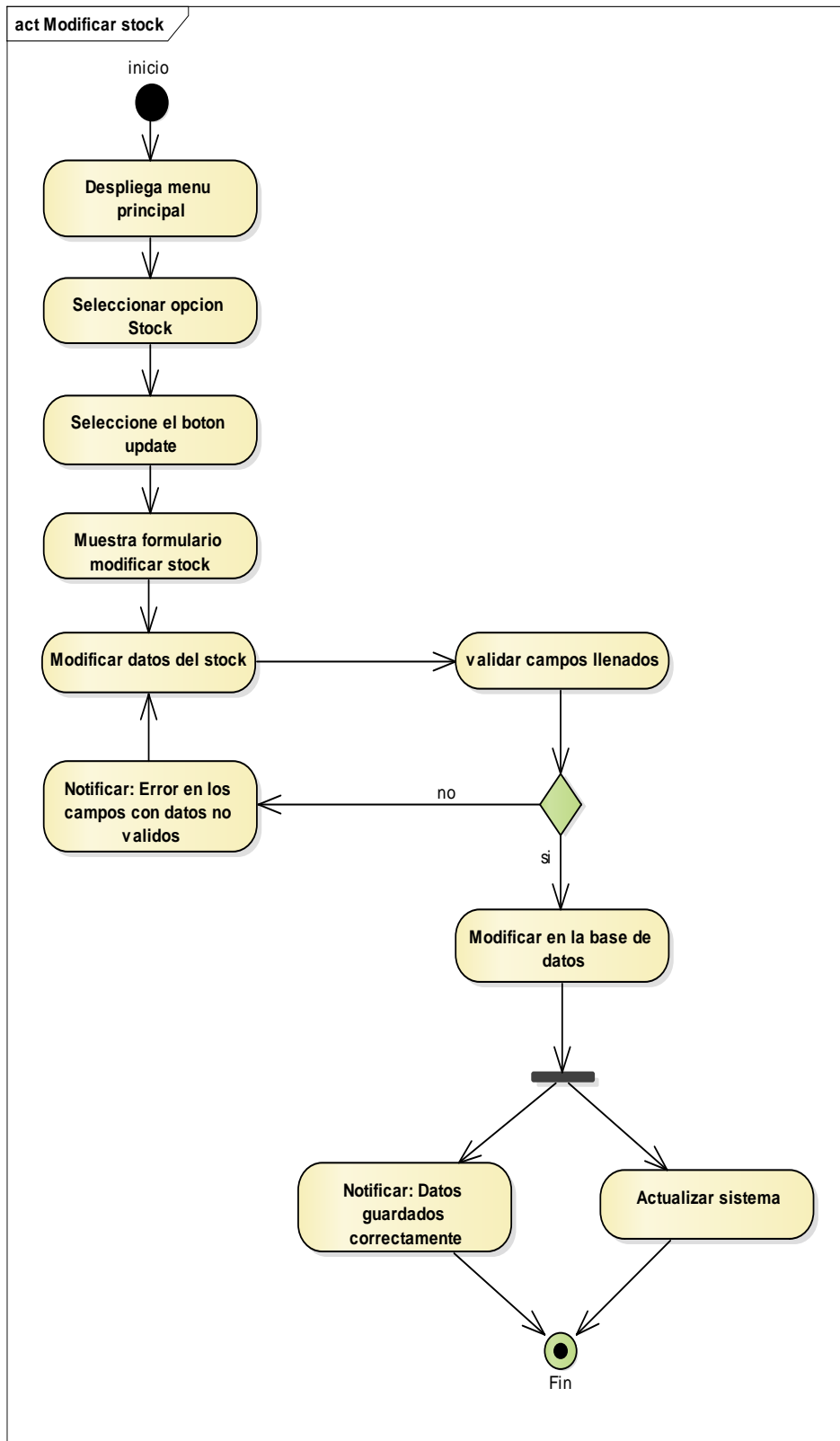
*Agregar Stock.*



## 2.13.20. Modificar Stock.

*Figura 61*

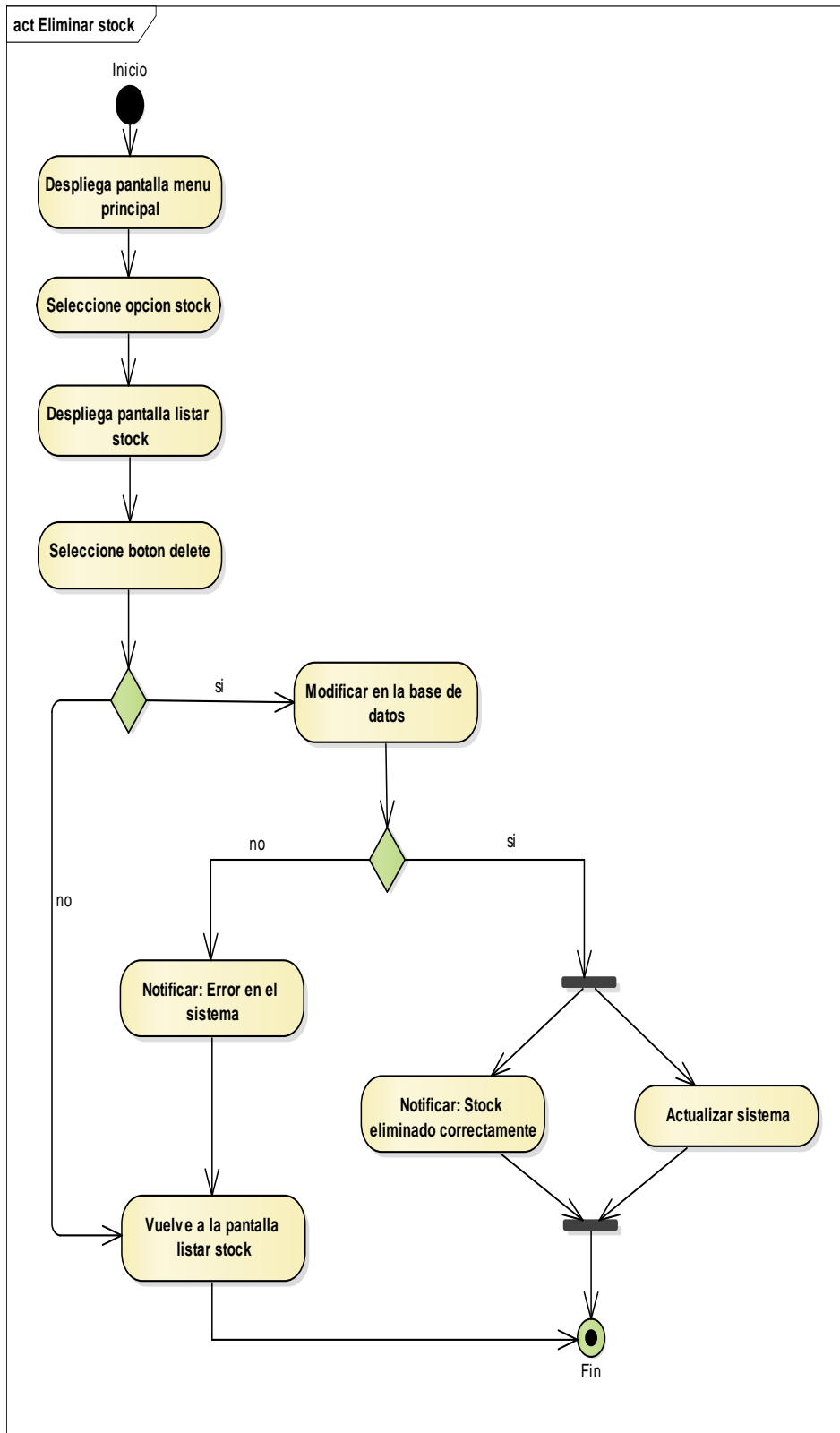
*Modificar Stock.*



### 2.13.21. Eliminar Stock.

*Figura 62*

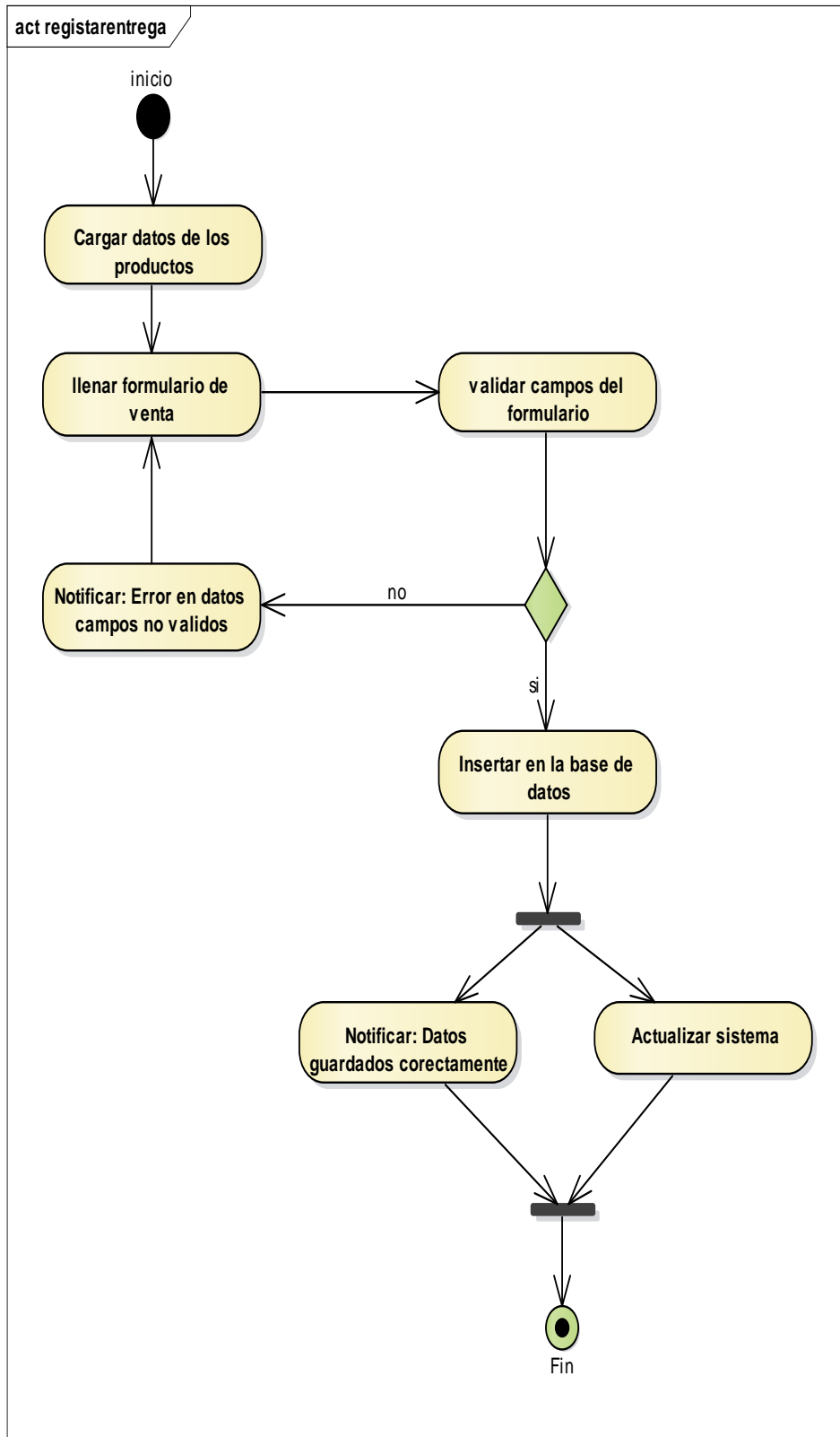
*Eliminar Stock.*



### 2.13.22. Registrar Entrega.

*Figura 63*

