

I. Capítulo I El Proyecto

I.1. Identificación del Proyecto

I.1.1. Título del Proyecto

Mejoramiento de la gestión de ventas en la imprenta “CORPOGRÁFIKA LTDA.”, con plataforma web.

I.1.2. Carrera

Ingeniería Informática

I.1.3. Facultad

Facultad de Ciencias y Tecnología

I.1.4. Duración del Proyecto

La realización del Proyecto tendrá una duración de 12 meses.

I.1.5. Área/Línea de investigación priorizada

Desarrollo de sistemas de gestión web.

I.1.6. Entidades Asociadas

I.1.6.1. Personal Vinculado al Proyecto

Director de Proyecto

Rodriguez	Balderrama	Jakeline	
Apellido Paterno	Apellido Materno	Nombre	
Estudiante	Ingeniería Informática	Ciencias y Tecnología	
Profesión	Carrera ó Unidad	Facultad	

Tabla 1. Director de Proyecto

I.1.6.2. Participantes equipo de trabajo

Categoría	Nombres y Apellidos	Profesión	Firma
Analista de Sistemas	JakelineRodriguez Balderrama	Estudiante	
Diseñador	JakelineRodriguez Balderrama	Estudiante	
Programador	JakelineRodriguez Balderrama	Estudiante	

Tabla 2. Participantes equipo de trabajo

I.1.6.3. Actividades previstas para los integrantes del equipo de investigación

Director:

- Planificación de actividades
- Establecimiento de las metodologías para el desarrollo del software
- Control y seguimiento del cumplimiento de las actividades planificadas.

Analista de Sistemas:

- Análisis de requerimientos

Diseñador:

- Diseño de software

Programador:

- Programación

I.2.Descripción del Proyecto

I.2.1. Resumen Ejecutivo del Proyecto

El propósito del proyecto es el mejoramiento de la gestión de ventas de la imprenta CORPOGRÁFIKA LTDA.

Los componentes del proyecto son:

- Sistema web de Administración de Ventas para la imprenta CORPOGRÁFIKA LTDA., desarrollado.
- Capacitación del uso del sistema web, implementado.

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

La imprenta CORPOGRÁFIKA LTDA., se dedica a la impresión de afiches, libros, revistas, etc., actualmente el manejo de la información para el área de ventas es realizado de forma manual junto a la utilización de planillas de Excel.

El actual manejo de la información no es rápido y oportuno sobre todo cuando se requiere realizar búsquedas de información pasada, esto conlleva mayor tiempo y esfuerzo, lo que constituye un aspecto negativo, ya que las empresas generalmente buscan minimizar estos factores para alcanzar mayores estándares de calidad y competitividad.

Para realizar una venta, previamente se debe cotizar el precio de un producto, dependiendo el tipo de producto se toman en cuenta varios aspectos como: la cantidad de hojas, el tipo de papel, el tamaño (alto, ancho) del producto, entre otras especificaciones.

La imprenta no cuenta con la generación de reportes, los cuales son de gran ayuda en toda empresa para saber de forma rápida las ventas generadas ya sea semanal o mensualmente.

En la actualidad, muchas empresas, ya sean grandes o pequeñas, utilizan las TIC no sólo para automatizar sus procesos administrativos, de producción, etc; sino también para expandir sus áreas de oferta de productos y/o servicios a nivel local, nacional e internacional, prueba de ello es la vasta cantidad de tiendas online que se puede encontrar navegando en la internet.

I.2.2. Descripción y Fundamentación del Proyecto

Un inadecuado almacenamiento de la información, es un obstáculo para un manejo eficiente de la información ya que se puede incurrir en la pérdida de información, para dar solución a este problema se debe registrar toda la información en un medio seguro para así poder clasificarla y tener acceso a la misma en el momento oportuno, para el logro de esto se procederá a guardar la información del sistema en una Base de Datos.

Poco aprovechamiento de las TIC para mejorar el tratamiento de la información, ya que se cuenta con equipos necesarios para la utilización de las TIC, sin embargo, éstos no son suficientemente aprovechados, con la capacitación necesaria, se dará a conocer las facilidades que ofrecen las TIC para el mejoramiento de cualquier empresa.

Los errores en el registro de información, toda información registrada de forma manual tiende a escribirse con errores, el sistema informático reducirá la cantidad de información registrada manualmente ya que una vez registrado un dato en particular, se podrá acceder a él tan sólo seleccionándolo, evitando así, introducirlo manualmente cada vez que se lo requiera.

Dificultad para realizar reportes de compras y ventas, realizar informes o reportes mensuales, anuales o similares, puede ser resultar una tarea algo desagradable para quien lo realice, ya que es muy lenta, pues se requiere acceder a varios datos almacenados en distintos lugares, y unir la información para realizar un reporte.

El acceso y tratamiento de la información es lento e inoportuno, ocasionando mayor tiempo y esfuerzo para el personal de la imprenta, con el desarrollo de un sistema informático se busca acelerar estos procesos.

Inadecuada utilización de tiempo y esfuerzo en los procesos de la gestión de ventas de la imprenta CORPOGRAFIKA LTDA, toda empresa tiene entre sus fines la productividad y competitividad, la mala utilización del tiempo, puede restar estos dos factores que son muy importantes ya que puede significar pérdidas y/o mal, por lo tanto, la conclusión del proyecto pretende mejorar esta situación a largo plazo.

I.2.2.1. Situación planteada Con y Sin Proyecto

Situación sin proyecto	Situación con proyecto
Inadecuado almacenamiento de la información. Poco aprovechamiento de las TIC para mejorar el tratamiento de la información del área de ventas de la imprenta “CORPOGRÁFIKA” LTDA. Errores en el registro de información. La búsqueda de información es demorosa. Dificultad para realizar reportes de compras y ventas. Inadecuada atención a clientes de la imprenta CORPOGRAFIKA LTDA para el personal de la imprenta.	Elaboración del Manual de Usuario. Implementación de la capacitación del sistema. Capacitación del uso del sistema web, implementado. Gestión de ventas en la imprenta “CORPOGRÁFIKA LTDA.”, mejorada. Adecuada atención a clientes de la imprenta CORPOGRAFIKA LTDA.

Tabla 3. Situación planteada Con y Sin Proyecto

I.2.3. Objetivos

I.2.3.1. Objetivo General

Mejorar la gestión de ventas de la imprenta “CORPOGRÁFIKA LTDA.”

I.2.3.2. Objetivos Específicos

- Desarrollo de un Sistema web, que otorgue confiabilidad y agilice los procesos en la Administración de Ventas.
- Capacitación a los usuarios en el uso del sistema, para el correcto manejo y buen aprovechamiento del sistema desarrollado.

I.2.4. Metodología de Trabajo

I.2.4.1. Desarrollo e implementación del Sistema.

El objetivo principal de este componente es desarrollar un sistema seguro, confiable y eficiente, el sistema tendrá mínimamente las siguientes funcionalidades:

- Administrar ventas
- Administrar compras
- Reportes

La metodología a utilizar para el desarrollo del software es RUP (Rational Unified Process), donde sólo se procederá a cumplir las tres primeras fases que marca dicha metodología.

RUP es un proceso de desarrollo que se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes.

El flujo de trabajo contempla los siguientes pasos:

Requerimientos: Trasladando las necesidades del negocio a un sistema automatizado.

Análisis y Diseño: Trasladando los requerimientos dentro de la arquitectura de software.

Programación e Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

I.2.4.2. Capacitación a los usuarios para el uso del sistema

Se realizarán 10 días de capacitación para el uso del sistema, donde cada clase tendrán una duración de 1 hora, dicha capacitación está dirigida al gerente de la imprenta, el objetivo de la capacitación es que el gerente tengan el conocimiento del funcionamiento del sistema, las clases serán prácticas.

I.2.4.3. Cronograma de Actividades

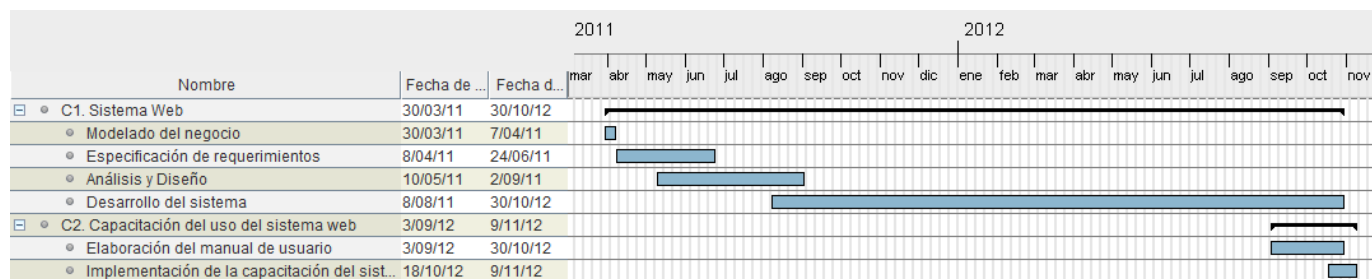


Tabla 4: Cronograma de Actividades

I.2.5. Marco Lógico del proyecto

I.2.5.1. Cuadro de Involucrados

GRUPOS	INTERESES	PROBLEMAS PERCIBIDOS	RECURSOS Y MANDATOS
Gerente General	Aumentar la competitividad de la empresa. Tener acceso a la información de forma oportuna.	- Dificultad y lentitud en listar reportes que permitan ver las compras y ventas de la empresa. - El acceso a la búsqueda de información es lenta e inoportuna. - El tratamiento de la información es poco confiable. - Poco aprovechamiento de los equipos de computación de la empresa.	- R. Aprueba el desarrollo del sistema. - M. Tomar la decisión final de utilizar o no el sistema una vez terminado el mismo.
Gerente de Ventas	Reducir el tiempo de realización de ventas.	- El control de ventas es lento y poco confiable. - El registro de la información es propensa a errores.	- R. Aprueba el desarrollo del sistema.

Personal de Almacenes	- Control de inventarios confiable.	- Control de inventarios poco confiable.	
-----------------------------	-------------------------------------	--	--

Tabla 5. Cuadro de Involucrados

I.2.5.2. Análisis de Problemas

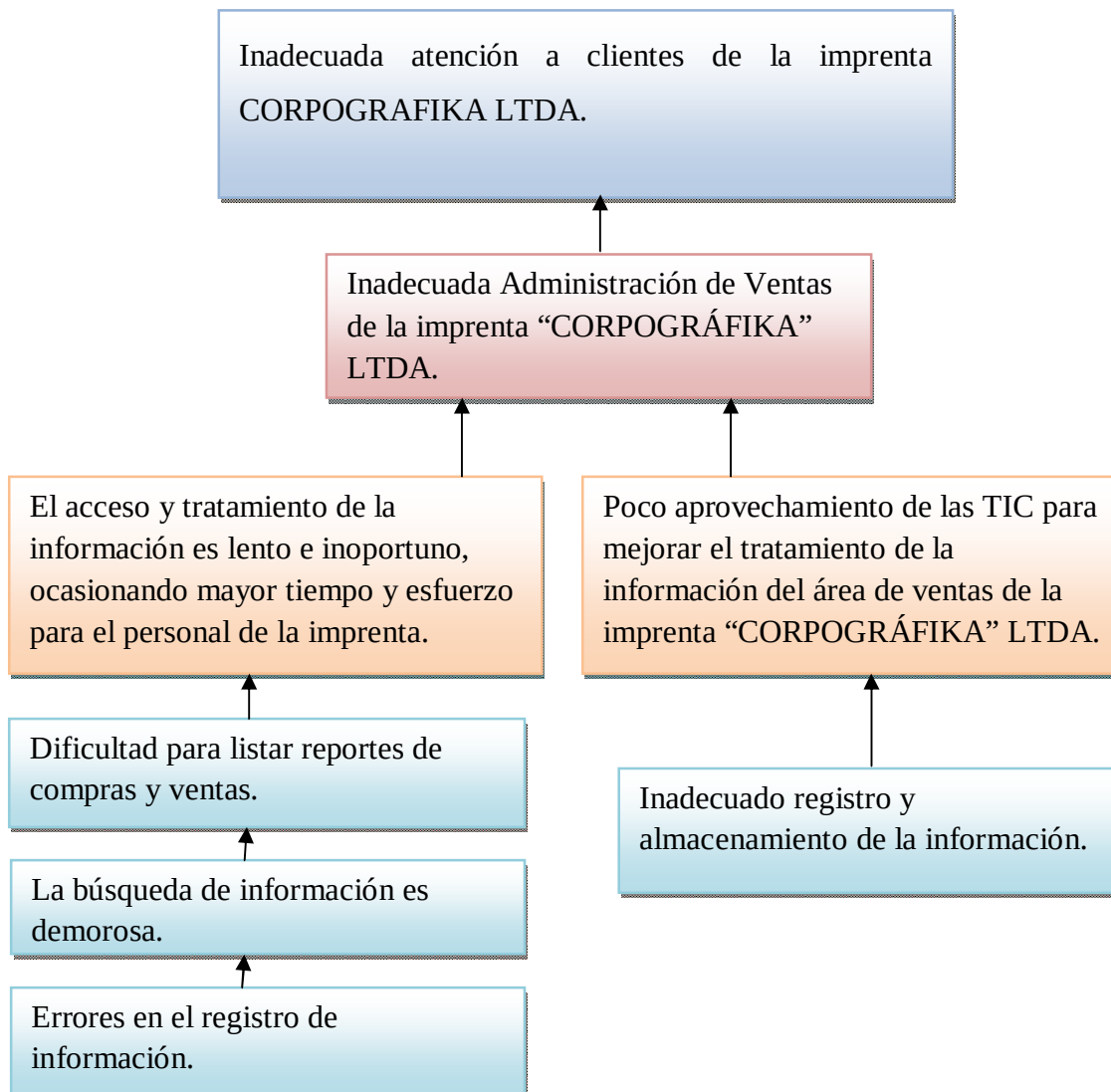


Figura 1. Árbol de Problemas

I.2.5.3. Análisis de Objetivos

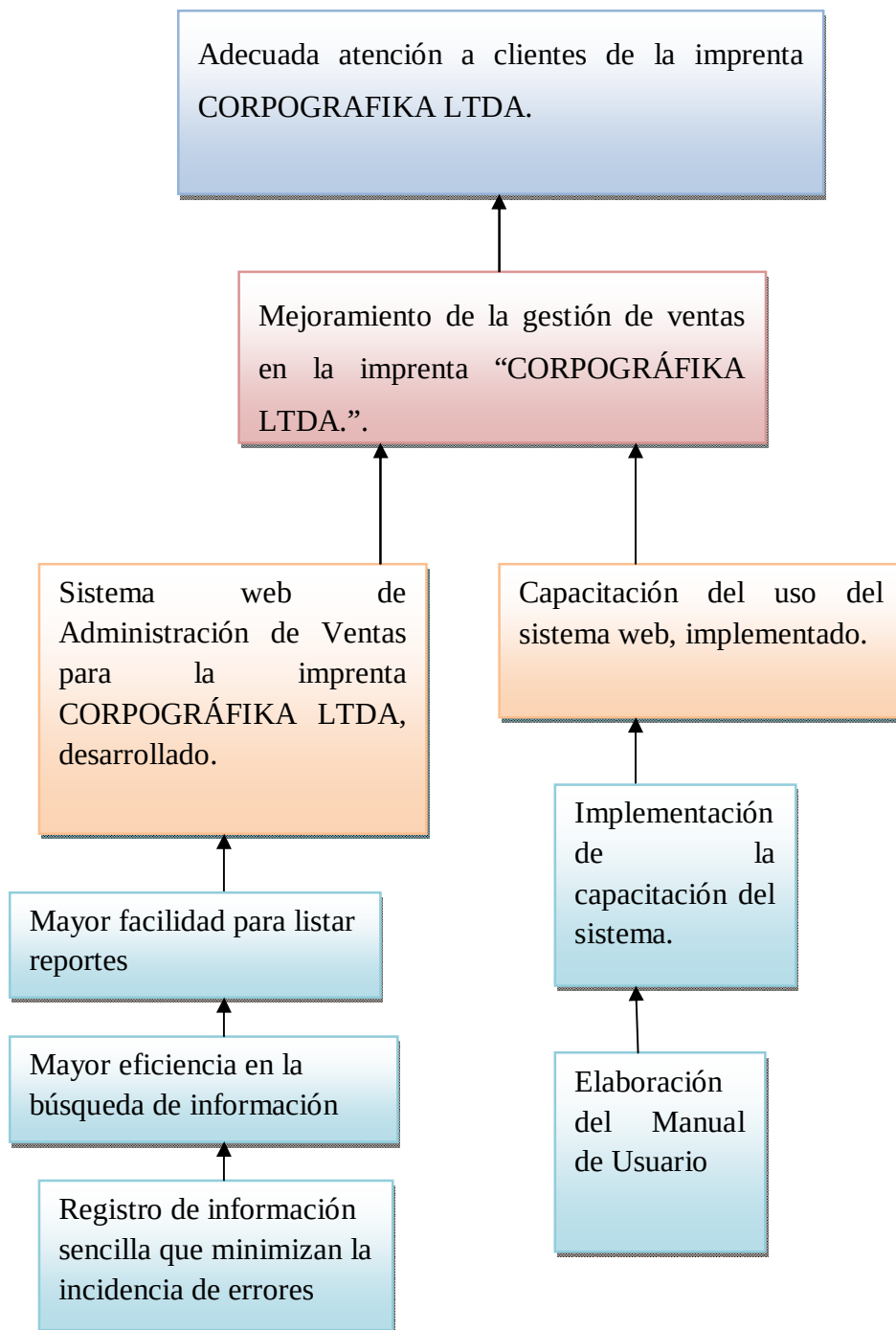


Figura 2. Árbol de Objetivos

I.2.5.4. Matriz de Marco Lógico

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir al mejoramiento de la atención a los clientes de la imprenta CORPOGRAFIKA LTDA.	A dos años de finalizado el proyecto, el número de clientes que expresan satisfacción por la atención recibida en la imprenta se incrementa al menos en un 50%.	Encuestas realizadas a los clientes de la imprenta CORPOGRAFIKA LTDA. con relación a la atención recibida durante el año base y a los dos años de finalizado el proyecto realizados por la gerencia de la empresa.	La empresa cuenta con los recursos económicos necesarios para garantizar la sostenibilidad del proyecto.
Objetivo General (Propósito) Gestión de ventas en la imprenta "CORPOGRÁFIKA	Al finalizar el proyecto, al menos un 70% de los procesos relacionados con la gestión de ventas han sido automatizados, mejorando el tratamiento de la información en el área de ventas de	Cuadro comparativo entre los procesos que involucra la gestión de ventas de la imprenta CORPOGRÁFIKA LTDA, y el total de	- La política de la empresa no afecta el desarrollo del proyecto. - Se poseen los medios y recursos económicos

LTDA.”, con plataforma web, mejorada.	la imprenta. Categoría: EFICACIA Expresión matemática <i>(Procesos automatizados/Total Procesos)x100</i>	procesos automatizados, avalado por la gerencia.	necesarios para el desarrollo del proyecto.
Objetivos Específicos (Componentes) 1. Sistema web de Administración de Ventas para la imprenta CORPOGRÁFIKA LTDA desarrollado.	1.1 Al finalizar el proyecto, se ha desarrollado un sistema web para la gestión de ventas de la imprenta CORPOGRÁFIKA LTDA, basado en los requerimientos bajo la norma IEEE830.	1.1.1. Cuestionarios y entrevistas realizados al gerente de la imprenta CORPOGRÁFIKA LTDA. 1.1.2 Documentación presentada a los docentes de la materia de Taller III (componente I de la presente documentación). 1.1.3. Carta firmada por el gerente de la imprenta CORPOGRÁFIKA LTDA,	• Usuarios brindan información necesaria para el desarrollo del proyecto. • Se cuenta con los recursos económicos necesarios para llevar a cabo el desarrollo de los componentes del proyecto • Usuarios participan en las capacitaciones.

2. Capacitación del uso del sistema web implementado.	2.1 Entre octubre y noviembre de 2012, el 50% de los usuarios involucrados con el sistema fueron capacitados para el uso del mismo. Categoría: EFICACIA	expresando su conformidad con el desarrollo del sistema. 2.1.1 Certificados de participación al personal capacitado en el uso del sistema. 2.1.2 Manual de usuario.	
Actividades C1. Sistema web 1.1 Modelado del negocio 1.2 Especificación de	<i>Servicios no personales</i> 720 <i>Materiales y suministros</i> 400 Total 1120 <i>Ver detalle en tabla Presupuesto</i>	Documentación del sistema web (componente 1 de la presente documentación)	El personal de la imprenta brinda la información necesaria para el desarrollo del sistema.

requerimientos 1.3 Análisis y Diseño 1.4 Desarrollo del Sistema C2. Capacitación del sistema web. 2.1 Elaboración del manual de usuario. 2.2. Implementación de la capacitación del sistema		• Manual de usuario impreso. • Documentación de la capacitación (componente 2 de la presente documentación)	• Se cuenta con los recursos necesarios para la realización de las actividades del proyecto • Los usuarios finales participan en las capacitaciones.
--	--	--	---

Tabla 6. Matriz de Marco Lógico

I.3.Presupuesto / Justificación

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
20000	SERVICIOS NO PERSONALES			
	21000. Servicios Básicos		350	350
	22000. Servicios de transporte		70	70
	22000. Servicio de internet		300	300
	Sub total rubro		720	720
30000	MATERIALES Y SUMINISTROS			
	32000. Productos de Papel, Cartón e Impresos		300	300
	39000. Productos Varios.		100	100
	Sub total rubro		400	400
	TOTAL		1120	1120
	TOTAL + 40% Incentivo		1568	1568

Tabla 7. Presupuesto

II. COMPONENTES

II.1.COMPONENTE I “Sistema web para la administración de ventas desarrollado.”.

II.1.1. Plan de desarrollo de Software

II.1.1.1. Introducción

El Plan de Desarrollo es el documento que proporciona la visión global del enfoque de desarrollo que propone el proyecto, este documento es elaborado e incluido como la versión final del proyecto de la asignatura de Taller III del Programa de Informática de la Universidad Autónoma Juan Misael Saracho.

El proyecto está basado en la metodología RUP, en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología.

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 artefactos o entregables, que serán generados. Este documento es a su vez uno de los artefactos de RUP.

II.1.1.1.1. Propósito

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

Los usuarios del Plan de Desarrollo del software son:

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

II.1.1.1.2. Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Sistema Informático de Gestión de Ventas de la imprenta CORPOGRÁFIKA LTDA.” Durante el proceso de desarrollo en el artefacto “Análisis y Diseño” se

definen las características del producto a desarrollar, mediante el estudio de los requerimientos funcionales y no funcionales, lo cual constituye la base para la planificación de las iteraciones. Para la primera versión la base está en la captura de requisitos realizada con la información proporcionada por el representante de la empresa. Posteriormente, el avance del proyecto y la realización de cada una de las iteraciones ocasionarán el ajuste de este documento produciendo nuevas versiones actualizadas.

II.1.1.2. Vista General del Proyecto

II.1.1.2.1. Propósito, Alcance y Objetivos.

a) Propósito

El presente proyecto tiene como propósito el desarrollo de un software a medida que ayude en el mejoramiento de la administración del Área de ventas de la imprenta “CORPOGRÁFIKA” LTDA, agilizando los procesos de la misma.

b) Objetivos

a. Objetivo General

Mejorar la Gestión de Ventas de la imprenta “CORPOGRÁFIKA” LTDA, mediante el desarrollo de un sistema web que automatice los procesos del área de ventas de la misma.

b. Objetivos Específicos

- Analizar y diseñar el sistema propuesto.
- Diseñar una interfaz gráfica y amigable para el usuario.
 - Instrumentar el sistema con las siguientes tecnologías:
 - JAVA
 - POSTGRESQL.
- Aplicar el lenguaje de modelado unificado (UML) en toda la fase del desarrollo del sistema.
- Aplicar la metodología de desarrollo RUP (Proceso Unificado Racional).

- o *Facilitar el acceso y manejo de la información:* El uso de una base de datos que nos permita registrar toda la información producida en la empresa nos permitirá a través de una consulta a la misma acceder a los datos deseados de manera rápida y oportuna en un tiempo eficiente evitando procesos tediosos y morosos.
- o *Proveer mayor seguridad en el manejo de la información:* El hecho de asignar roles a cada actor (usuarios) que interviene dentro del sistema nos permitirá tener cierto grado de seguridad en el manejo y acceso a la información dentro de nuestro sistema.

c. Alcances.

El alcance general del presente proyecto es implementar un sistema web para la Imprenta CORPOGRÁFIKA LTDA. que proporcione al usuario una interfaz amigable para que este pueda tener información clasificada, actualizada y acorde a sus necesidades y requerimiento.

- Mediante el sistema se podrán realizar altas, cambios de estado y modificaciones de usuarios, clientes y otras funciones necesarias
- Mediante el sistema se podrán realizar ventas.
- El sistema contará con un sistema de búsqueda.
- El sistema tendrá la opción de administración de clientes y proveedores.
- El sistema podrá listar reportes de ventas y compras por fechas

II.1.1.2.2. Suposiciones y Restricciones

Suposiciones

- Información actualizada.
- Los trabajadores de la imprenta cuentan con conocimientos básicos de computación.
- Reducción y detección de errores en cuanto a la información.

Restricciones

- El sistema no soporta un módulo contable.

- El sistema debe ser utilizado por el personal capacitado.
- El sistema requiere mantenimiento.
- El sistema será implementado en la plataforma Windows.
- El sistema no emitirá facturas.
- El sistema será diseñado según los requerimientos de la empresa.

II.1.1.2.3. Entregables del proyecto

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

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

- **Plan de Desarrollo del Software**

Es el presente documento.

- **Glosario**

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

- **Modelo de Casos de Uso del Negocio**

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.). Permite situar al sistema en el contexto organizacional haciendo énfasis en los

objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

- **Modelo de Objetos del Negocio**

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio.

- **Modelo de Casos de Uso**

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

- **Especificaciones de Casos de Uso**

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluye el flujo de eventos de los mismos.

- **Visión**

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

- **Modelo de Análisis y Diseño**

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

- **Modelo de Datos**

Este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

- **Modelo de Implementación**

Este modelo es una colección de componentes y los subsistemas que lo contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implementación y despliegue del sistema.

- **Prototipos de Interfaces de Usuario**

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

- **Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Manual de Usuario.

- **Producto**

Los ficheros del producto empaquetados y almacenadas en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la

primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva versión al final de cada iteración.

II.1.1.3. Organización del Proyecto

II.1.1.3.1. Participantes en el Proyecto

Tan sólo se cuenta con única persona como partícipe del proyecto, y estará a cargo de la directora del proyecto, quien tomará los roles que se disponen para el proyecto.

- Jefe de Proyecto.
- Analista de Sistemas.
- Diseñador.
- Analista - Programador

II.1.1.3.2. Roles y Responsabilidades

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

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Elaboración del modelo de datos.

Diseñador	Diseña el modelo de la interfaz que tendrá el sistema web.
Programador	Construcción de prototipos.

Tabla 1. Roles y responsabilidades

II.1.2. GLOSARIO

II.1.2.1. Introducción

Este documento recoge los principales términos manejados durante la elaboración del proyecto de desarrollo del sistema de información para la imprenta CORPOGRÁFIKA, se trata de un diccionario informal de datos y de definiciones que se maneja.

II.1.2.2. Propósito

El propósito de este glosario es definir la terminología manejada en el proyecto del sistema para “Administración de ventas vía web para la imprenta CORPOGRÁFIKA LTDA”. También sirve como guía de consulta para la clarificación de los puntos pocos esclarecedores del proyecto.

II.1.2.3. Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos para el área de ventas de la imprenta CORPOGRÁFIKA LTDA”. De tal modo que la terminología empleada se refleje con claridad en este documento.

II.1.2.4. Organización del Glosario

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

II.1.2.5. Definiciones

A continuación se presentan los términos manejados a lo largo del proyecto de desarrollo.

- Administración de ventas: Actividades, procesos y decisiones que abarca la función de la administración de ventas de una empresa.
- Confiable: Capacidad del sistema de realizar su función de la manera prevista.
- Cotización: Precio que se establece a determinado producto según sus características.
- Encargado de almacenes: Usuario que realiza el control del almacén.
- Gerente General: Usuarios que realiza el seguimiento de la institución.
- Gerente de Ventas: Usuario que realiza las cotizaciones ventas.
- Transacciones: Operación de diverso tipo que se realiza entre dos o más partes y que supone el intercambio de bienes o servicios a cambio del capital correspondiente.
- Dato: Unidad o cantidad mínima de información no elaborada, sin sentido en sí misma, pero que convenientemente tratada se puede utilizar en la realización de cálculos o toma de decisiones. Es de empleo muy común en el ámbito informático.
- Diagrama: Gráfico empleado en muchos campos para mostrar los procedimientos detallados que se deben seguir al a realizar una tarea, como un proceso de fabricación. También se utilizan en la resolución de problemas, como por ejemplo en algoritmos. Los diagramas de flujo se usan normalmente para seguir la secuencia lógica de las acciones en el diseño de programas de computadoras.
- Diseño: Etapa o parte del desarrollo de aplicaciones donde se ha de expresar el problema y la solución en términos informáticos para que los programadores puedan realizar su trabajo.
- Interfaz:
- Menú: En Informática, lista de opciones dentro las cuales el usuario de un programa puede seleccionar una acción, por ejemplo, eligiendo un comando o dando un formato particular aparte de un documento. Muchos

programas de aplicaciones, especialmente aquellos que ofrecen una interfaz gráfica de usuario, utilizan menús como un medio que proporciona al usuario una alternativa fácil de aprender y de usar, diferente de la memorización de comandos de programa y su uso apropiado.

- **Proceso:** Un proceso puede informalmente entenderse como un programa en ejecución. Formalmente un proceso es "Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados"
- **Programación:** La Programación es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales.
- **Registro:** En informática, o concretamente en el contexto de una base de datos relacional, un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla. En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas o campos. Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.
- **Sistema:** Un sistema es un conjunto de partes o elementos organizadas y relacionadas que interactúan entre sí para lograr un objetivo
- **Sistema informático:** Un sistema informático es un conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso. Sus partes son: hardware, software y las personas que lo usan.
- **Gramaje:** El peso en gramos del papel.
- **Materiales:** Material utilizado para la realización de un producto.

II.1.3. Requerimientos Funcionales

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

A continuación se describen las funcionalidades que debe proporcionar el sistema.

ID del Requerimiento:	REQ-01
Nombre del Requerimiento:	Otorgar facilidad de ingreso al sistema
Características:	El sistema será de fácil acceso para los usuarios.
Atributo:Prioridad Normal	

Tabla 2. Requisito Funcional 1

ID del Requerimiento:	REQ-02
Nombre del Requerimiento:	Permitir al usuario administrador crear usuarios y asignar roles a los mismos
Características:	El sistema permitirá al administrador del mismo crear usuarios y asignarles los roles respectivos para que estos últimos puedan realizar tan sólo las tareas que les corresponden.
Atributo:Prioridad Alta	

Tabla 3. Requisito funcional 2

ID del Requerimiento:	REQ-03
Nombre del Requerimiento:	Permitir al usuario los procesos del sistema según el rol asignado
Características:	El sistema deberá realizar la autenticación del usuario y permitir que sólo pueda realizar las tareas que les corresponden.
Atributo:Prioridad Normal	

Tabla 4. Requisito funcional 3

ID del Requerimiento:	REQ-04
Nombre del Requerimiento:	Administrar los datos de los usuarios, clientes y proveedores.
Características:	El sistema deberá permitir adicionar nuevos usuarios, clientes y proveedores, permitiendo identificar a las personas de las empresas, también podrá modificar los datos de los mismos y darlos de baja (ponerlos en estado inactivo).
Atributo:Prioridad Normal	

Tabla 5. Requisito funcional 4

ID del Requerimiento:	REQ-05
Nombre del Requerimiento:	Administrar datos de materiales
Características:	El sistema deberá permitir adicionar nuevos materiales, modificarlos y darlos de baja (ponerlos en estado inactivo),.
Atributo:Prioridad	
Normal	

Tabla 6. Requisito funcional 5

ID del Requerimiento:	REQ-06
Nombre del Requerimiento:	Administrar ventas
Características:	El sistema deberá llevar un registro de las ventas realizadas por la empresa.
Atributo:Prioridad	
Normal	

Tabla 7. Requisito funcional 6

ID del Requerimiento:	REQ-07
Nombre del Requerimiento:	Listar reportes
Características:	Obtener reportes de compras y ventas por fechas requeridas por el

	usuario
Atributo:Prioridad Normal	

Tabla 8. Requisito funcional 7

II.1.4. Requerimientos no Funcionales

Estos requerimientos tienen que ver con las características que una u otra forma puedan limitar el sistema como son: el rendimiento (en tiempo y espacio), confiabilidad, interfaces, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, etc.

- **Seguridad**

1. El usuario que intente conectarse al sistema deberá introducir su nombre de usuario y contraseña de acceso, el sistema deberá comprobar que se trata de un usuario autorizado. En caso de que no sea así, se le negará el acceso.
2. El sistema tendrá distintos tipos de usuarios ya a cada uno de ellos se le permitirá únicamente el acceso a las funciones que le correspondan.

- **Fiabilidad**

Cualquier transacción finalizada por un usuario deberá ser procesada exitosamente con una tasa de error del 0%. En caso de un error de procesamiento, la transacción no se deberá considerar finalizada.

II.1.4.1. Estereotipos UML

Actor del Negocio	
-------------------	---

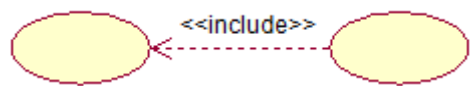
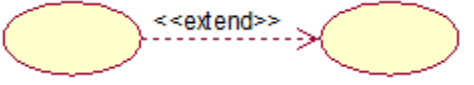
Caso de Uso del Negocio	
Comunicación	
Trabajador del Negocio	
Entidad del Negocio	
Actor	
Caso de Uso	
Relación de Inclusión	
Relación de Extensión	

Tabla 9. Estereotipos UML

II.1.5. MODELO DE CASO DE USO DEL NEGOCIO

II.1.5.1. Introducción

El modelo de casos de Uso del Negocio es un artefacto de la disciplina Requisitos en la metodología RUP la cual se está implementando.

II.1.5.2. Propósito

- Comprender la estructura y la dinámica de la imprenta
- Comprender los procesos del negocio de la imprenta

II.1.5.3. Alcance

- Identificar y definir los procesos del negocio.
- Describir los procesos de los usuarios.