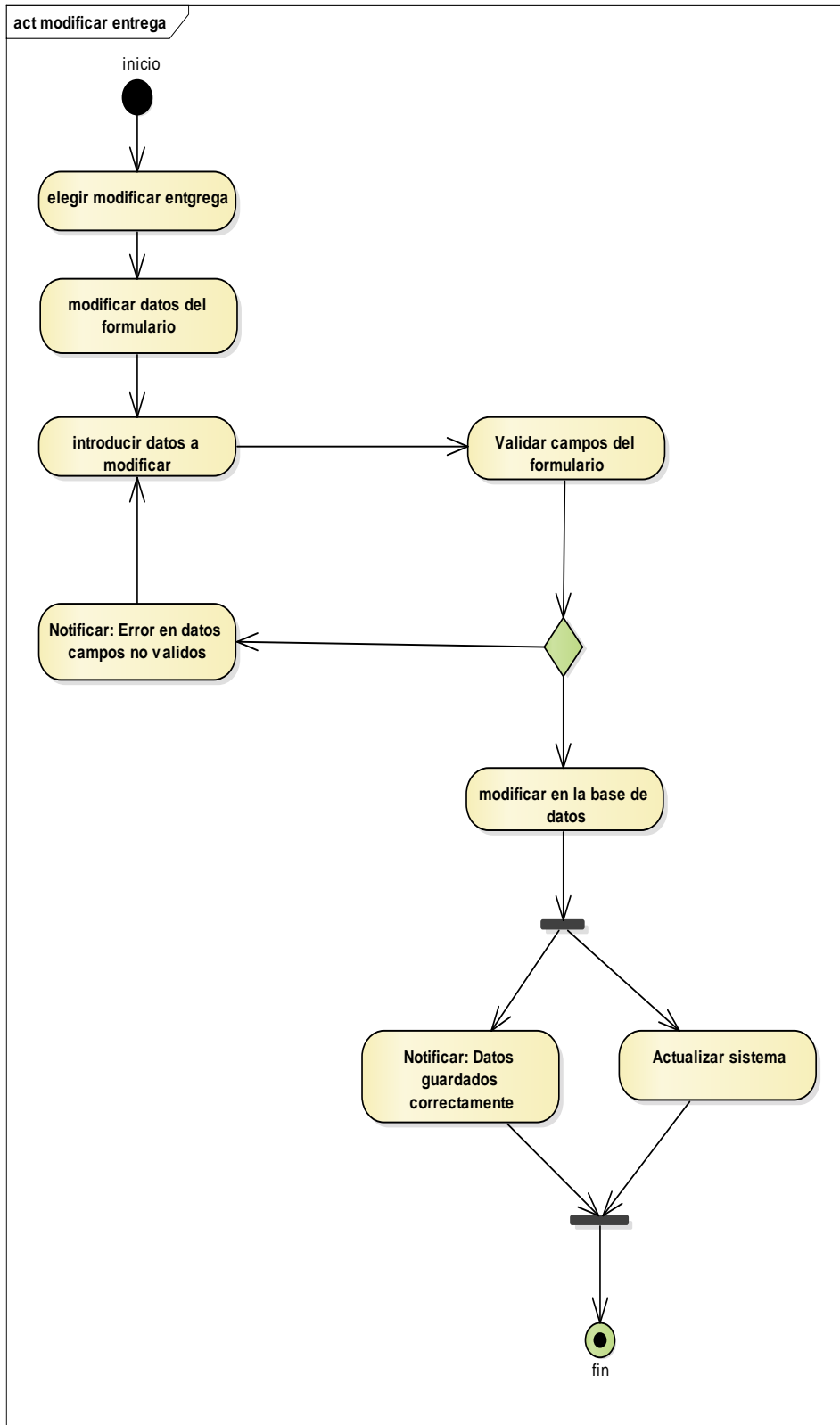
*Registrar Entrega.*



### 2.13.23. Modificar Entrega.

*Figura 64*

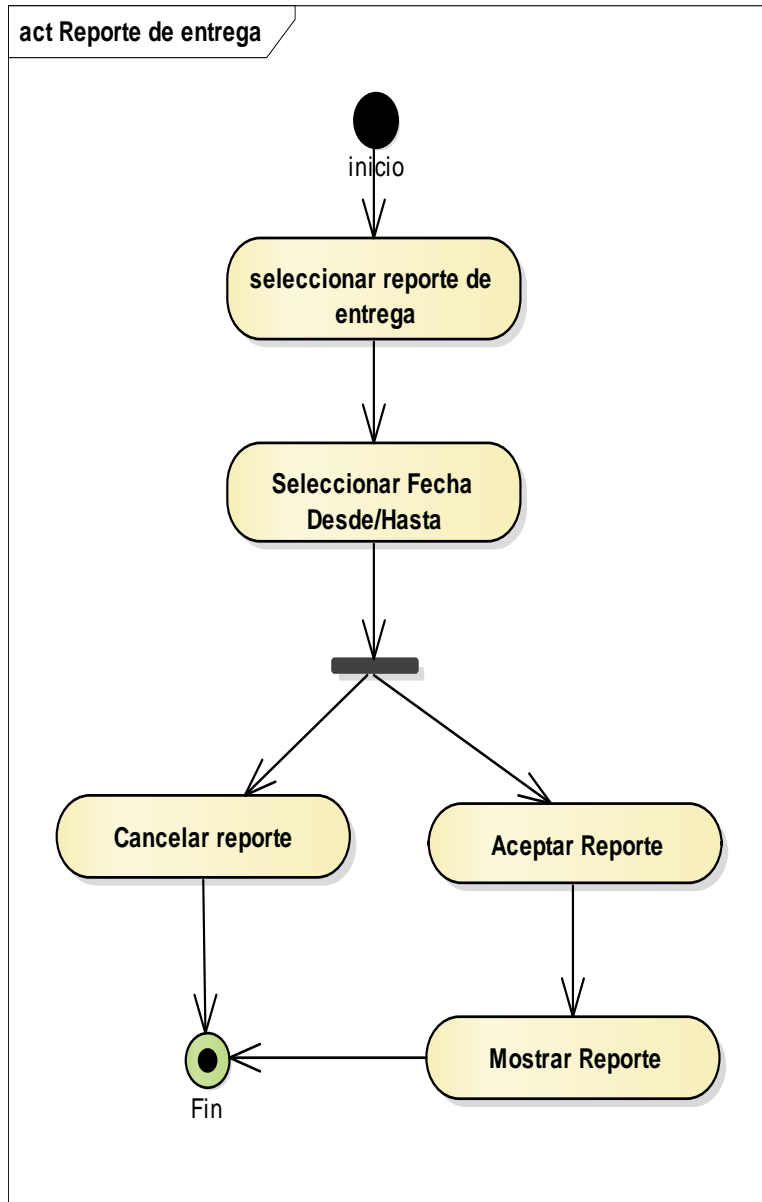
*Modificar Entrega.*



### 2.13.24. Reporte de Entrega.

*Figura 65*

*Reporte de Entrega.*



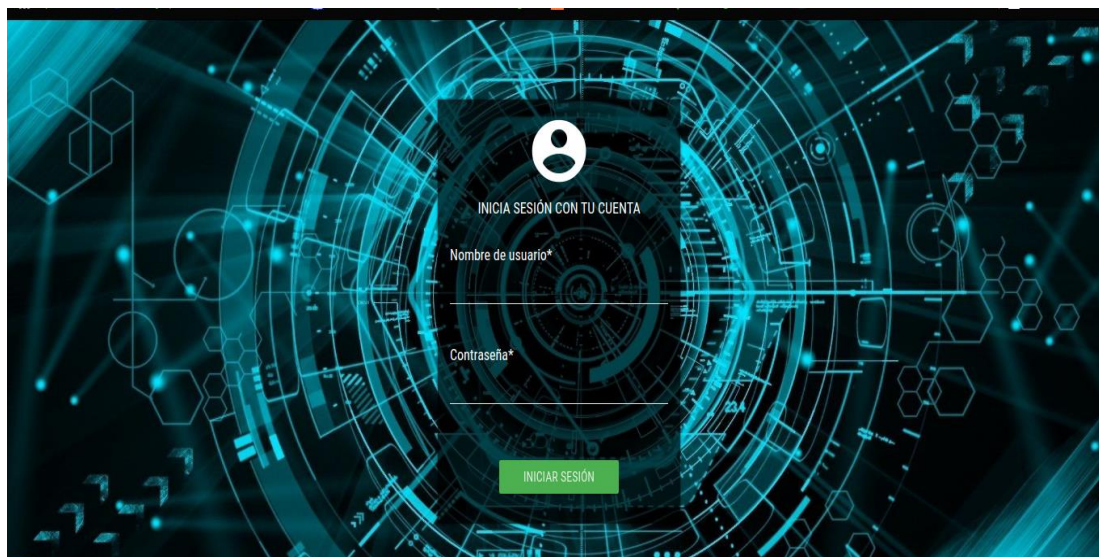
## 2.14. Interfaz de Usuario.