II.1.5.4. Diagrama de Casos de Uso del Negocio

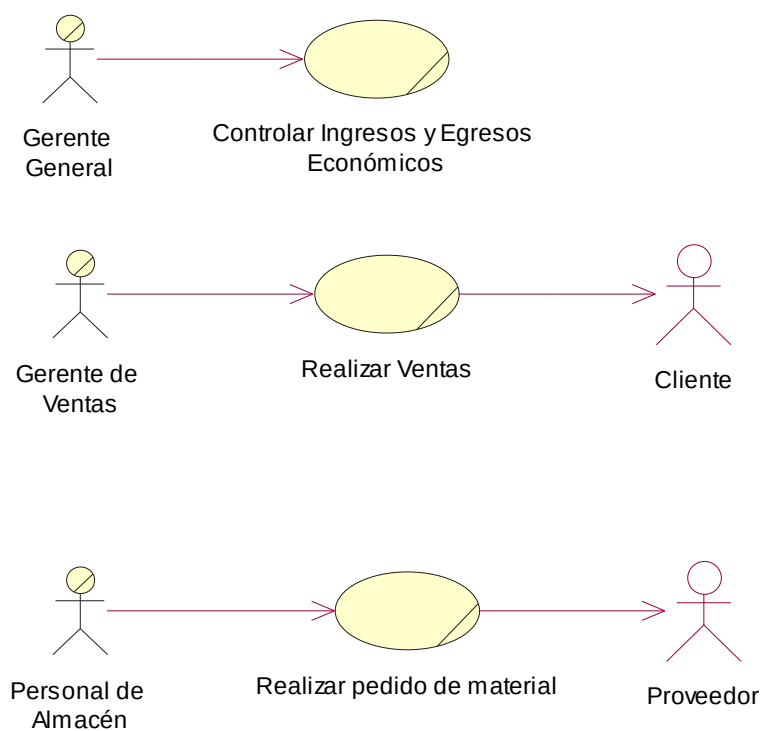


Figura 1. Diagrama de Casos de Uso del Negocio

- Realizar Ventas

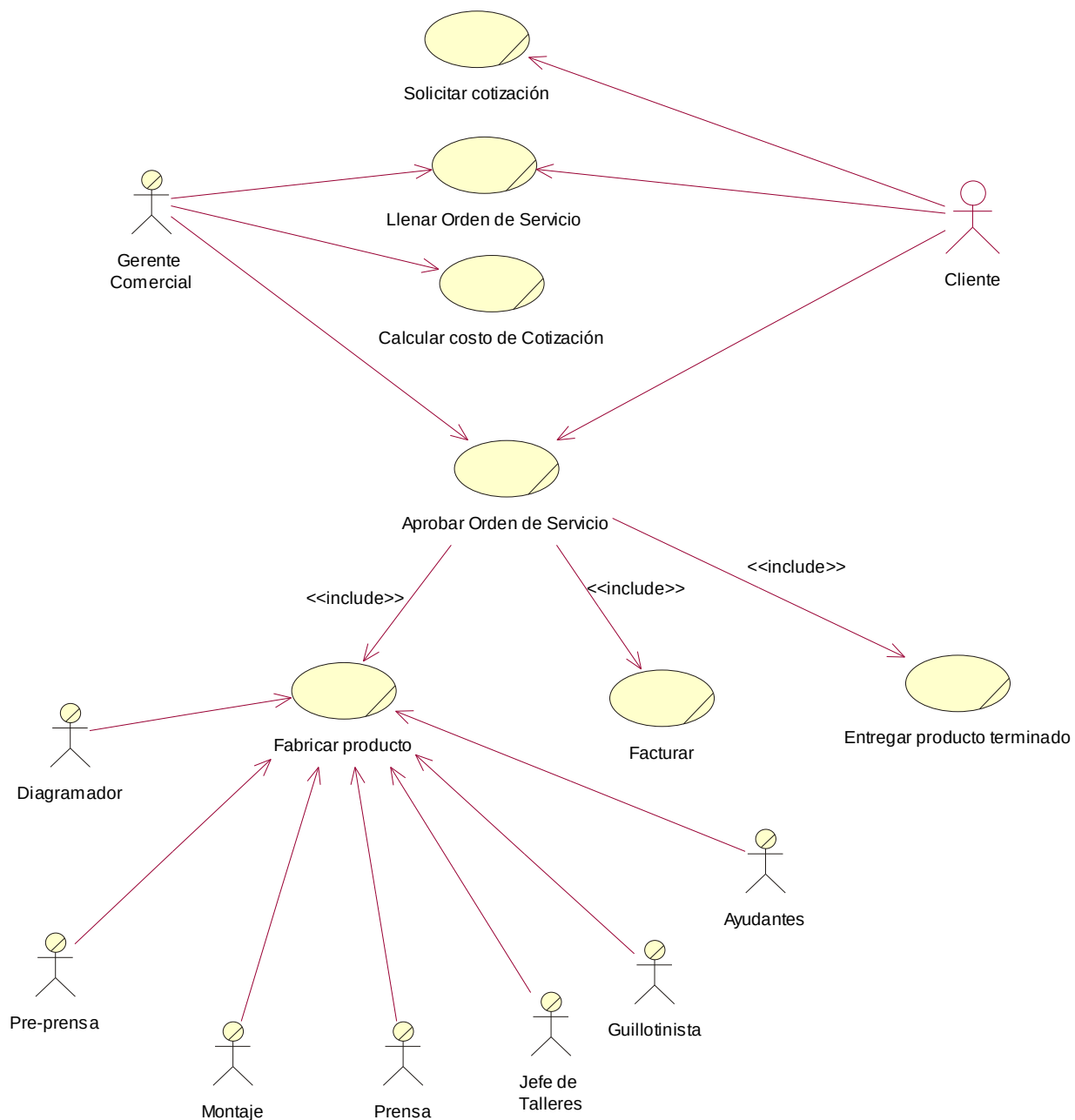


Figura 2. D.C.U.N Realizar Ventas

- Realizar pedido de material

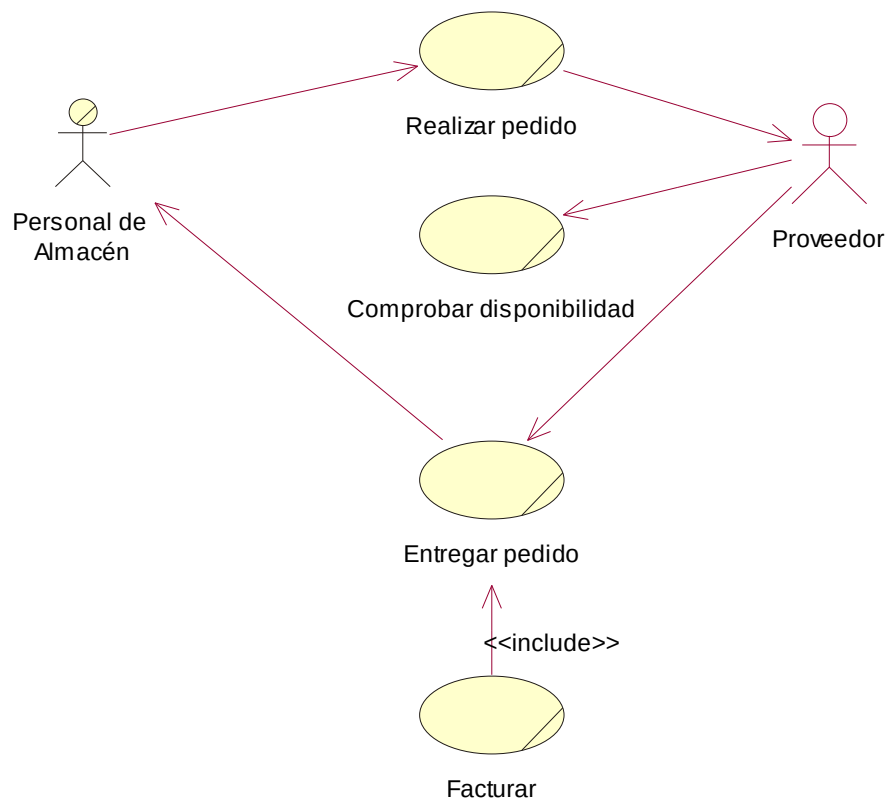


Figura 3. D.C.U.N Realizar pedido de material

- Solicitar reportes

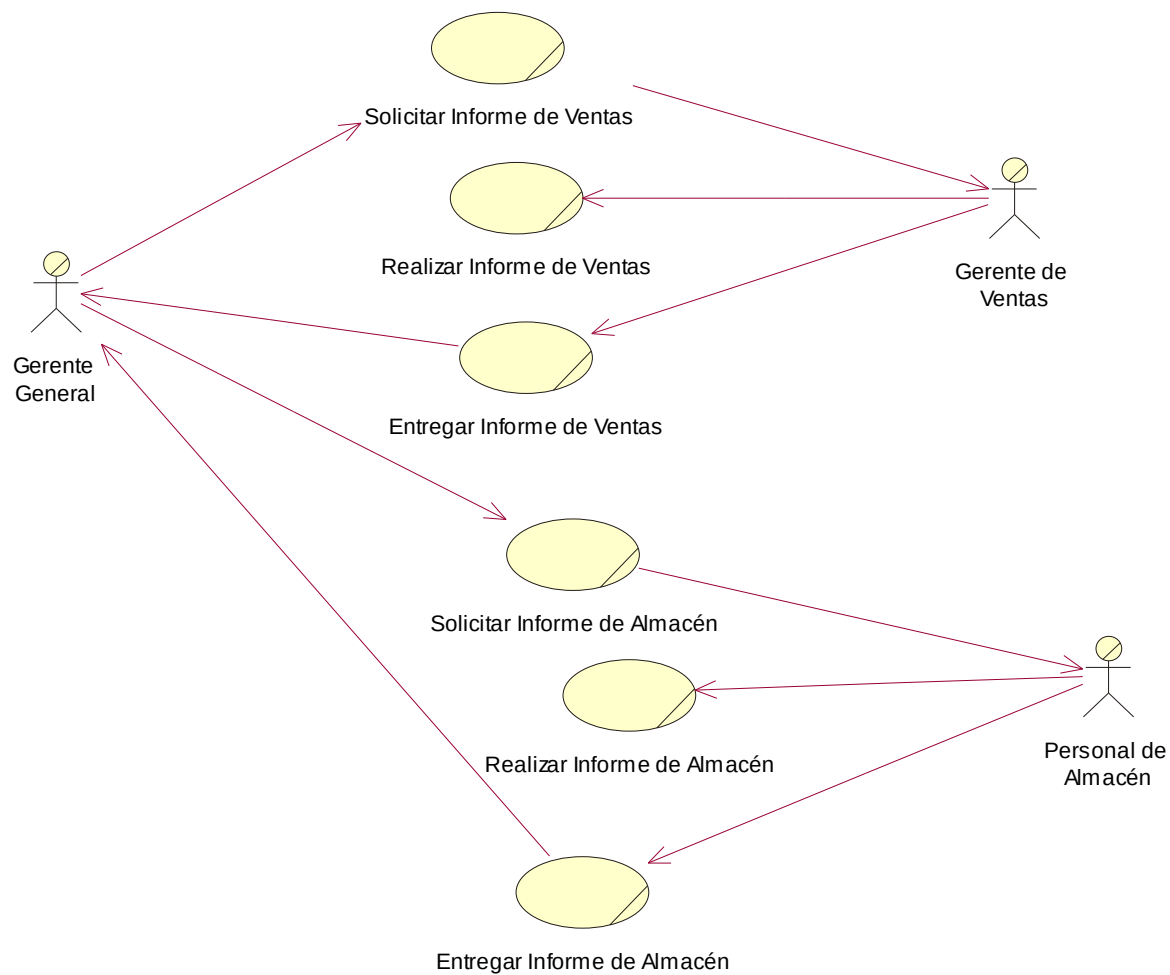


Figura 4. D.C.U.N Solicitar reportes

II.1.6. MODELO DE OBJETOS DEL NEGOCIO

II.1.6.1. Introducción

El Modelo de Objetos del Negocio es un artefacto de la disciplina Requisitos en la metodología RUP la cual se está implementando.

II.1.6.2. Propósito

- Comprender la estructura dinámica de los casos de uso del negocio de la organización
- Comprender los procesos de negocio de la organización.

II.1.6.3. Alcance

- Describe el comportamiento de los procesos de negocio
- Identificar y definir los *objetos* del negocio

II.1.6.4. Diagramas de Objetos del Negocio

- **Realizar Ventas**

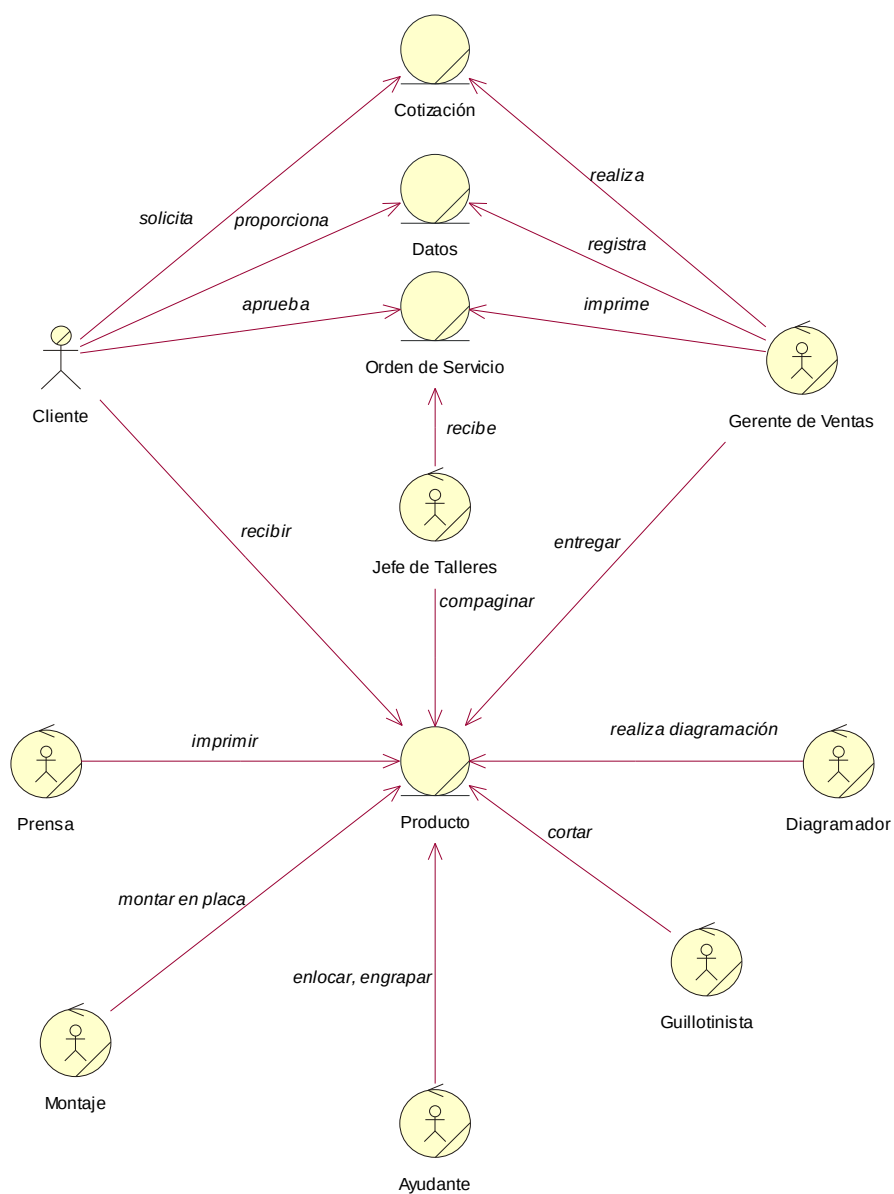


Figura 5. Diagramas de Objetos del Negocio

- *Realizar pedido de material*

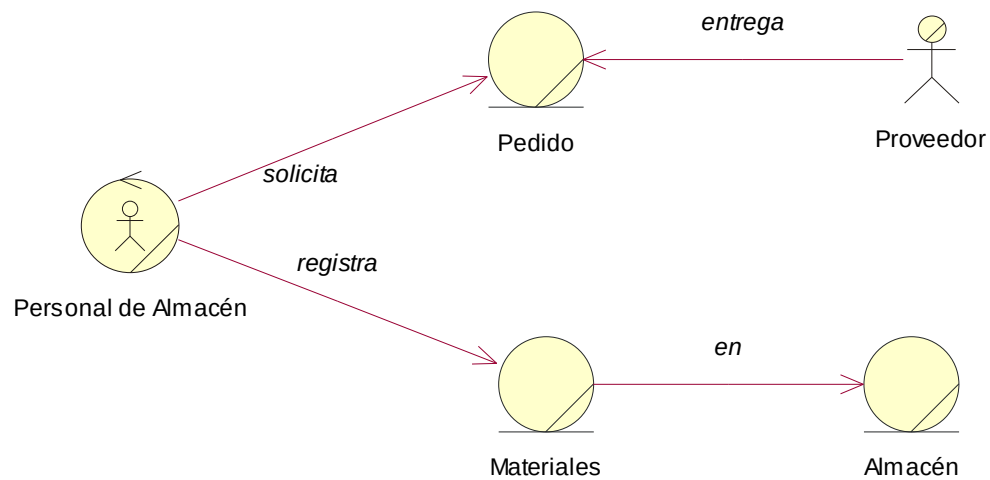


Figura 6. Realizar pedido material

- *Solicitar reportes*

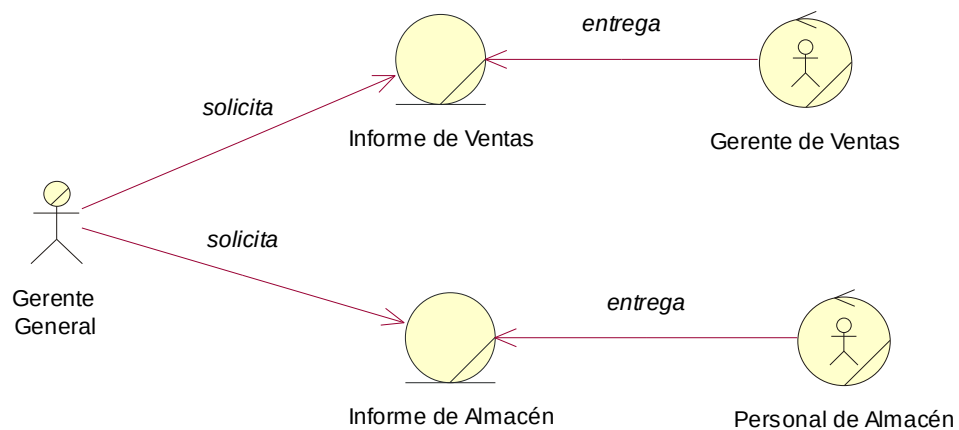


Figura 7. Solicitar reportes

II.1.7. MODELO DE CASOS DE USO

II.1.7.1. Introducción

Los casos de uso se utilizan para modelar el funcionamiento del sistema.

II.1.7.2. Propósito:

- Modelar los requerimientos del sistema
- Identificar los procesos del sistema

II.1.7.3. Alcance:

Identificar y definir los procesos del sistema.

II.1.7.4. Diagramas de Casos de Uso

Caso de uso Administrar Usuarios:

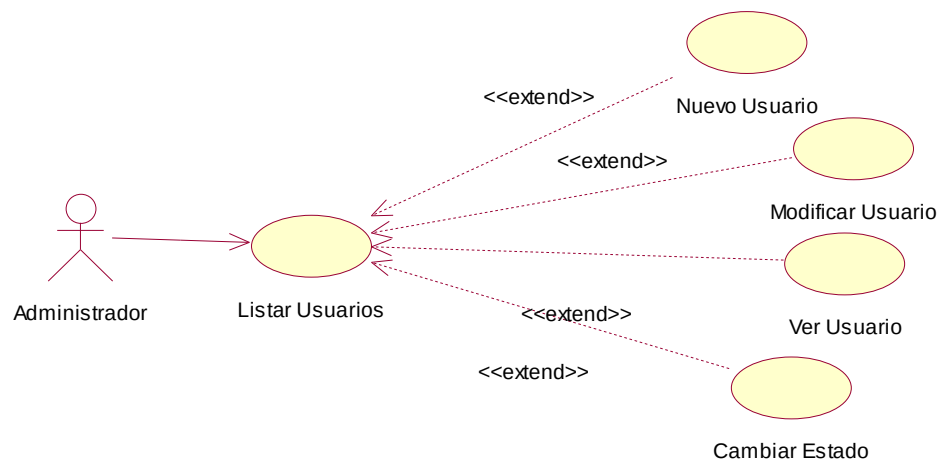


Figura 8. C.U. Administrar Usuarios

Caso de uso Administrar Roles:

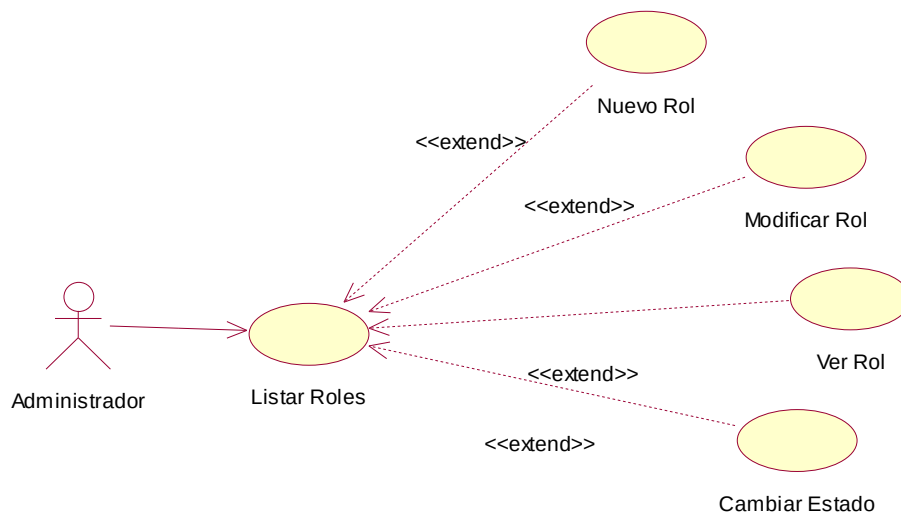


Figura 9. C.U. Administrar Roles

a. Caso de uso Administrar Procesos:

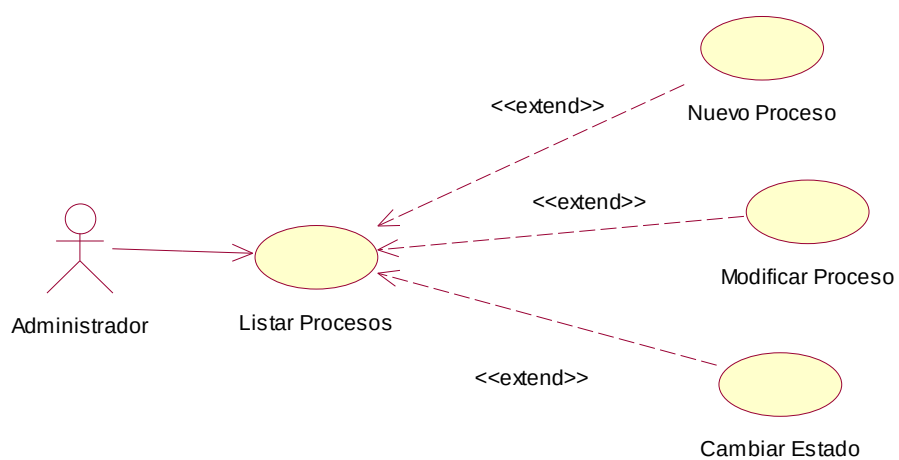


Figura 10. C.U. Administrar Procesos

b. Caso de uso Administrar Proveedores:

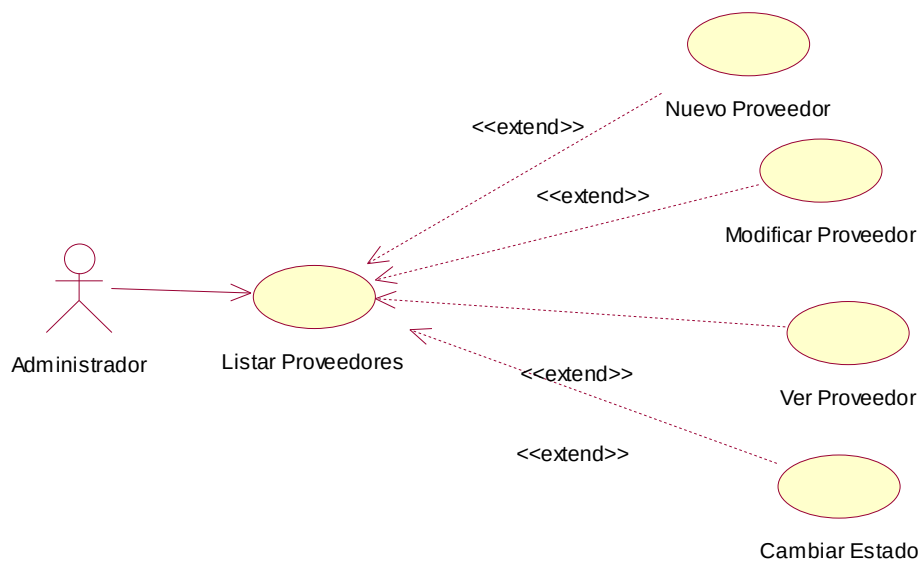


Figura 11. C.U. Administrar Proveedores

c. Caso de uso Administrar Clientes:

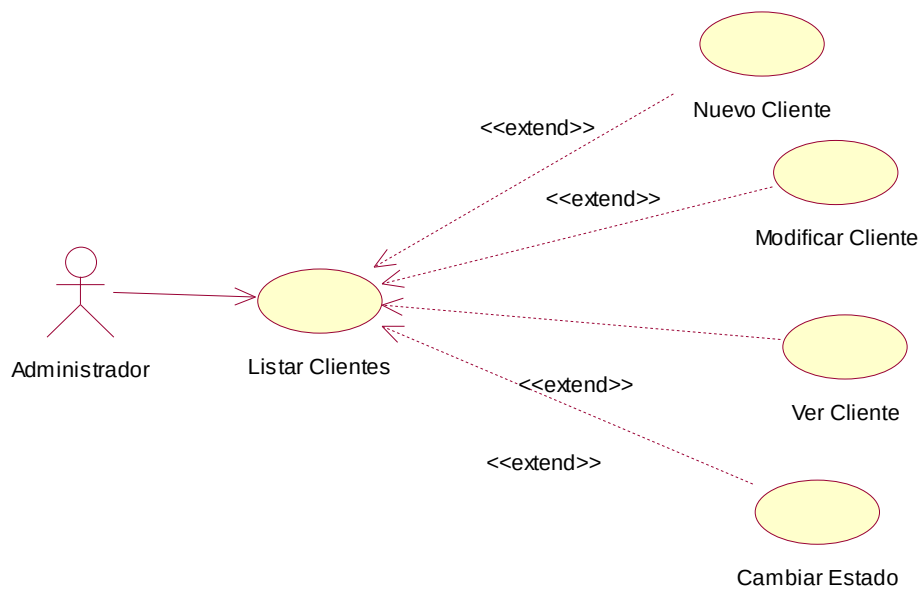


Figura 12. C.U. Administrar Clientes

d. Caso de uso Administrar Materiales:

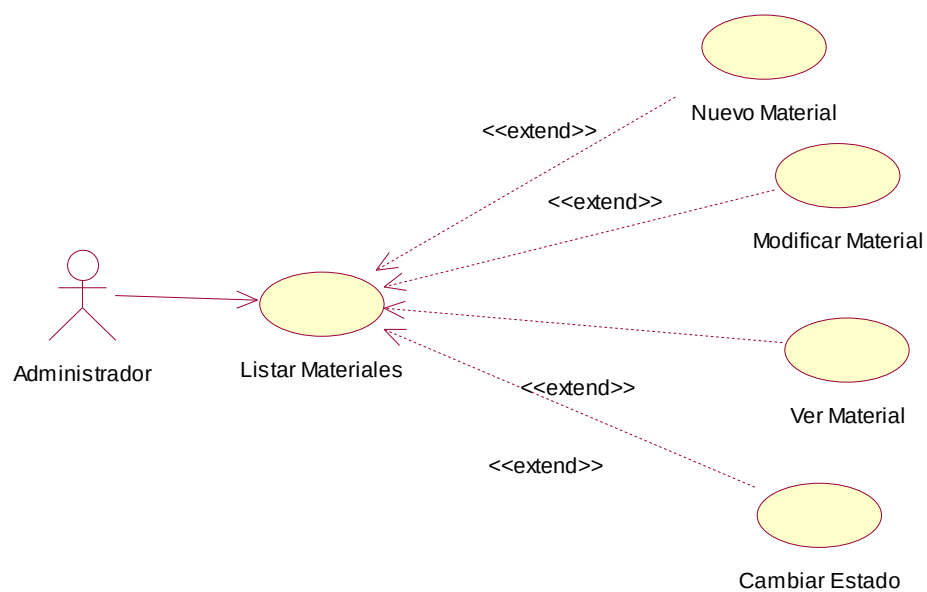


Figura 13. C.U. Administrar Material

e. Caso de uso Administrar Tipo Material:

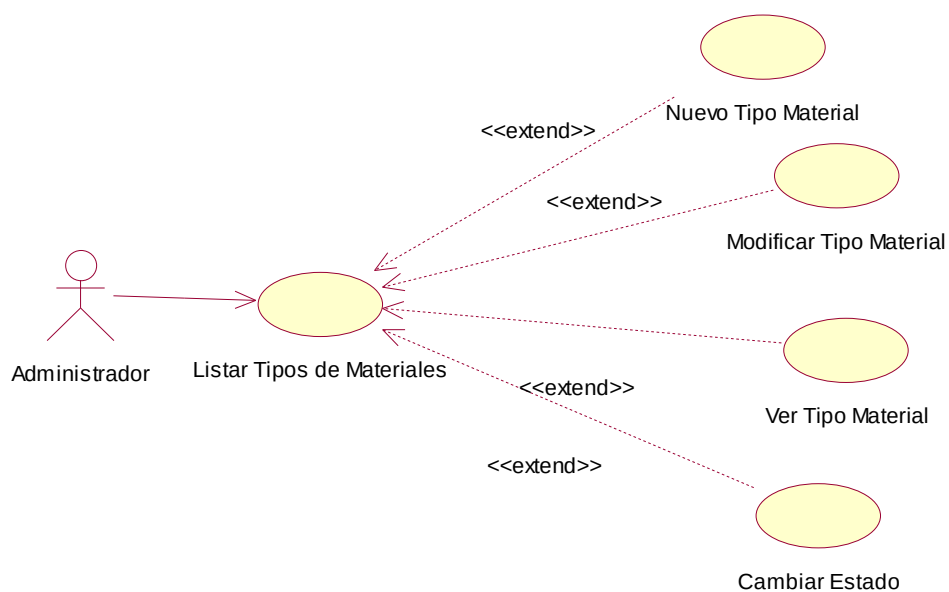


Figura 14. C.U. Administrar Tipo Material

f. Caso de uso Administrar Notas:

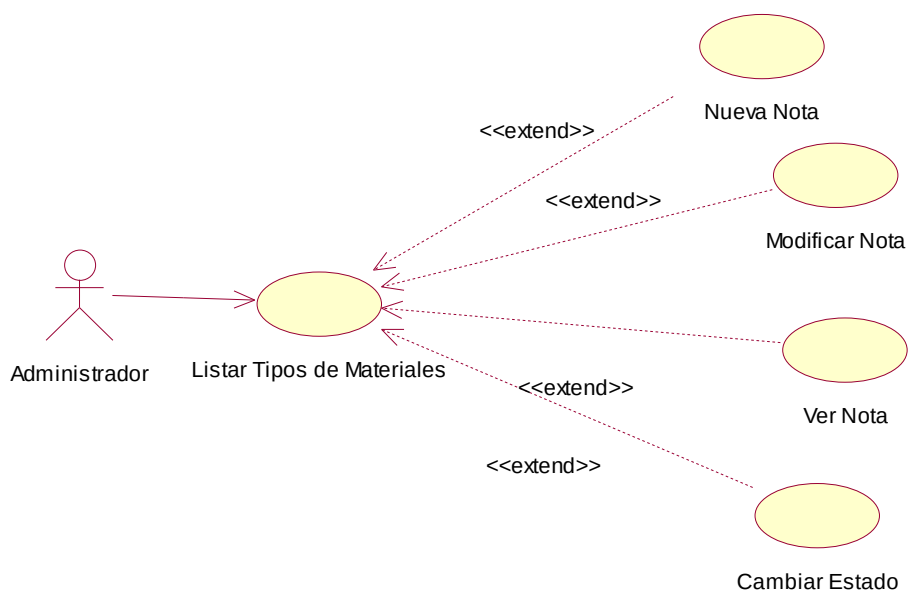


Figura 15. C.U. Administrar Notas

g. Caso de uso Administrar Categoría Materiales:

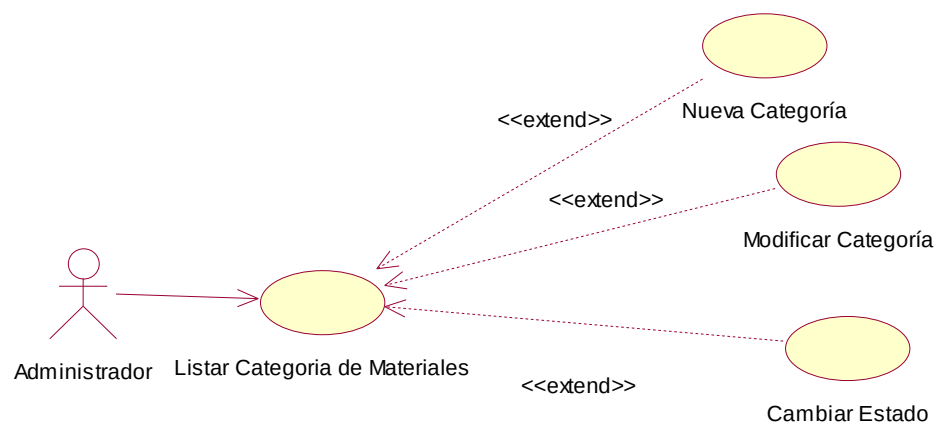


Figura 16. Administrar Categoría Materiales

h. Caso de uso Administrar Compras:

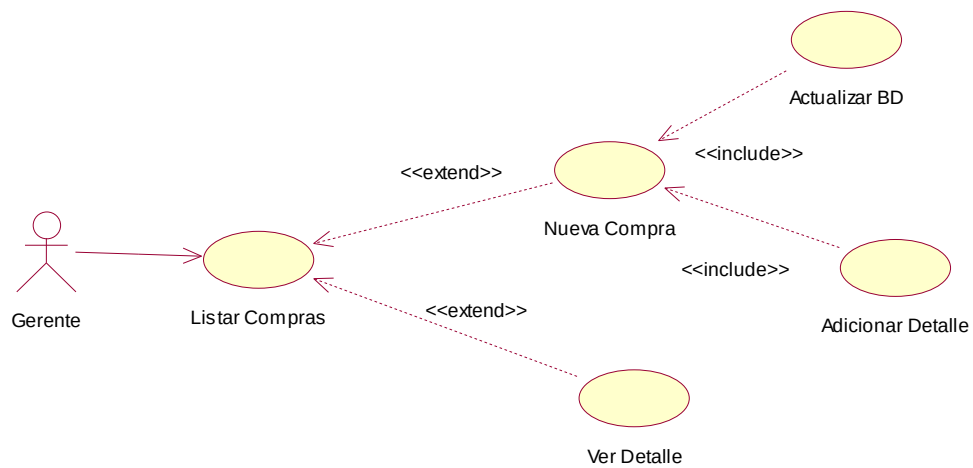


Figura 17. Administrar Compras

i. Caso de uso Administrar Ventas:

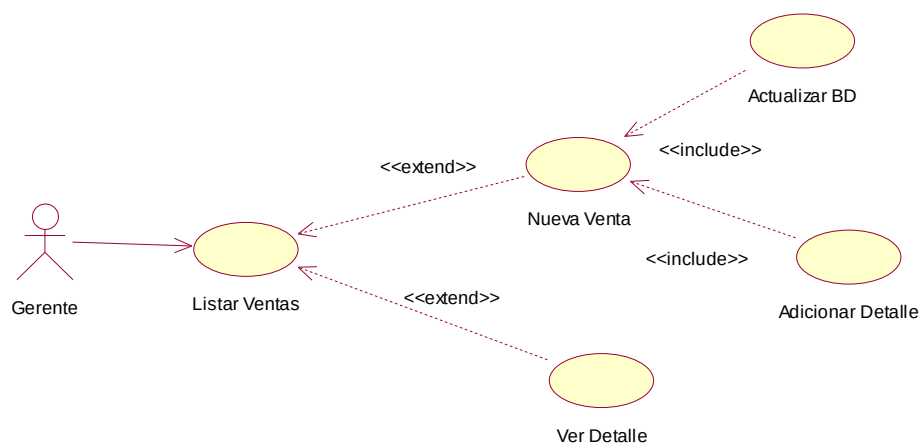


Figura 18. Administrar Ventas

j. Caso de uso Listar Reportes:

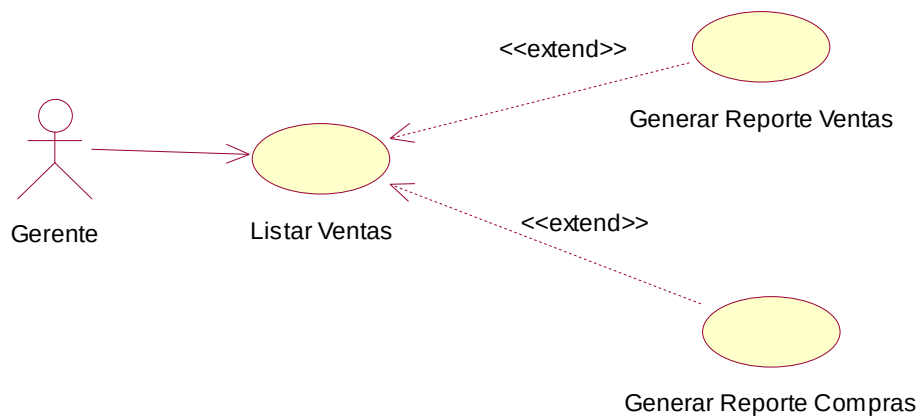


Figura 19. Listar Reportes

II.1.8. ESPECIFICACIONES DE CASOS DE USO

II.1.8.1. Introducción

La Especificación de los Casos de Uso es una especificación detallada de los casos de uso del sistema

II.1.8.2. Propósito

- Interpretar y describir los casos de uso

II.1.8.3. Alcance

- Describe los procesos internos de los casos de uso.
- Detalla los flujos de los casos de uso según lo establecido por la empresa.

II.1.8.4. Especificación de los Casos de Uso

Administrar Usuarios

Especificación de Caso de Uso: Nuevo Usuario

Caso de Uso: Nuevo Usuario
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Usuario.
2. El sistema guarda los datos del Nuevo Usuario.
3. El administrador visualiza la lista actualizada de Usuarios.
4. Fin del Caso

Tabla 10. Especificación de C. U. Nuevo Usuario

Especificación de Caso de Uso: Modificar Usuario

Caso de Uso: Modificar Usuario
Actor: Administrador
1. El administrador selecciona el Usuario a Modificar
2. El sistema devuelve una pantalla para Modificar datos del Usuario seleccionado.
3. El administrador modifica datos del Usuario.
4. El sistema guarda los datos del Usuario modificado.
5. El administrador visualiza la lista actualizada de Usuarios.
6. Fin del Caso

Tabla 11. Especificación de C. U. Modificar Usuario

Especificación de Caso de Uso: Ver Usuario

Caso de Uso: Ver Usuario
Actor: Administrador
1. El administrador selecciona el Usuario a Visualizar
2. El sistema devuelve una pantalla con los datos del Usuario seleccionado.
3. El administrador visualiza los datos del Usuario.
4. Fin del Caso

Tabla 12. Especificación de C. U. Ver Usuario

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado del Usuario deseado.
2. El sistema cambia el estado del Usuario.
3. El Administrador visualiza la Lista Actualizada de Usuarios.
4. Fin del Caso

Tabla 13. Especificación de C. U. Cambiar Estado

Administrar Roles

Especificación de Caso de Uso: Nuevo Rol

Caso de Uso: Nuevo Rol
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Rol.
2. El sistema guarda los datos del Nuevo Rol
3. El administrador visualiza la lista actualizada de Roles
4. Fin del Caso.

Tabla 14: Especificación de C. U. Nuevo Rol***Especificación de Caso de Uso: Modificar Rol***

Caso de Uso: Modificar Rol
Actor: Administrador
1. El administrador hace clic en el botón Modificar del Rol a Modificar
2. El sistema devuelve una pantalla para Modificar datos del Rol seleccionado.
3. El administrador modifica datos del Rol.
4. El sistema guarda los datos del Rol, vuelve a la pantalla Administrar Roles.
5. El administrador visualiza la lista actualizada de Roles
6. Fin del Caso

Tabla 15: Especificación de C. U. Modificar Rol***Especificación de Caso de Uso: Ver Rol***

Caso de Uso: Ver Rol
Actor: Administrador
1. El administrador selecciona el Rol a Visualizar
2. El sistema devuelve una pantalla con los datos del Rol seleccionado.
3. El Administrador visualiza los datos del Rol.
4. Fin del Caso

Tabla 16: Especificación de C. U. Ver Rol

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
5. El administrador hace clic en el botón Cambiar Estado del Rol deseado.
6. El sistema cambia el estado del Rol.
7. El Administrador visualiza la Lista Actualizada de Roles.
8. Fin del Caso

Tabla 17: Especificación de C. U. Cambiar Estado

Administrar Procesos

Especificación de Caso de Uso: Nuevo Proceso

Caso de Uso: Nuevo Proceso
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Proceso.
2. El sistema guarda los datos del Nuevo Proceso.
3. El administrador visualiza la lista actualizada de Procesos
4. Fin del Caso

Tabla 18: Especificación de C. U. Nuevo Proceso

Especificación de Caso de Uso: Modificar Proceso

Caso de Uso: Modificar Proceso
Actor: Administrador
1. El administrador hace clic en el botón Modificar del Proceso a Modificar
2. El sistema devuelve una pantalla para Modificar datos del Proceso seleccionado.
3. El administrador modifica datos del Proceso.
4. El sistema guarda los datos del Proceso y muestra la pantalla Administrar Procesos.
5. El administrador visualiza la lista actualizada de Procesos
6. Fin del Caso.

Tabla 19: Especificación de C. U. Modificar Proceso

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado del Proceso deseado.
2. El sistema cambia el estado del Proceso.
3. El Administrador visualiza la Lista Actualizada de Roles.
4. Fin del Caso

Tabla 20: Especificación de C. U. Cambiar Estado

Administrar Clientes

Especificación de Caso de Uso: Nuevo Cliente

Caso de Uso: Nuevo Cliente
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Cliente.
2. El sistema guarda los datos del Nuevo Cliente
3. El administrador visualiza la lista actualizada de Clientes
4. Fin del Caso

Tabla 21: Especificación de C. U. Nuevo Cliente

Especificación de Caso de Uso: Modificar Cliente

Caso de Uso: Modificar Cliente
Actor: Administrador
1. El administrador hace clic en el botón Modificar del Cliente a Modificar.
2. El sistema devuelve una pantalla para Modificar datos del Cliente seleccionado.
3. El administrador modifica datos del Cliente.
4. El sistema guarda los datos del Cliente modificado.
5. El administrador visualiza la lista actualizada de Clientes
6. Fin del Caso.

Tabla 22: Especificación de C. U. Modificar Cliente

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado del Cliente deseado.
2. El sistema cambia el estado del Cliente
3. El Administrador visualiza la Lista Actualizada de Clientes
4. Fin del Caso

Tabla 23: Especificación de C. U. Cambiar estado

Especificación de Caso de Uso: Ver Cliente

Caso de Uso: Ver cliente
Actor: Administrador
1. El administrador selecciona el Cliente a Visualizar
2. El sistema devuelve una pantalla con los datos del Cliente
3. El Administrador visualiza los datos del Cliente
4. Fin del Caso

Tabla 24: Especificación de C. U. Ver Rol

Administrar Proveedores

Especificación de Caso de Uso: Nuevo Proveedor

Caso de Uso: Nuevo Proveedor
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Proveedor.
2. El sistema guarda los datos del Nuevo Proveedor
3. El administrador visualiza la lista actualizada de Proveedores
4. Fin del Caso.

Tabla 25: Especificación de C. U. Nuevo Proveedor

Especificación de Caso de Uso: Modificar Proveedor

Caso de Uso: Modificar Proveedor
Actor: Administrador
1. El administrador selecciona el botón Modificar del Proveedor a Modificar.
2. El sistema devuelve una pantalla para Modificar datos del Proveedor seleccionado.
3. El administrador modifica datos del Proveedor.
4. El sistema guarda los datos del Proveedor modificado, vuelve a la pantalla Administrar Proveedores.
5. El administrador visualiza la lista actualizada de Proveedores
6. Fin del Caso.

Tabla 26: Especificación de C. U. Modificar Proveedor

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
5. El administrador hace clic en el botón Cambiar Estado del Proveedor deseado.
6. El sistema cambia el estado del Proveedor
7. El Administrador visualiza la Lista Actualizada de Proveedores
8. Fin del Caso

Tabla 27: Especificación de C. U. Cambiar Estado

Especificación de Caso de Uso: Ver Proveedor

Caso de Uso: Ver Proveedor
Actor: Administrador
1. El administrador selecciona el Proveedor a Visualizar
2. El sistema devuelve una pantalla con los datos del Proveedor
3. El Administrador visualiza los datos del Proveedor
4. Fin del Caso

Tabla 28: Especificación de C. U. Ver Proveedor

Administrar Notas

Especificación de Caso de Uso: Nueva Nota

Caso de Uso: Nueva Nota
Actor: Administrador
1. El administrador ingresa los datos de la Nueva Nota.
2. El sistema guarda los datos de la Nueva Nota
3. El administrador visualiza la lista actualizada de Notas
4. Fin del Caso.