### 2.14.1 Pantalla Login.

*Figura 66*

#### *Pantalla Login.*

En esta pantalla el usuario debe introducir su usuario y contraseña para poder acceder al sistema, presionando el botón iniciar sesión.

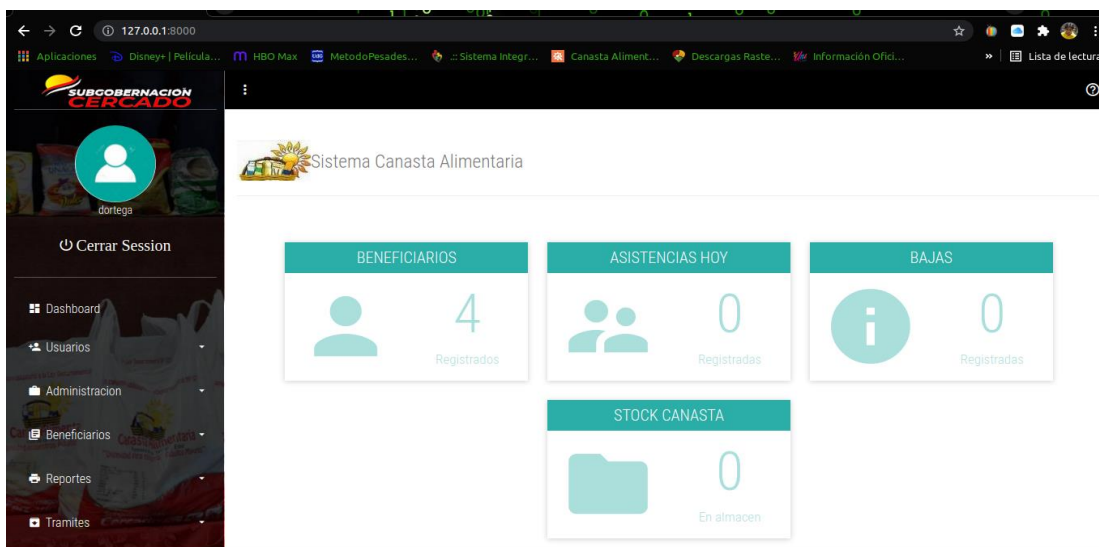


### 2.14.2. Pantalla Principal.

*Figura 67*

#### *Pantalla Principal.*

En esta pantalla los usuarios del sistema podrán elegir las diferentes opciones de manejo de sistema, cerrar sesión y observar el dashboard.





### 2.14.3. Pantalla Usuarios.

*Figura 68*

*Pantalla Usuarios.*

En esta pantalla se puede ver los filtros de búsqueda de los usuarios “Lista de Usuarios”, luego un listado de todas los Usuarios registrados en el sistema y se podrá su apellidos y nombres, un botón de agregar usuario, un botón para modificar sus datos, un botón para eliminar el usuario, un botón para habilitar ese usuario, y en la parte inferior un paginador.

The screenshot shows a web application interface for user management. The sidebar on the left contains the following menu items: Dashboard, Usuarios, Administración, Beneficiarios, Reportes, and Tramites. The main content area is titled 'Usuarios' and features a search bar with a 'Buscar' button. Below the search bar is a table with the following data:

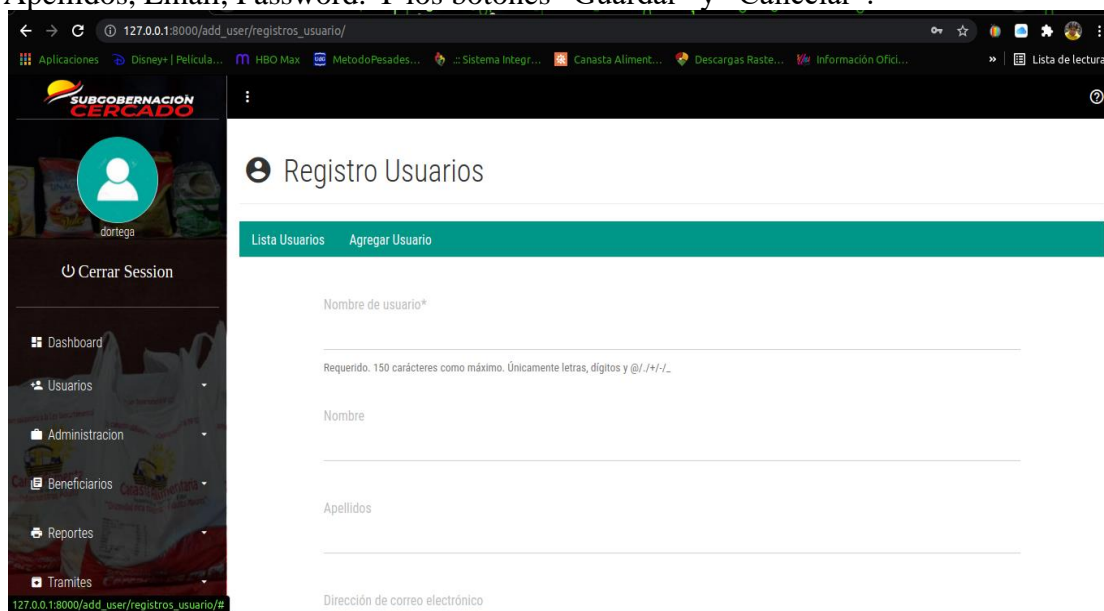
#	Nombre Usuario	Nombres	Apellidos	Email	Estado	Update	Delete
1	dortega	Daniel Fabian	Ortega Galean	danielortega782@gmail.com	ACTIVO		
2	brianbhl	BRIAN JHONNY	PEREZ GONZALES	30secondsxd@gmail.com	ACTIVO		
3	brianhl	BRIAN	HUMACATA	bhlbrianhumacata@gmail.com	ACTIVO		

## 2.14.4. Pantalla Agregar Usuarios.

*Figura 69*

### *Pantalla Agregar Usuarios.*

Este formulario nos mostrara los campos de nombre usuario, Nombres, Apellidos, Email, Password. Y los botones “Guardar” y “Cancelar”.



The screenshot shows a web browser window with the URL 127.0.0.1:8000/add\_user/registros\_usuario/. The page title is 'Registro Usuarios'. On the left is a dark sidebar with the 'SUBGOBERNACION CERCADO' logo, a user profile for 'dortega', and a menu with options: 'Cerrar Session', 'Dashboard', 'Usuarios', 'Administracion', 'Beneficiarios', 'Reportes', and 'Tramites'. The main content area has a green header with 'Lista Usuarios' and 'Agregar Usuario'. Below this are four input fields: 'Nombre de usuario\*' (with a note: 'Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @./+/-/\_'), 'Nombre', 'Apellidos', and 'Dirección de correo electrónico'.

## 2.14.5 Formulario Modificar Usuarios.

*Figura 70*

### *Formulario Modificar Usuarios.*

Este formulario nos mostrara los campos de nombre de usuario, Nombres, Apellidos, Email. Con los datos que están guardados en la tabla “user”. Y los botones “Modificar” y “Cancelar”.



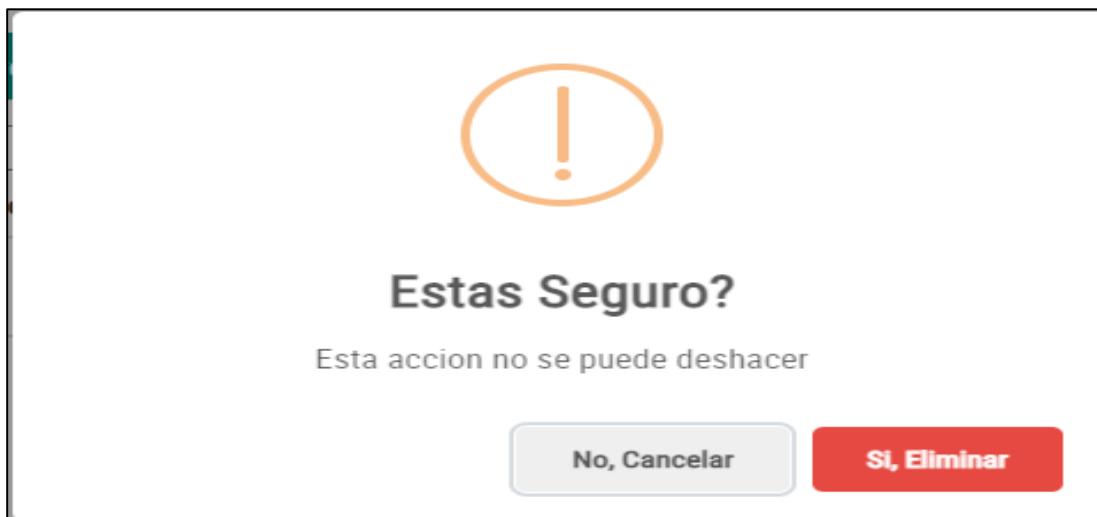
The screenshot shows the 'Modificar Usuarios' form. The sidebar is identical to the previous screenshot. The main content area has a green header with 'Modificar Usuarios'. Below this are four input fields with pre-filled data: 'Nombre de usuario\*' (value: 'dortega'), 'Nombre' (value: 'Daniel Fabian'), 'Apellidos' (value: 'Ortega Galean'), and 'Dirección de correo electrónico' (value: 'danielortega782@gmail.com').

#### 2.14.6. Eliminar Usuario.

*Figura 71*

*Eliminar Usuario.*

En este formulario se podrá Eliminar al usuario o no con los botones “Si” o “No”

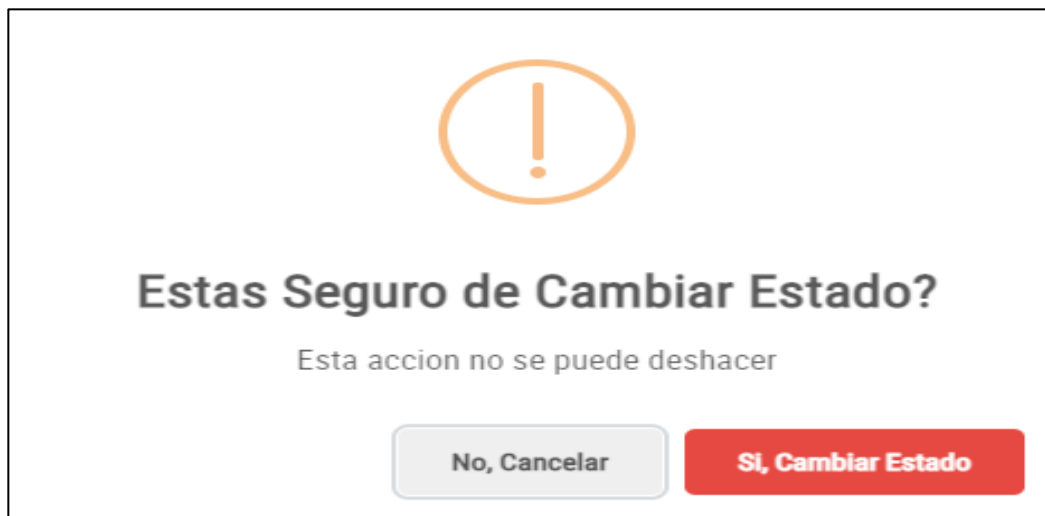


#### 2.14.7. Estado Usuario.

*Figura 72*

*Estado Usuario.*

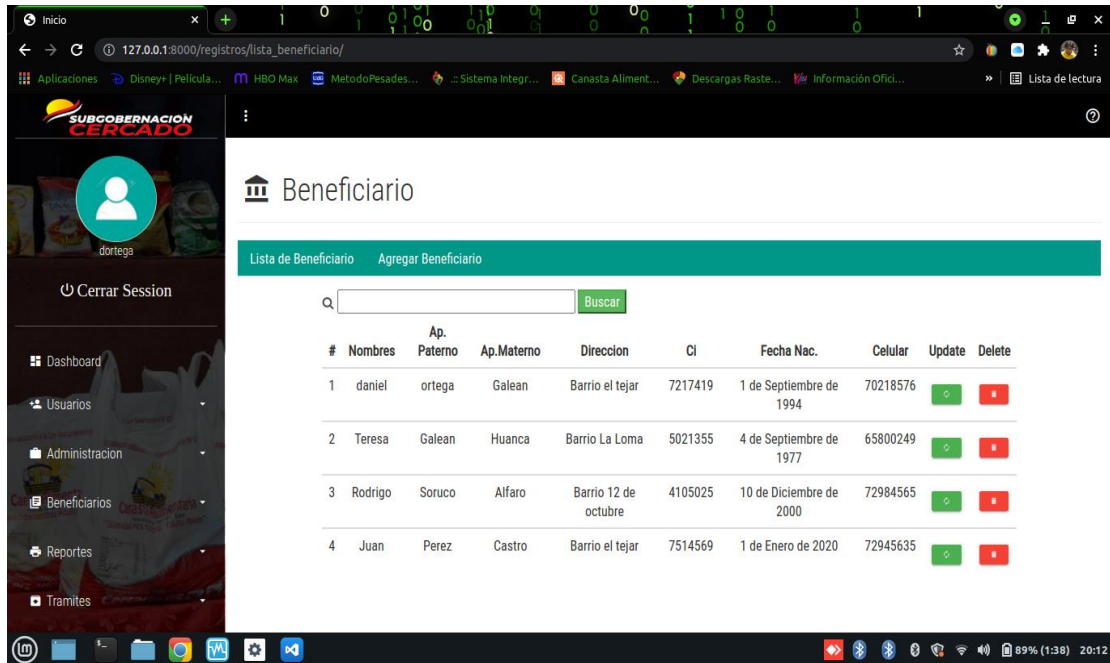
En este formulario se podrá Cambiar estado del Usuario o no con los botones “Si” o “No”



## 2.14.8. Pantalla Beneficiario.

*Figura 73*

*Pantalla Beneficiario.*



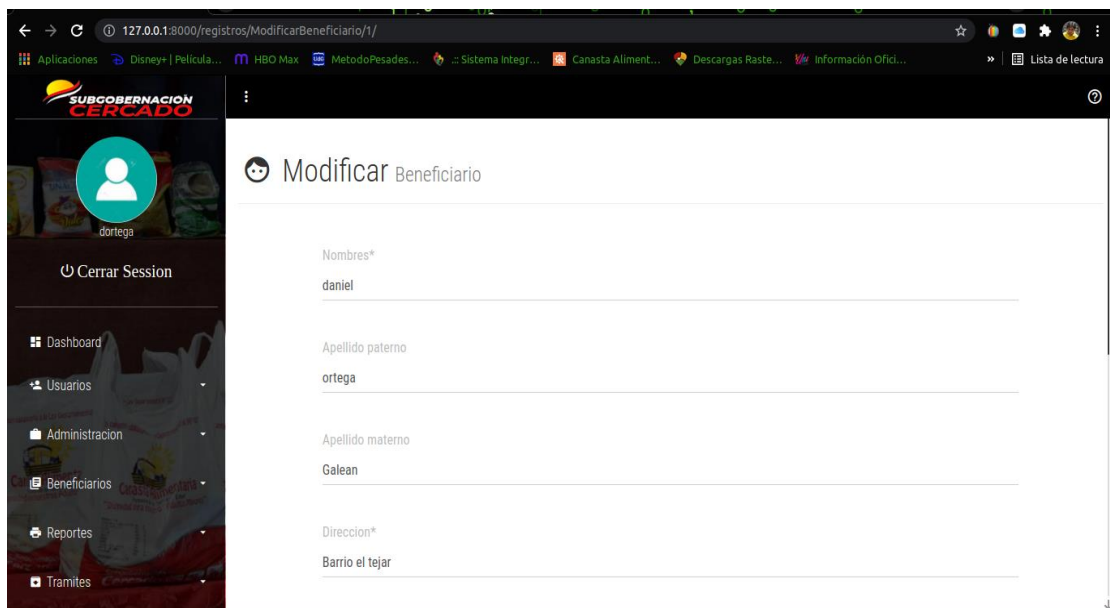
The screenshot displays the 'Beneficiario' page of a web application. The page features a sidebar on the left with a user profile for 'dortega' and a 'Cerrar Session' button. The main content area is titled 'Beneficiario' and includes a search bar and a 'Buscar' button. Below the search bar is a table with the following data:

#	Nombres	Ap. Paterno	Ap. Materno	Direccion	CI	Fecha Nac.	Celular	Update	Delete
1	daniel	ortega	Galean	Barrio el tejlar	7217419	1 de Septiembre de 1994	70218576		
2	Teresa	Galean	Huanca	Barrio La Loma	5021355	4 de Septiembre de 1977	65800249		
3	Rodrigo	Soruco	Alfaro	Barrio 12 de octubre	4105025	10 de Diciembre de 2000	72984565		
4	Juan	Perez	Castro	Barrio el tejlar	7514569	1 de Enero de 2020	72945635		

## 2.14.9. Pantalla Modificar Beneficiario.

*Figura 74*

*Pantalla Modificar Beneficiario.*



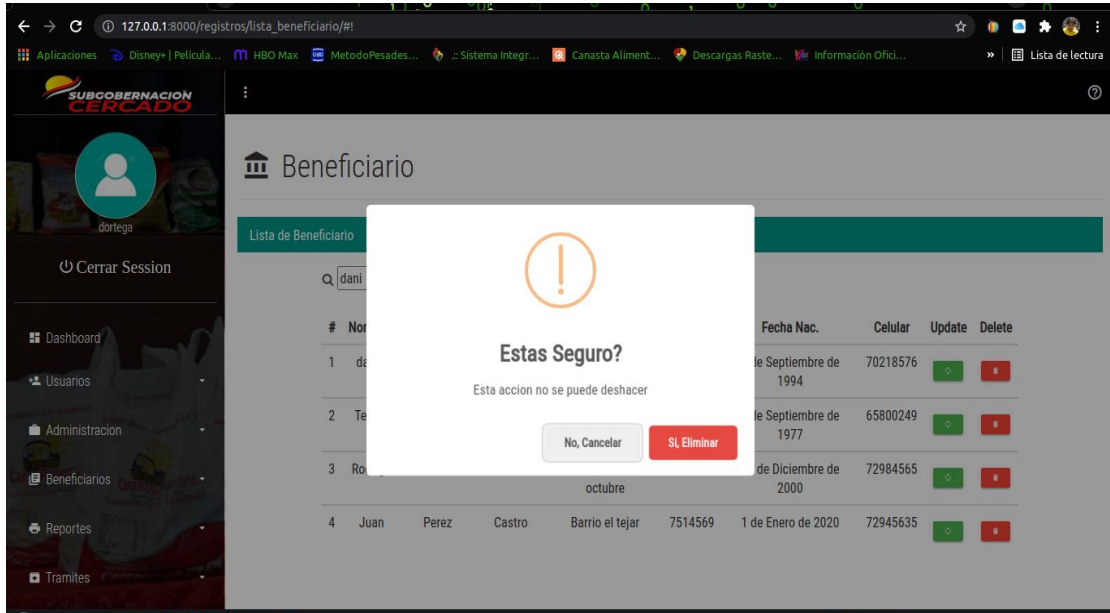
The screenshot displays the 'Modificar Beneficiario' page of a web application. The page features a sidebar on the left with a user profile for 'dortega' and a 'Cerrar Session' button. The main content area is titled 'Modificar Beneficiario' and includes a form with the following fields:

- Nombres\*: daniel
- Apellido paterno: ortega
- Apellido materno: Galean
- Direccion\*: Barrio el tejlar