Tabla 29: Especificación de C. U. Nueva Nota

Especificación de Caso de Uso: Modificar Nota

Caso de Uso: Modificar Nota
Actor: Administrador
1. El administrador hace clic en el botón Modificar de la Nota a Modificar
2. El sistema devuelve una pantalla para Modificar datos de la Nota seleccionada.
3. El administrador modifica datos de la Nota.
4. El sistema guarda los datos de la Nota modificada y vuelve a la pantalla Administrar Notas.
5. El administrador visualiza la lista actualizada de Notas
6. Fin del Caso

Tabla 30: Especificación de C. U. Modificar Nota

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado de la Nota deseada.
2. El sistema cambia el estado de la Nota
3. El Administrador visualiza la Lista Actualizada de Notas
4. Fin del Caso

Tabla 31: Especificación de C. U. Cambiar Estado

Especificación de Caso de Uso: Ver Nota

Caso de Uso: Ver Nota
Actor: Administrador
1. El administrador selecciona la Nota a Visualizar
2. El sistema devuelve una pantalla con los datos de la Nota
3. El Administrador visualiza los datos de la Nota
4. Fin del Caso

Tabla 32: Especificación de C. U. Ver Nota

Administrar Foto Productos

Especificación de Caso de Uso: Nueva Foto

Caso de Uso: Nueva Foto
Actor: Administrador
1. El administrador ingresa los datos de la Nueva Foto.
2. El sistema guarda los datos de la Nueva Foto
3. El administrador visualiza la lista actualizada de Fotos
4. Fin del Caso

Tabla 33: Especificación de C. U. Nueva Foto

Especificación de Caso de Uso: Modificar Foto

Caso de Uso: Modificar Foto
Actor: Administrador
1. El administrador hace clic en el botón Modificar de la Foto a Modificar
2. El sistema devuelve una pantalla para Modificar datos de la foto seleccionada.
3. El administrador modifica datos de la foto.
4. El sistema guarda los datos de la Foto modificada, vuelve a la pantalla Administrar Fotos.
5. El administrador visualiza la lista actualizada de Fotos
6. Fin del Caso

Tabla 34: Especificación de C. U. Modificar Foto

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado de la Foto deseada.
2. El sistema cambia el estado de la Foto
3. El Administrador visualiza la Lista Actualizada de Fotos
4. Fin del Caso

Tabla 35: Especificación de C. U. Cambiar Estado

Especificación de Caso de Uso: Ver Foto

Caso de Uso: Ver Foto
Actor: Administrador
1. El administrador selecciona la Foto a Visualizar
2. El sistema devuelve una pantalla con los datos de la Foto
3. El Administrador visualiza los datos de la Foto
4. Fin del Caso

Tabla 36: Especificación de C. U. Ver Foto

Administrar Tipo Productos

Especificación de Caso de Uso: Nuevo Tipo Producto

Caso de Uso: Nuevo Tipo Producto
Actor: Administrador
5. El administrador ingresa los datos del Nuevo Tipo Producto
6. El sistema guarda los datos del Nuevo Tipo Producto
7. El administrador visualiza la lista actualizada de Tipo Productos
8. Fin del Caso

Tabla 37: Especificación de C. U. Nuevo Tipo Producto

Especificación de Caso de Uso: Modificar Tipo Producto

Caso de Uso: Modificar Tipo Producto
Actor: Administrador
7. El administrador hace clic en el botón Modificar del Tipo Producto a Modificar
8. El sistema devuelve una pantalla para Modificar datos del Tipo Producto seleccionado.
9. El administrador modifica datos del Tipo Producto.
10. El sistema guarda los datos del Tipo Producto modificado, vuelve a la pantalla Administrar Tipo Producto.
11. El administrador visualiza la lista actualizada de Tipo Productos
12. Fin del Caso

Tabla 38: Especificación de C. U. Modificar Tipo Producto

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado del Tipo de Producto deseado.
2. El sistema cambia el estado del Tipo de Producto
3. El Administrador visualiza la Lista Actualizada de Tipo de Productos
4. Fin del Caso

Tabla 39: Especificación de C. U. Cambiar Estado

Especificación de Caso de Uso: Ver Tipo Producto

Caso de Uso: Ver Tipo Producto
Actor: Administrador
1. El administrador selecciona el Tipo de Producto a Visualizar
2. El sistema devuelve una pantalla con los datos del Tipo de Producto
3. El Administrador visualiza los datos del Tipo de Producto
4. Fin del Caso

Tabla 40: Especificación de C. U. Ver Tipo Producto

Administrar Compras

Especificación de Caso de Uso: Nueva Compra

Caso de Uso: Nueva Compra
Actor: Gerente
1. El gerente ingresa los datos de Nueva Compra
2. El sistema guarda los datos generales de Nueva Compra y muestra una pantalla para adicionar los detalles de Compra.
3. El gerente adiciona un detalle.
4. El sistema guarda de forma temporánea los datos de detalle introducido
5. Se repite el paso 3 hasta que no se desee adicionar más detalles.
6. El sistema guarda en la base de datos los datos de Compra.
7. Fin del Caso

Tabla 41: Especificación de C. U. Nueva Compra

Especificación de Caso de Uso: Ver Detalle Compra

Caso de Uso: Ver Detalle Compra
Actor: Gerente
1. El gerente hace clic en Ver detalle de la compra deseada.
2. El sistema devuelve una pantalla con los detalles de la compra seleccionada.
3. El gerente visualiza los detalles de la compra.
4. Fin del Caso

Tabla 42: Especificación de C. U. Ver detalle Compra

Administrar Ventas

Especificación de Caso de Uso: Nueva Venta

Caso de Uso: Nueva venta
Actor: gerente
1. El gerente ingresa los datos de Nueva Venta
2. El sistema guarda los datos generales de Nueva Venta y muestra una pantalla para adicionar los detalles de Venta.
3. El gerente adiciona un detalle de venta.
4. El sistema guarda de forma temporal el detalle de venta.
5. Se repite el paso 3 hasta que no se desee adicionar más productos.
6. El sistema guarda en la base de datos los datos de la Nueva Venta.
7. Fin del Caso

Tabla 43: Especificación de C. U. Nueva Venta

Especificación de Caso de Uso: Ver Detalle Venta

Caso de Uso: Ver Detalle Venta
Actor: gerente
1. El gerente hace clic en Ver Detalle de la venta deseada.
2. El sistema devuelve una pantalla con los detalles de la venta seleccionada.
3. El gerente visualiza los detalles de la venta.
4. Fin del Caso

Tabla 44: Especificación de C. U. Ver detalle Venta

Administrar Materiales

Especificación de Caso de Uso: Nuevo Material

Caso de Uso: Nuevo Material
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Material.
2. El sistema guarda los datos del Nuevo Material.
3. El administrador visualiza la lista actualizada de Materiales
4. Fin del Caso

Tabla 45: Especificación de C. U. Nuevo Material

Especificación de Caso de Uso: Modificar Material

Caso de Uso: Modificar Material
Actor: Administrador
1. El administrador hace clic en el botón Modificar del Material a Modificar
2. El sistema devuelve una pantalla para Modificar datos del Material seleccionado.
3. El administrador modifica datos del Material.
4. El sistema guarda los datos del Material modificado, vuelve a la pantalla Administrar Materiales.
5. El administrador visualiza la lista actualizada de Materiales
6. Fin del Caso

Tabla 46: Especificación de C. U. Modificar Material

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado del Material Producto deseado.
2. El sistema cambia el estado del Material
3. El Administrador visualiza la Lista Actualizada de Materiales
4. Fin del Caso

Tabla 47: Especificación de C. U. Cambiar Estado

Especificación de Caso de Uso: Ver Material

Caso de Uso: Ver Material
Actor: Administrador
1. El administrador selecciona el Material a Visualizar
2. El sistema devuelve una pantalla con los datos del Material
3. El Administrador visualiza los datos del Material
4. Fin del Caso

Tabla 48: Especificación de C. U. Ver Material

Administrar Categoría Materiales

Especificación de Caso de Uso: Nueva Categoría Materiales

Caso de Uso: Nueva Categoría Materiales
Actor: Administrador
1. El administrador ingresa los datos de la Nueva categoría.
2. El sistema guarda los datos de la categoría
3. El administrador visualiza la lista actualizada de Categoría Materiales
4. Fin del Caso

Tabla 49: Especificación de C. U. Nueva Categoría Materiales

Especificación de Caso de Uso: Modificar Categoría Materiales

Caso de Uso: Modificar Categoría Materiales
Actor: Administrador
1. El administrador hace clic en el botón Modificar de la categoría a Modificar
2. El sistema devuelve una pantalla para Modificar datos de la categoría seleccionada.
3. El administrador modifica datos de la Categoría.
4. El sistema guarda los datos de la categoría modificada, vuelve a la pantalla Administrar Categoría Materiales
5. El administrador visualiza la lista actualizada de Categoría Materiales
6. Fin del Caso

Tabla 50: Especificación de C. U. Modificar Categoría Materiales

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado de la Categoría deseada.
2. El sistema cambia el estado de la Categoría
3. El Administrador visualiza la Lista Actualizada de Categoría Materiales
4. Fin del Caso

Tabla 51: Especificación de C. U. Cambiar Estado

Administrar Tipo Materiales

Especificación de Caso de Uso: Nuevo Tipo Material

Caso de Uso: Nuevo Tipo Material
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Tipo Material
2. El sistema guarda los datos del Nuevo Tipo Material
3. El administrador visualiza la lista actualizada de Tipo Materiales
4. Fin del Caso

Tabla 52: Especificación de C. U. Nuevo Tipo Material

Especificación de Caso de Uso: Modificar Tipo Material

Caso de Uso: Modificar Tipo Material
Actor: Administrador
1. El administrador hace clic en el botón Modificar del Tipo Material a Modificar
2. El sistema devuelve una pantalla para Modificar datos del Tipo Material seleccionado.
3. El administrador modifica datos del Tipo Material.
4. El sistema guarda los datos del Tipo Material modificado, vuelve a la pantalla Administrar Tipo Materiales.
5. El administrador visualiza la lista actualizada de Tipo Materiales
6. Fin del Caso

Tabla 53: Especificación de C. U. Modificar Tipo Material

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado del Tipo de Material deseado.
2. El sistema cambia el estado del Tipo de Material
3. El Administrador visualiza la Lista Actualizada de Tipos de Materiales
4. Fin del Caso

Tabla 54: Especificación de C. U. Cambiar Estado

Administrar Medida Materiales

Especificación de Caso de Uso: Nueva Medida Material

Caso de Uso: Nueva Medida Material
Actor: Administrador
1. El administrador ingresa los datos de la Nueva Medida Material
2. El sistema guarda los datos de la Nueva Medida Material
3. El administrador visualiza la lista actualizada de Medidas de Materiales
4. Fin del Caso

Tabla 55: Especificación de C. U. Nueva Medida Material

Especificación de Caso de Uso: Modificar Medida Material

Caso de Uso: Modificar Medida Material
Actor: Administrador
1. El administrador hace clic en el botón Modificar de la Medida Material a Modificar
2. El sistema devuelve una pantalla para Modificar datos de la Medida Material seleccionada.
3. El administrador modifica datos de la Medida Material.
4. El sistema guarda los datos de la Medida Material modificada, vuelve a la pantalla Administrar Medidas.
5. El administrador visualiza la lista actualizada de Medidas de Materiales
6. Fin del Caso

Tabla 56: Especificación de C. U. Modificar Medida Material

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado de la Medida de Material deseada.
2. El sistema cambia el estado de la Medida de Material
3. El Administrador visualiza la Lista Actualizada de Medidas de Materiales
4. Fin del Caso

Tabla 57: Especificación de C. U. Cambiar Estado

Administrar Gramajes

Especificación de Caso de Uso: Nuevo Gramaje

Caso de Uso: Nuevo Gramaje
Actor: Administrador
1. El administrador ingresa los datos del Nuevo Gramaje.
2. El sistema guarda los datos del Nuevo Gramaje
3. El administrador visualiza la lista actualizada de Gramajes
4. Fin del Caso

Tabla 58: Especificación de C. U. Nuevo Gramaje

Especificación de Caso de Uso: Modificar Gramaje

Caso de Uso: Modificar Gramaje
Actor: Administrador
1. El administrador hace clic en el botón Modificar del Gramaje a Modificar
2. El sistema devuelve una pantalla para Modificar datos del Gramaje seleccionado.
3. El administrador modifica datos del Gramaje.
4. El sistema guarda los datos del Gramaje modificado, vuelve a la pantalla Administrar Gramajes.
5. El administrador visualiza la lista actualizada de Gramajes
6. Fin del Caso

Tabla 59: Especificación de C. U. Modificar Gramaje

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
5. El administrador hace clic en el botón Cambiar Estado del Gramaje deseado.
6. El sistema cambia el estado del Gramaje.
7. El Administrador visualiza la Lista Actualizada de Gramajes
8. Fin del Caso

Tabla 60: Especificación de C. U. Cambiar Estado

Administrar Máquinas

Especificación de Caso de Uso: Nueva Máquina

Caso de Uso: Nueva Máquina
Actor: Administrador
1. El administrador ingresa los datos de la Nueva Máquina.
2. El sistema guarda los datos de la Nueva Máquina.
3. El administrador visualiza la lista actualizada de Máquinas
4. Fin del Caso

Tabla 61: Especificación de C. U. Nueva Máquina

Especificación de Caso de Uso: Modificar Máquina

Caso de Uso: Modificar Máquina
Actor: Administrador
1. El administrador hace clic en el botón Modificar de la Máquina a Modificar
2. El sistema devuelve una pantalla para Modificar datos de la Máquina seleccionada.
3. El administrador modifica datos de la Máquina.
4. El sistema guarda los datos de la Máquina modificada, vuelve a la pantalla Administrar Máquinas.
5. El administrador visualiza la lista actualizada de Máquinas
6. Fin del Caso

Tabla 62: Especificación de C. U. Modificar Máquina

Especificación de Caso de Uso: Cambiar Estado

Caso de Uso: Cambiar Estado
Actor: Administrador
1. El administrador hace clic en el botón Cambiar Estado de la Máquina deseada.
2. El sistema cambia el estado de la Máquina
3. El Administrador visualiza la Lista Actualizada de Máquinas
4. Fin del Caso

Tabla 63: Especificación de C. U. Cambiar Estado

II.1.9. VISION

II.1.9.1. Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las características del Sistema Web desarrollado para la Administración de ventas de la imprenta CORPOGRÁFIKA LTDA.. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

II.1.9.2. Alcance

El documento visión se ocupa, como ya se ha apuntado, del Sistema Web desarrollado para la Administración de ventas de la imprenta CORPOGRÁFIKA LTDA. Dicho sistema será desarrollado por la Universitaria: JakelineRodriguez Balderrama. Alumna de quinto año de la carrera de Ingeniería Informática, facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho (U.A.J.M.S.) Tarija.

El sistema permitirá a los Administradores de la imprenta, controlar todo lo referente gestión de usuarios y clientes, ventas, reportes y almacenes.

II.1.9.3. Definiciones Acrónicas

RUP: Son las siglas de Racional UnifiedProcess. Se trata de una metodología para describir el proceso de desarrollo del software.

II.1.9.4. Referencias

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

II.1.10. MODELADO DE DIAGRAMAS DE ACTIVIDADES

II.1.10.1. Introducción

- El Diagrama de Actividades es un artefacto de la disciplina Requisitos en la metodología RUP la cual estamos implementando.
- Los diagramas de actividad se utilizan para modelar los aspectos dinámicos de un sistema, esto implica modelar los pasos secuenciales de un proceso.

II.1.10.2. Propósito

- Comprender la estructura y la dinámica del sistema deseado para la imprenta.
- Identificar posibles mejoras en el sistema.

II.1.10.3. Alcance

- Describe los procesos de sistema y los clientes.
- Identificar y definir los *procesos de los casos de uso* según los *objetivos* de la imprenta.

II.1.10.4. Diagrama de Actividades

Diagrama de Actividad: Administrar Usuarios

Diagrama de Actividad: Nuevo Usuario

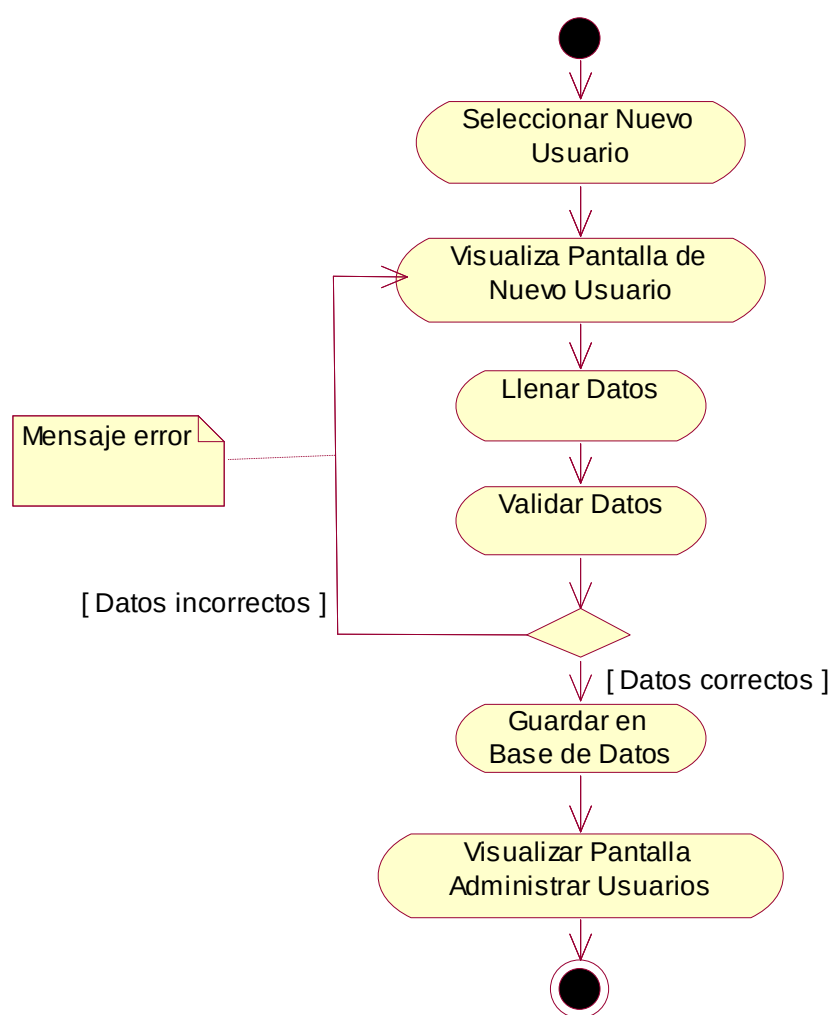


Figura 20. D.A. Nuevo Usuario

Diagrama de Actividad: Modificar Usuario

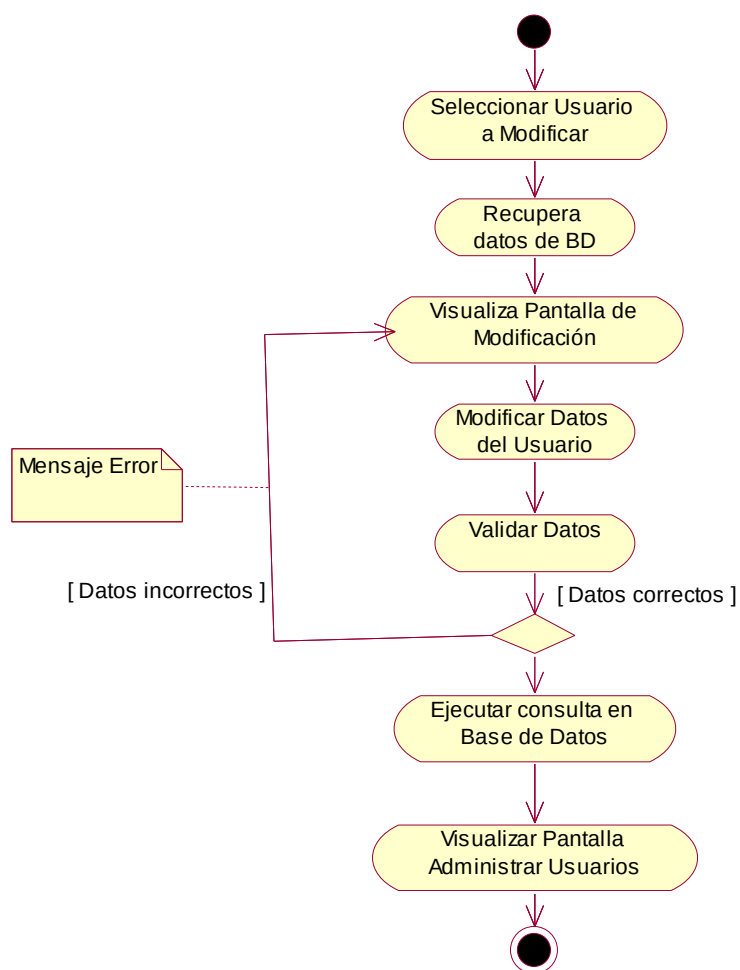


Figura 21. D.A. Modificar Usuario

Diagrama de Actividad: Cambiar estado

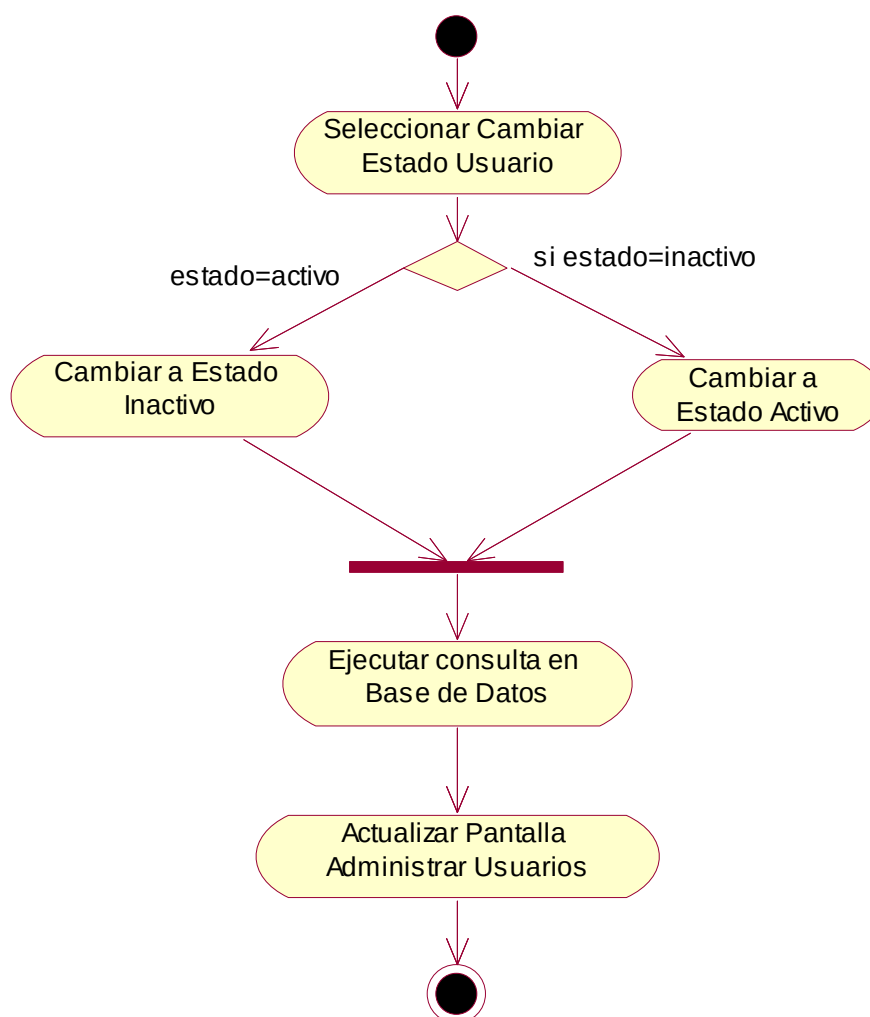


Figura 22.D.A. Cambiar Estado

Diagrama de Actividad: Cambiar Clave

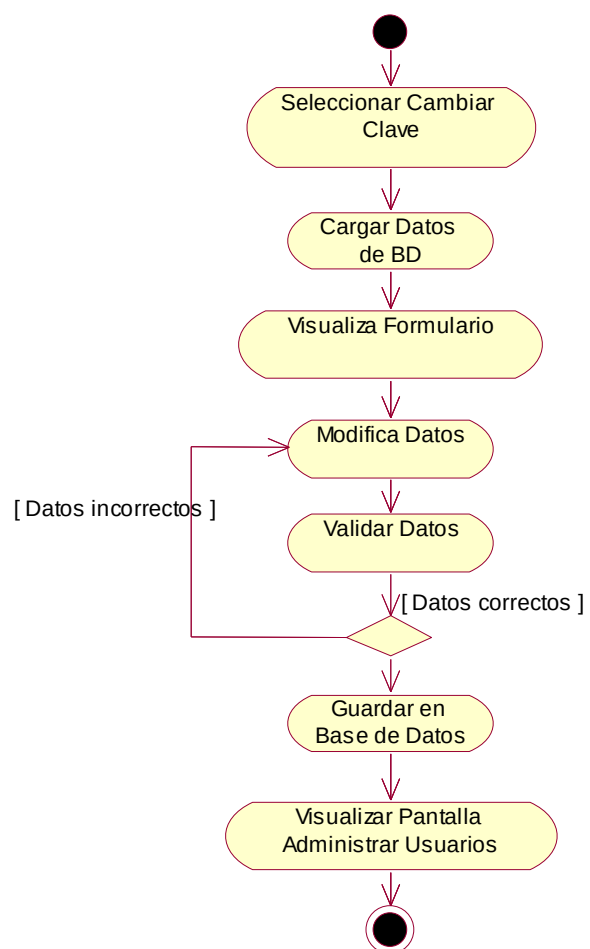


Figura 23.D.A. Cambiar Clave

Diagrama de Actividad: Ver Usuario

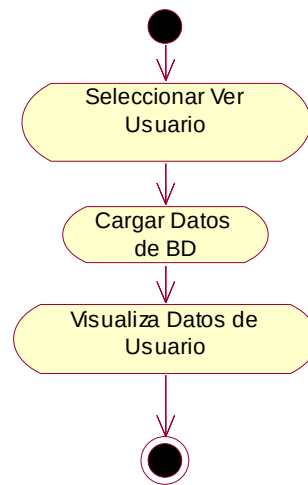


Figura 24. D.A. Ver Usuario

Administrar Roles

Diagrama de Actividad: Nuevo rol

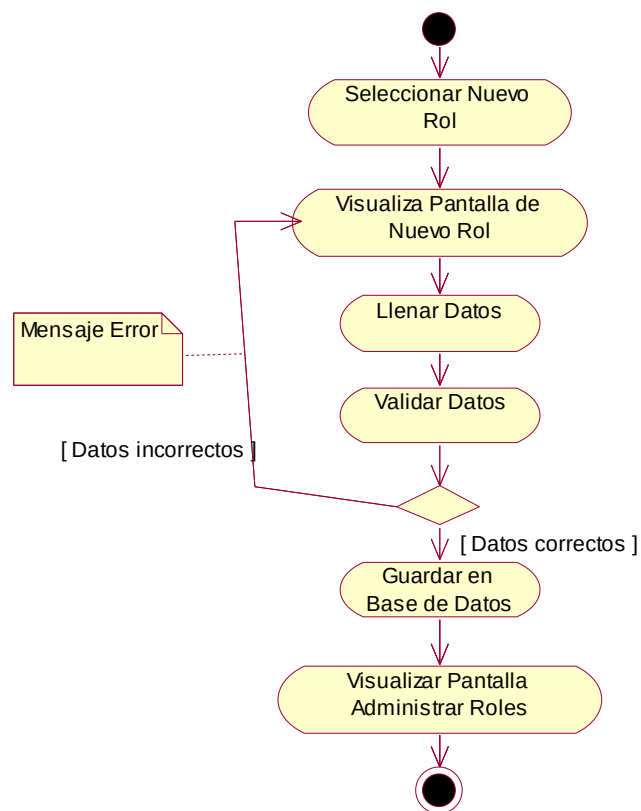


Figura 25. D.A. Nuevo Rol

Diagrama de Actividad: Modificar Rol

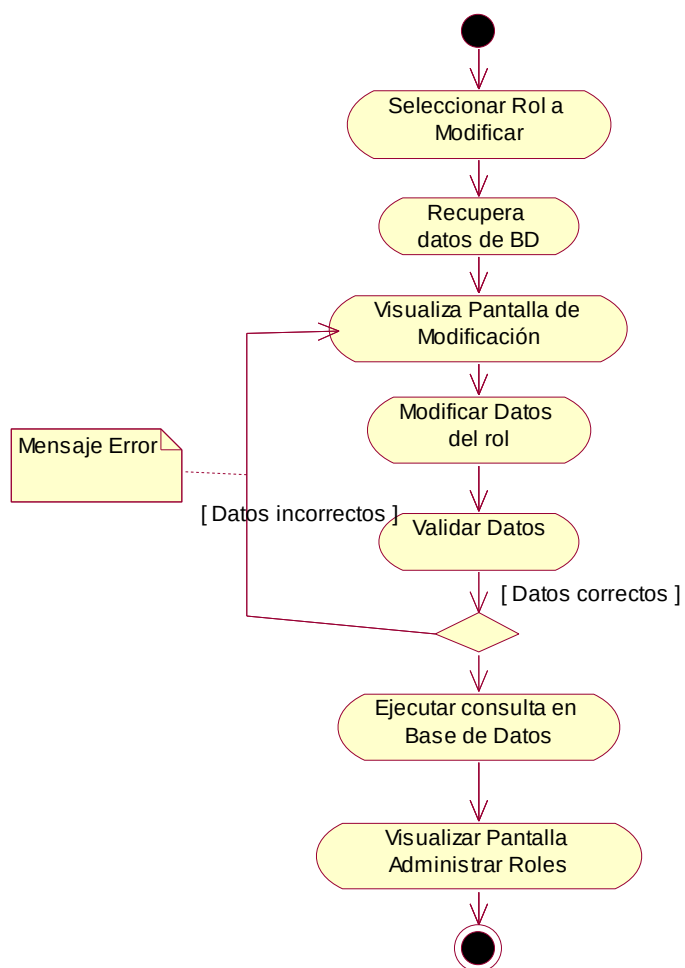


Figura 26. D.A. Modificar Rol

Diagrama de Actividad: Cambiar Estado

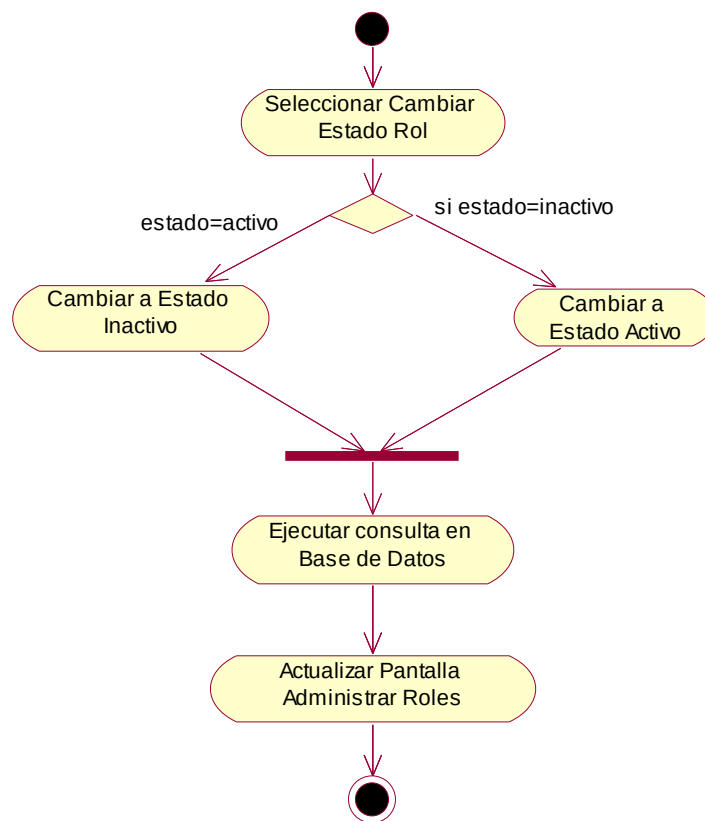


Figura 27.D.A. Cambiar Estado

Diagrama de Actividad: Ver Rol

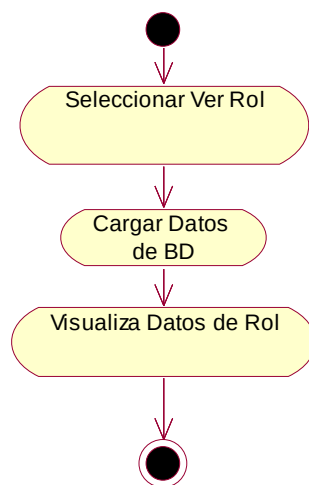


Figura 28.D.A. Ver Rol

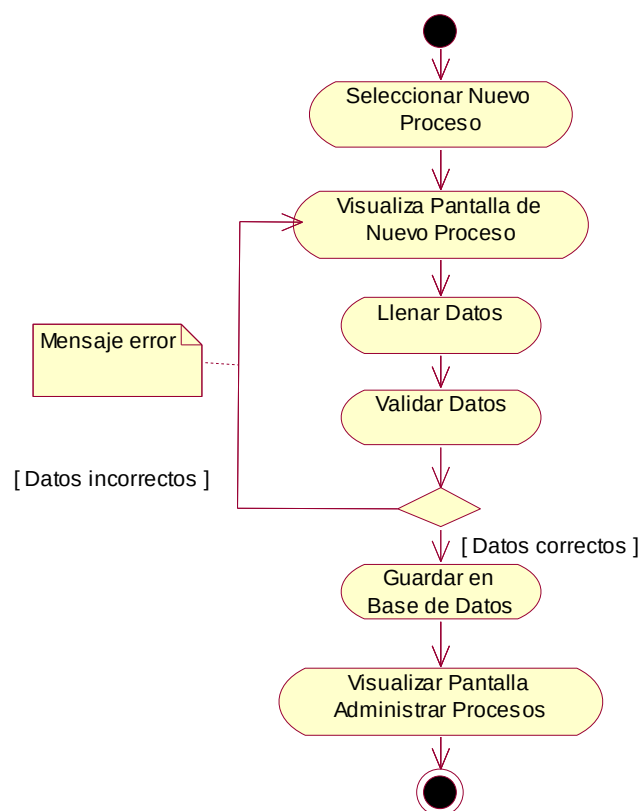
Diagrama de Actividad: Administrar Procesos***Diagrama de Actividad: Nuevo Proceso*****Figura 29. D.A. Nuevo Proceso**

Diagrama de Actividad: Modificar Proceso

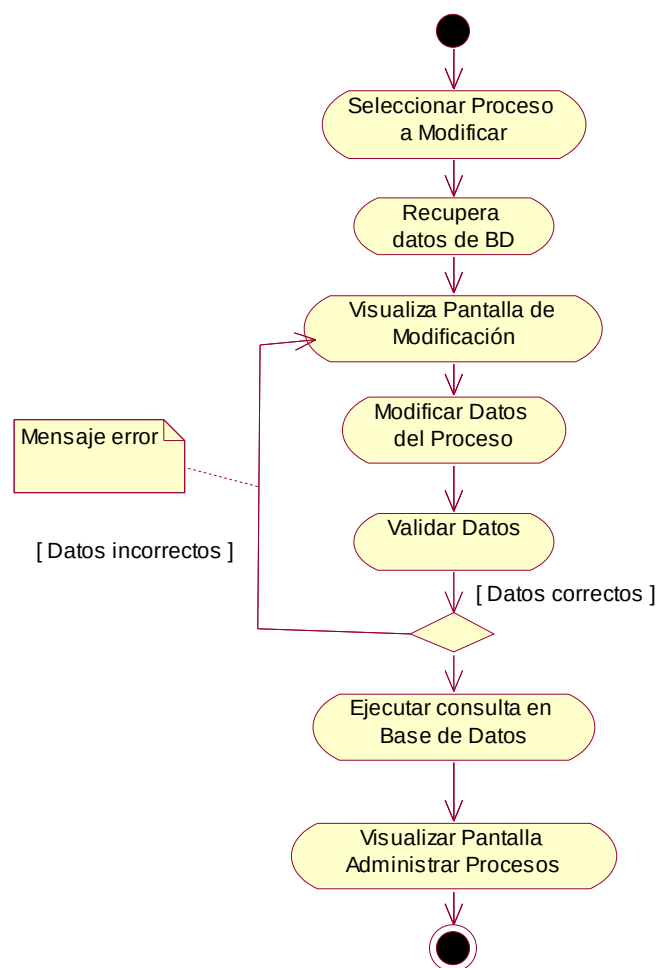


Figura 30. D.A. Modificar Proceso

Diagrama de Actividad: Cambiar Estado

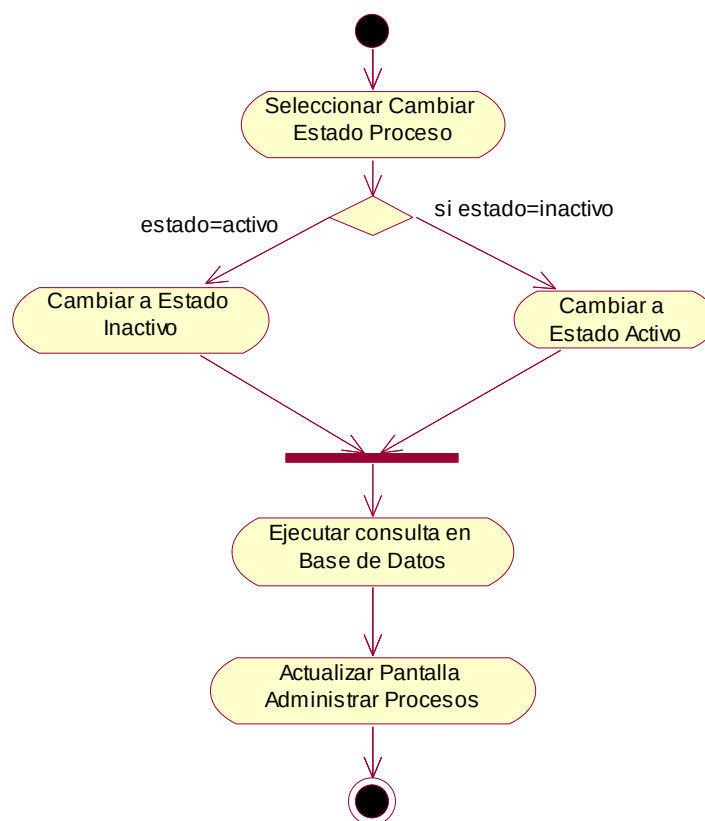


Figura 31.D.A. Cambiar Estado

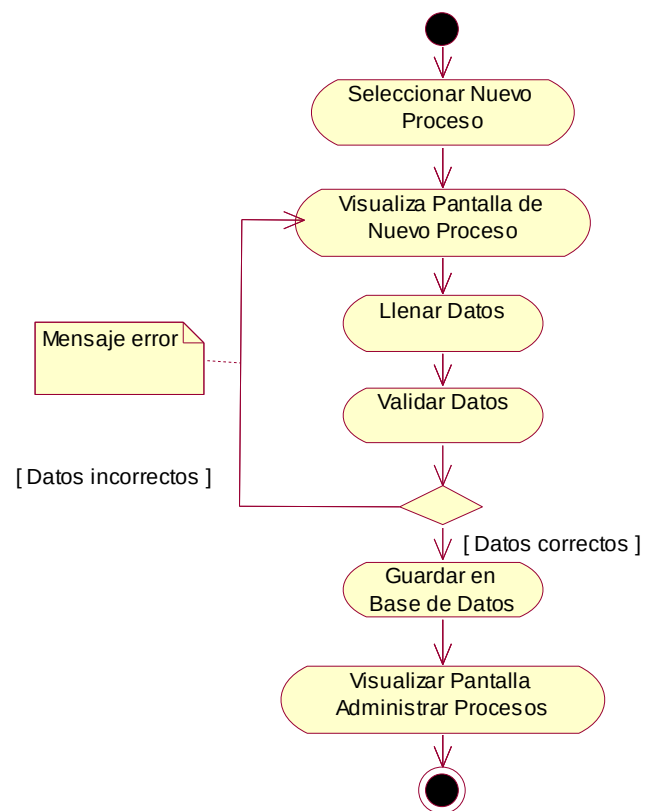
Diagrama de Actividad: Administrar Cliente***Diagrama de Actividad: Nuevo Cliente*****Figura 32. D.A. Nuevo Cliente**

Diagrama de Actividad: Modificar Cliente

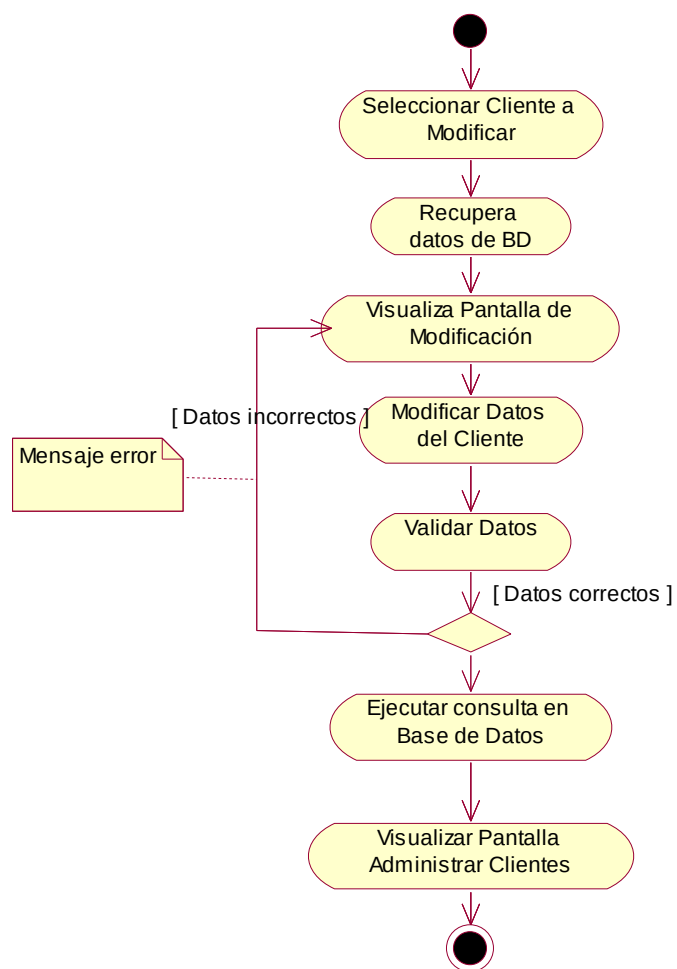


Figura 33. D.A. Modificar Cliente

Diagrama de Actividad: Cambiar Estado

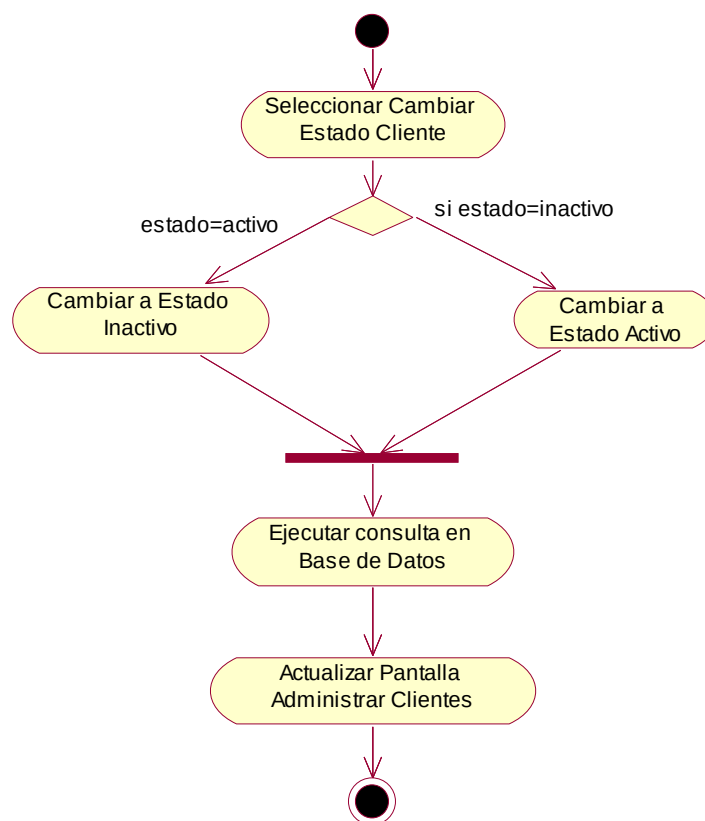


Figura 34. D.A. Cambiar Estado

Diagrama de Actividad: Ver Cliente

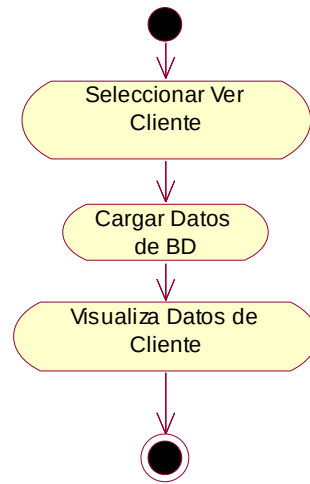


Figura 35.D. A. Ver Cliente

Administrar Proveedores

Diagrama de Actividad: Nuevo Proveedor

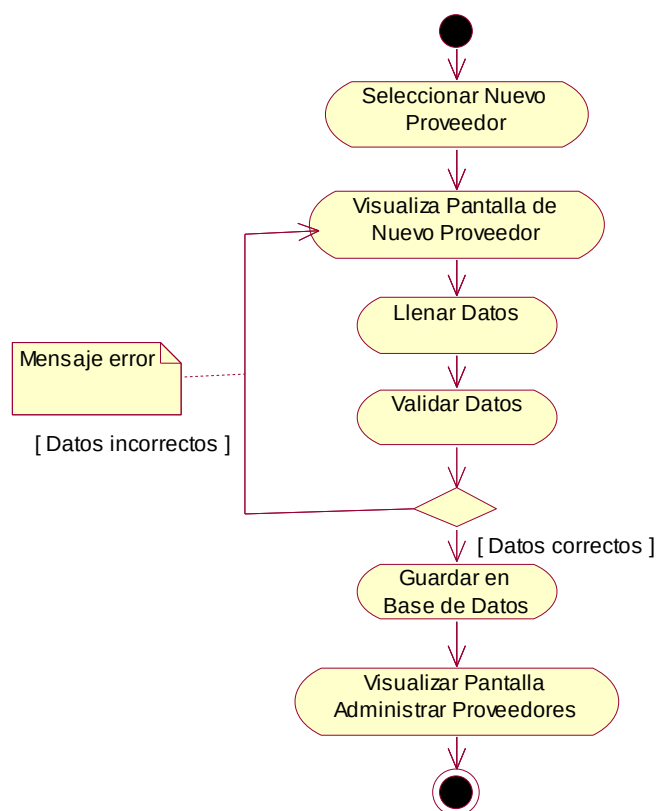


Figura 36. D.A. Nuevo Proveedor

Diagrama de Actividad: Modificar Proveedor

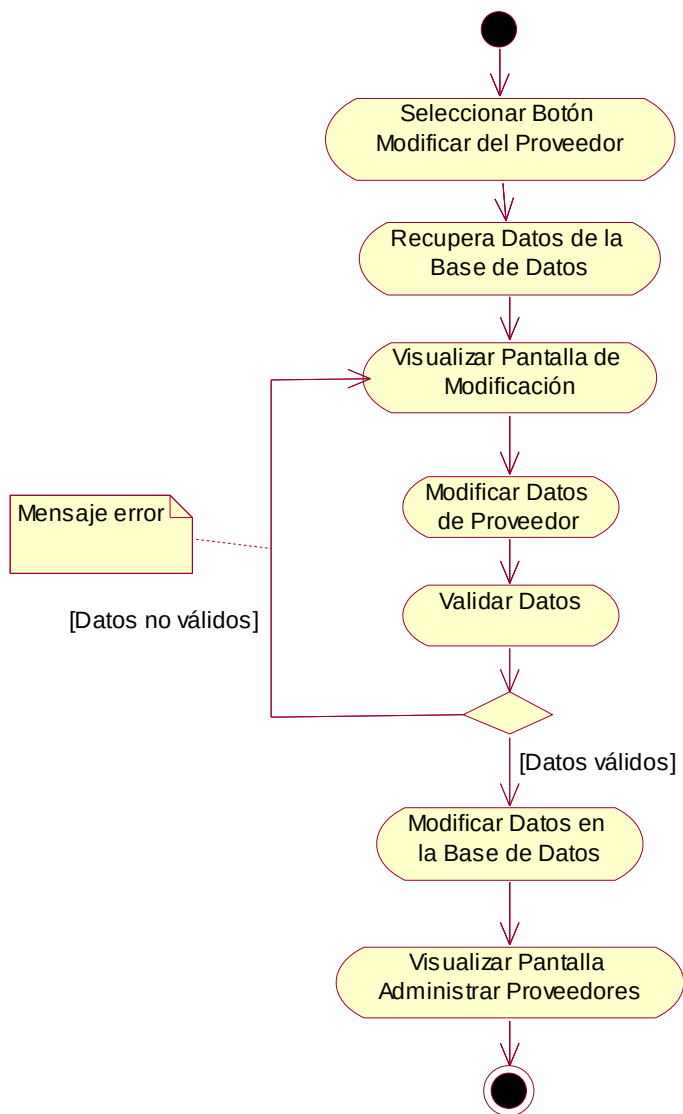


Figura 37. D.A. Modificar Proveedor

Diagrama de Actividad: Cambiar Estado

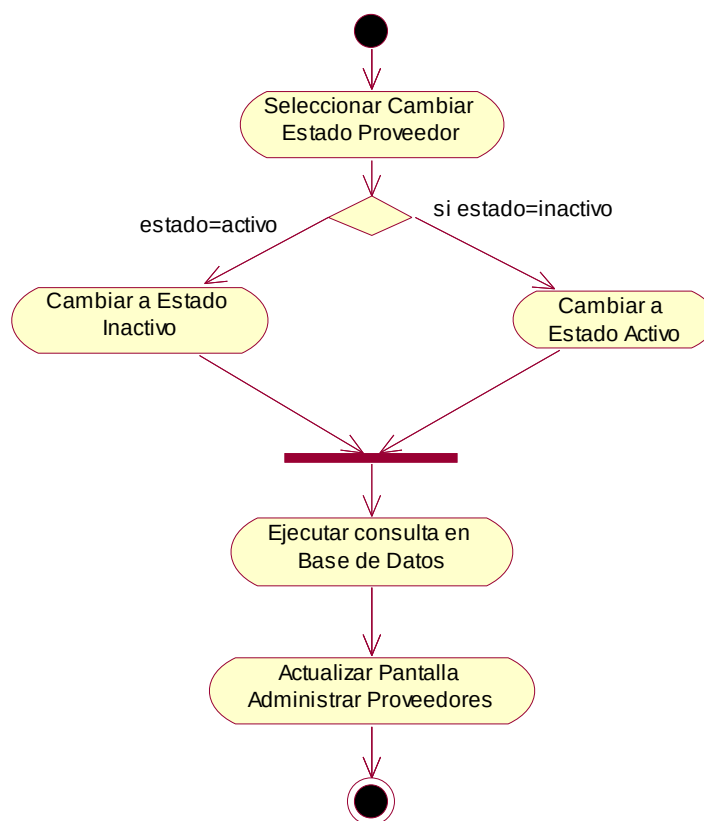


Figura 38.D.A. Cambiar Estado

Diagrama de Actividad: Ver Proveedor

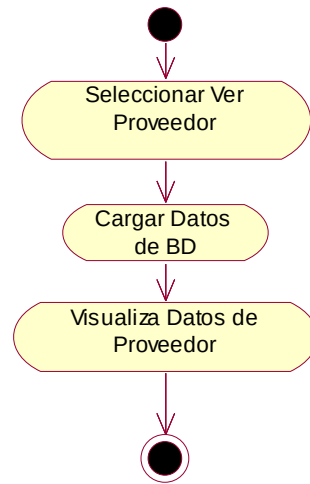


Figura 39.D.A. Ver Proveedor

Administrar Categoría Materiales

Diagrama de Actividad: Nueva Categoría Material

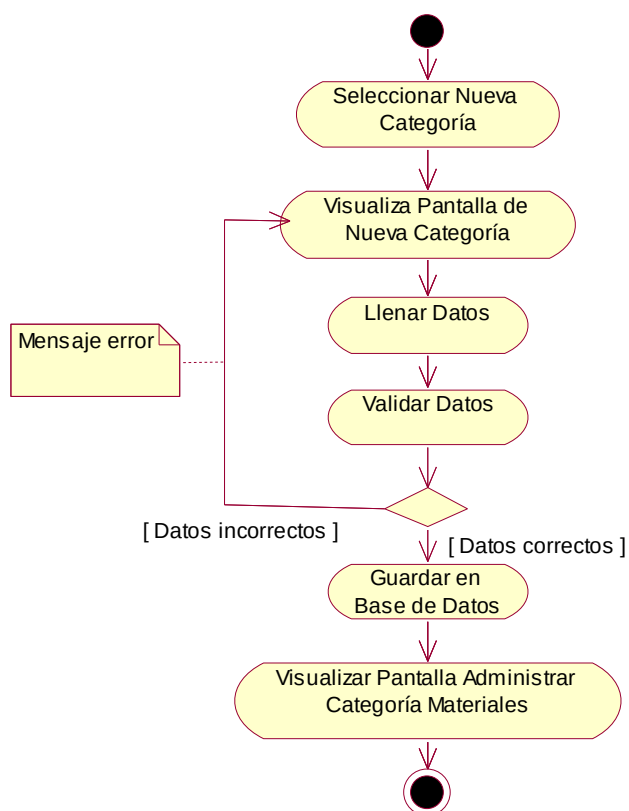


Figura 40. D.A. Nueva Categoría Material

Diagrama de Actividad: Modificar Categoría Material

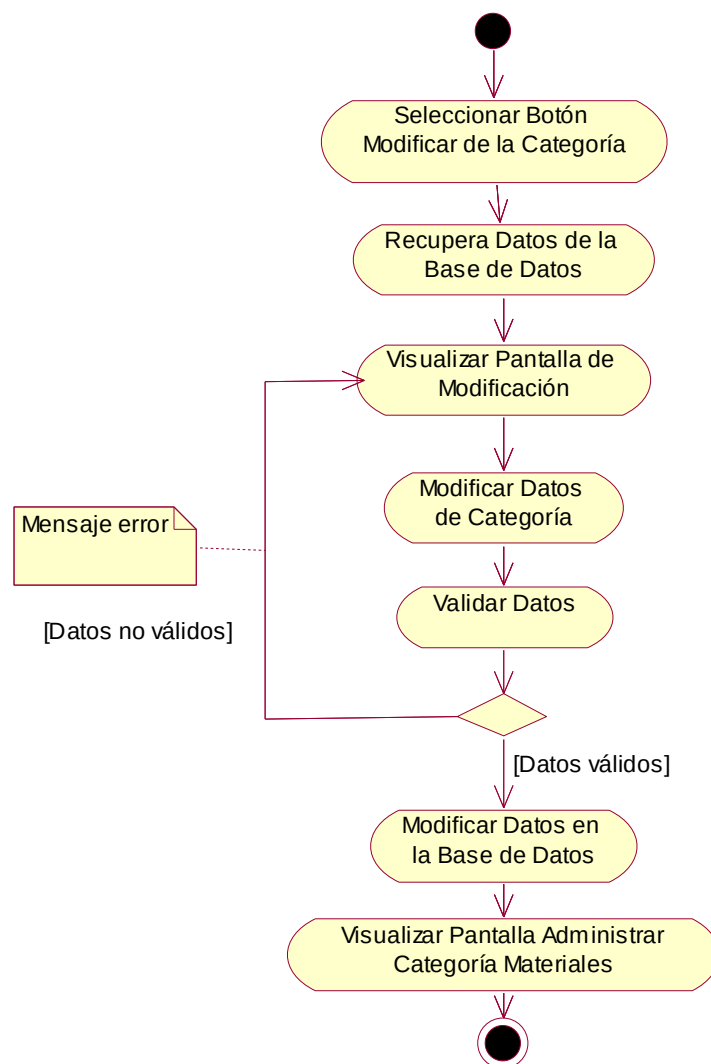


Figura 41. D.A. Modificar Categoría Material

Diagrama de Actividad: Cambiar Estado

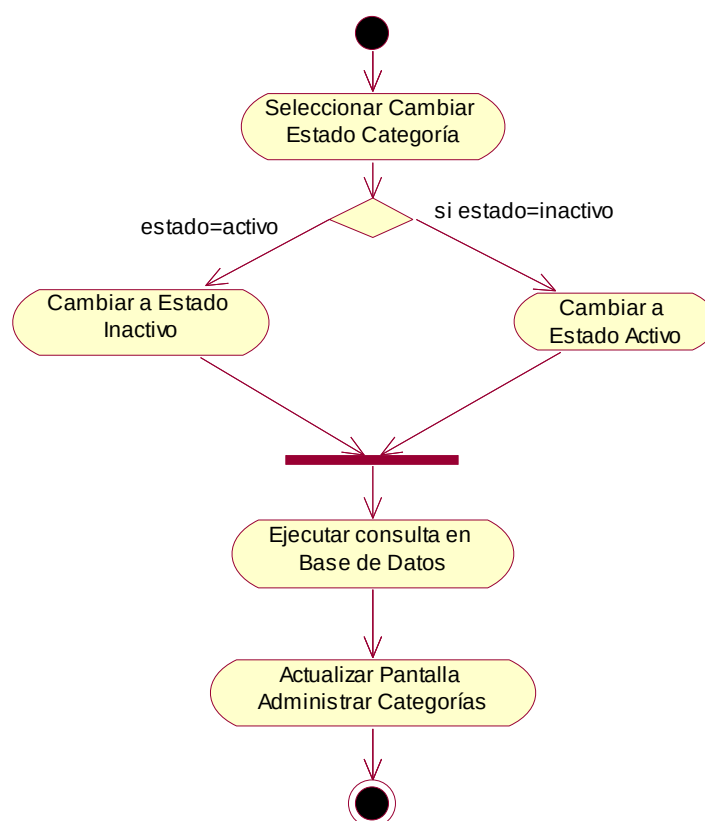


Figura 42.D.A. Cambiar Estado

Administrar Tipo Materiales

Diagrama de Actividad: Nuevo Tipo Material

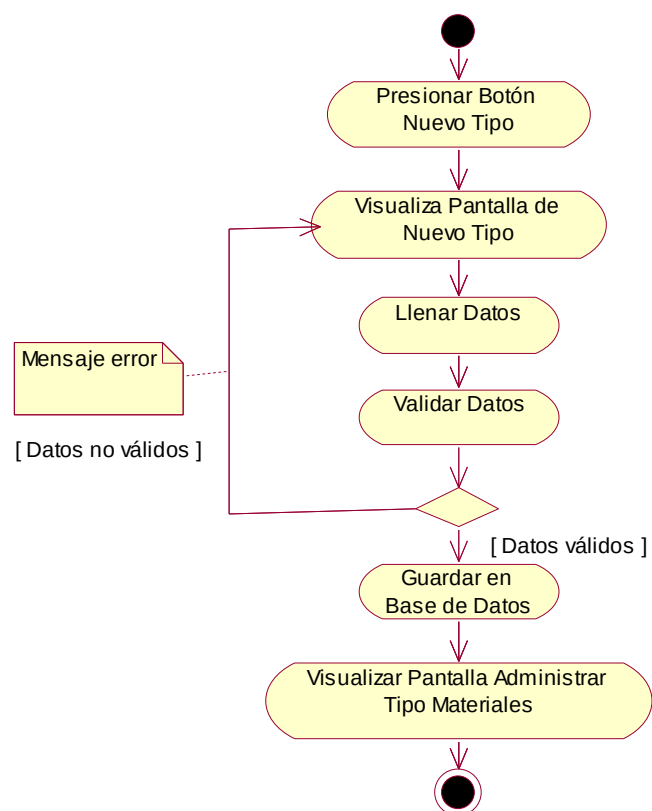


Figura 43.D.A Nuevo Tipo Material

Diagrama de Actividad: Modificar Tipo Material

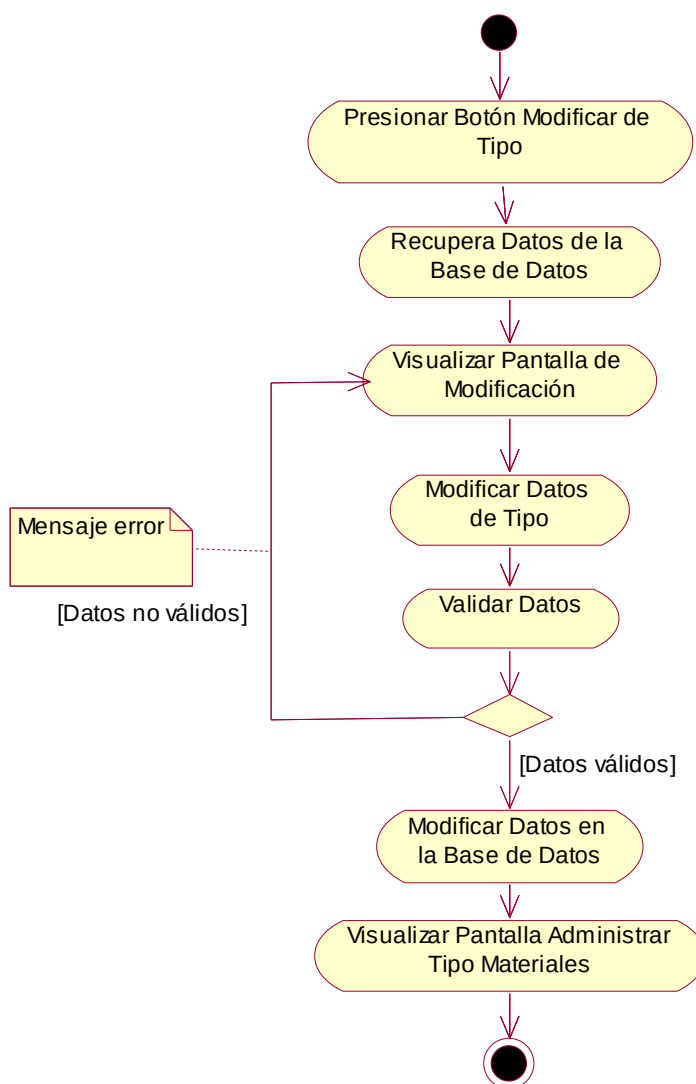


Figura 44. D.A Modificar Tipo Material

Diagrama de Actividad: Cambiar Estado

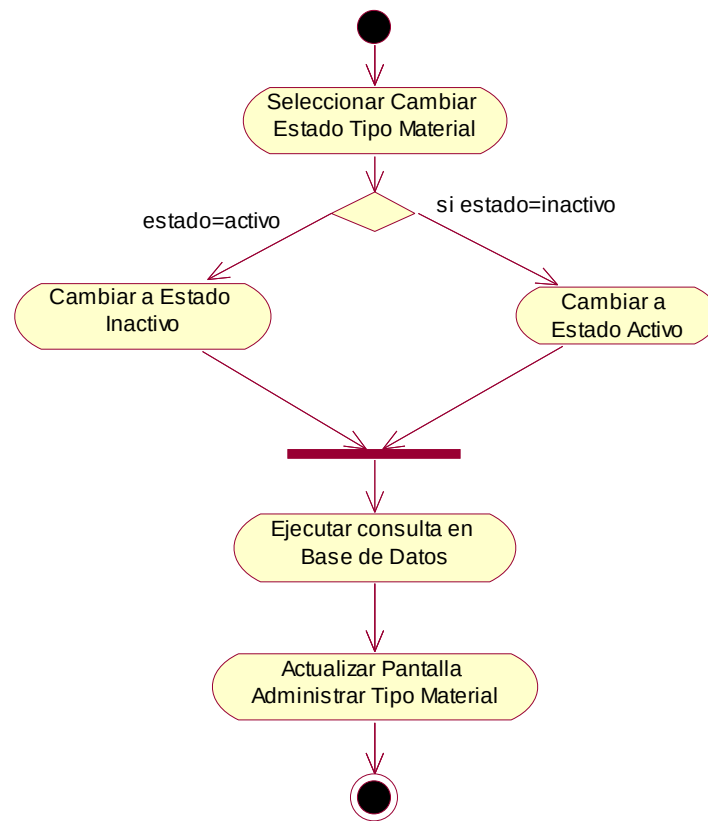


Figura 45.D.A. Cambiar Estado

Diagrama de Actividad: Ver Tipo Material

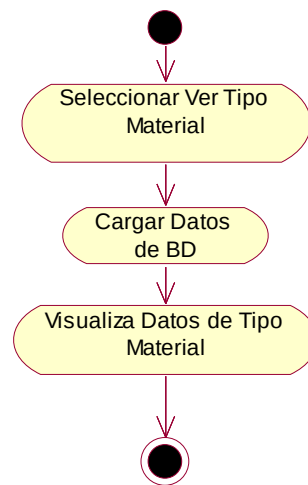


Figura 46.D.A. Ver Tipo Material

Administrar Notas

Diagrama de Actividad: Nueva Nota

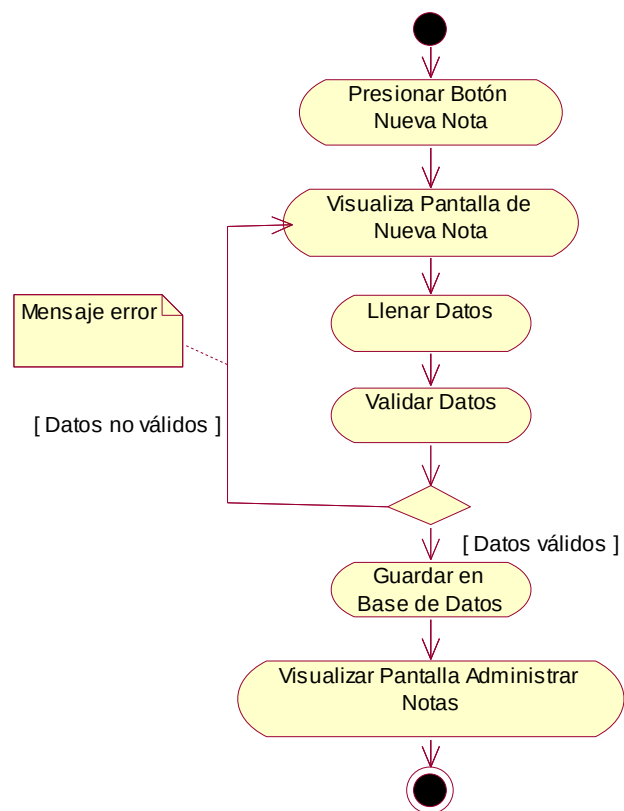


Figura 47. D.A. Nueva Nota

Diagrama de Actividad: Modificar Nota

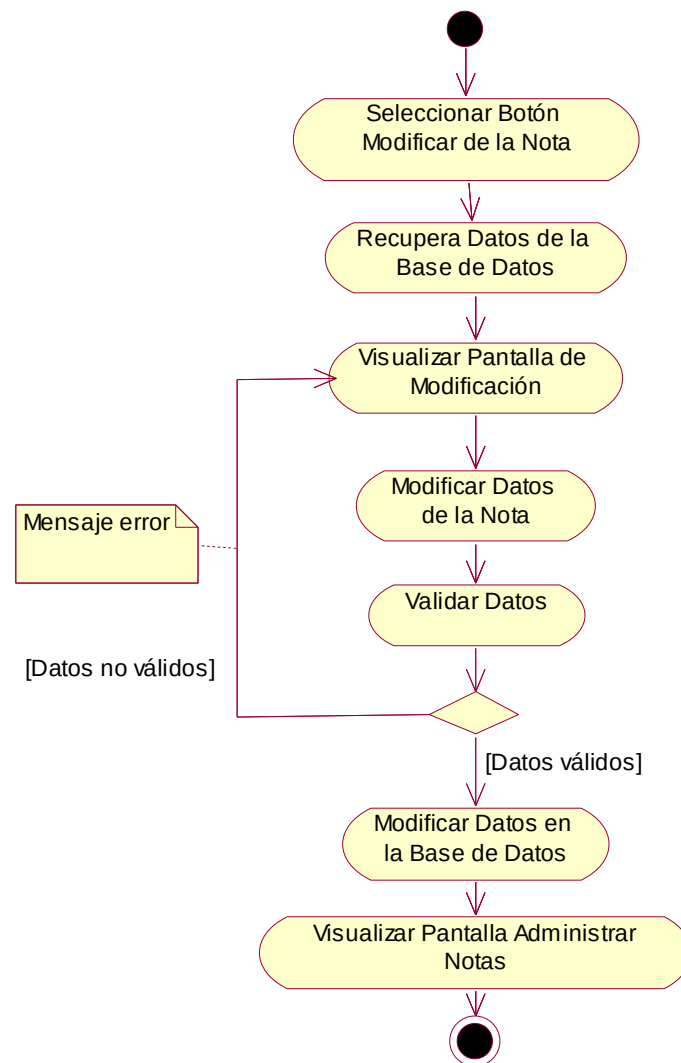


Figura 48. D.A. Modificar Nota

Diagrama de Actividad: Cambiar Estado

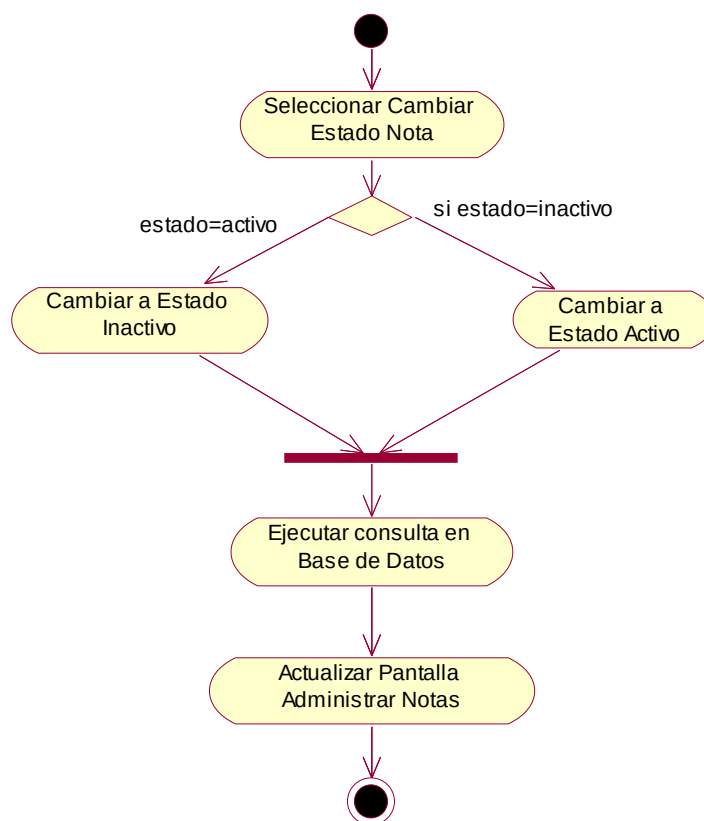


Figura 49.D.A. Cambiar Estado

Diagrama de Actividad: Ver Nota

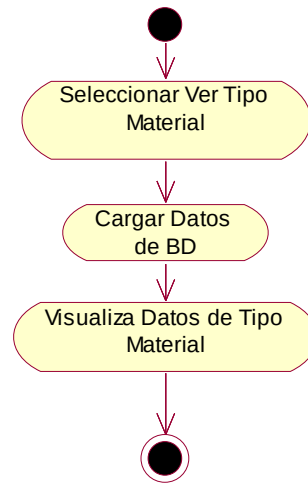


Figura 50.D.A. Ver Nota

Administrar Reportes

Diagrama de Actividad: Generar Reporte

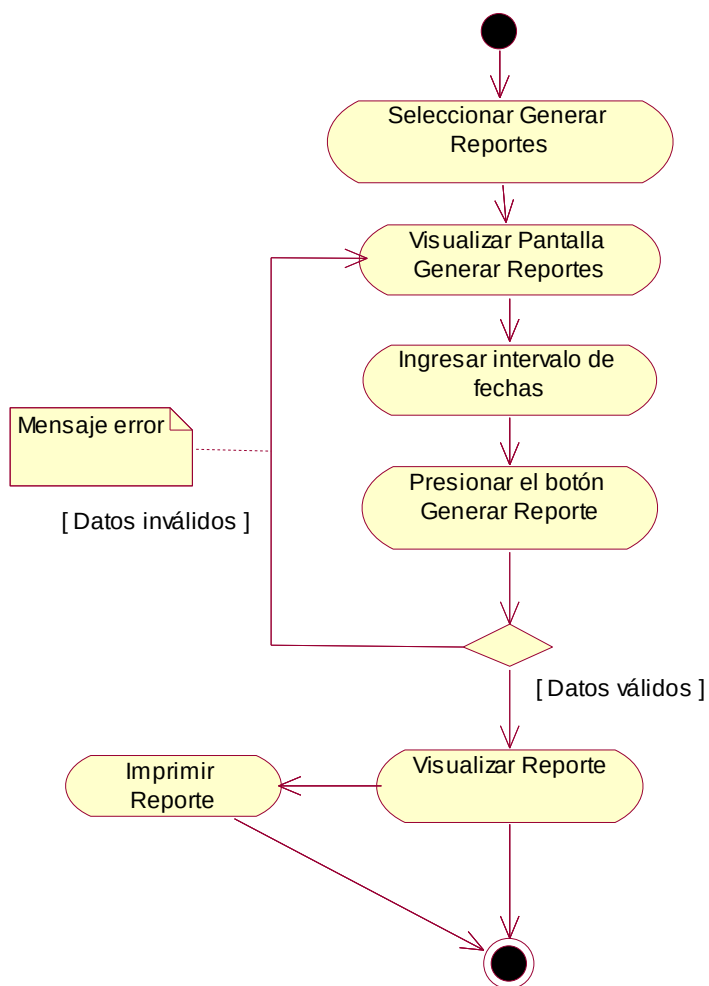


Figura 51. D.A. Generar Reporte

Administrar Compras

Diagrama de Actividad: Nueva Compra

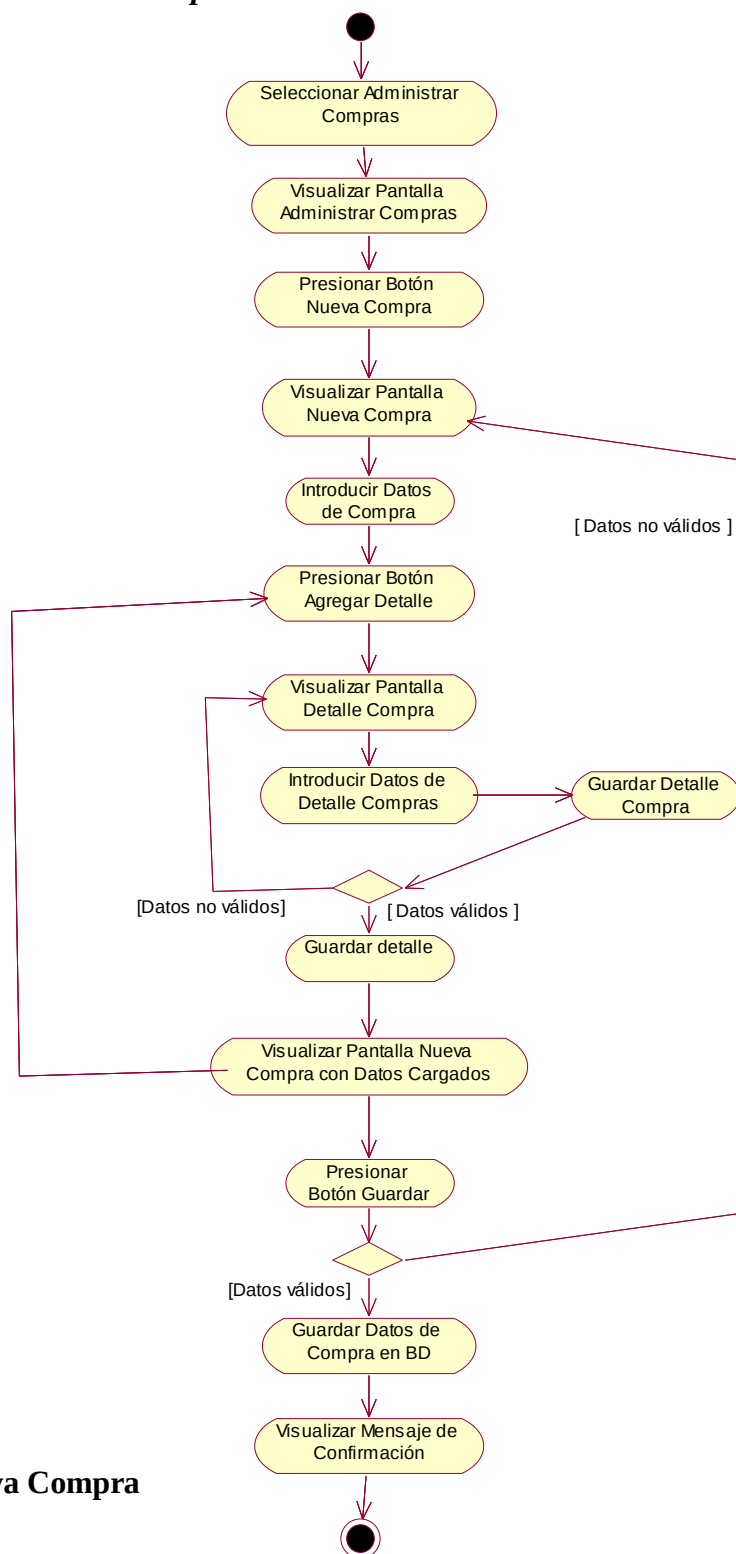


Figura 52. D.A. Nueva Compra

Diagrama de Actividad: Ver Detalle

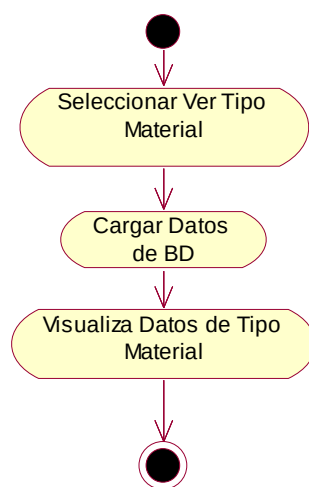


Figura 53.D.A. Ver Detalle

Administrar Ventas

Diagrama de Actividad: Nueva Venta

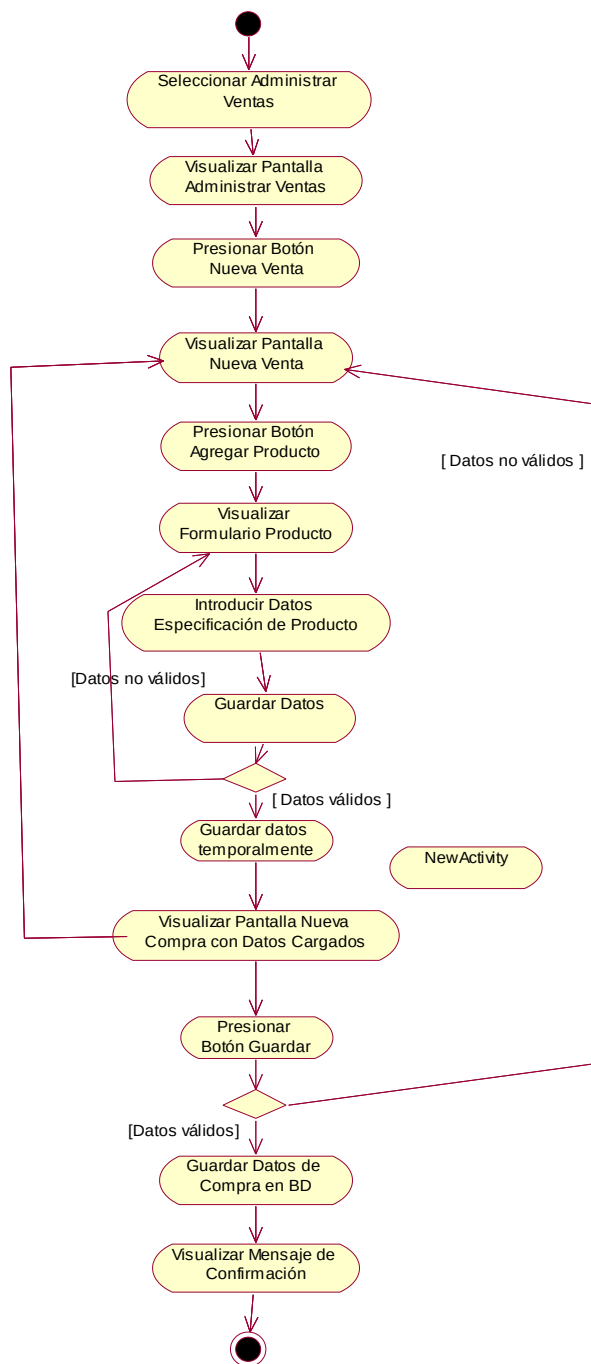


Figura 54.D.A. Nueva Venta

Diagrama de Actividad: Ver Detalle

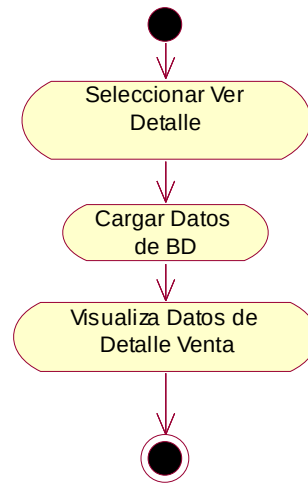


Figura 55.D.A. Ver Detalle

II.1.11. MODELADO DE DIAGRAMA DE INTERACCIÓN

II.1.11.1. Introducción

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario.