## 2.14.10. Eliminar Beneficiario.

*Figura 75*

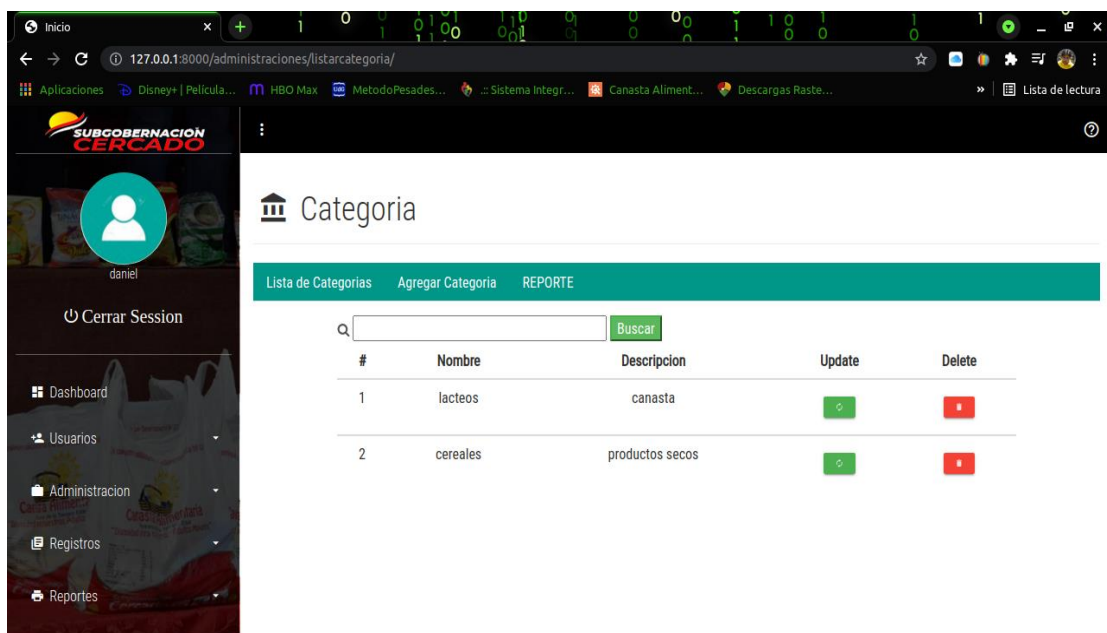
*Eliminar Beneficiario.*



## 2.14.11. Pantalla Categoría.

*Figura 76*

*Pantalla Categoría.*



## 2.14.12. Pantalla Producto.

*Figura 77*

*Pantalla Producto.*

Productos

Productos Agregar Producto REPORTE

Buscar

#	Producto	Precio	Stock	Categoria	Update	Delete
1	leche pil	50	100	lacteos		
2	cerelac	200	10	cereales		

## 2.14.13. Pantalla Área de Entrega.

*Figura 78*

*Pantalla Área de Entrega.*

Area de Entrega

Lista de Areas Agregar Area de Entrega Reporte

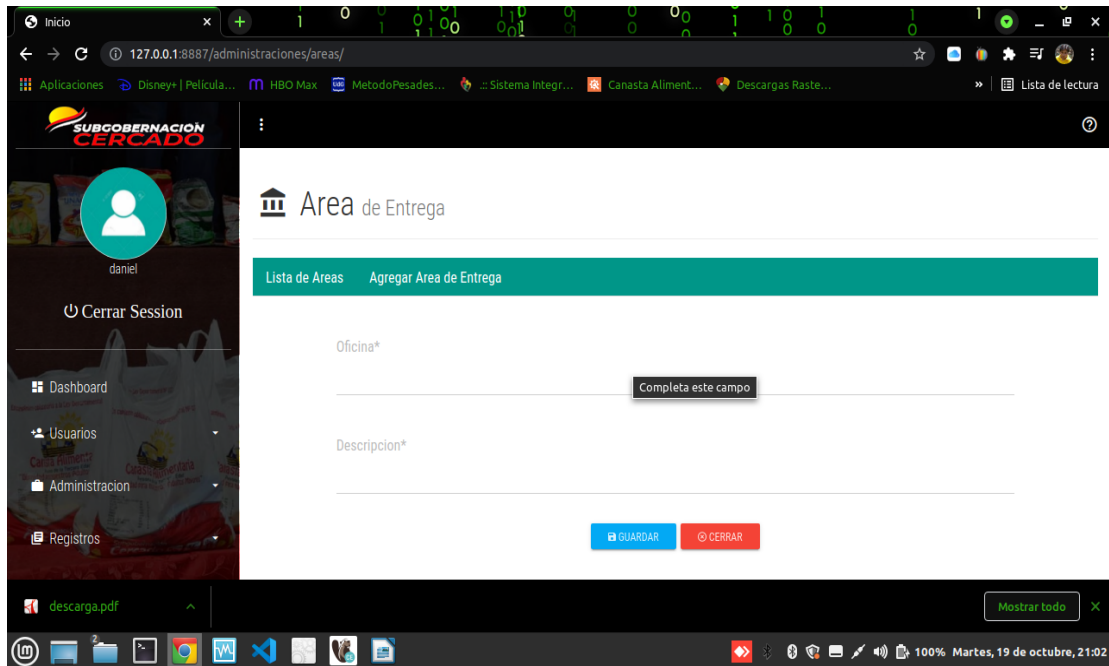
Buscar

#	Oficina	Descripcion	Update	Delete
1	central de adulto mayor	Oficina central avenida la paz		
2	central de adulto mayor de 60	mayores		

## 2.14.14. Agregar Área de Entrega.

*Figura 79*

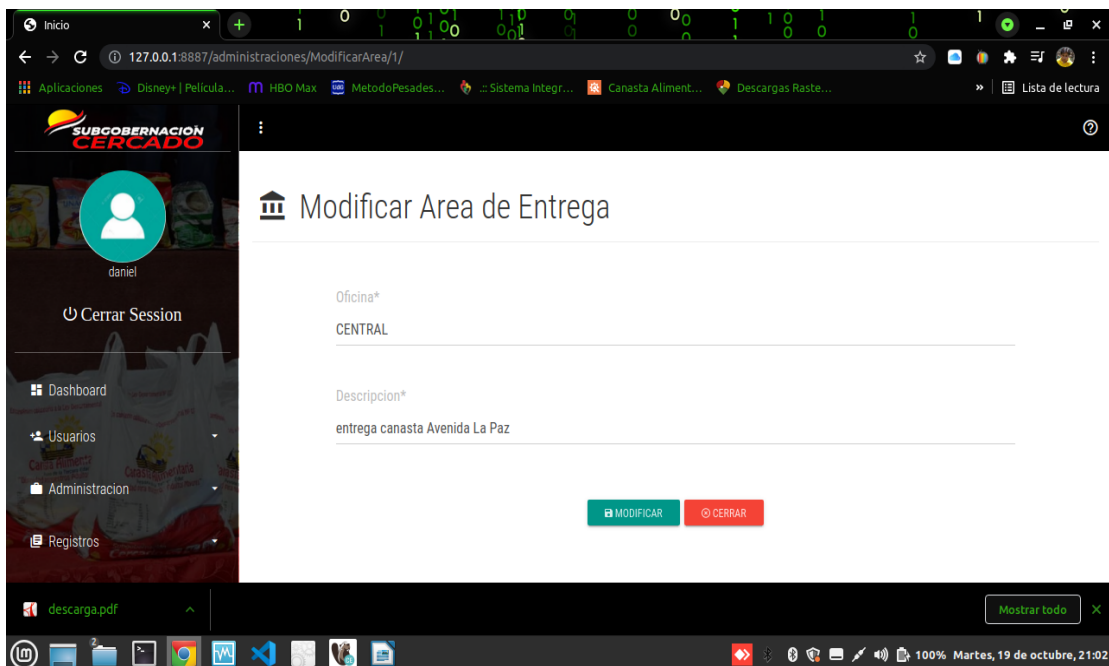
*Agregar Área de Entrega.*



## 2.14.15. Modificar Área de Entrega.

*Figura 80*

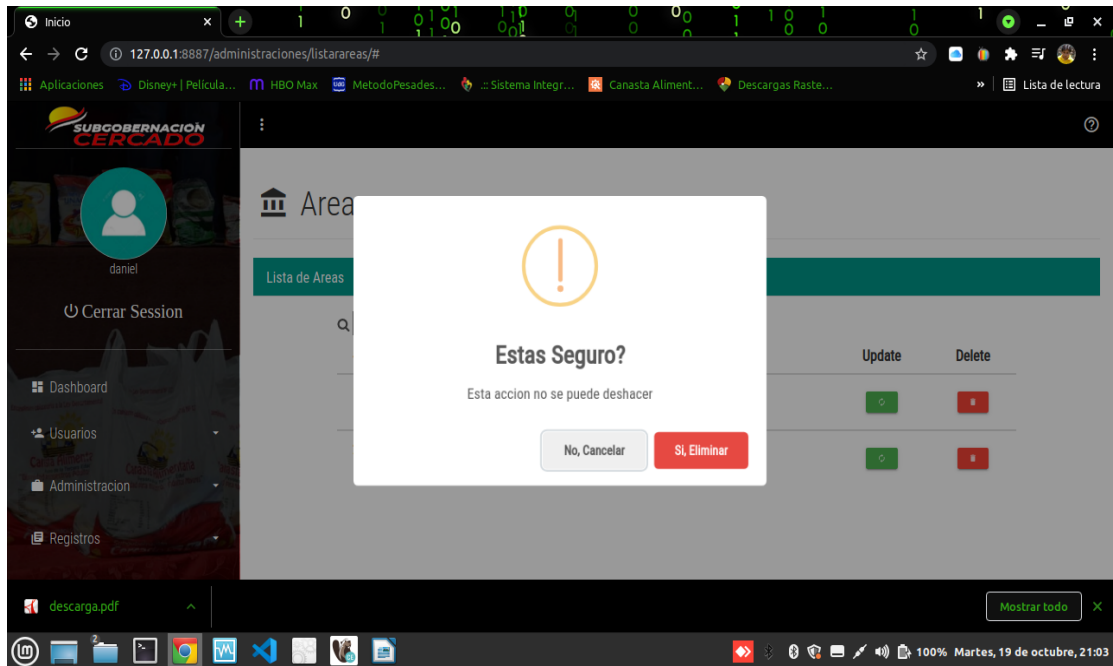
*Modificar Área de Entrega.*



## 2.14.16. Eliminar Área de Entrega.

*Figura 81*

*Eliminar Área de Entrega.*





## 2.14.17. Reporte de Asistencia.

### Figura 82

### Reporte de Asistencia.



## INFORME DE CONTROL DE ASISTENCIA

Fecha 19 de Octubre de 2021 a las 21:00  
Beneficiario Daniel Ortega Galean  
L. Entrega CENTRAL

#	Fecha de Presentacion	Fecha de Asistencia	Observacion
1	20 de Octubre de 2021	None	None
2	19 de Octubre de 2021	19 de Octubre de 2021 a las 16:41	None
3	18 de Octubre de 2021	18 de Octubre de 2021 a las 03:14	None
4	17 de Octubre de 2021	16 de Octubre de 2021 a las 21:13	None
5	16 de Octubre de 2021	15 de Octubre de 2021 a las 20:49	None
6	15 de Octubre de 2021	14 de Octubre de 2021 a las 20:45	None

Atte:  
Ingenieria Informatica - Taller III.