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

Los diagramas de secuencia, formalmente diagramas de traza de eventos o de interacción de objetos, se utilizan con frecuencia para validar los casos de uso documentan el diseño desde el punto de vista de los casos de uso. Observando que mensajes se envían a los objetos, componentes o casos de uso y viendo grosso modo cuanto tiempo consume el método invocado, los diagramas de secuencia nos ayudan a comprender los cuellos de botella potenciales, para así de esta manera poderlos eliminar. A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del diagrama de clases.

II.1.11.2. Propósito

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

II.1.11.3. Alcance

- Definir la dinámica del sistema en el tiempo de vida de las clases u objetos

II.1.11.4. Diagramas de Secuencia

Diagrama de secuencia: Ingresar al sistema

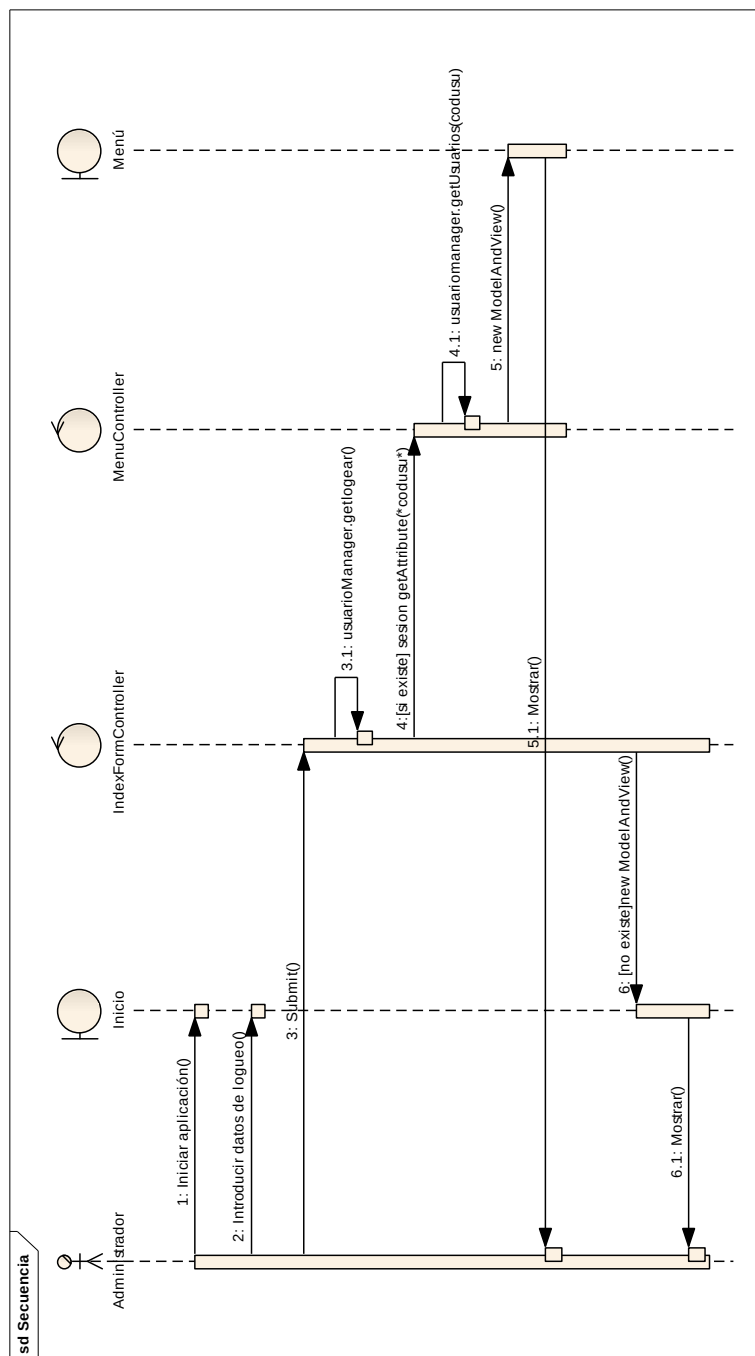


Figura 56. Ingresar al Sistema

Diagrama de Secuencia: Administrar Usuarios

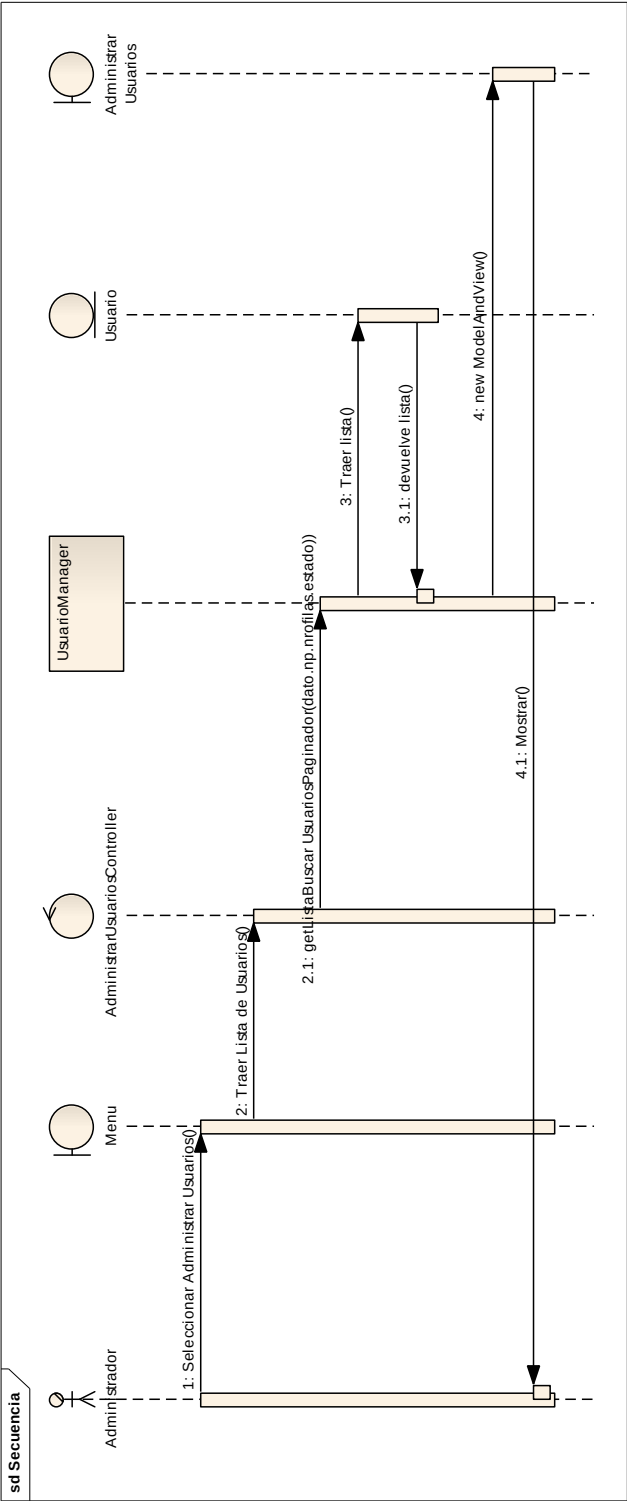


Figura 57. D.S. Administrar Usuarios

Diagrama de Secuencia: Nuevo Usuario

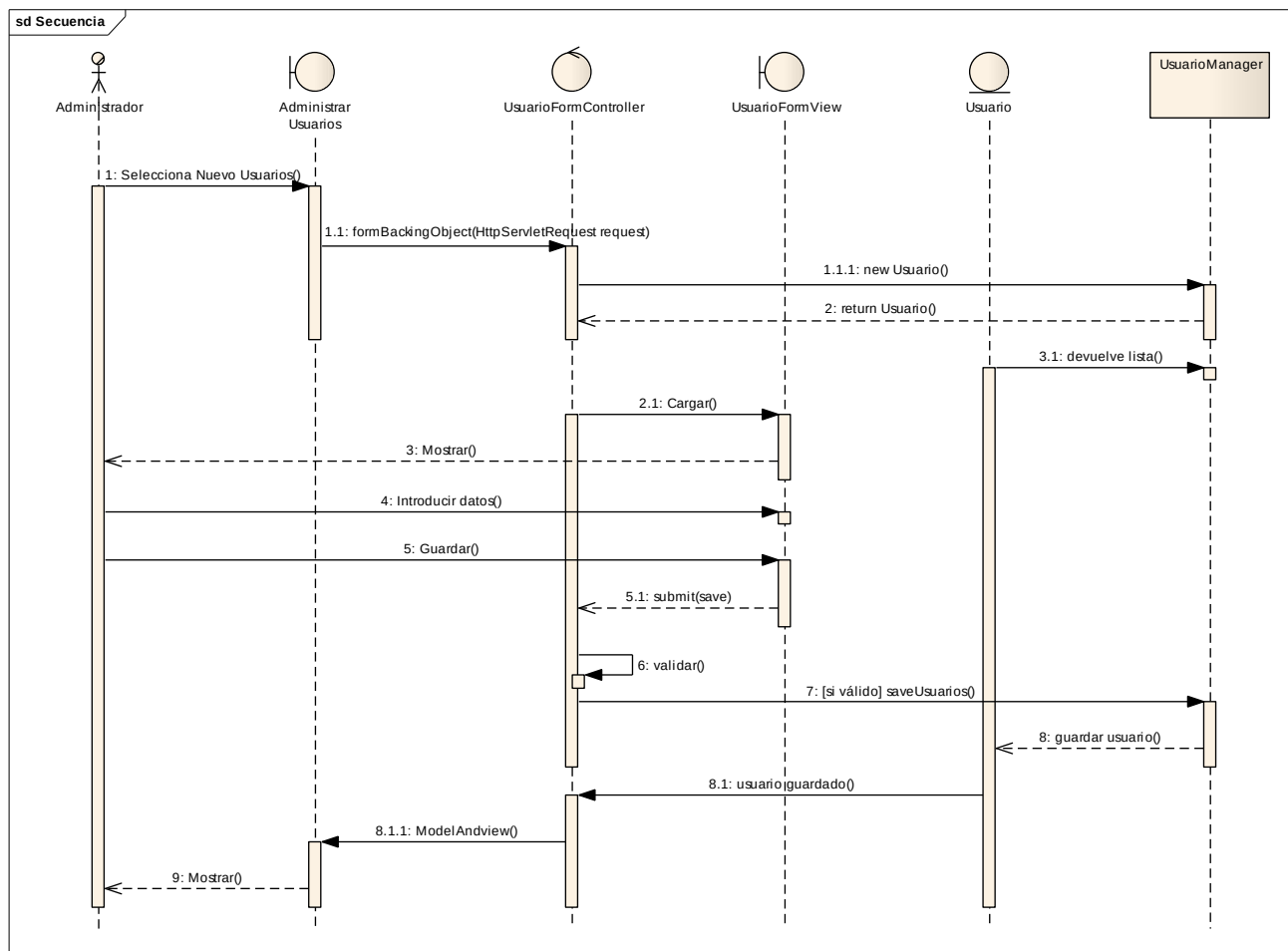


Figura 58. D.S. Nuevo Usuario

Diagrama de Secuencia: Modificar Usuario

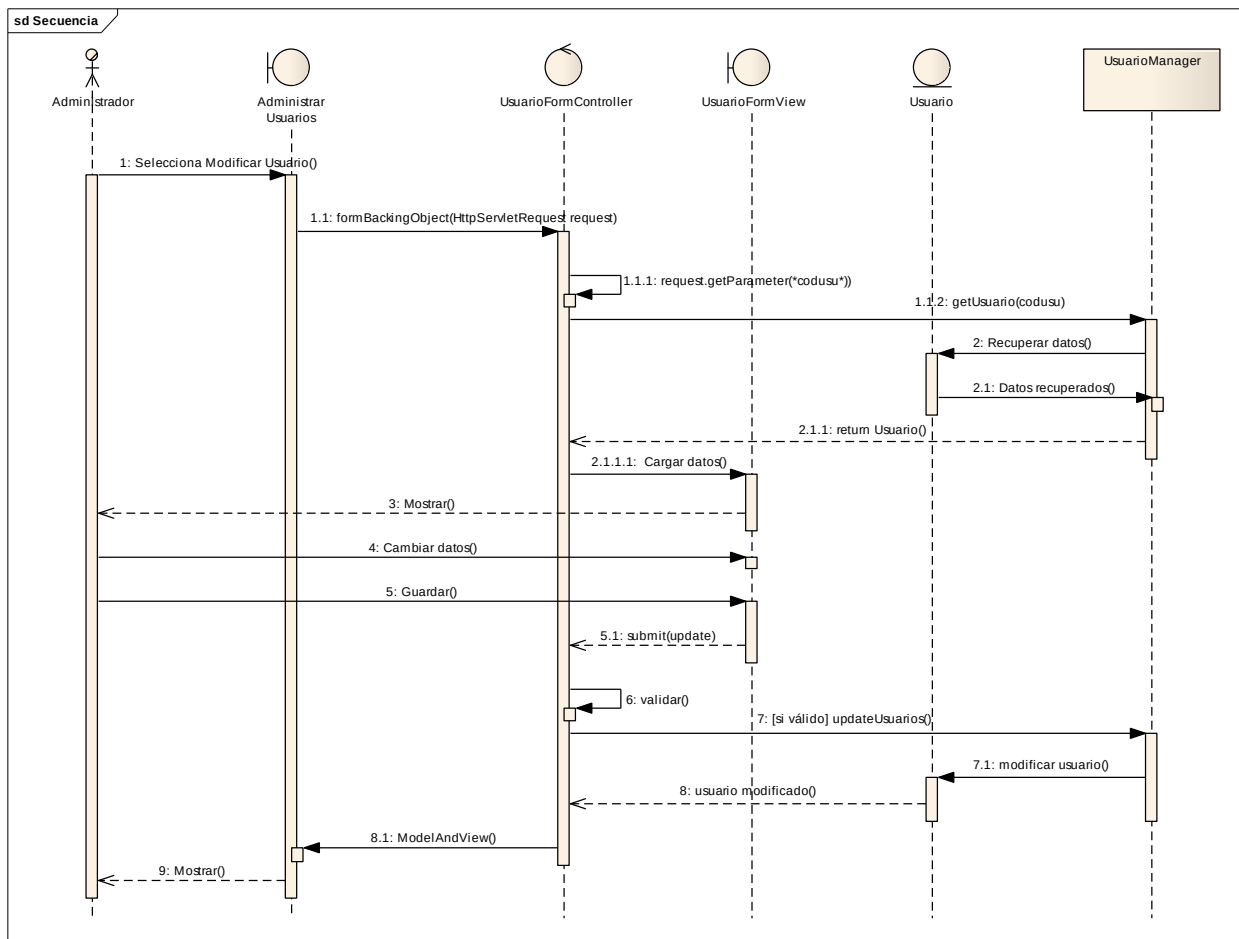


Figura 59. D.S. Modificar Usuario

Diagrama de Secuencia: Cambiar estado

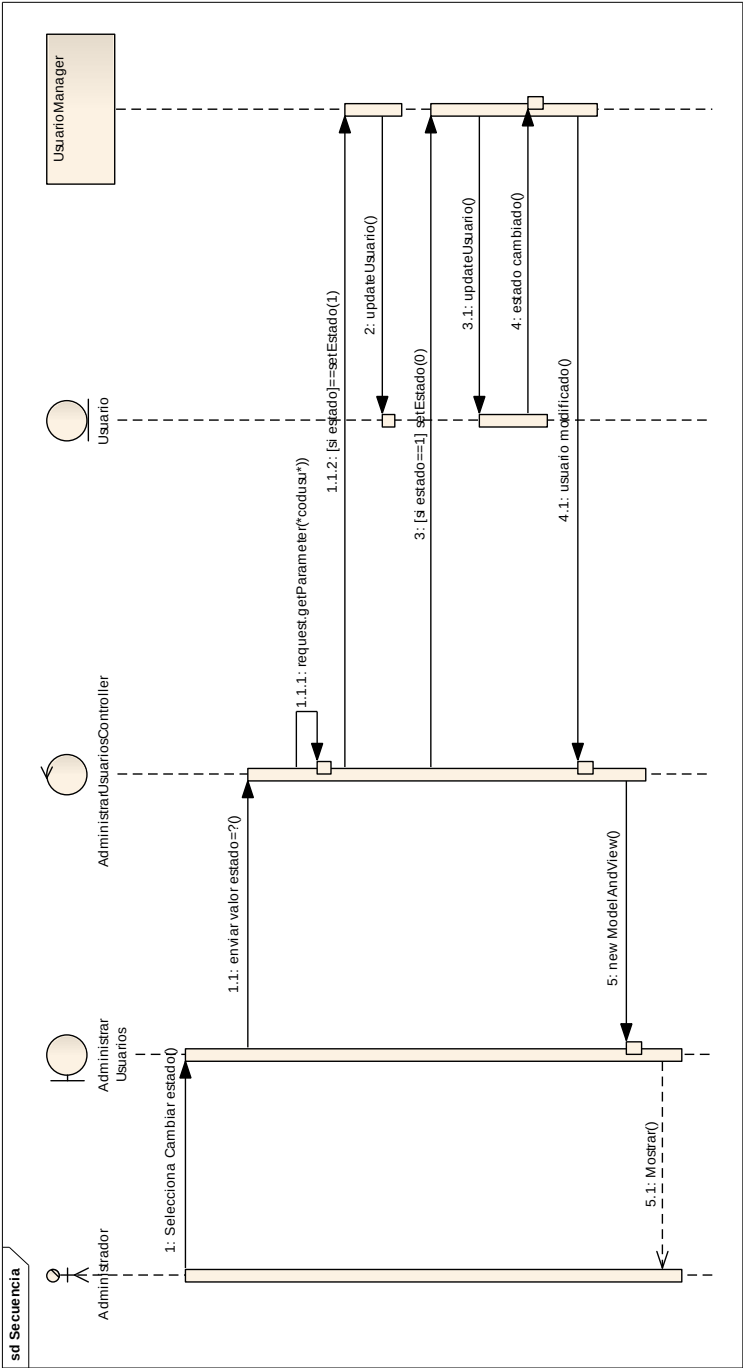


Figura 60. Cambiar estado

Diagrama de Secuencia: Administrar Roles

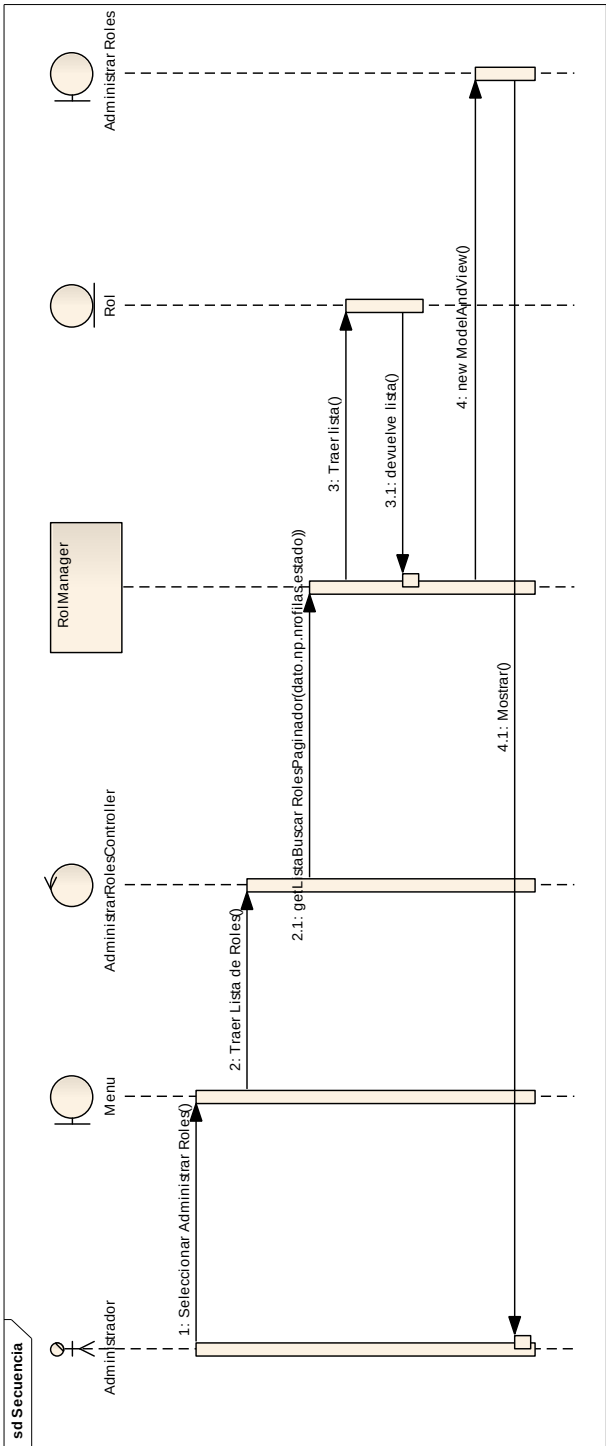


Figura 61. D.S. Administrar Roles

Diagrama de Secuencia: Nuevo Rol

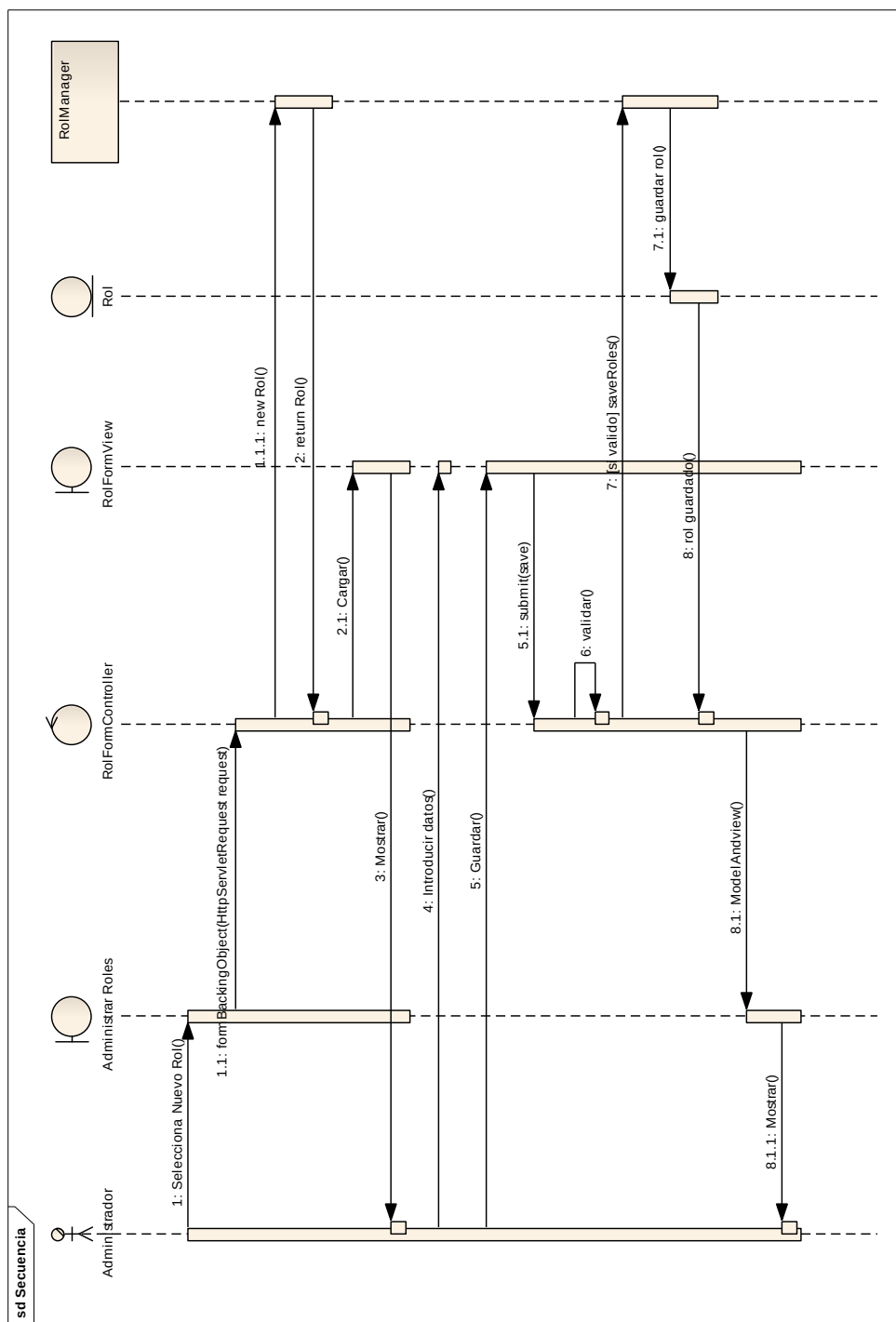


Figura 62. D.S. Nuevo Rol

Diagrama de Secuencia: Modificar Rol

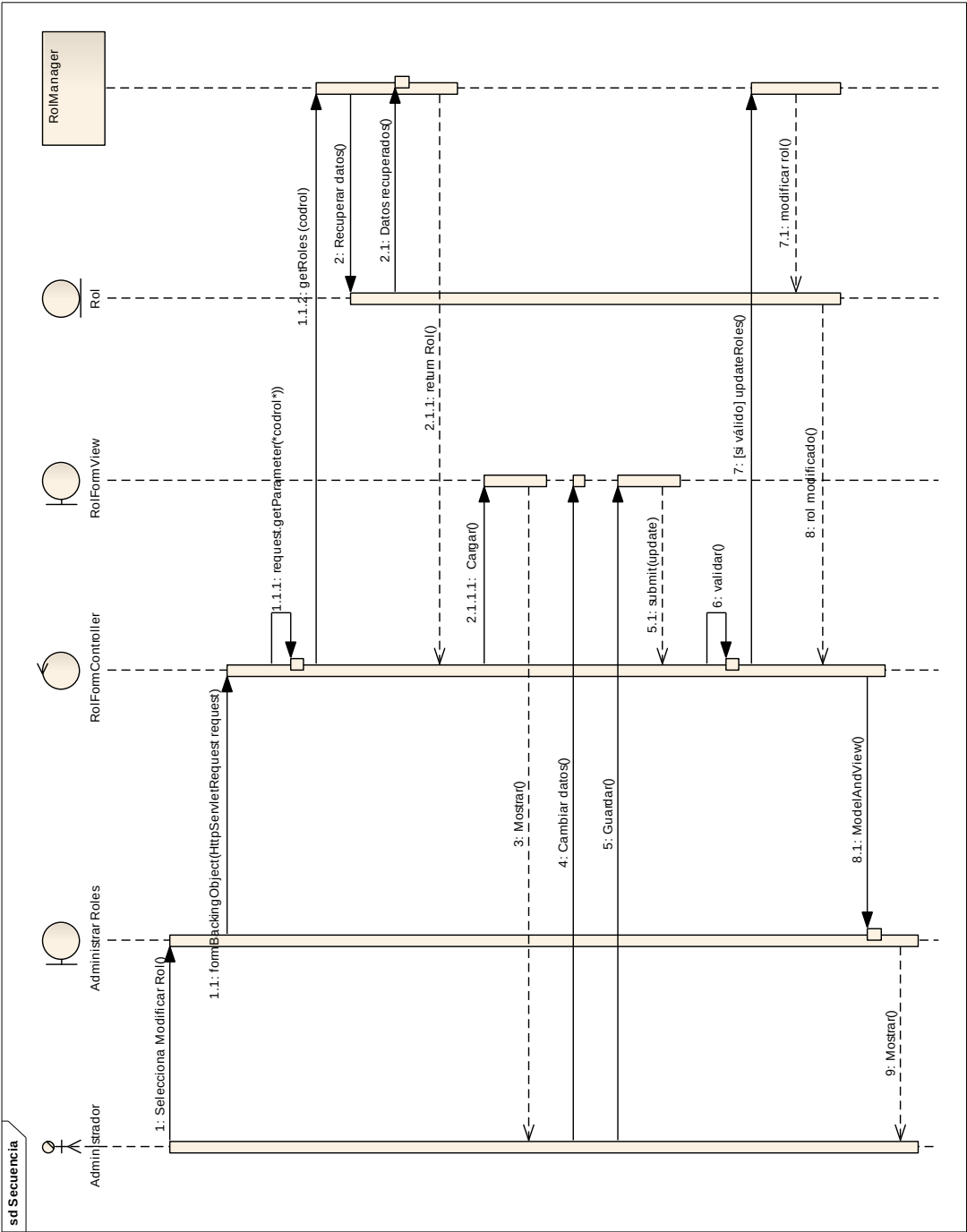


Figura 63. D.S. Modificar Rol

Diagrama de Secuencia: Administrar Procesos

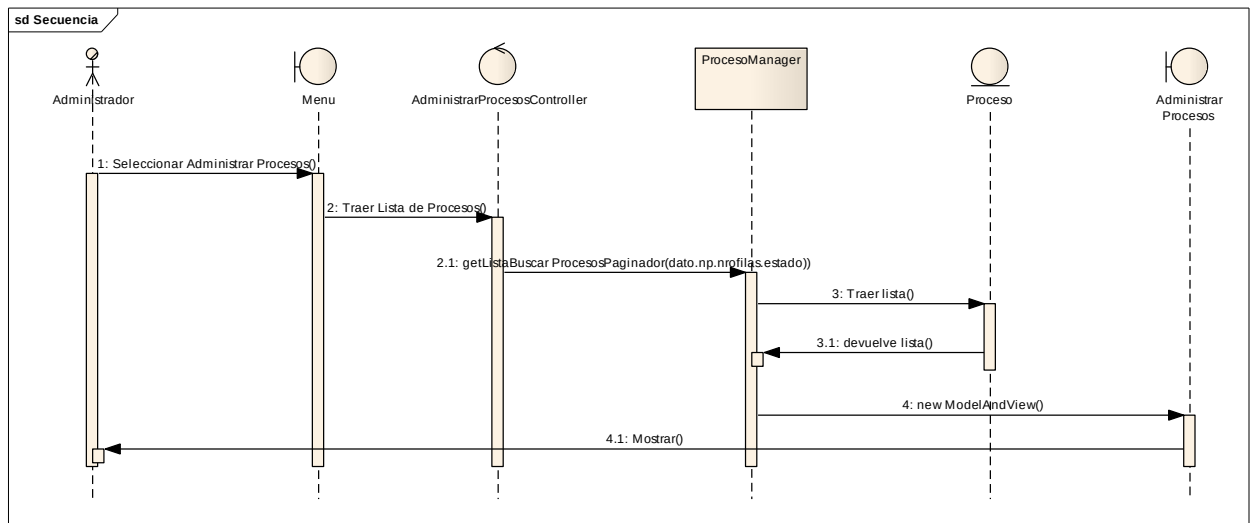


Figura 64. D.S. Administrar Procesos

Diagrama de Secuencia: Nuevo Proceso

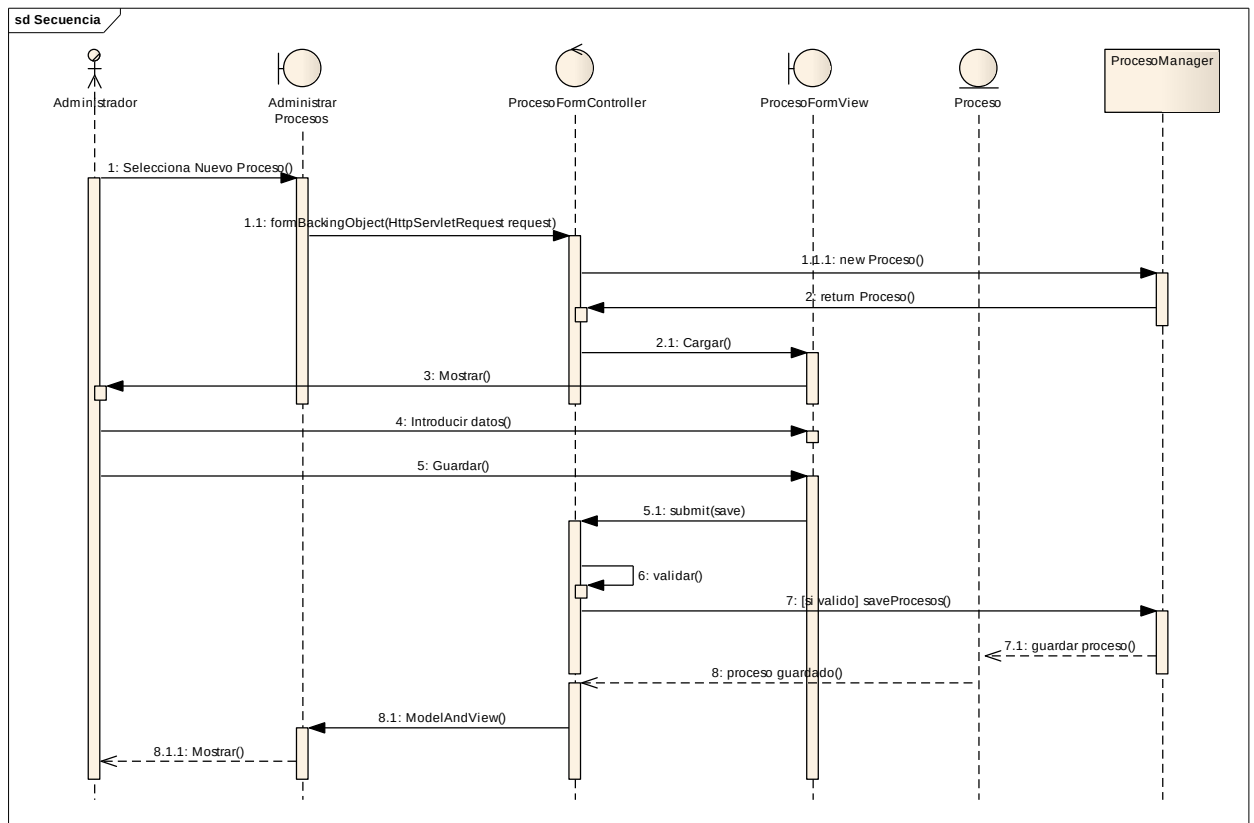


Figura 65. D.S. Nuevo Proceso

Diagrama de Secuencia: Modificar Proceso

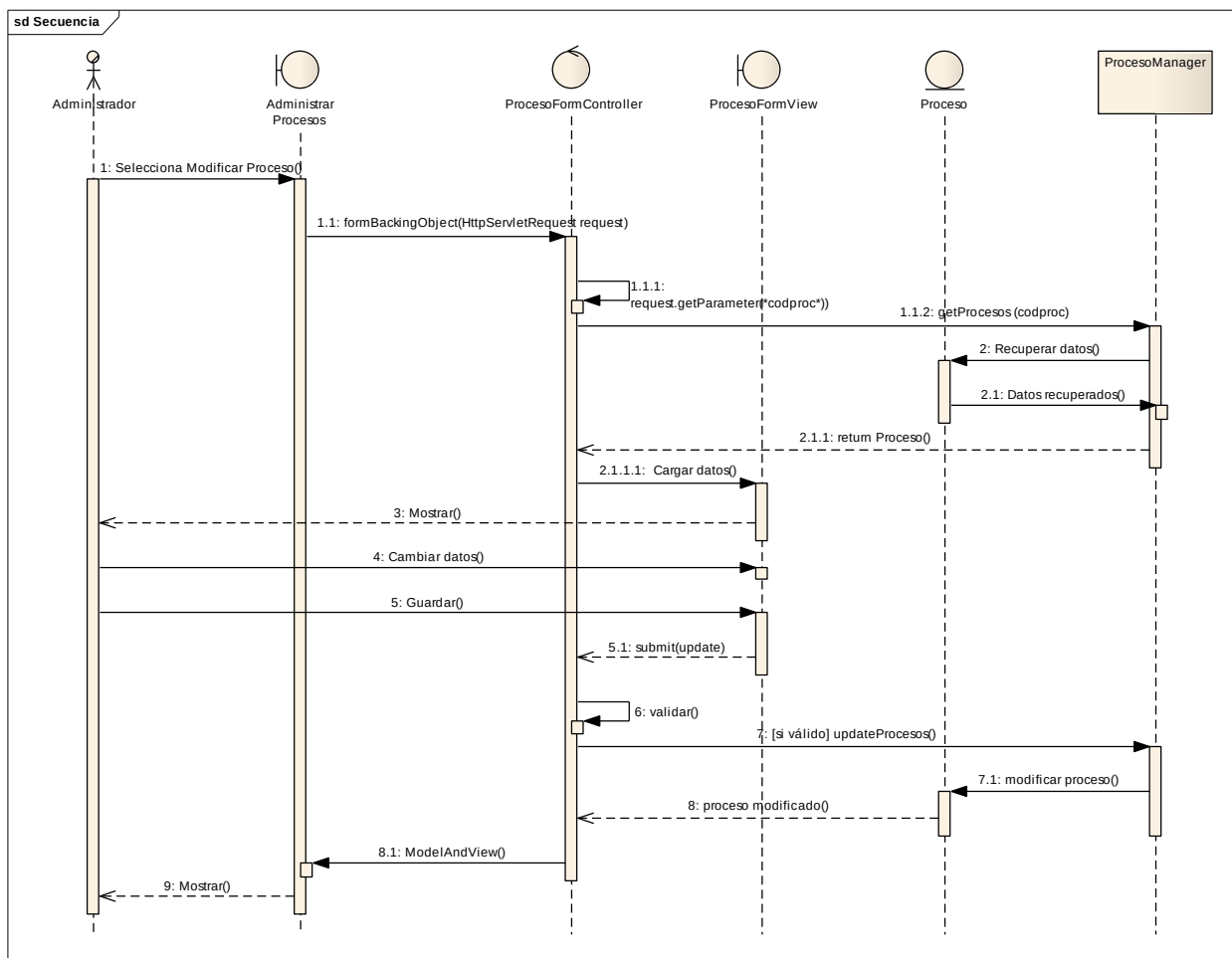


Figura 66. D.S. Modificar Proceso

Diagrama de Secuencia: Administrar Clientes

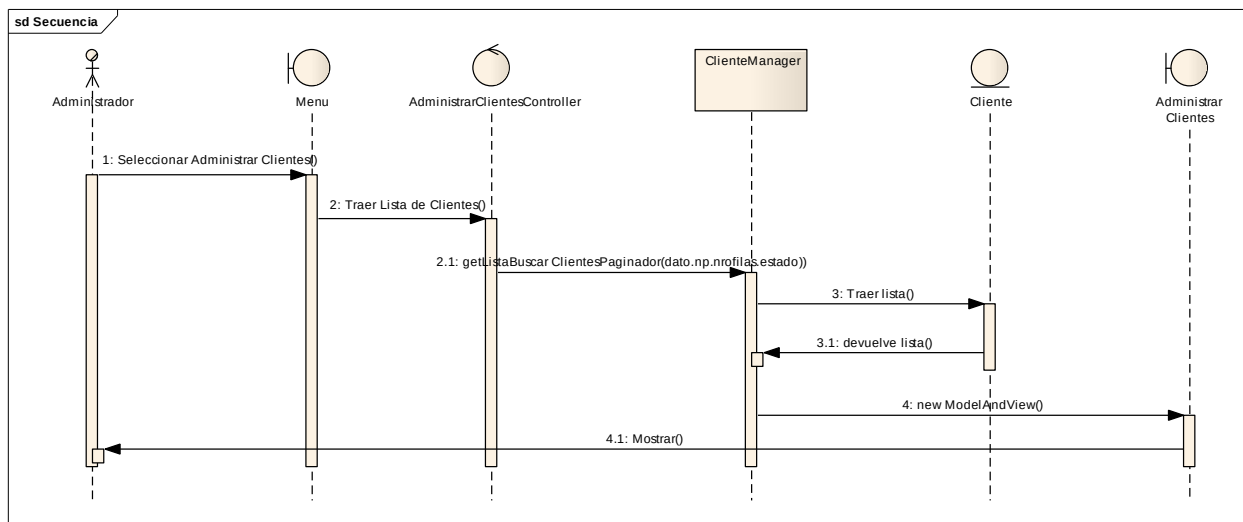


Figura 67. D.S. Administrar Clientes

Diagrama de Secuencia: Nuevo Cliente Persona

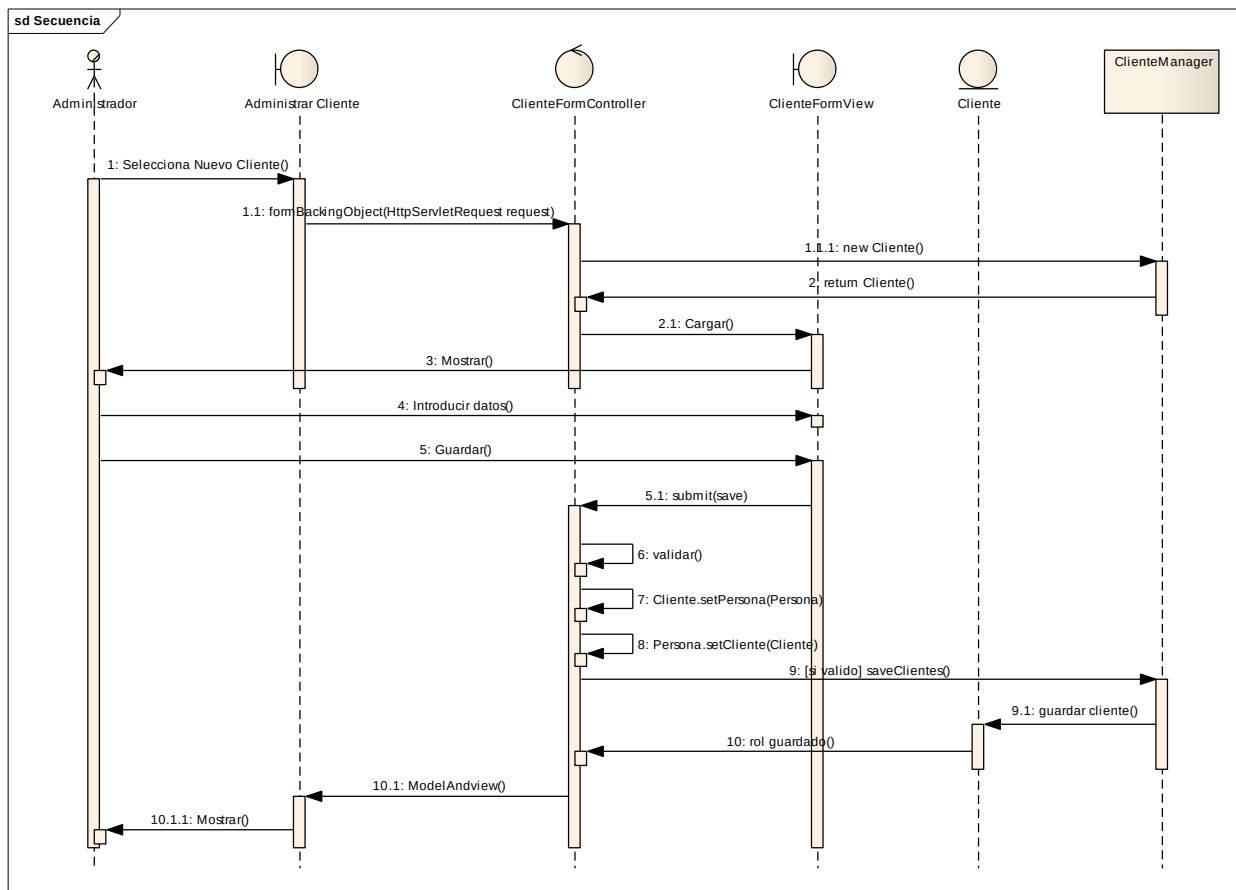


Figura 68. D.S. Nuevo Cliente Persona

Diagrama de Secuencia: Nuevo Cliente Empresa

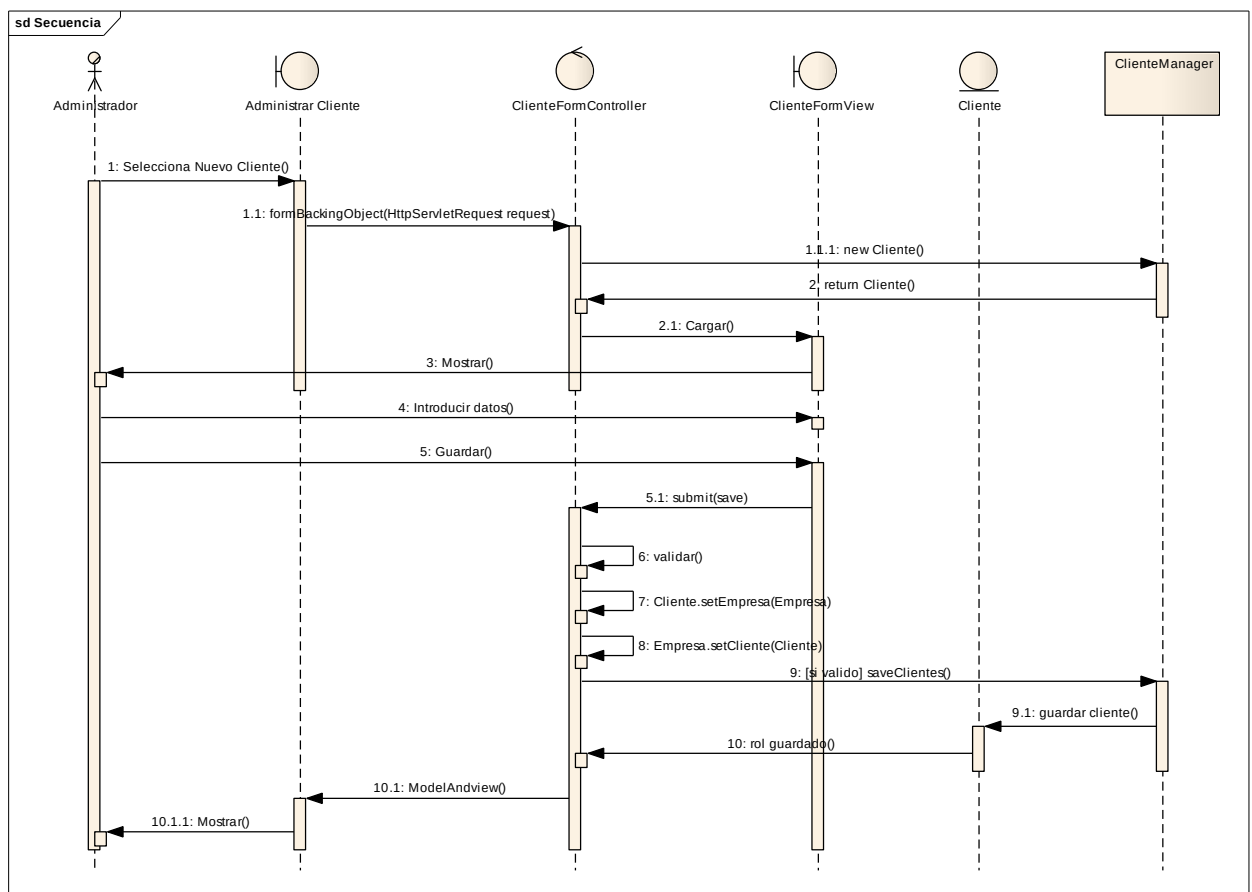


Figura 69. D.S. Nuevo Cliente Empresa

Diagrama de Secuencia: Administrar Proveedores

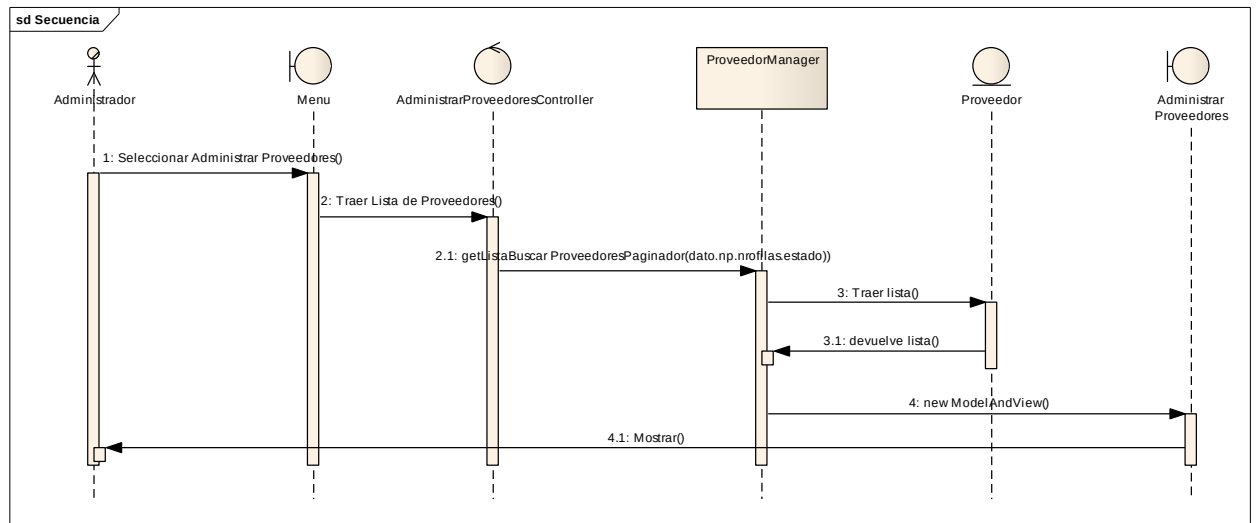


Figura 70. D.S. Administrar Proveedores

Diagrama de Secuencia: Nuevo Proveedor Persona

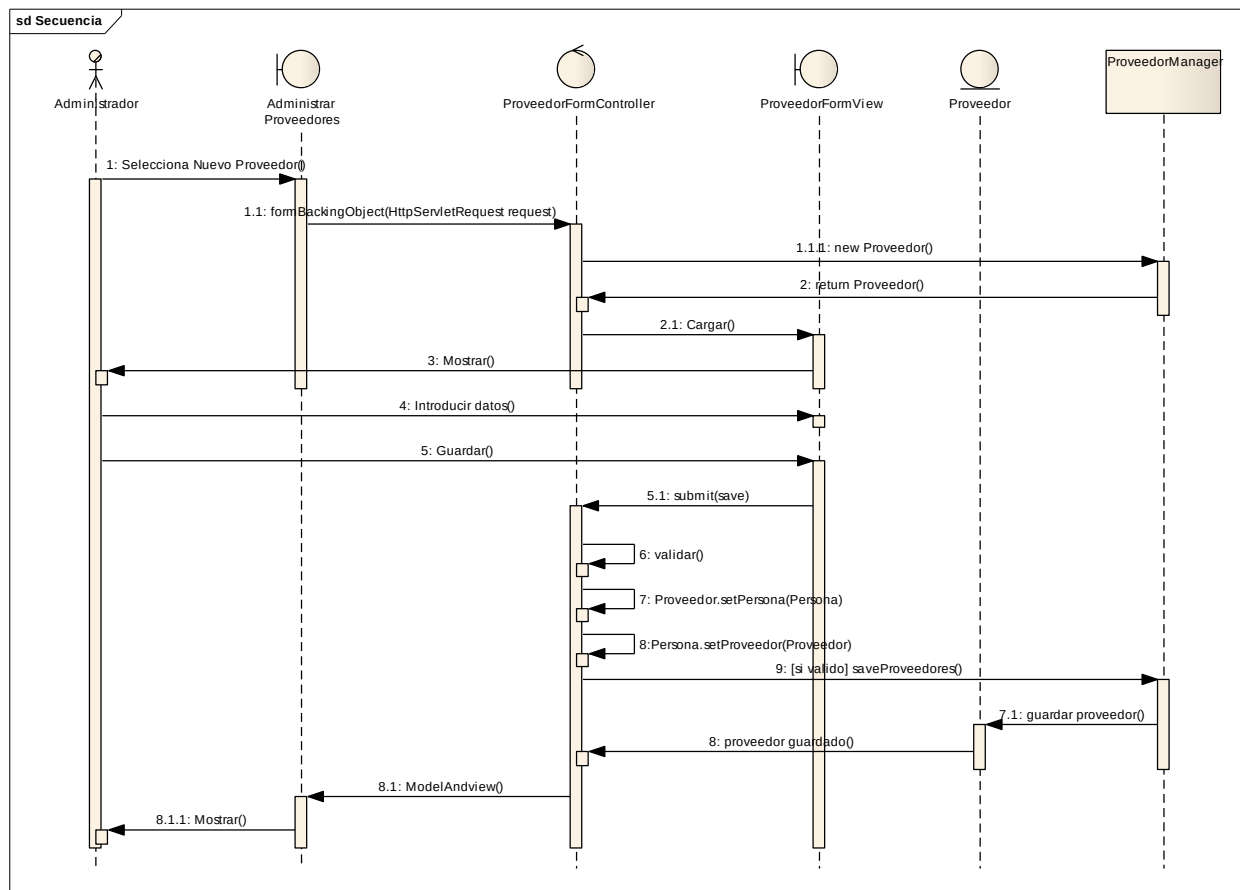


Figura 71. D.S. Nuevo Proveedor Persona

Diagrama de Secuencia: Nuevo Proveedor Empresa

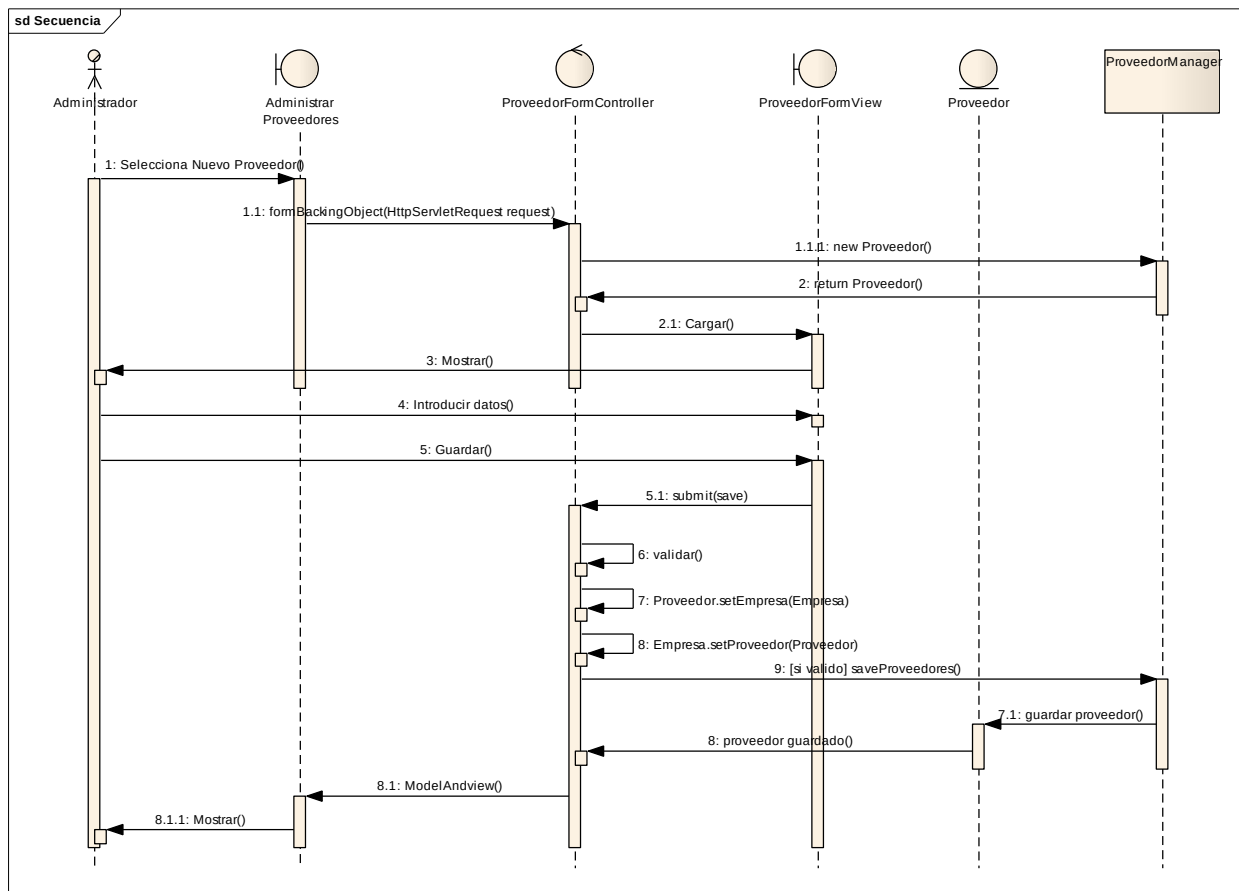


Figura 72. D.S. Nuevo Proveedor Empresa

Diagrama de Secuencia: Nueva Categoría

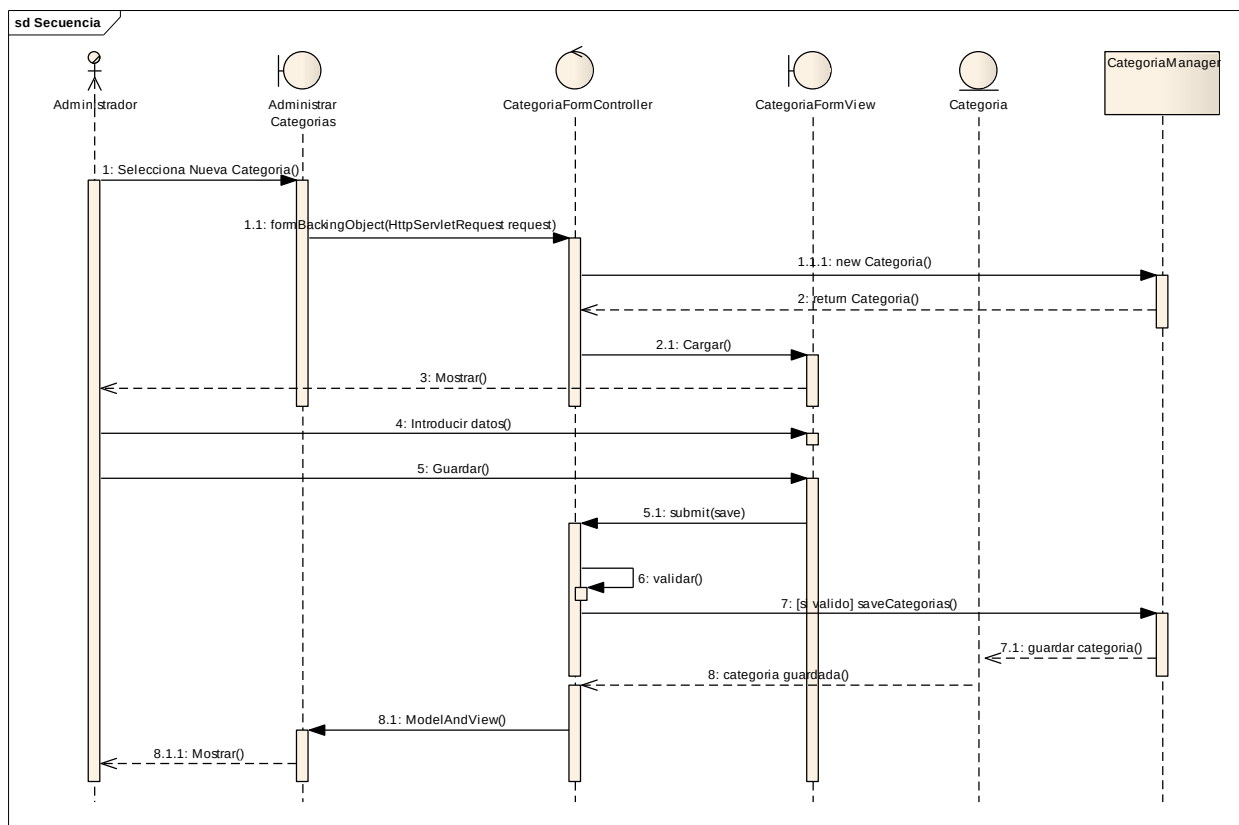


Figura 74. D.S. Nueva Categoría

Diagrama de Secuencia: Modificar Categoría

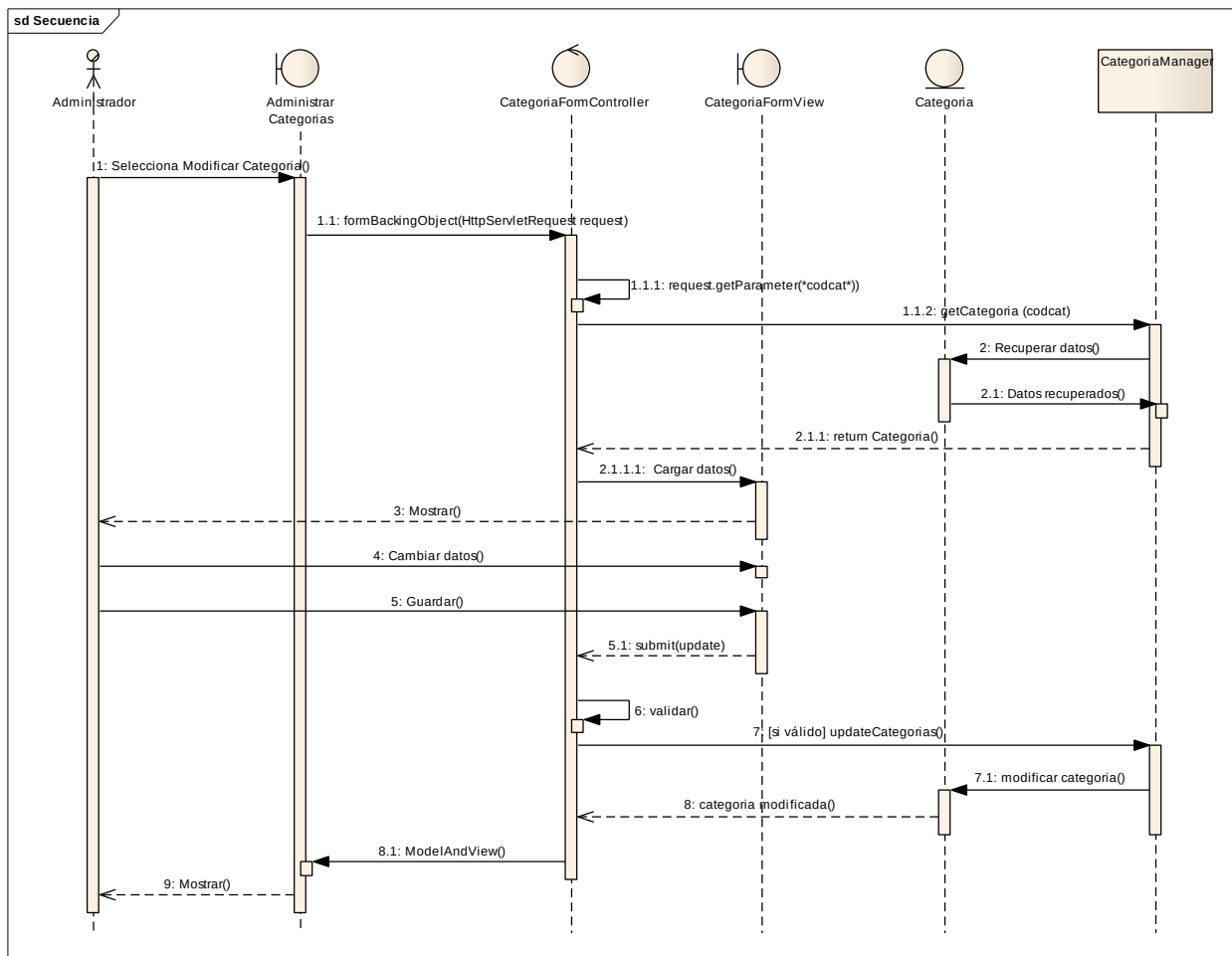


Figura 75. D.S. Modificar Categoría

Diagrama de Secuencia: Administrar Notas

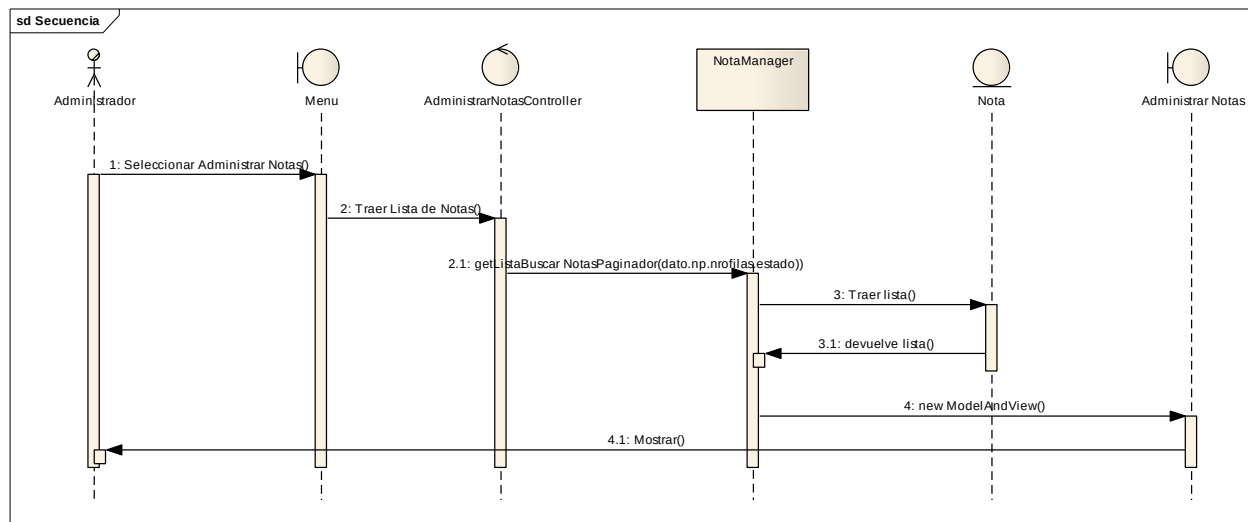


Figura 76. D.S. Administrar Notas

Diagrama de Secuencia: Modificar Nota

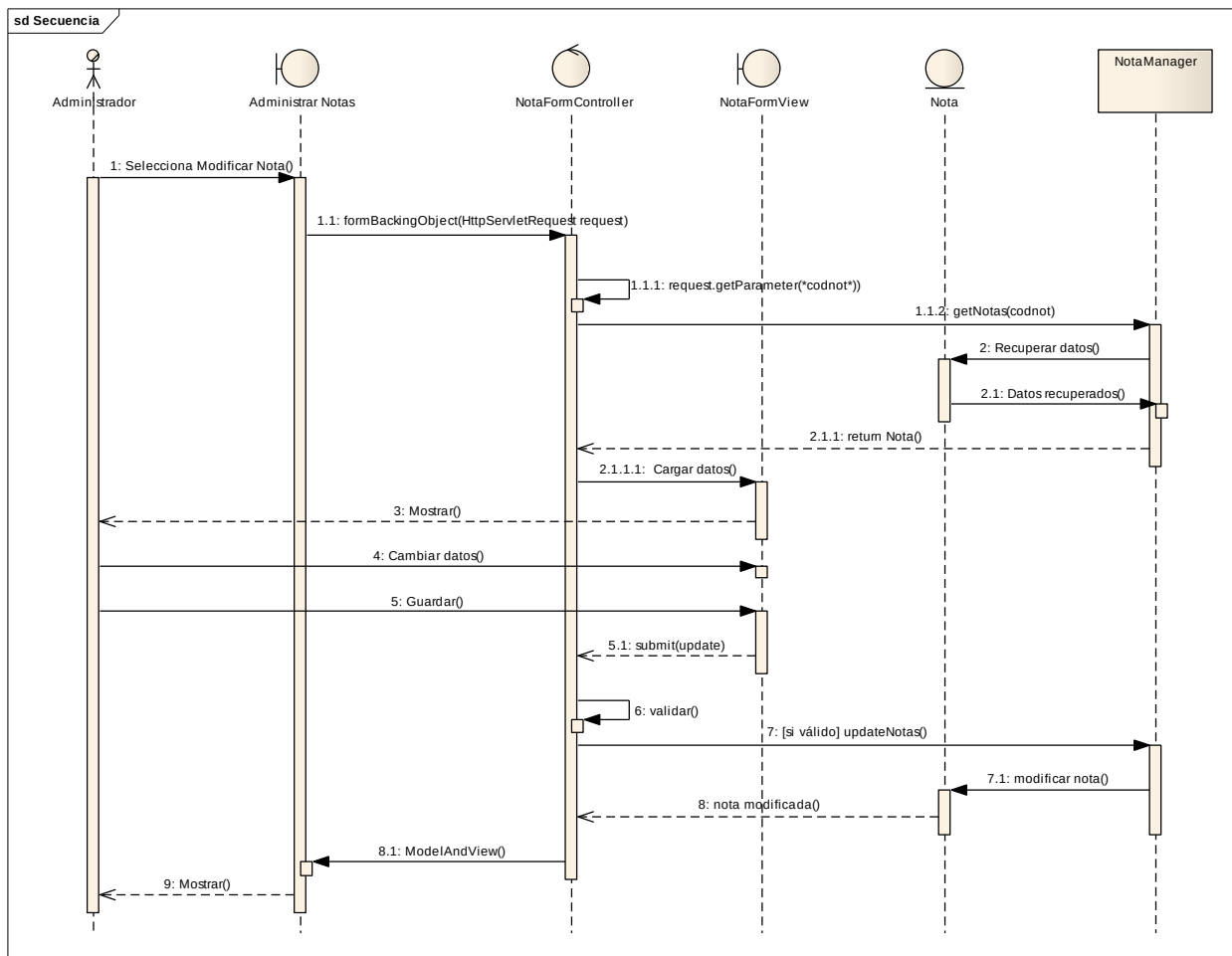


Figura 77. D.S. Nueva Nota

Diagrama de Secuencia: Administrar Tipos

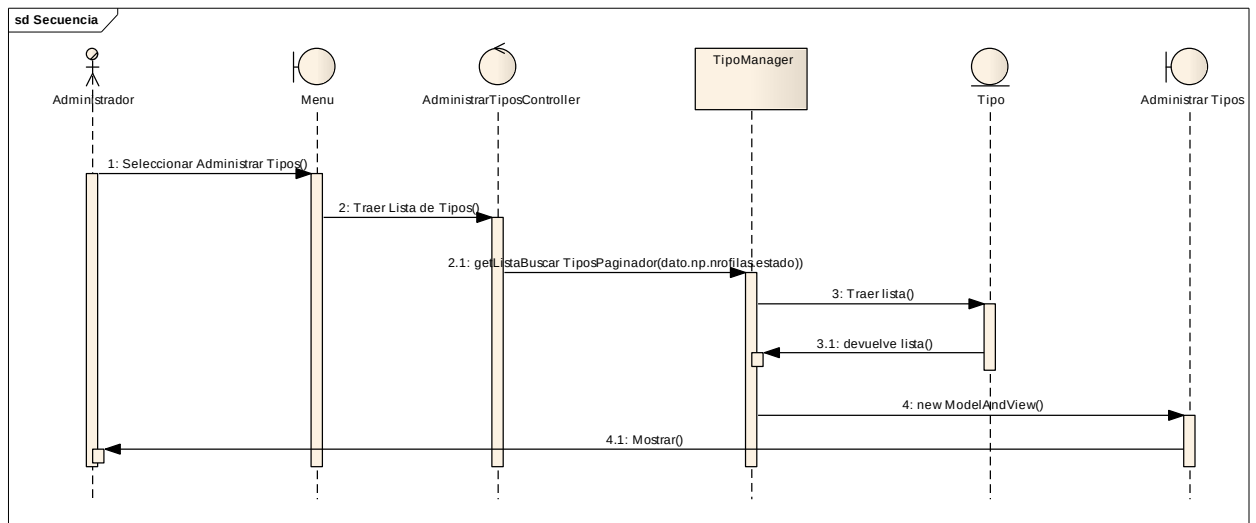


Figura 78. D.S. Administrar Tipos

Diagrama de Secuencia: Nuevo Tipo

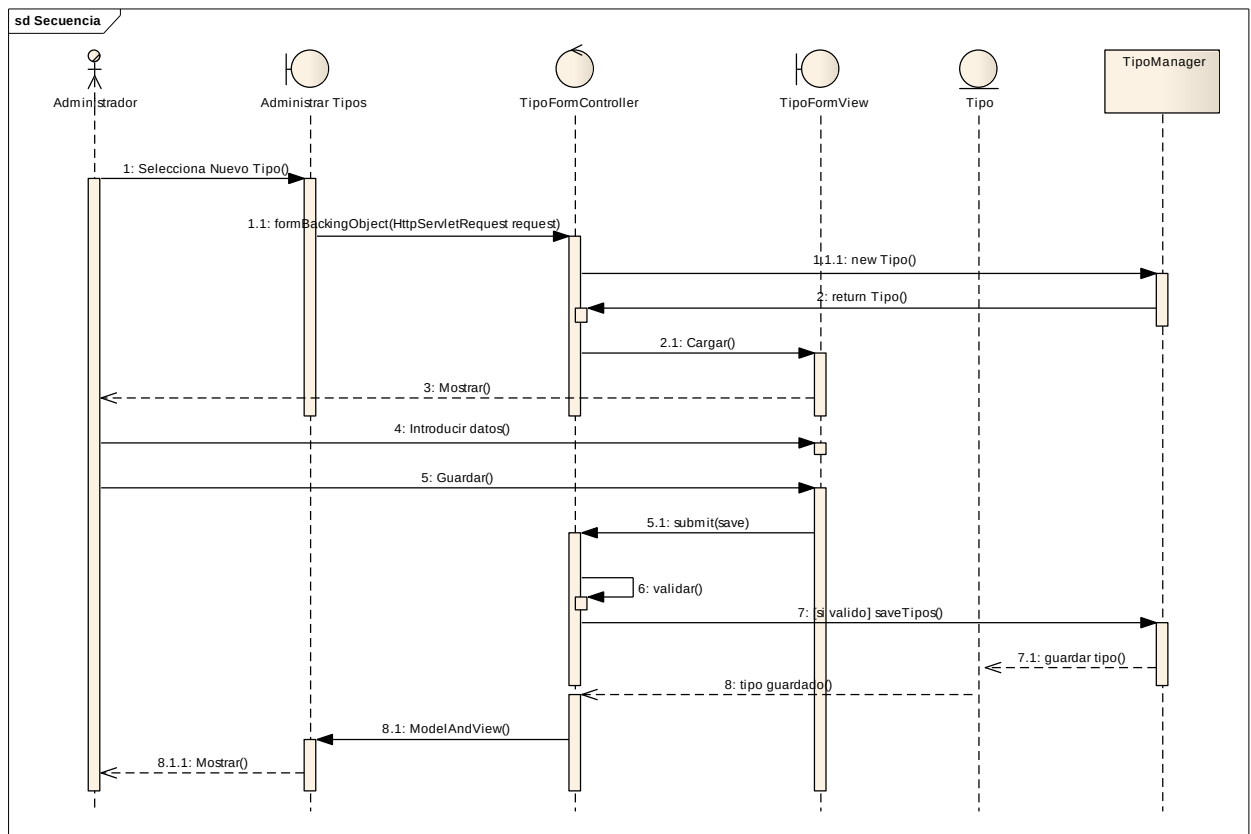


Figura 79. D.S. Nuevo Tipo

Diagrama de Secuencia: Modificar Tipo

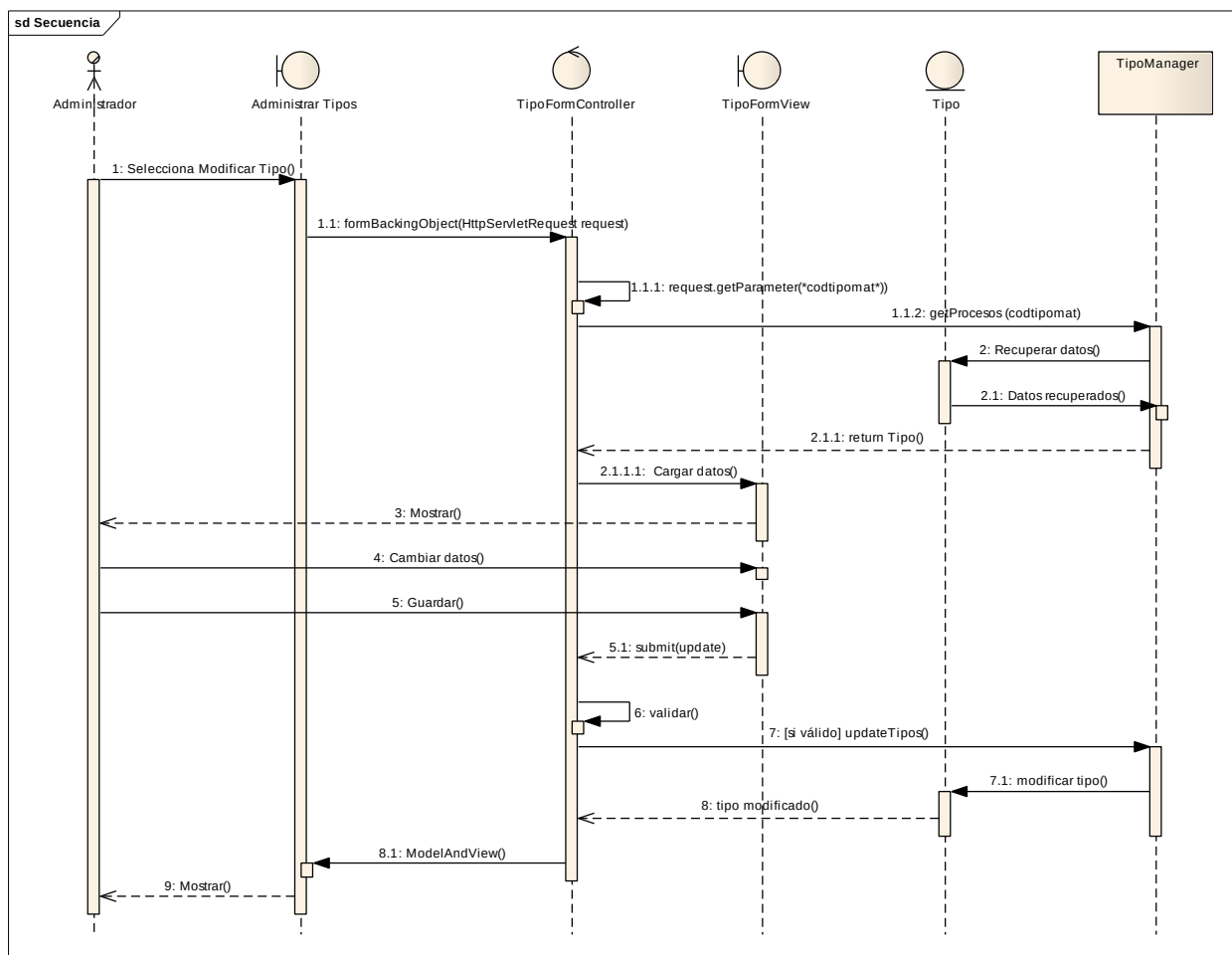


Figura 80 D.S. Modificar Tipo

Diagrama de Secuencia: Administrar Tipo Productos

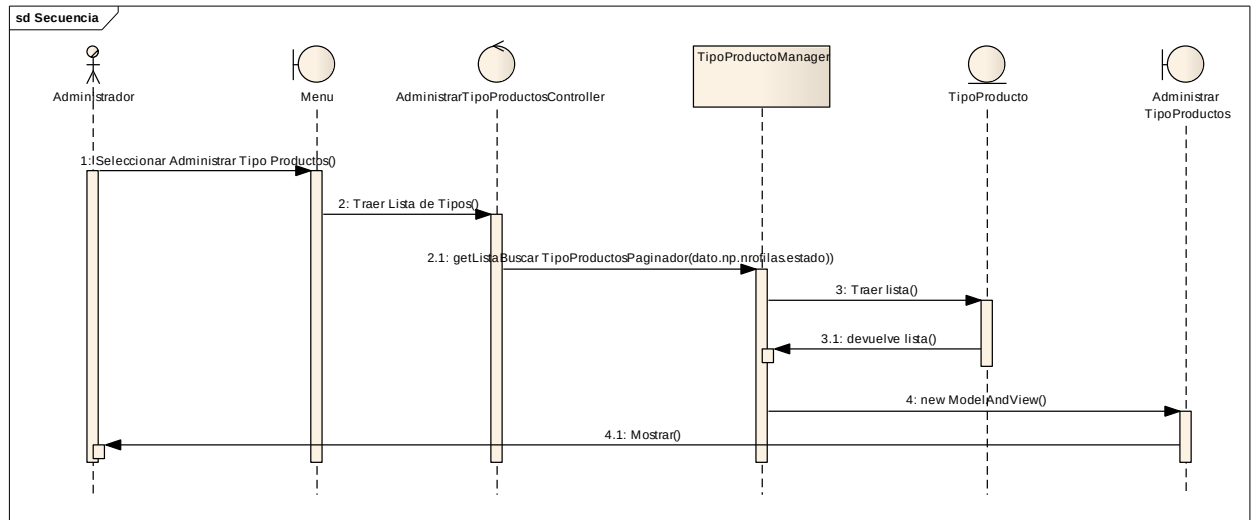


Figura 81. D.S. Administrar Tipo Productos

Nuevo Tipo Producto

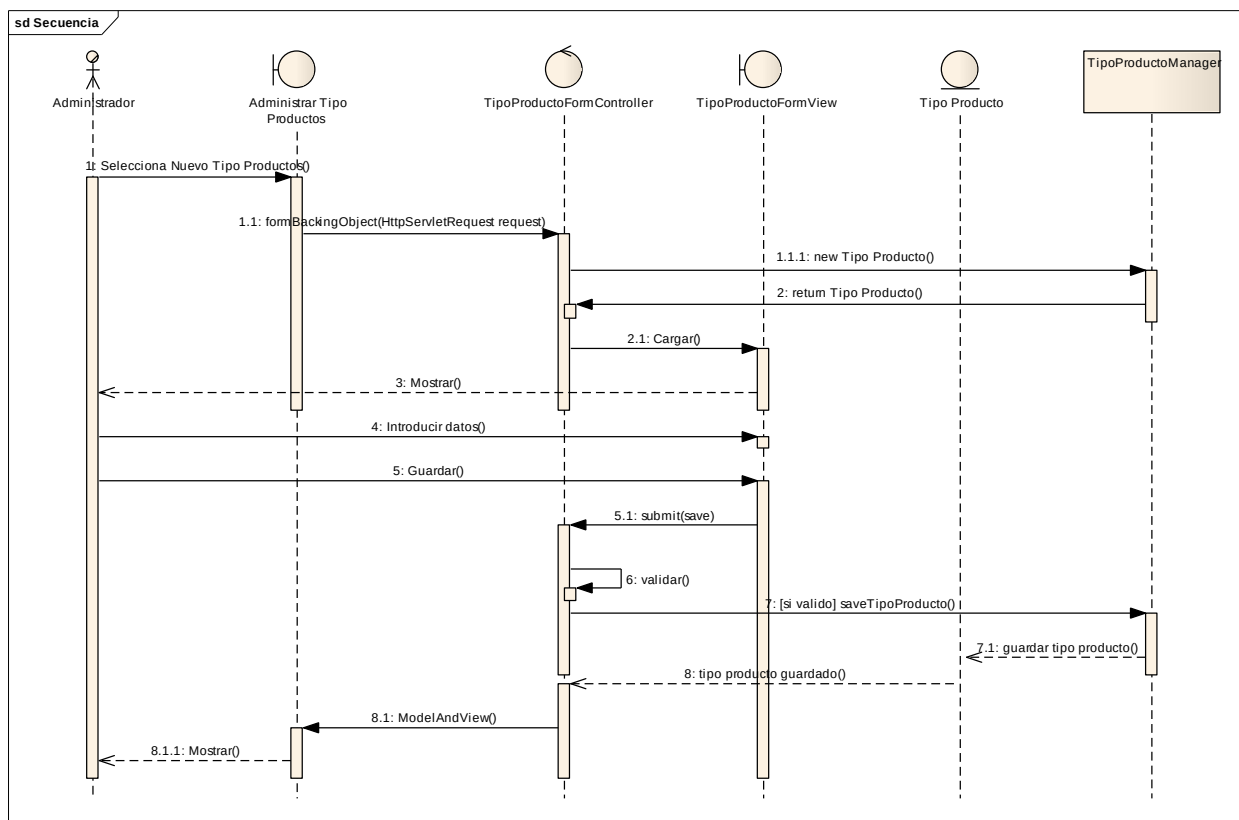


Figura 82. D.S. Nuevo Tipo Producto

Diagrama de Secuencia: Modificar Tipo Producto

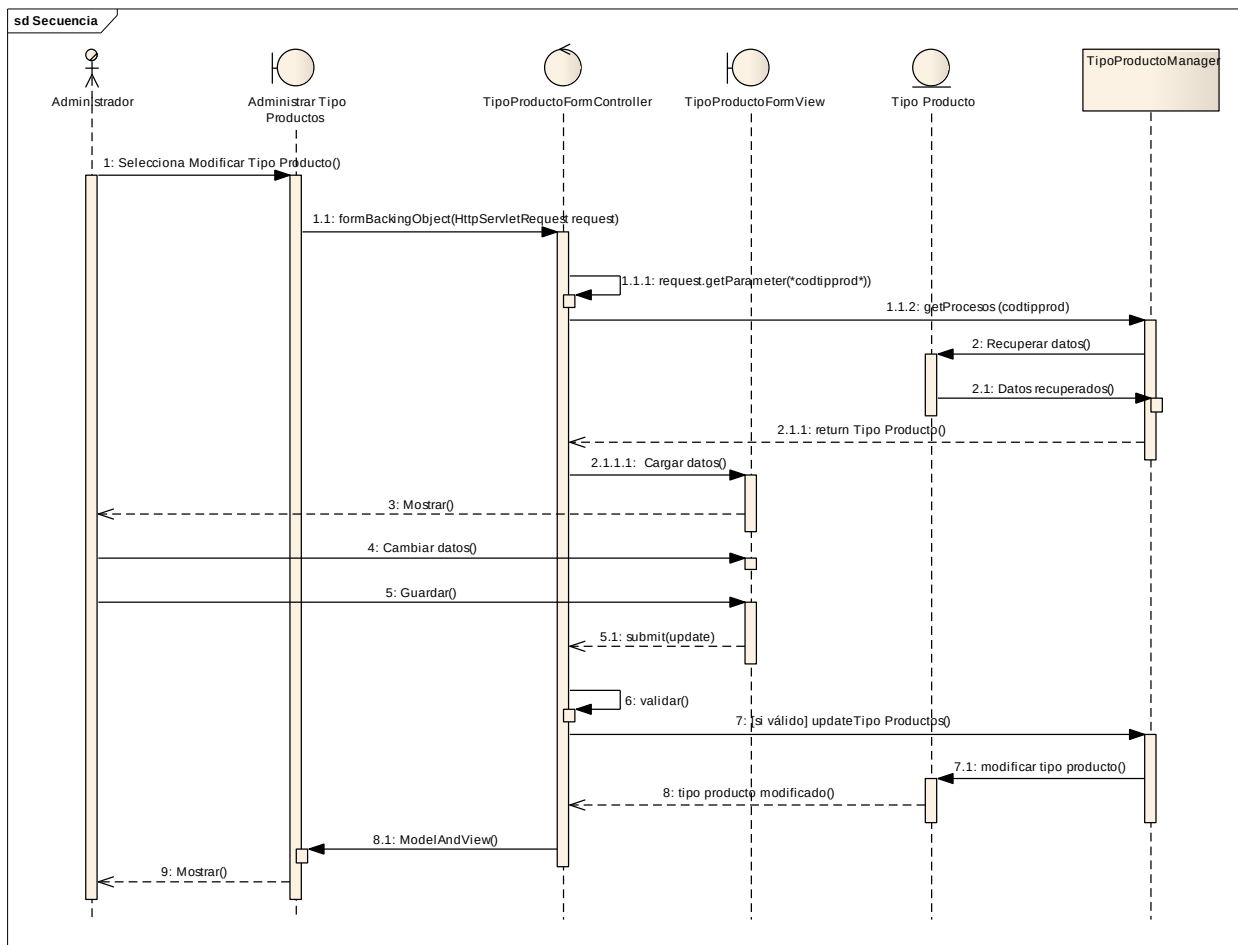


Figura 83. D.S. Modificar Tipo Producto

Diagrama de Secuencia: Administrar Materiales

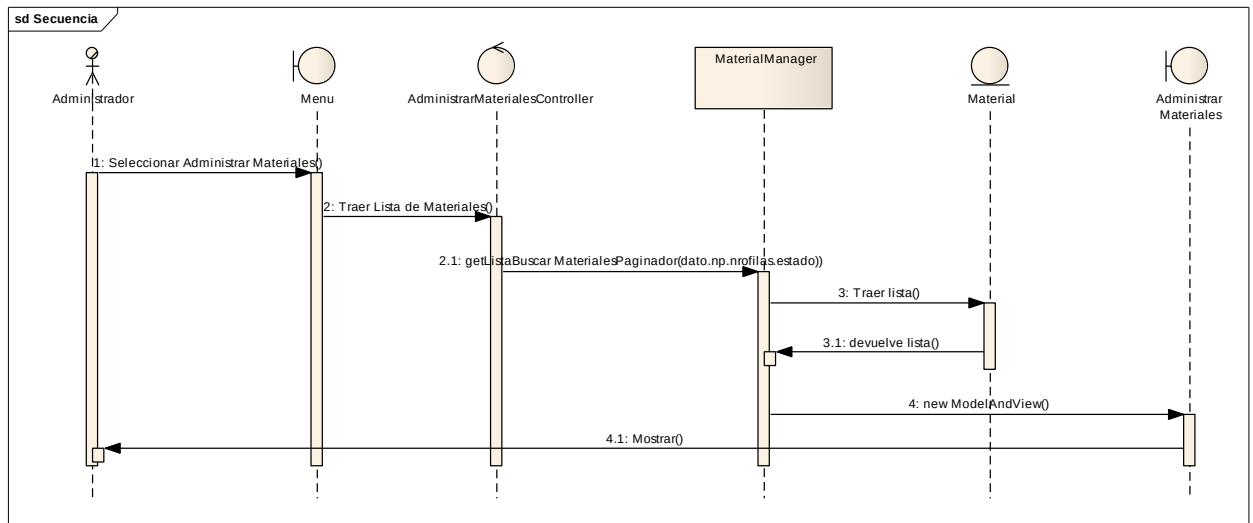


Figura 84. D.S. Administrar Materiales

Diagrama de Secuencia: Nuevo Material

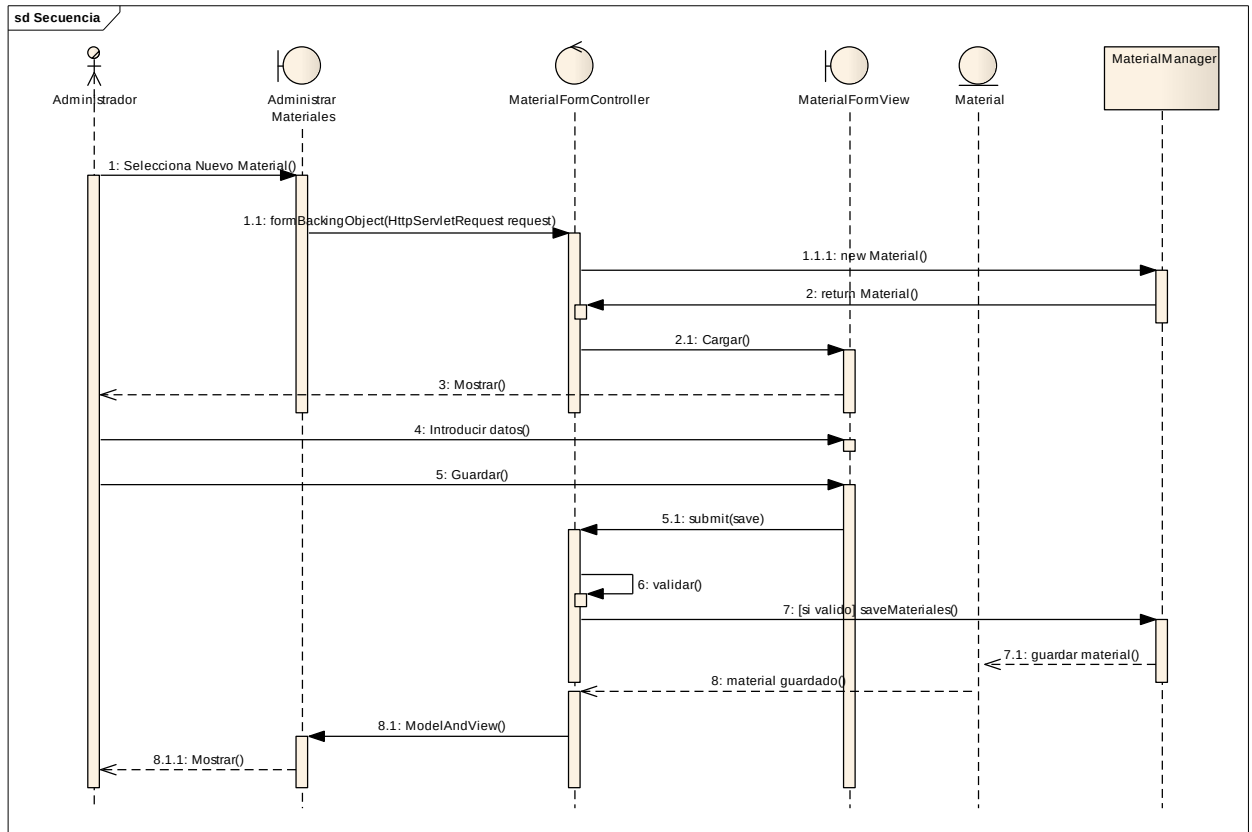


Figura 85. D.S. Nuevo Material

Diagrama de Secuencia: Modificar Material

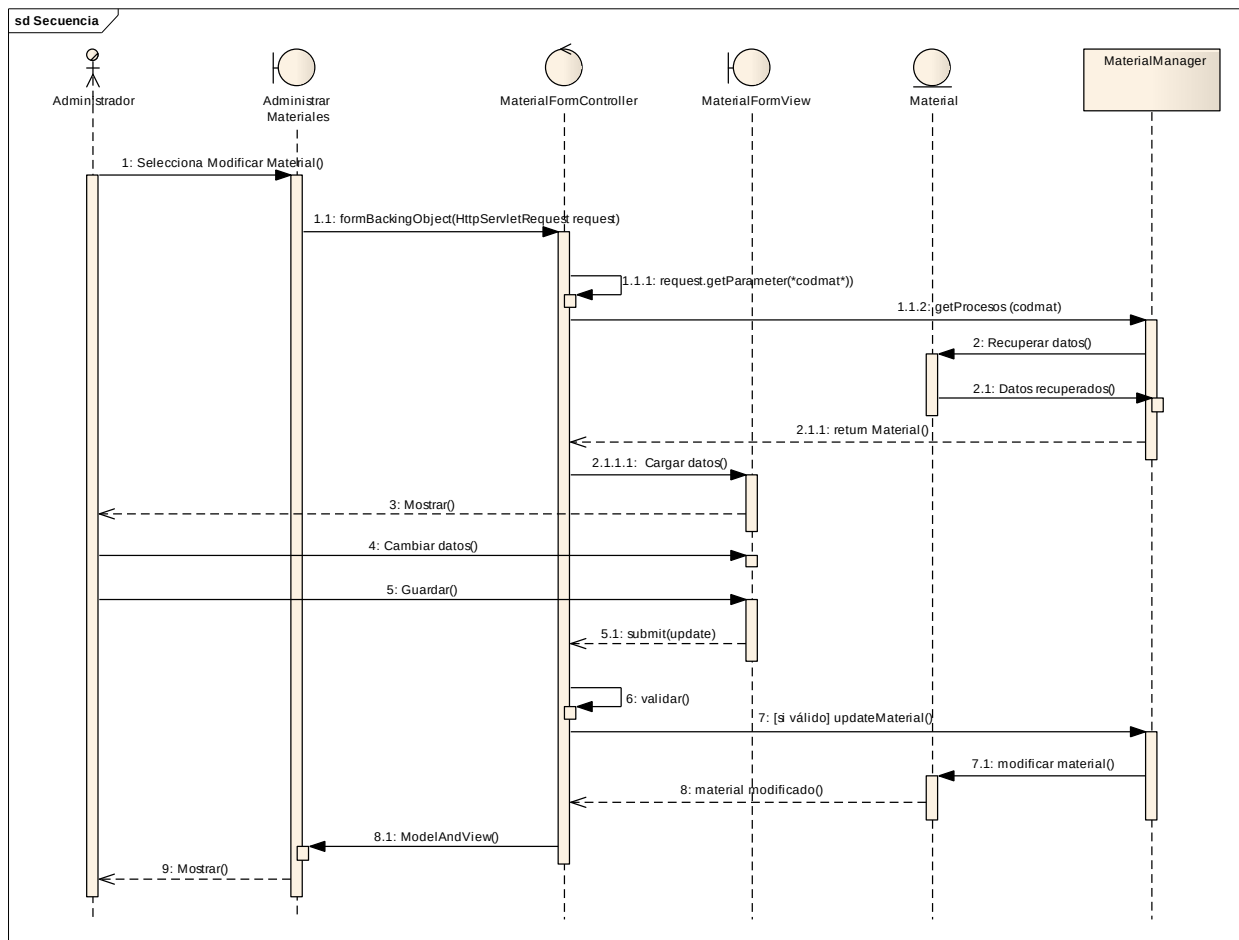


Figura 86. D.S. Modificar Material

Diagrama de Secuencia: Ver Material

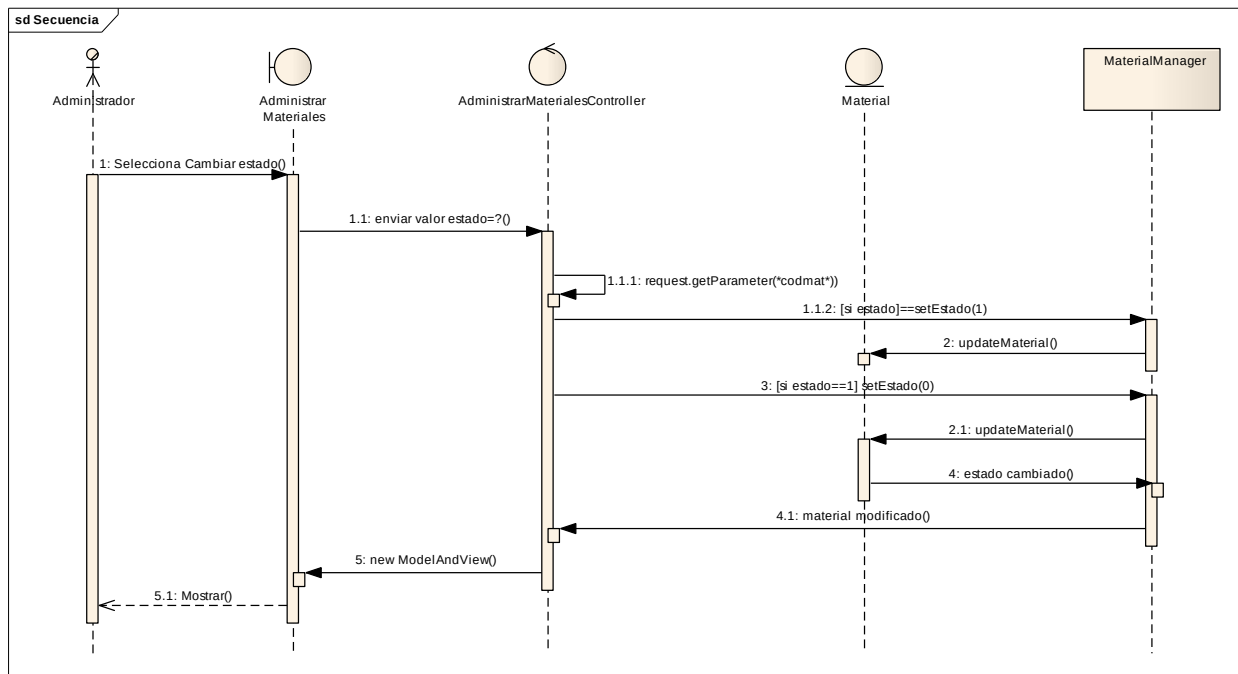


Figura 87. D.S. Ver Material

Diagrama de Secuencia: Administrar Medidas

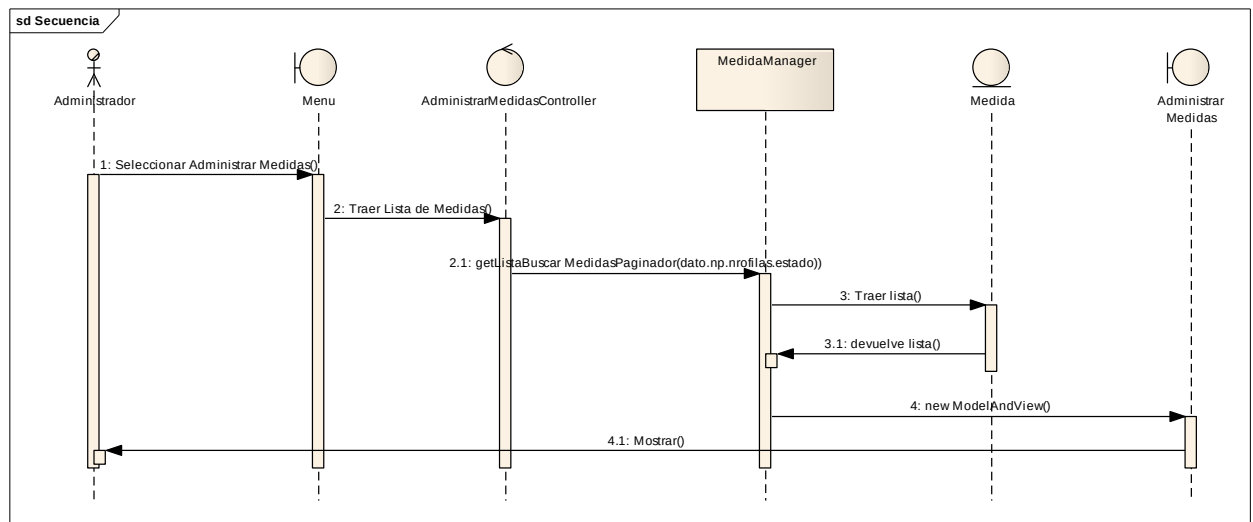


Figura 88. D.S. Administrar Medidas