## 2.14.18. Pantalla Reconocimiento Facial

*Figura 83*

*Pantalla Reconocimiento Facial.*



## 2.14.19. Pantalla Mercado de Asistencia

*Figura 84*

*Pantalla Mercado de Asistencia.*



## 2.15. Pruebas de caja negra

Su objetivo es probar la funcionalidad del código intentan encontrar casos en los que el módulo no atiende a su especificación especialmente indicadas en los módulos que van a ser interfaz con el usuario.

Las pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa.

Los casos de prueba de caja negra pretenden demostrar que:

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

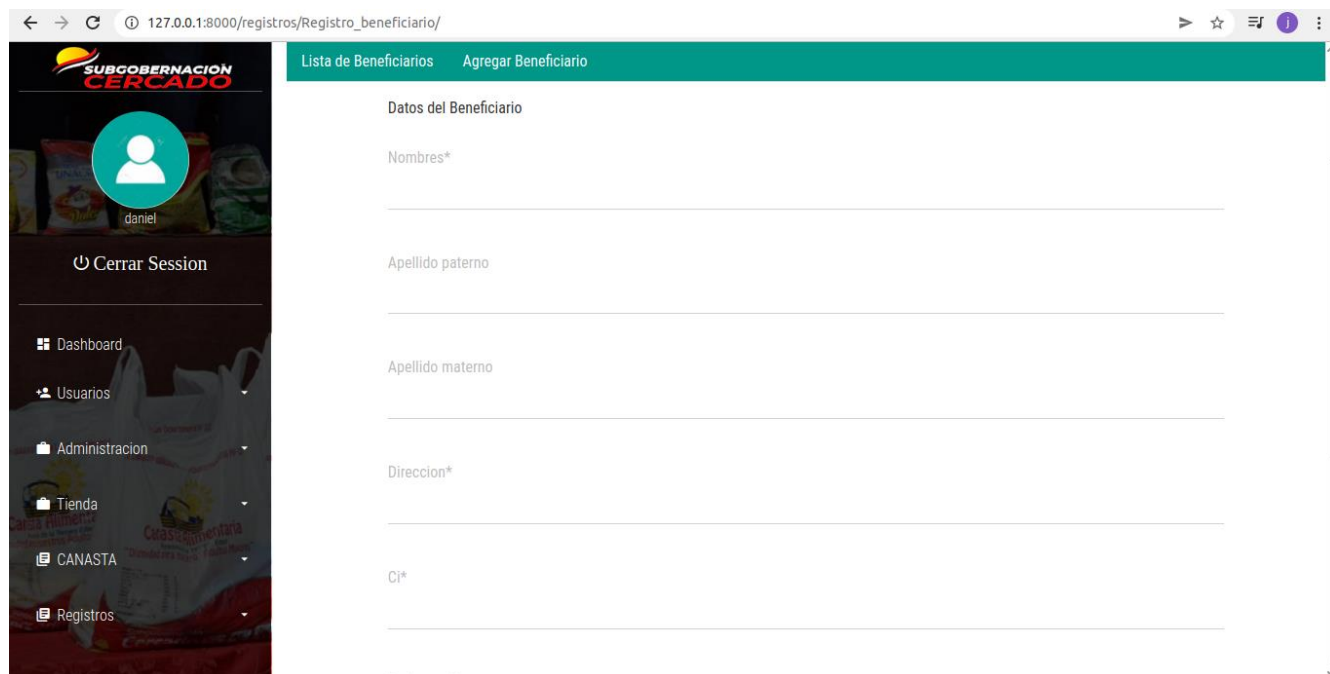
### Técnica Partición Equivalente

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de de entrada.

#### 2.15.1. Interfaz: Registro de Beneficiario

*Figura 85*

*Pantalla Registro Beneficiario.*



Se identifican las siguientes entradas:

**Datos del Beneficiario:**

**CI:** Número de 8 dígitos

**Nombre:** Letra, tamaño 50 como máximo

**Ap. Paterno:** Letra, tamaño 20 como máximo

**Am. Materno:** Letra, tamaño 20 como máximo

**Fecha Nacimiento:** Números, 10 como máximo y el carácter barra 2

**Teléfono:** Número, 10 máximo

**Dirección:** Caracteres Alfa numéricos

Celular : Número,10 máximo

<b>CONDICIÓN DE ENTRADA</b>	<b>CLASE EQUIVALENCIA VALIDA</b>	<b>CLASE EQUIVALENCIA NO VALIDA</b>
CI	1.- En blanco 2.- Si 3.- número 4.- $\geq 8$	5.- Cualquier cosa 6.- $< 6$ ó $> 9$
Nombre	7.- Si 8.- Letra 9.- $1 < \text{valor} \leq 50$	10.- No 11.- cualquier cosa 12.- $< 1$ ó $> 50$
Apellido paterno	13.- Si 14.- Letra 15.- $1 < \text{valor} \leq 20$	16.- No 17.- Cualquier cosa 18.- $< 1$ ó $> 20$
Apellido materno	19.- Si 20.- Letra 21.- $1 < \text{valor} \leq 20$	22.- No 23.-Cualquier Cosa 24.- $< 1$ ó $> 20$
Fecha nacimiento	25.- Si 26.- Números y carácter barra	29.- No 30.- Cualquier Cosa

	<b>27.-</b> Números $\leq 10$ <b>28.-</b> Carácter barra = 2	<b>31.-</b> Números $> 10$ y que no estén en el rango de los meses años días. <b>32.-</b> $2 < \text{Carácter barra} > 2$
Teléfono	<b>33.-</b> En blanco <b>34.-</b> Si <b>35.-</b> Números <b>36.-</b> Número $\leq 10$	<b>37.-</b> Cualquier cosa <b>38.-</b> Numero $> 10$
Dirección	<b>39.-</b> Si	<b>38.-</b> No

### Identificación de los Casos de Prueba que cubran uno o más clases de equivalencia

#### Clases Válidas

Clases de prueba:

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
5985215	Marcos	Armella	Duran	12/08/1961	6665897	Av. La Paz N° 547	Casado

Cubre las clases de equivalencia válidas: 1-2-3-6-7-8-12-13-14-18-19-20-24-25-26-27-32-33-34-35-38-39-40-44-45

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7217419	Juan	Arce	Martínez	12/08/1958	6632051	Av. Cochabamba N° 1489	Soltero

Cubre las clases de equivalencia válidas: 1-7-8-9-13-14-15-19-20-21-25-26-27-28-24-25-26-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
5796321	Isaura	Padilla	Escalante	01/03/1955		C. Litoral N° 489	Casada

Cubre las clases de equivalencia válidas: 2-3-4-7-8-9-13-14-15-19-20-21-25-26-27-28-33-40-44

### **Clases No Válidas**

Clases de prueba:

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
ytufjh	Isaura	Padilla	Escalante	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no válidas: 5-7-8-9-13-14-15-19-20-21-25-26-27-28-33-40-46

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7895674512	Isaura	Padilla	Escalante	01/03/1980	6645897	C. Litoral N° 489	545463

Cubre las clases de equivalencia no válidas: 6-7-8-9-13-14-15-19-20-21-25-26-27-28-33-40-47

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7956745		Padilla	Escalante	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no válidas: 2-3-4-10-13-14-15-19-20-21-25-26-27-28-33-40-46

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7956745	457%&7	Padilla	Escalante	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no válidas: 2-3-4-11-13-14-15-19-20-21-25-26-27-28-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7956745	Nertyuremnasdfertyuikk	Padilla	Escalante	01/03/1980	6645897	C. Litoral N° 489	

Clases de Prueba: 2-3-4-12-13-14-15-19-20-21-25-26-27-28-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	Carlos		Escalante	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no válidas: 1-7-8-9-16-25-26-27-28-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	Carlos	Tre145*%\$	Escalante	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no válidas: 1-7-8-9-17-25-26-27-28-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	Carlos	uturiengfdersadwqeryijhggf	Escalante	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no validas: 1-7-8-9-18-25-26-27-28-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	Carlos	Sanchez		01/03/1980	6645897	C. Litoral N° 489	44242

Cubre las clases de equivalencia no válidas: 1-7-8-9-22-25-26-27-28-33-40-47

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	Carlos	Figueroa	14556jfkd	01/03/1980	6645897	C. Litoral N° 489	

Cubre las clases de equivalencia no válidas: 1-7-8-9-23-25-26-27-28-33-40-46

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	Carlos		utryewqasdfcgthyuioplkn	01/03/1980		C. Litoral N° 489	

Cubre las clases de equivalencia no validas: 1-7-16-24-25-26-27-28-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres			C. Litoral N° 489	Soltero

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-29-33-40



CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres	12-12-1985		C. Litoral N° 489	Casado

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-30-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres	12/13/2		C. Litoral N° 489	divorciado

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-31-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres	05/06/2000/		C. Litoral N° 489	viudo

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-32-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres	05/06/2000	5Kdl7984g	C. Litoral N° 489	5414541

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-25-26-27-28-37-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres	05/06/2000	72985256784	C. Litoral N° 489	casado

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-25-26-27-28-38-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
7896523	Carlos	Figueroa	Cáceres	05/06/2000	6645898		viudo

Cubre las clases de equivalencia no validas: 2-7-8-15-19-20-21-25-26-27-28-34-35-36-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil

Cubre las clases de equivalencia no validas: 1-10-16-22-29-33-40-46

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
5897456	Carlos					C. Santa Cruz	

Cubre las clases de equivalencia no validas: 2-3-4-7-14-15-16-22-29-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
	José	Rivera	Torrez				

Cubre las clases de equivalencia no validas: 1-7-8-9-13-14-15-19-20-21-29-33-40

CI	Nombre	Ap. Paterno	Ap. Materno	Fecha nacimiento	Teléfono	Dirección	Estado civil
78ytert6	Jegr789	789uyt	Tyt65rt	478ewtu	7%&/yt7*	78/*42412	

Cubre las clases de equivalencia no validas: 5-11-17-23-30-37-39

## Interfaz: Nuevo Producto

*Figura 86*

*Pantalla Nuevo Producto.*

The screenshot shows a web browser window with the URL `127.0.0.1:8000/administraciones/producto/`. The page title is "Productos" and the sub-header is "Agregar Producto". The form contains the following fields:

- Producto\* (text input)
- Precio\* (text input)
- Fecha de vencimiento\* (date picker with format dd/mm/aaaa)
- Stock\* (text input)
- Categoria id\* (text input)

At the bottom of the form, there are two buttons: "GUARDAR" (blue) and "CERRAR" (red). The sidebar on the left includes a user profile for "daniel" and a "Cerrar Session" button. The sidebar menu items are: Dashboard, Usuarios, Administracion, Tienda, CANASTA, and Registros.