Diagrama de Secuencia: Nueva Medida

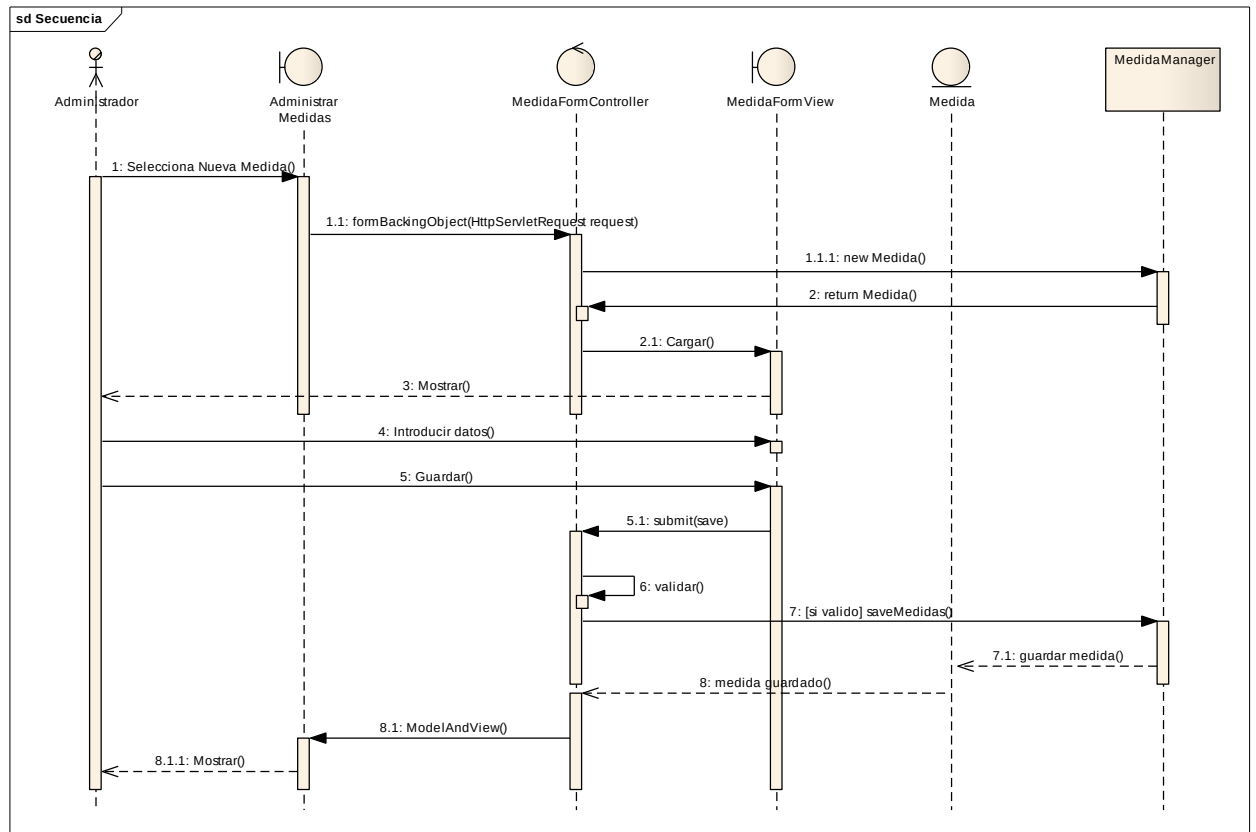


Figura 89. D.S. Nueva Medida

Diagrama de Secuencia: Modificar Medida

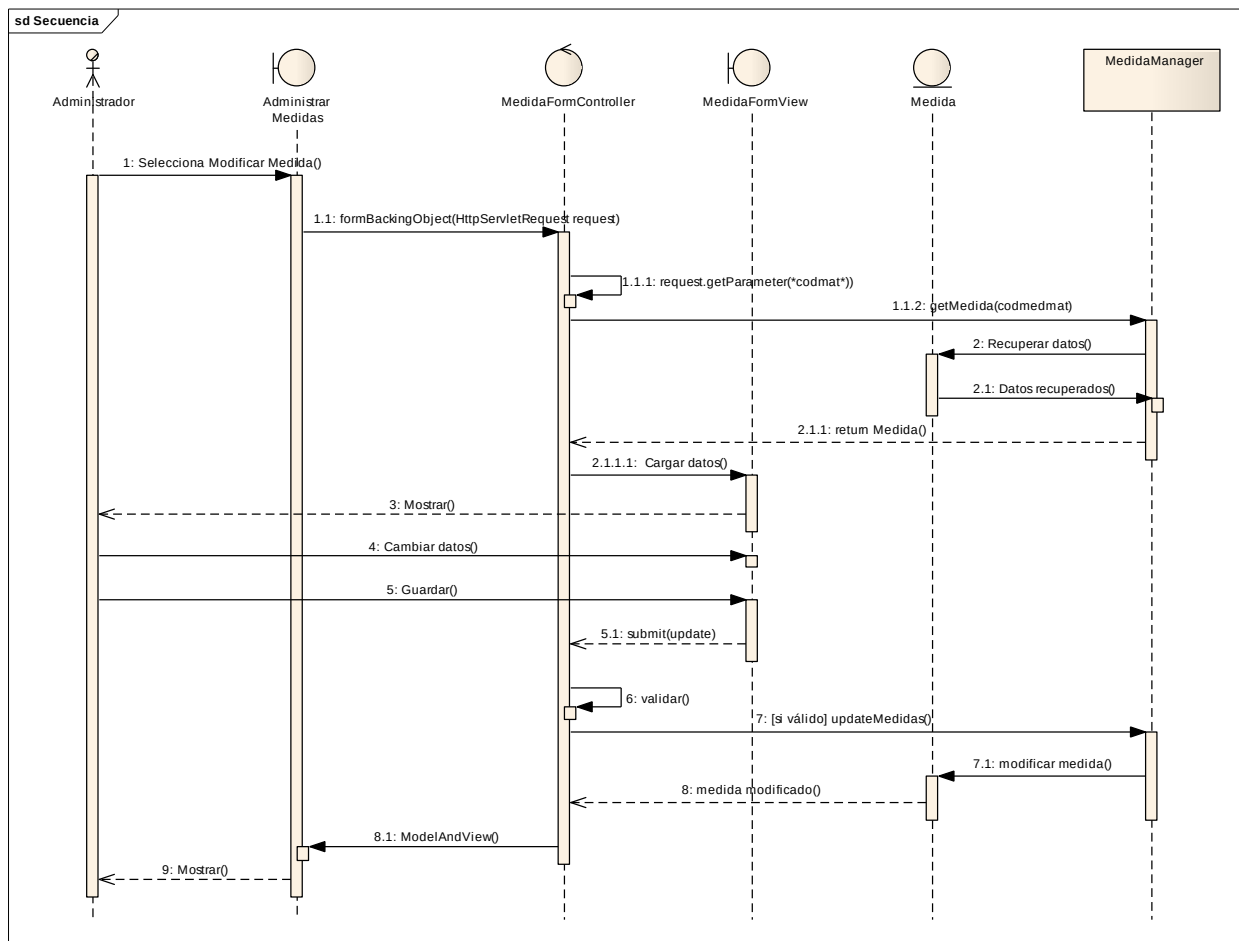


Figura 90. D.S. Modificar Medida

Diagrama de Secuencia: Administrar Gramajes

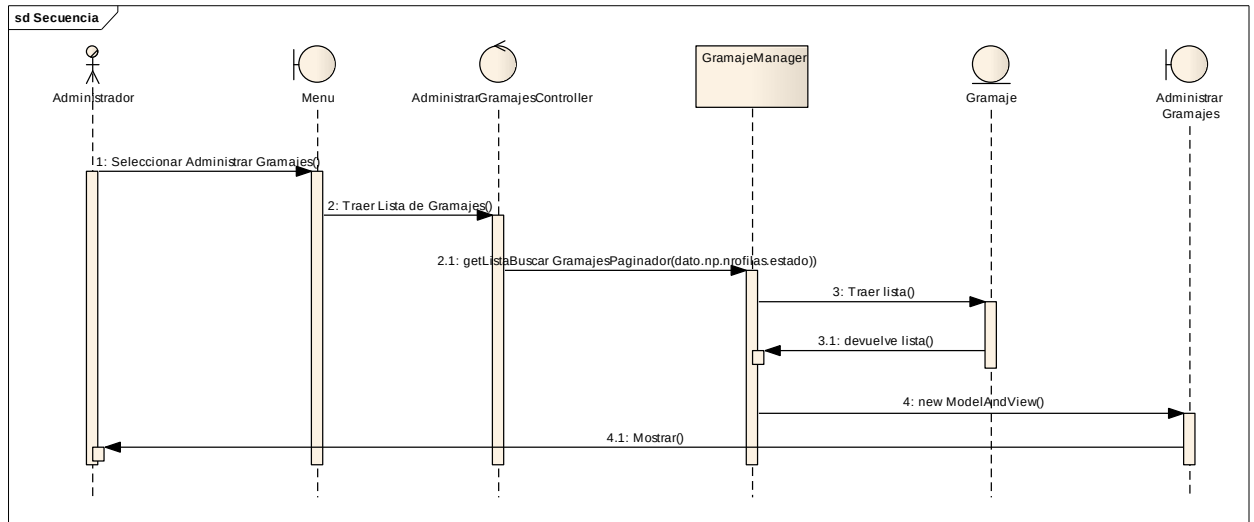


Figura 91. D.S. Administrar Gramajes

Diagrama de Secuencia: Nuevo Gramaje

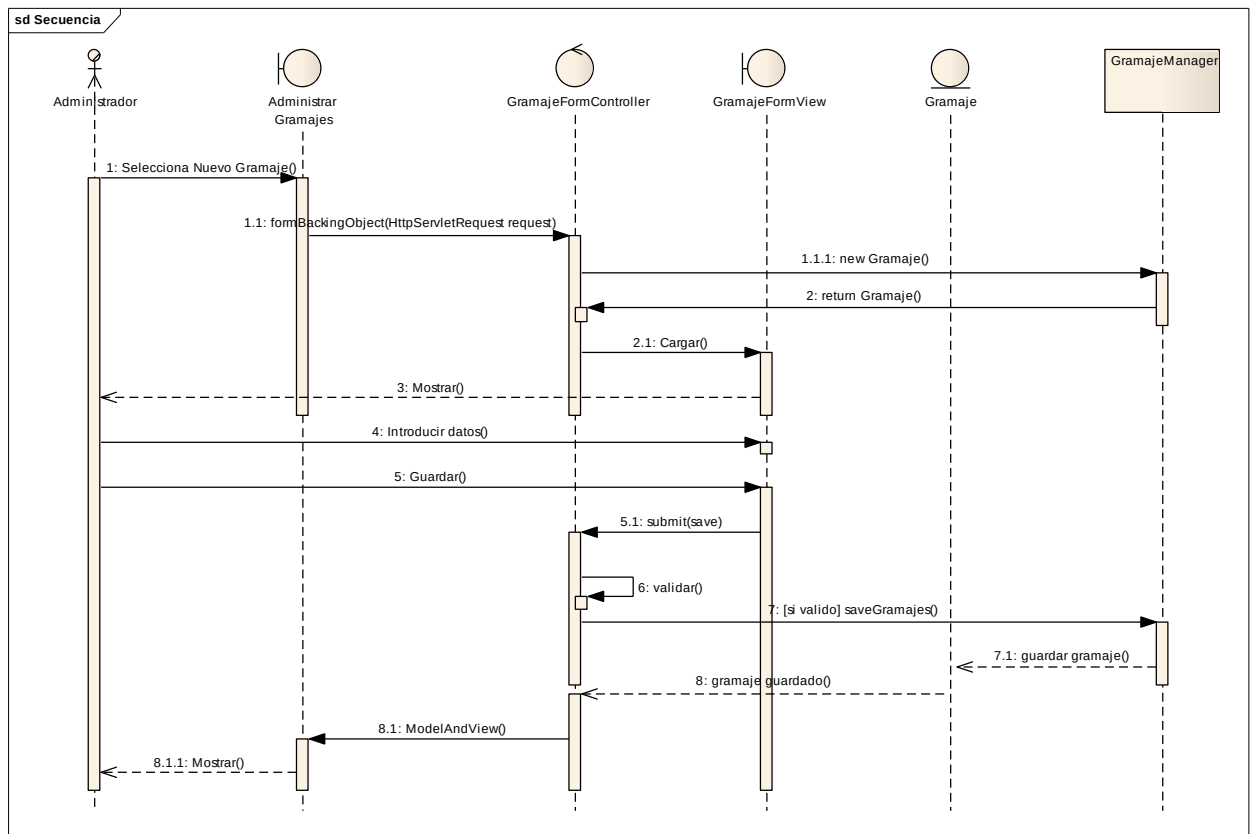


Figura 92. D.S. Nuevo Gramaje

Diagrama de Secuencia: Modificar Gramaje

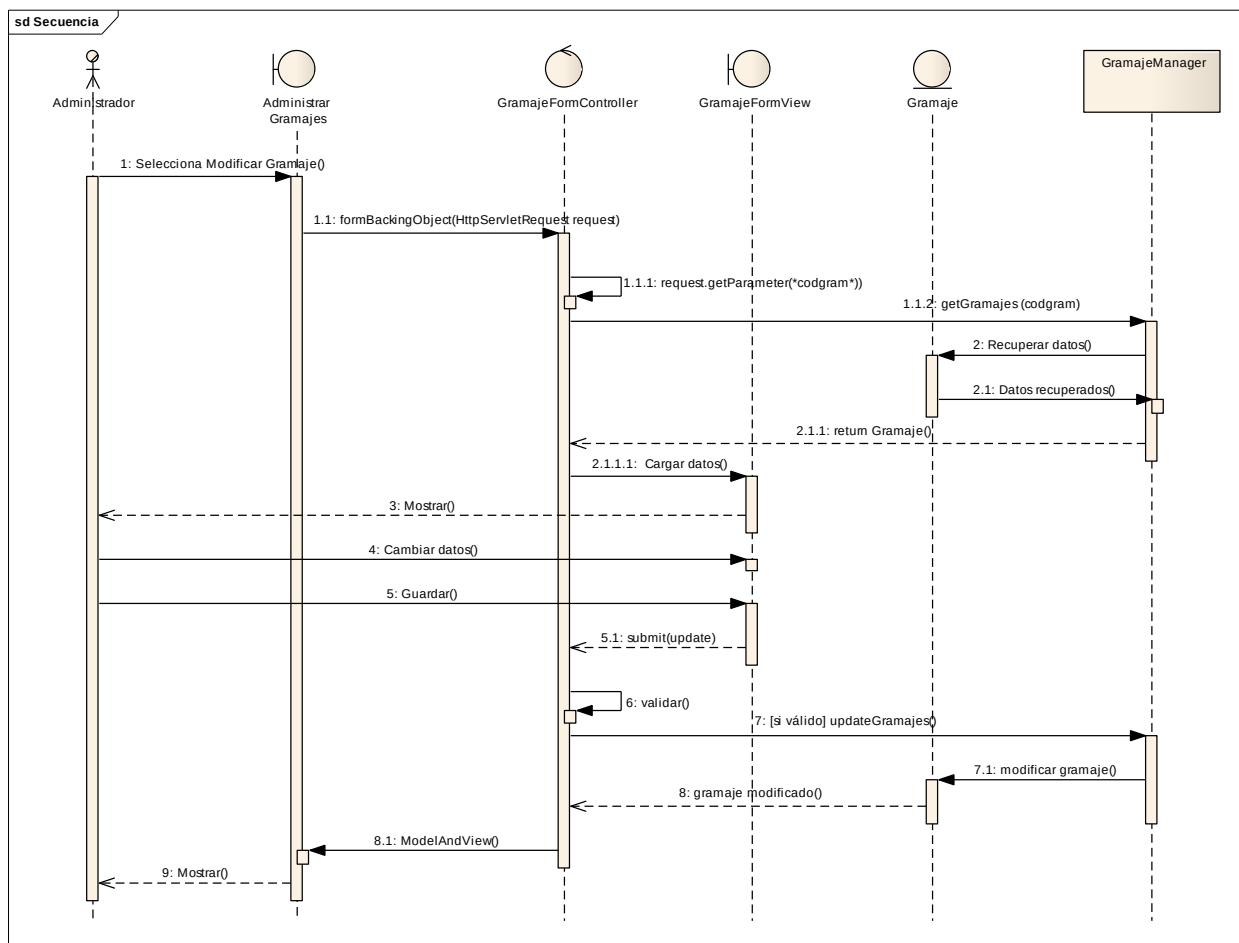


Figura 93. D.S. Modificar Gramaje

Diagrama de Secuencia: Administrar Marcas

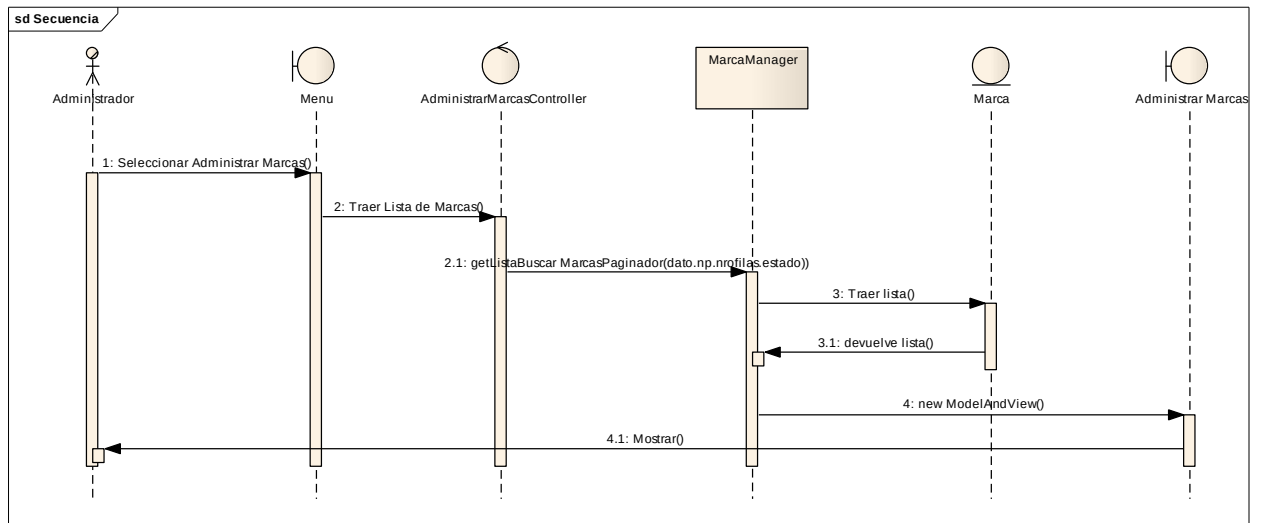


Figura 94. D.S. Administrar Marcas

Diagrama de Secuencia: Nueva Marca

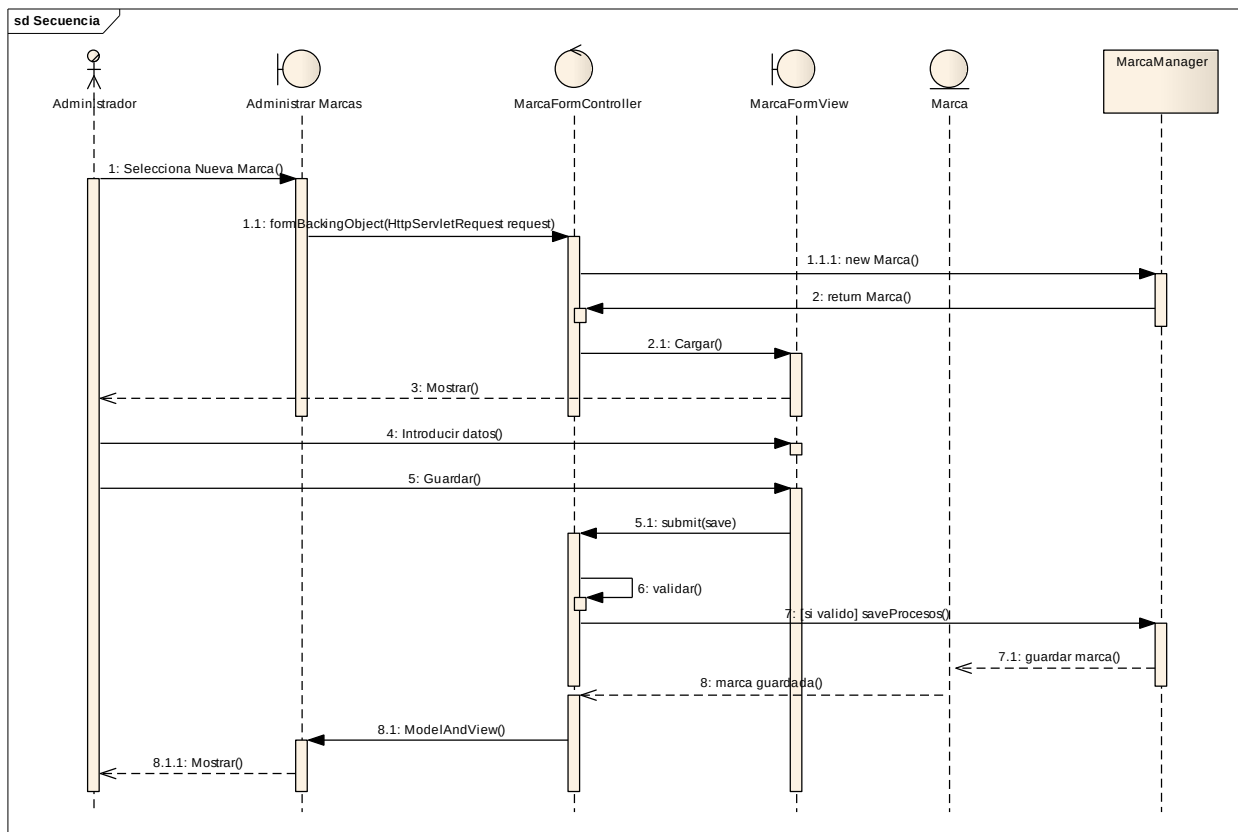


Figura 95. D.S. Nueva Marca

Diagrama de Secuencia: Modificar Marca

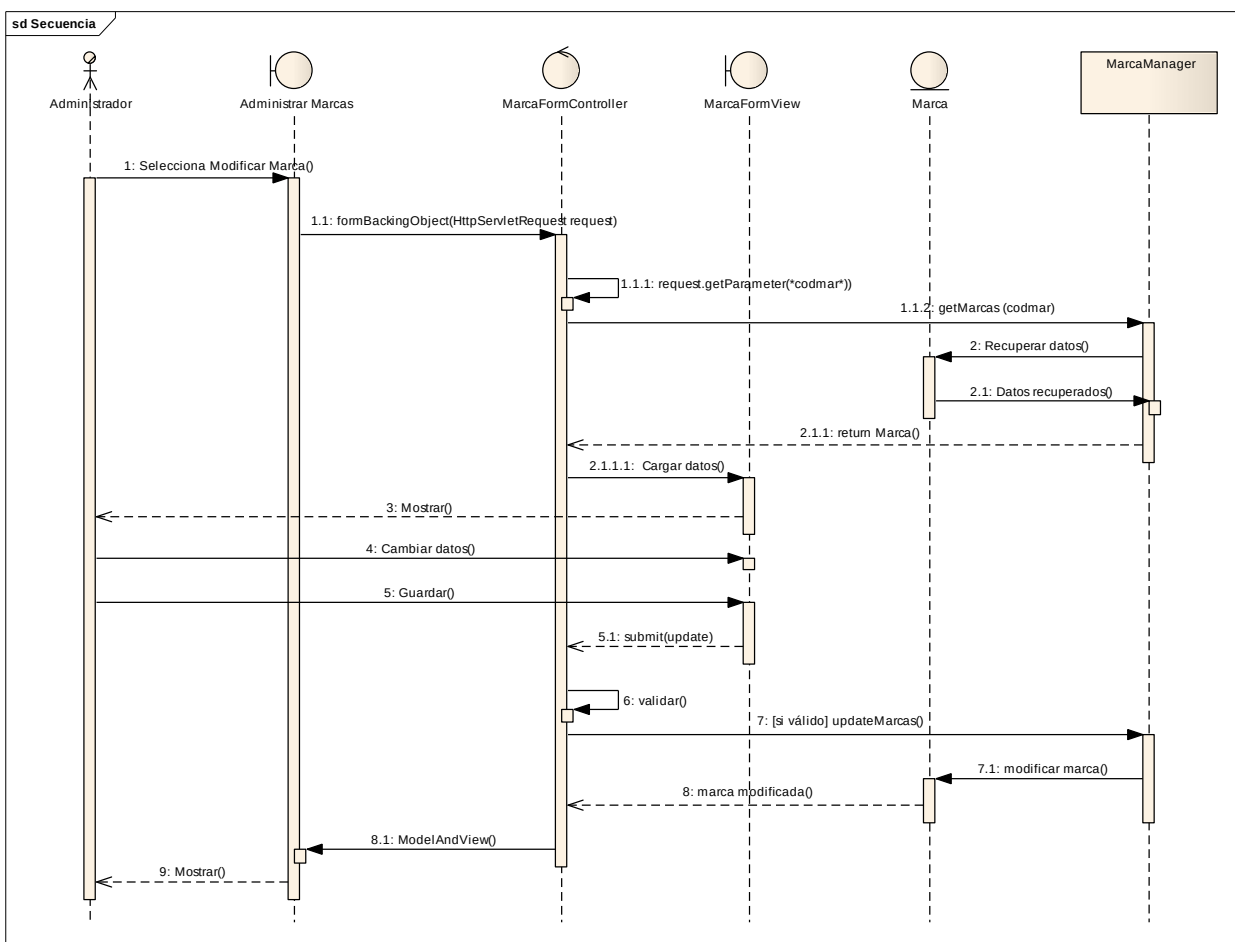


Figura 96. D.S. Modificar Marca

Diagrama de Secuencia: Administrar Máquinas

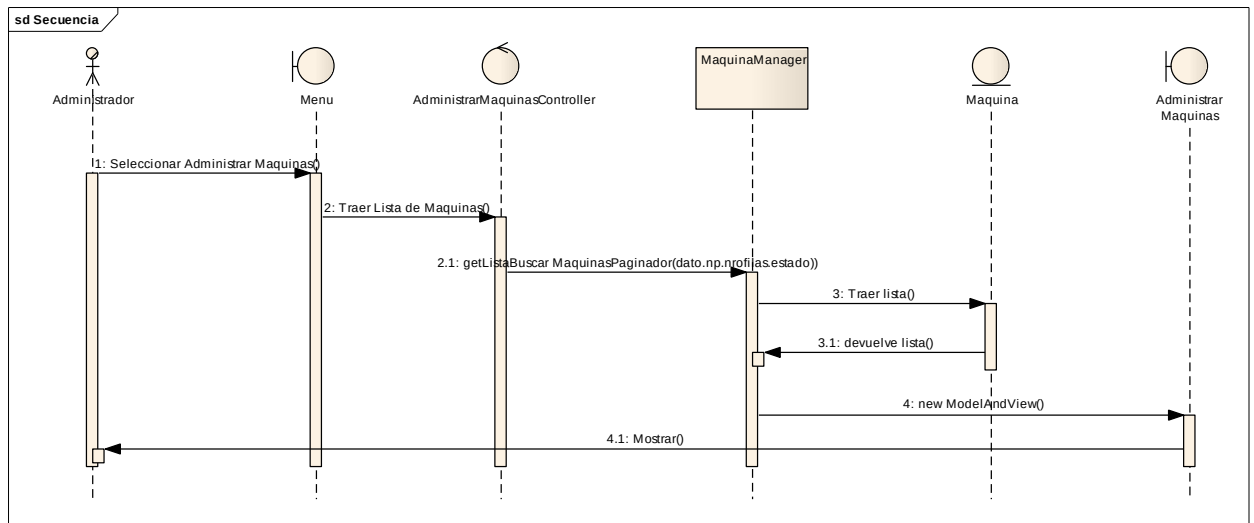


Figura 97. D.S. Administrar Máquinas

Diagrama de Secuencia: Nueva Máquina

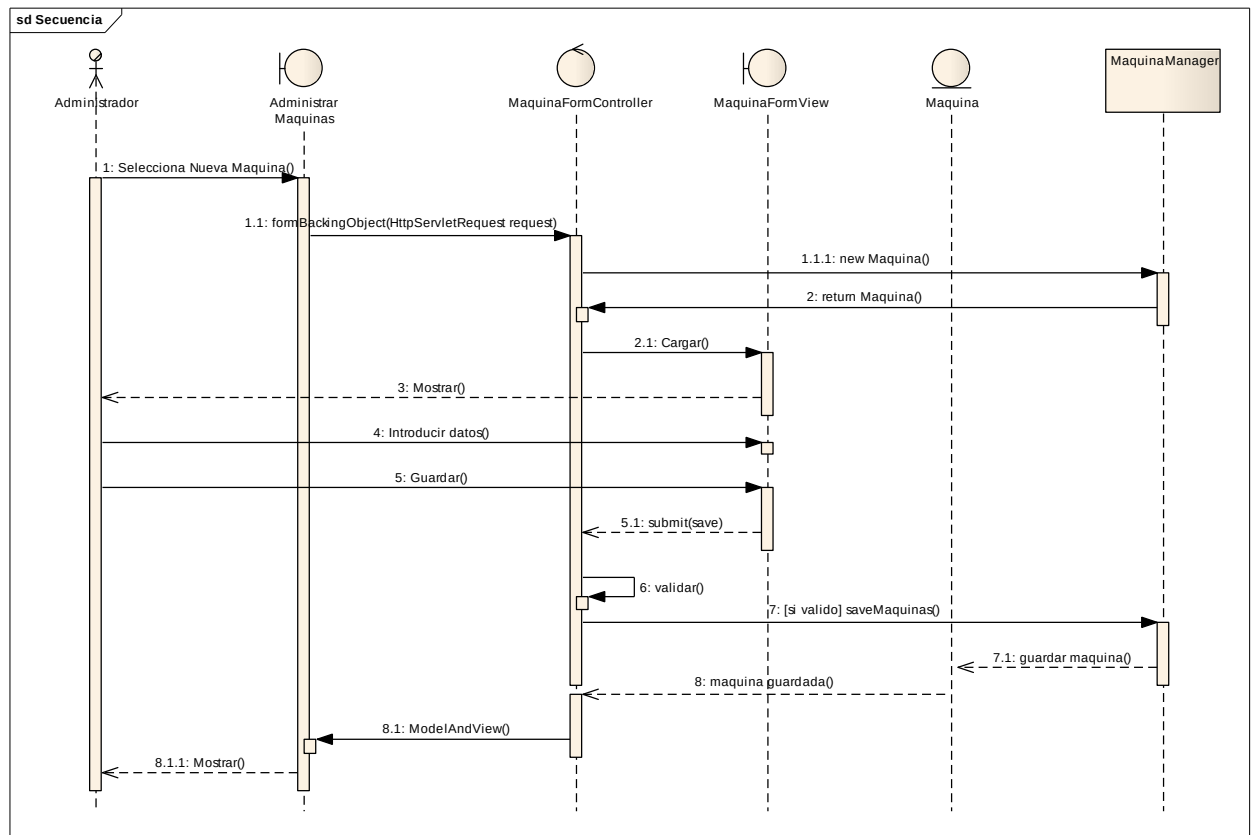


Figura 98. D.S. Nueva Máquina

Diagrama de Secuencia: Modificar Máquina

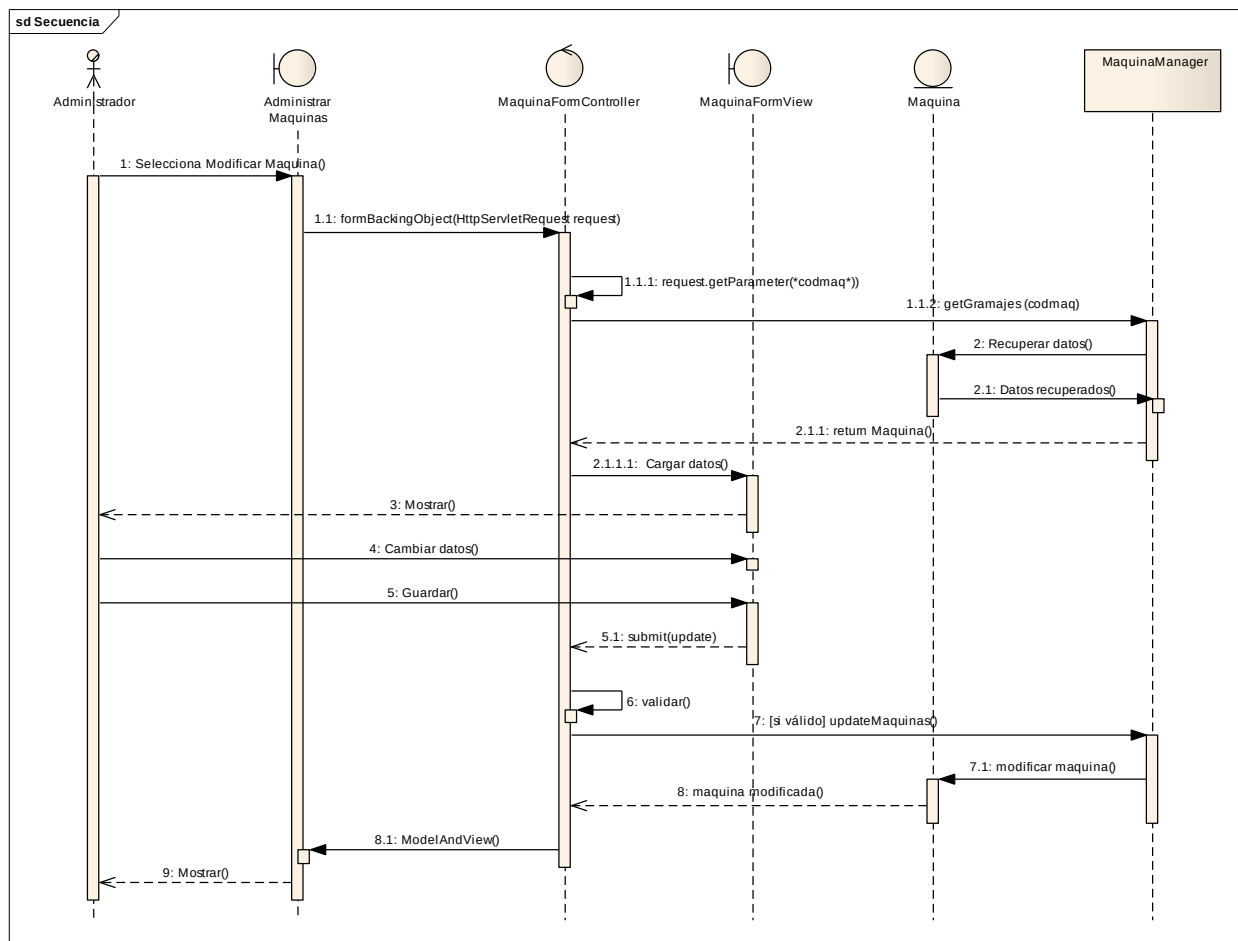


Figura 99. D.S. Modificar Máquina

Diagrama de Secuencia: Administrar Ventas

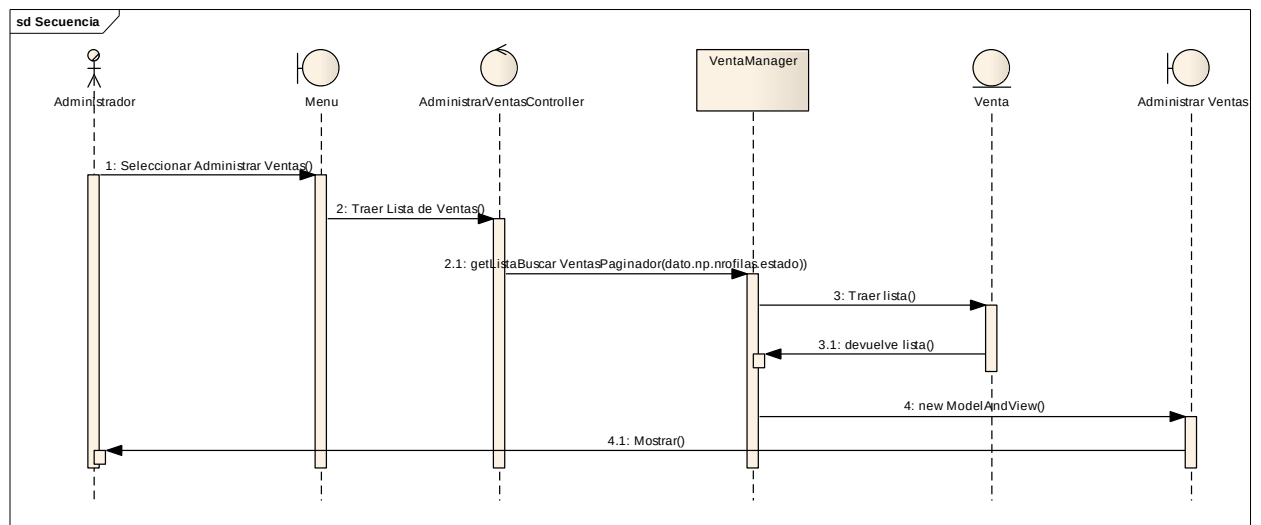


Figura 100. D.S. Administrar Ventas

Diagrama de Secuencia: Administrar Compras

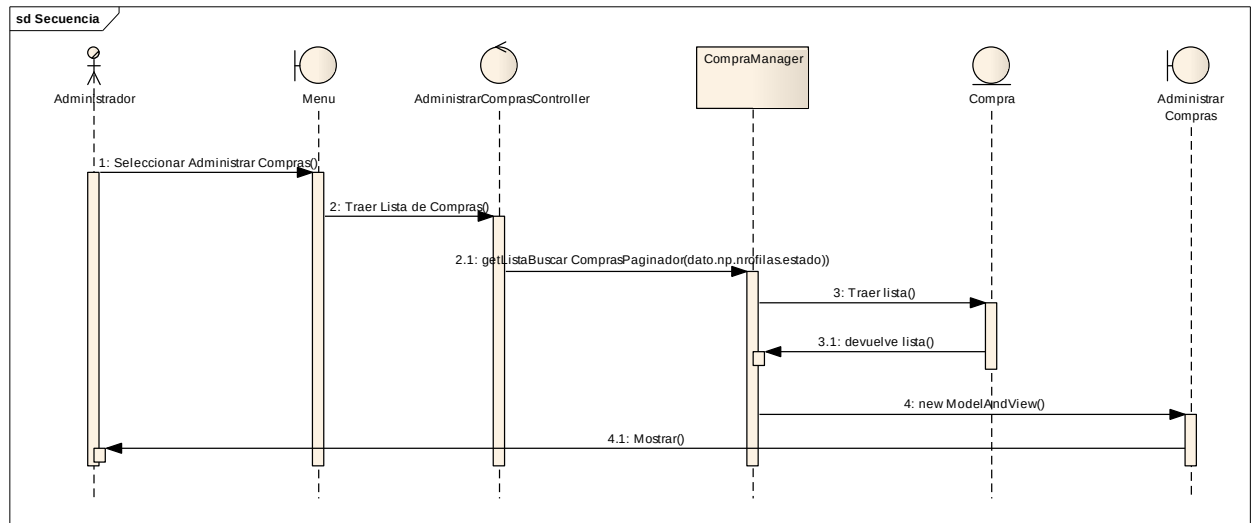


Figura 101. D.S. Administrar Compras

Diagrama de Secuencia: Administrar Productos

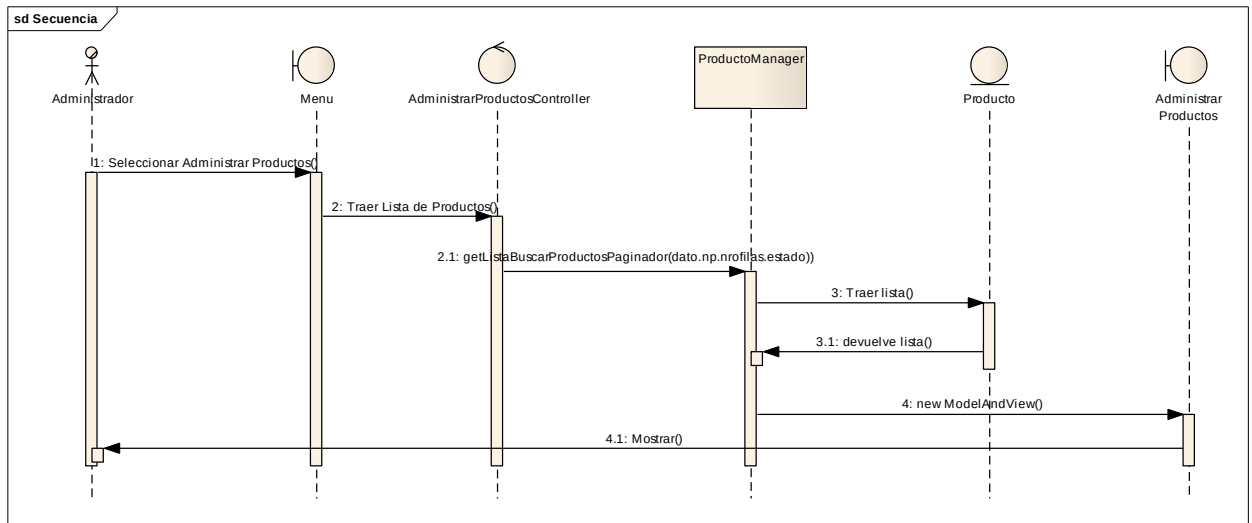


Figura 102. D.S. Administrar Productos

II.1.11.5. MODELO DE ANÁLISIS Y DISEÑO

II.1.11.6. Introducción

El Diagrama de Clases es un artefacto de la disciplina Análisis/Diseño en la metodología RUP la cual se está implementando.

II.1.11.7. Propósito

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

II.1.11.8. Alcance

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

II.1.11.9. Diagrama de clases

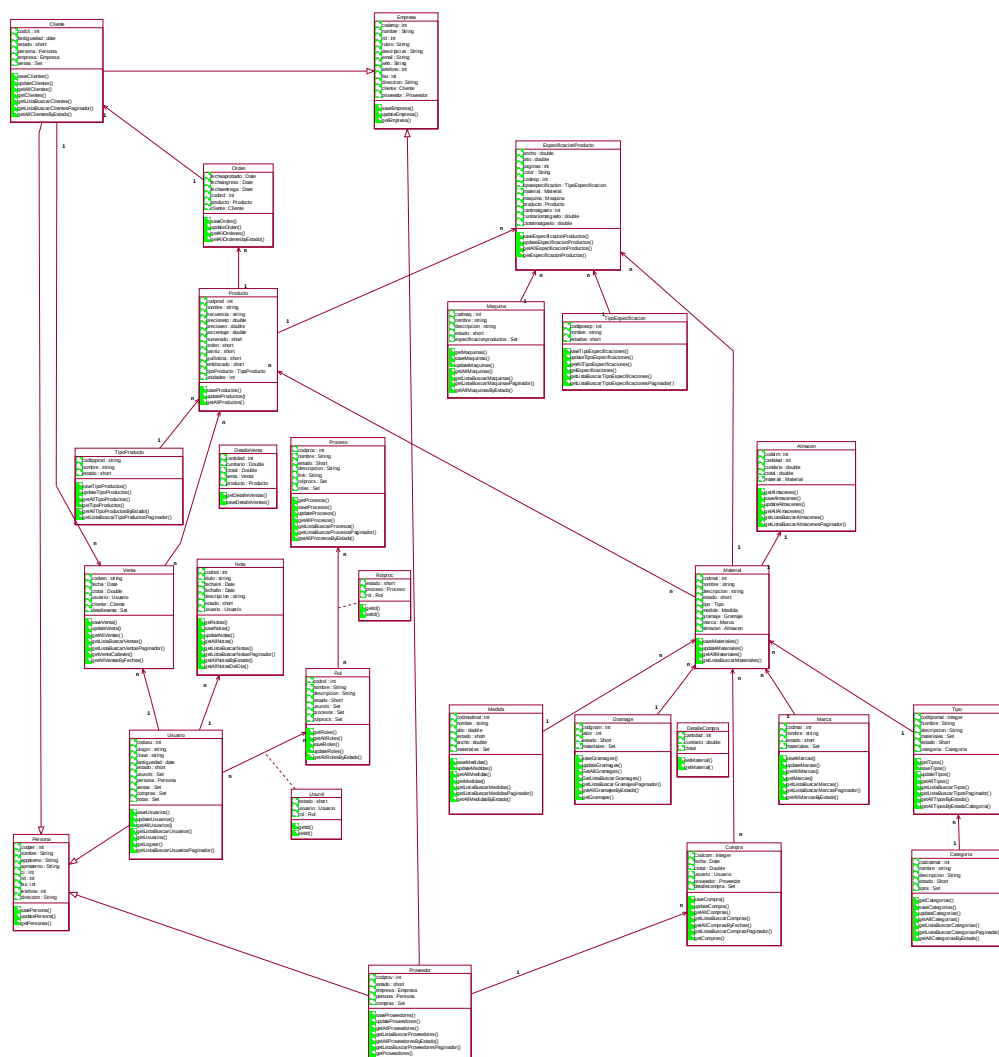


Figura 103. Diagrama de Clases

II.1.12. MODELO DE DATOS

II.1.12.1. Introducción

Un modelo de datos es un lenguaje orientado a descubrir una base de datos. Típicamente un Modelo de Datos permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- Operaciones de manipulación de los datos: Típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Otro enfoque es pensar que un modelo de datos permite describir los elementos que intervienen en una realidad dada, y la forma en que se relacionan esos elementos entre sí.

No hay que perder de vista que una Base de Datos siempre está orientada a resolver un problema determinado, por lo que los dos enfoques propuestos son necesarios en cualquier desarrollo de software.

El Modelado de Datos nos sirve para tener un detalle de las tablas con sus respectivos campos de la base de datos.

II.1.12.2. Propósito

- Comprender la estructura de las tablas y sus campos, en la base de datos del sistema.
- Identificar los tipos de campos de cada tabla de la base de datos

II.1.12.3. Alcance

- Identificar y definir las relaciones entre las diferentes tablas de la base de datos de nuestro sistema deseado y aprobado por la organización.

II.1.12.4. Creación de las tablas

```
--
-- PostgreSQL database dump
--

--

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

--
-- TOC entry 407 (class 2612 OID 23499)
-- Name: plpgsql; Type: PROCEDURAL LANGUAGE; Schema: -; Owner:
postgres
--

CREATE PROCEDURAL LANGUAGE plpgsql;

ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;

SET search_path = public, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = false;

--
-- TOC entry 1594 (class 1259 OID 23500)
-- Dependencies: 6
-- Name: almacén; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE almacén (
    codalm integer NOT NULL,
    codmat integer NOT NULL,
    cantidad integer NOT NULL,
    cunitario double precision NOT NULL,
    cttotal double precision
);

ALTER TABLE public.almacén OWNER TO postgres;
```

```

--
-- TOC entry 1595 (class 1259 OID 23503)
-- Dependencies: 1594 6
-- Name: almacén_codalm_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE almacén_codalm_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.almacén_codalm_seq OWNER TO postgres;

--
-- TOC entry 2110 (class 0 OID 0)
-- Dependencies: 1595
-- Name: almacén_codalm_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE almacén_codalm_seq OWNED BY almacén.codalm;

--
-- TOC entry 1596 (class 1259 OID 23505)
-- Dependencies: 6
-- Name: categoriamat; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE categoriamat (
    codcatmat integer NOT NULL,
    nombre character varying(150),
    descripción character varying(200),
    estado smallint
);

ALTER TABLE public.categoriamat OWNER TO postgres;

--
-- TOC entry 1597 (class 1259 OID 23508)
-- Dependencies: 1596 6
-- Name: categoriamat_codcatmat_seq; Type: SEQUENCE; Schema:
public; Owner: postgres

```

```
--
CREATE SEQUENCE categoriamat_codcatmat_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.categoriamat_codcatmat_seq OWNER TO
postgres;

--
-- TOC entry 2111 (class 0 OID 0)
-- Dependencies: 1597
-- Name: categoriamat_codcatmat_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE categoriamat_codcatmat_seq OWNED BY
categoriamat.codcatmat;

--
-- TOC entry 1598 (class 1259 OID 23510)
-- Dependencies: 6
-- Name: cliente; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE cliente (
    codcli integer NOT NULL,
    fantiguedad date,
    estado smallint
);

ALTER TABLE public.cliente OWNER TO postgres;

--
-- TOC entry 1599 (class 1259 OID 23513)
-- Dependencies: 1598 6
-- Name: cliente_codcli_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE cliente_codcli_seq
    START WITH 1
```

```

INCREMENT BY 1
NO MAXVALUE
NO MINVALUE
CACHE 1;

```

```

ALTER TABLE public.cliente_codcli_seq OWNER TO postgres;

```

```

--
-- TOC entry 2112 (class 0 OID 0)
-- Dependencies: 1599
-- Name: cliente_codcli_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

```

```

ALTER SEQUENCE cliente_codcli_seq OWNED BY cliente.codcli;

```

```

--
-- TOC entry 1600 (class 1259 OID 23515)
-- Dependencies: 6
-- Name: clienteempresa; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

```

```

CREATE TABLE clienteempresa (
    codcli integer NOT NULL,
    codemp integer NOT NULL
);

```

```

ALTER TABLE public.clienteempresa OWNER TO postgres;

```

```

--
-- TOC entry 1601 (class 1259 OID 23518)
-- Dependencies: 6
-- Name: clientepersona; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

```

```

CREATE TABLE clientepersona (
    codper integer NOT NULL,
    codcli integer NOT NULL
);

```

```

ALTER TABLE public.clientepersona OWNER TO postgres;

```

```

--
-- TOC entry 1602 (class 1259 OID 23526)

```

```
-- Dependencies: 6
-- Name: compra; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE compra (
    codcom integer NOT NULL,
    codusu integer NOT NULL,
    codprov integer,
    fecha date NOT NULL,
    ctototal double precision NOT NULL
);
```

```
ALTER TABLE public.compra OWNER TO postgres;
```

```
--
-- TOC entry 1603 (class 1259 OID 23529)
-- Dependencies: 6 1602
-- Name: compra_codcom_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--
```

```
CREATE SEQUENCE compra_codcom_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.compra_codcom_seq OWNER TO postgres;
```

```
--
-- TOC entry 2113 (class 0 OID 0)
-- Dependencies: 1603
-- Name: compra_codcom_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--
```

```
ALTER SEQUENCE compra_codcom_seq OWNED BY compra.codcom;
```

```
--
-- TOC entry 1604 (class 1259 OID 23531)
-- Dependencies: 6
-- Name: datos; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE datos (
    codusu integer NOT NULL,
    login character varying(50) NOT NULL,
    clave character varying(50) NOT NULL,
    estado smallint NOT NULL
);
```

```
ALTER TABLE public.datos OWNER TO postgres;
```

```
--
-- TOC entry 1605 (class 1259 OID 23534)
-- Dependencies: 6 1604
-- Name: datos_codper_seq; Type: SEQUENCE; Schema: public;
-- Owner: postgres
--
```

```
CREATE SEQUENCE datos_codper_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.datos_codper_seq OWNER TO postgres;
```

```
--
-- TOC entry 2114 (class 0 OID 0)
-- Dependencies: 1605
-- Name: datos_codper_seq; Type: SEQUENCE OWNED BY; Schema:
-- public; Owner: postgres
--
```

```
ALTER SEQUENCE datos_codper_seq OWNED BY datos.codusu;
```

```
--
-- TOC entry 1606 (class 1259 OID 23536)
-- Dependencies: 6
-- Name: detallecompra; Type: TABLE; Schema: public; Owner:
-- postgres; Tablespace:
--
```

```
CREATE TABLE detallecompra (
    codcom integer NOT NULL,
    codmat integer NOT NULL,
    cantidad integer NOT NULL,
```

```

        cttotal double precision NOT NULL,
        cunitario double precision NOT NULL
    );

```

```

ALTER TABLE public.detallecompra OWNER TO postgres;

```

```

--
-- TOC entry 1607 (class 1259 OID 23539)
-- Dependencies: 6
-- Name: detalleproducto; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

```

```

CREATE TABLE detalleproducto (
    codprod integer NOT NULL,
    codmat integer NOT NULL,
    cantidad integer,
    cunitario double precision,
    cttotal double precision,
    codesp integer NOT NULL
);

```

```