Se identifican las siguientes entradas:

**Producto** Letras como tamaño máximo de 22 caracteres

**Precio:** Números

**Fecha de Nacimiento:** Números, 10 como máximo y el carácter barra 2

**Categoría:** Letras como tamaño máximo de 22 caracteres

**Cantidad:** Numero

**Descripción:** Caracteres alfanuméricos

CONDICIÓN DE ENTRADA	CLASE EQUIVALENCIA VALIDA	CLASE EQUIVALENCIA NO VALIDA
----------------------	---------------------------	------------------------------

Producto	<b>1.- Si</b> <b>2.- Letra</b> <b>3.- <math>1 &lt; \text{valor} &gt; = 22</math></b>	<b>4.- No</b> <b>5.- Cualquier cosa</b> <b>6.- <math>&lt; 1 \text{ ó } &gt; 22</math></b>
Precio	<b>7.- Si</b> <b>8.- Numero</b>	<b>9.- No</b> <b>10.- cualquier cosa</b>
Fecha Vencimiento	<b>11.-si</b> <b>12.-Números <math>\leq 10</math></b> <b>13.-Carácter barra = 2</b>	<b>14.- Cualquier Cosa</b> <b>15.- Números <math>&gt; 10</math> y que no estén en el rango de los meses años días.</b> <b>16.- <math>2 &lt; \text{Carácter barra} &gt; 2</math></b>
Stock	<b>13.- Si</b> <b>14.- Número</b>	<b>15.- No</b> <b>16.- Cualquier cosa</b>
Categoría	<b>17.- Si</b> <b>18.- Letra</b> <b>19.- <math>1 &lt; \text{valor} &gt; = 22</math></b>	<b>20.- No</b> <b>21.- cualquier cosa</b> <b>22.- <math>&lt; 1 \text{ ó } &gt; 22</math></b>

### Clases Válidas

Clases de prueba:

Producto	Precio	Fecha vencimiento	Stock	Categoria
Leche	5	10/11/2021	50	Lacteos

Cubre las clases de equivalencia válidas: 1-2-3-7-8-11-12-13-17-18-21-22-23

Producto	Precio	Fecha vencimiento	Stock	Categoria

Arroz	7	25/12/2021	30	Abarrotes
-------	---	------------	----	-----------

Cubre las clases de equivalencia válidas: 1-2-3-7-8-11-12-13-17-18-21-22-23

### Clases No Validas

Clases de prueba:

Producto	Precio	Fecha vencimiento	Stock	Categoria
	hffh	25//05/2022	50	lacteos

Cubre las clases de equivalencia no válidas: 4-10-16

Producto	Precio	Fecha vencimiento	Stock	Categoria
5985215	svc	25082021	5j	dfgsdff

Cubre las clases de equivalencia no válidas: 6-10-16-20-25

Producto	Precio	Fecha vencimiento	Stock	Categoria

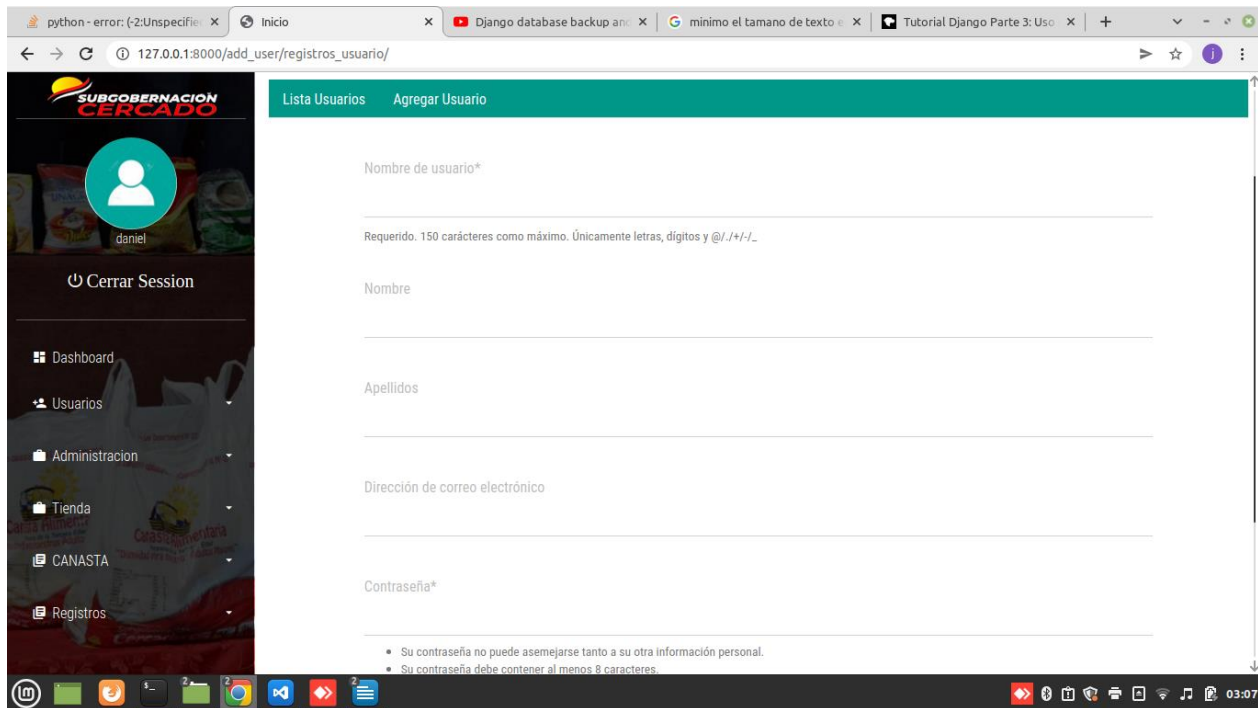
Cubre las clases de equivalencia no válidas: 4-9-19-24

Producto	Precio	Fecha vencimiento	Stock	Categoria
5985215	Marcos	Armella	Duran	12/08/1961

## Interfaz: Nuevo Usuario

*Figura 87*

*Pantalla Nuevo Usuario.*



Se identifican las siguientes entradas:

### Datos del Usuario

**Nombre:** Letra, tamaño 50 como máximo

**Ap. Paterno:** Letra, tamaño 20 como máximo

**Am. Materno:** Letra, tamaño 20 como máximo

**Correo Electronico:** Números,letras 10 como máximo y el carácter @

Contraseña:

<b>CONDICIÓN DE ENTRADA</b>	<b>CLASE EQUIVALENCIA VALIDA</b>	<b>CLASE EQUIVALENCIA NO VALIDA</b>
Nombre	1.- Si 2.- Letra 3.- $1 < \text{valor} \leq 50$	4.- No 5.- cualquier cosa 6.- $< 1 \text{ ó } > 50$
Apellido paterno	7.- Si 8.- Letra 9.- $1 < \text{valor} \leq 20$	10.- No 11.- Cualquier cosa 12.- $< 1 \text{ ó } > 20$
Apellido materno	13.- Si 14.- Letra 15.- $1 < \text{valor} \leq 20$	16.- No 17.-Cualquier Cosa 18.- $< 1 \text{ ó } > 20$
Correo Electrónico	19.- si 20.- $1 < \text{valor} \leq 20$ 21.- Letra,carácter @	22.- No 23.-Cualquier Cosa 24.- $< 1 \text{ ó } > 20$
Contraseña	25.- Si 26.-Numero,Letras 27.- Números $\leq 10$	28.- No 29.- Cualquier Cosa 30.- Números $> 10$

### **Identificación de los Casos de Prueba que cubran uno o más clases de equivalencia**

#### **Clases Válidas**

Clases de prueba:

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
daniel	Daniel	Ortega	Galean	Daniel782@gmail.com	142585Da

Cubre las clases de equivalencia válidas: 1-2-3-6-7-8-12-13-14-18-19-20-24-25-26-27-32-33-34-35-38-39-40-44-45

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Cubre las clases de equivalencia válidas: 1-7-8-9-13-14-15-19-20-21-25-26-27-28-24-25-26-40

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Cubre las clases de equivalencia válidas: 2-3-4-7-8-9-13-14-15-19-20-21-25-26-27-28-33-40-44

### **Clases No Validas**

Clases de prueba:

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Cubre las clases de equivalencia no válidas: 5-7-8-9-13-14-15-19-20-21-25-26-27-28-33-40-46

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Cubre las clases de equivalencia no válidas: 6-7-8-9-13-14-15-19-20-21-25-26-27-28-33-40-47



Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Cubre las clases de equivalencia no válidas: 2-3-4-10-13-14-15-19-20-21-25-26-27-28-33-40-46

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Cubre las clases de equivalencia no válidas: 2-3-4-11-13-14-15-19-20-21-25-26-27-28-33-40

Nombre usuario	Nombre	Ap. Paterno	Ap. Materno	Correo electronico	contraseña
5985215	Marcos	Armella	Duran	12/08/1961	6665897

Clases de Prueba: 2-3-4-12-13-14-15-19-20-21-25-26-27-28-33-40

## **2.16. Programa de capacitación para el personal administrativo en el uso y manipulación del Software, Mejora del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC.**

### **2.16.1. Introducción**

La determinada capacitación y sistema de información están cambiando la forma de trabajo de las empresas. Luego de realizar visitas y entrevistas al personal de informática, se pudo percibir que no cuentan con la información necesaria, pero poseen una gran motivación por automatizar sus registros y aprender el correcto manejo del mismo. Los Sistemas de Información ayudan a acelerar los procesos, por tanto, las organizaciones que los implantan logran ventajas competitivas al adoptarlos en sus funciones. La capacitación se refiere a las metodologías que se usan para enseñar a las personas dentro de las instituciones, las habilidades necesarias que necesitan para realizar su trabajo de una manera más eficiente, esto contempla desde pequeños cursos que le permitan al usuario entender el funcionamiento básico del sistema nuevo hasta capacitaciones más profundas y avanzadas a base de prácticas y material didáctico como libros, exámenes, etc.

Es muy importante la realización de este componente para que el personal de Informática obtenga el conocimiento suficiente sobre las características que ofrece el sistema. Para que luego ellos sepan la correcta manipulación del sistema evitando problemas posteriores.

### **2.16.2. Contexto**

La capacitación se desarrollará en un paso:

Se desarrollará una capacitación global del sistema mostrando al usuario de manera general lo que contiene el sistema y el alcance que se puede lograr con cada módulo, todo esto para que el usuario vea el beneficio del Sistema Informático.

### **2.16.3. Propuesta Pedagógica**

La propuesta pedagógica a utilizar dadas las características de los usuarios del sistema, los cuales cuentan con una carrera profesional, tendrá en cuenta sus particularidades individuales debido a que el personal al que se impartirá la capacitación son adultos jóvenes, que tienen conocimiento avanzado en el uso de páginas web, la propuesta pedagógica que se tomó en cuenta es la metodología de instrucción directa.

## 2.16.4. Programa de Capacitación

**Tabla**

Programa de capacitación

N.º	CONTENIDO	OBJETIVO	DURACIÓN	MATERIAL DIDÁCTICO	MEDIOS DE ENSEÑANZA - APRENDIZAJE	DESTINATARIO
1	Lección 1.- Ingresar al Sistema	Que el usuario se identifique y pueda entrar al sistema.	10 min	Manual de Usuario y diapositivas	Computadora, data display	Administradores, Operador
2	Lección 2.- Presentación global de la navegación por el sistema	Que el usuario pueda conocer y practicar como las opciones que presenta el sistema.	15 min	Manual de Usuario y diapositivas	Computadora, data display	Administradores,
3	Lección 3.- Registro de Usuarios,	Que el usuario pueda añadir, modificar, eliminar y Habilitar a usuarios,	20 min	Manual de Usuario y diapositivas	Computadora, data display	Administradores
4	Lección 4.- Registro de Beneficiarios y planificaciones	Que el usuario pueda añadir, modificar, eliminar y habilitar Beneficiarios, y planificaciones	20 min	Manual de Usuario y diapositivas	Computadora, data display	Administradores
5	Lección 5.- Ver Reportes	Que el usuario pueda generar los distintos	10 min	Manual de Usuario y diapositivas	Computadora, data display	Administradores

		reportes que ofrece el sistema				
--	--	--------------------------------------	--	--	--	--

Fuente Propia

## 2.17. Medios de Verificación

### 2.17.1. Carta de cumplimiento

Tarija, 6 de diciembre del 2021

**Señores:**

**Presente**

A quien corresponda:

Mediante la presente se certifica que el proyecto de grado "MEJORAMIENTO DEL PROCESO DE REGISTRO Y DISTRIBUCION, DE LA CANASTA ALIMENTARIA DEL ADULTO MAYOR EN LA SUBGOBERNACION DE CERCADO, INCORPORANDO LAS TIC", por parte del estudiante DANIEL FABIAN ORTEGA GALEAN de la carrera de Ingeniería Informática, cumplió con los requerimientos pactados al inicio del mismo, realizando satisfactoriamente las funcionalidades acordadas.

Se expide la presente a solicitud verbal del interesado para los fines que estime convenientemente.



Lic. Carlos Antonio Knez Ponce

**RESPONSABLE DEL PROGRAMA**

Lic. Odt. Carlos A. Knez Ponce  
RESPONSABLE DEL PROYECTO DE INVESTIGACIÓN PRESENTADA  
SUBGOBERNACIÓN CERCADO  
Gobierno Autónomo Deptal. de Tarija

## 2.17.2. Carta de confirmación de capacitación

Tarija, 6 de diciembre del 2021

**Señores:**

**Presente**

A quien corresponda:

Mediante la presente hago constatar que el estudiante universitario del último año Daniel Fabian Ortega Galean, realizó la capacitación correspondiente de grado "MEJORAMIENTO DEL PROCESO DE REGISTRO Y DISTRIBUCION, DE LA CANASTA ALIMENTARIA DEL ADULTO MAYOR EN LA SUBGOBERNACION DE CERCADO, INCORPORANDO LAS TIC".

Se expide la presente a solicitud verbal del interesado para los fines que estime conveniente.



Lic. Carlos Antonio Knez Ponce

**RESPONSABLE DEL PROGRAMA**

Lic. Odf. Carlos A. Knez Ponce  
RESPONSABLE DEL PROYECTO DE INVESTIGACIÓN PREVENCIÓN  
SUBGOBERNACIÓN CERCADO  
Gobierno Autónomo Deptal. de Tarija

### 2.17.3. Lista de participantes

#### **LISTA DE ASISTENCIA A LA CAPACITACIÓN DEL PROYECTO DE “MEJORAMIENTO DEL PROCESO DE REGISTRO Y DISTRIBUCION, DE LA CANASTA ALIMENTARIA DEL ADULTO MAYOR EN LA SUBGOBERNACION DE CERCADO, INCORPORANDO LAS TIC”.**

La presente lista avalada y registra la presencia a la capacitación virtual mediante zoom, del “mejoramiento del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernacion de cercado, incorporando las tic”, por parte del personal de la Subgobernacion de cercado que realizo el seguimiento del proyecto.

**Nombre:** Lic. Carlos Antonio Knez Ponce

**Cargo:** Responsable del Proyecto

**Firma:**

*Lic. Odt. Carlos A. Knez Ponce*  
RESPONSABLE DEL PROYECTO DE INVESTIGACION PREVENCIÓN  
SUBGOBERNACIÓN CERCADO  
Gobierno Autónomo Dptal. de Tarija

**Nombre:** Lic. Malena Martinez Coro

**Cargo:** Atencion al beneficiario

**Firma:**

**Nombre:** Rodrigo Soruco Alfaro

**Cargo:** Técnico de Caja

**Firma:**

**Nombre:** Gilda Vargas Fernandez

**Cargo:** Técnico de Caja

**Firma:**

## 2.17.4. Fotografías de la capacitación

Figura 88

Capacitacion.

Sistema para Canasta Alimentaria

BENEFICIARIOS	ASISTENCIAS HOY	ENTREGAS
2 Registrados	0 Registradas	0

FALTANTES	CANASTAS DISPONIBLES
2 Por venir	0 Por Entregar

Participants in the video call: Daniel Ortega, Martinez Malena, Gilda Vargas, Rodrigo Soruco.

Agregar Beneficiario

Rellenar todos los campos sin excepcion de uno

Lista de Beneficiarios | **Agregar Beneficiario**

Datos del Beneficiario

Nombres\*

Apellido paterno

Apellido materno

Participants in the video call: Daniel Ortega, Martinez Malena, Gilda Vargas, Rodrigo Soruco.



127.0.0.1:8000/report/report/1

Aplicaciones Disney+ Pelicula... HBO Max MetodoPesades... Sistema Integr... Canasta Aliment... Descargas Raste... Lista de lectura

Example 1 1 / 1 100%

1

### INFORME DE CANASTA ENTREGADAS

Fecha 7 de Diciembre de 2021 a las 01:48

#	Beneficiario	Barrio	Canasta	Asistio
1	Juan Carlos Galean Huanca	la loma	CANASTA ENERO	5 de Diciembre de 2021 a las 20:33

Atte:  
Ingenieria Informatica - Taller III.

Daniel Ortega

Martinez Malena

Martinez Malena

Gilda Vargas

Gilda Vargas

Rodrigo Soruco

Rodrigo Soruco

## **CAPÍTULO III**

### **3.1 Conclusiones**

Con el desarrollo del presente proyecto se logró cumplir con el objetivo del proyecto Mejora del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC, para el logro de los siguientes componentes.

-Se logró mejorar el proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC con el cual se logró automatizar los procesos que se realizaba en forma manual demandando mucho tiempo y propensa a errores no voluntarios como son de registro, actualización de datos personales del beneficiario, gestión de usuario, productos, categoría, entrega, generar reportes, que será de gran ayuda para la toma de decisiones.

-Con el sistema se logró facilitar el acceso a la información a través de la agilización de sus procesos. Al concluir el presente proyecto se debe decir que al cumplir con sus objetivos pudo aportarse de manera significativa en la difusión de las TIC dentro de la sociedad adulta mayor beneficiaria, sobre todo se logró mejorar la gestión de información que genera el programa canasta alimentaria de la subgobernación de Cercado Tarija.

Para el logro de los resultados ha sido de vital importancia la participación y colaboración del personal administrativo del programa canasta alimentaria de la subgobernación de Cercado Tarija brindando la información oportuna.

### **3.2 Recomendaciones**

Al concluir el proyecto se recomienda tener en cuenta los siguientes aspectos:

-Las autoridades administrativas del programa canasta alimentaria deben fomentar y fortalecer el uso de las TIC mediante talleres de capacitación para poder contar con un personal más efectivo.

-Se recomienda el uso del Sistema Mejora del proceso de registro y distribución, de la canasta alimentaria del adulto mayor en la Subgobernación de Cercado, incorporando las TIC, de esta manera los usuarios cuenten con un medio que le permita la agilidad en sus procesos, teniendo información actualizada y de manera oportuna.

-Se recomienda documentar en lo posible las políticas de seguridad a implementar y comunicar a todo el personal involucrado incluyendo los beneficios y riesgos relacionados con los recursos y bienes y sus elementos de seguridad.