ALTER TABLE public.detalleproducto OWNER TO postgres;

```

```

--
-- TOC entry 1608 (class 1259 OID 23542)
-- Dependencies: 6
-- Name: detalleventa; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

```

```

CREATE TABLE detalleventa (
    codven integer NOT NULL,
    codprod integer NOT NULL,
    cantidad integer NOT NULL,
    cunitario double precision NOT NULL,
    cttotal double precision
);

```

```

ALTER TABLE public.detalleventa OWNER TO postgres;

```

```

--
-- TOC entry 1609 (class 1259 OID 23545)
-- Dependencies: 6
-- Name: empresa; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:

```

```

--

CREATE TABLE empresa (
    codemp integer NOT NULL,
    nombre character varying(100),
    nit bigint NOT NULL,
    rubro character varying(50),
    email character varying(50),
    web character varying(50),
    telefono integer,
    fax integer,
    direccion character varying(100)
);

ALTER TABLE public.empresa OWNER TO postgres;

--
-- TOC entry 1610 (class 1259 OID 23551)
-- Dependencies: 1609 6
-- Name: empresa_codemp_seq; Type: SEQUENCE; Schema: public;
-- Owner: postgres
--

CREATE SEQUENCE empresa_codemp_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.empresa_codemp_seq OWNER TO postgres;

--
-- TOC entry 2115 (class 0 OID 0)
-- Dependencies: 1610
-- Name: empresa_codemp_seq; Type: SEQUENCE OWNED BY; Schema:
-- public; Owner: postgres
--

ALTER SEQUENCE empresa_codemp_seq OWNED BY empresa.codemp;

--
-- TOC entry 1653 (class 1259 OID 24021)
-- Dependencies: 6
-- Name: especificacionprod; Type: TABLE; Schema: public;
-- Owner: postgres; Tablespace:

```



```
--
```

```
CREATE TABLE especificacionprod (
    codtipoesp integer NOT NULL,
    ancho double precision NOT NULL,
    alto double precision NOT NULL,
    paginas integer,
    codmaq integer,
    color character varying NOT NULL,
    codesp integer NOT NULL,
    codprod integer NOT NULL,
    cantmatgasto integer,
    cunitariomatgasto double precision,
    ctotalmatgasto double precision,
    codmat integer
);
```

```
ALTER TABLE public.especificacionprod OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1652 (class 1259 OID 24019)
-- Dependencies: 6 1653
-- Name: especificacionprod_codesp_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE especificacionprod_codesp_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.especificacionprod_codesp_seq OWNER TO
postgres;
```

```
--
```

```
-- TOC entry 2116 (class 0 OID 0)
-- Dependencies: 1652
-- Name: especificacionprod_codesp_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE especificacionprod_codesp_seq OWNED BY
especificacionprod.codesp;
```

```

--
-- TOC entry 1611 (class 1259 OID 23561)
-- Dependencies: 6
-- Name: fotosprod; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE fotosprod (
    codfoto integer NOT NULL,
    codtipprod integer NOT NULL,
    link character varying(200) NOT NULL,
    nombre character varying,
    estado smallint
);

ALTER TABLE public.fotosprod OWNER TO postgres;

--
-- TOC entry 1612 (class 1259 OID 23564)
-- Dependencies: 6 1611
-- Name: fotosprod_codfoto_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE fotosprod_codfoto_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.fotosprod_codfoto_seq OWNER TO postgres;

--
-- TOC entry 2117 (class 0 OID 0)
-- Dependencies: 1612
-- Name: fotosprod_codfoto_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE fotosprod_codfoto_seq OWNED BY
fotosprod.codfoto;

--
-- TOC entry 1613 (class 1259 OID 23566)
-- Dependencies: 1939 6

```

```
-- Name: gramaje; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--
```

```
CREATE TABLE gramaje (
    codgram integer NOT NULL,
    estado smallint DEFAULT 1 NOT NULL,
    valor character varying NOT NULL
);
```

```
ALTER TABLE public.gramaje OWNER TO postgres;
```

```
--
-- TOC entry 1614 (class 1259 OID 23570)
-- Dependencies: 1613 6
-- Name: gramaje_codgram_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--
```

```
CREATE SEQUENCE gramaje_codgram_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.gramaje_codgram_seq OWNER TO postgres;
```

```
--
-- TOC entry 2118 (class 0 OID 0)
-- Dependencies: 1614
-- Name: gramaje_codgram_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--
```

```
ALTER SEQUENCE gramaje_codgram_seq OWNED BY gramaje.codgram;
```

```
--
-- TOC entry 1615 (class 1259 OID 23572)
-- Dependencies: 6
-- Name: gramajemat; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--
```

```
CREATE TABLE gramajemat (
    codgram integer NOT NULL,
```

```

        codmat integer NOT NULL
    );

ALTER TABLE public.gramajemat OWNER TO postgres;

--
-- TOC entry 1616 (class 1259 OID 23575)
-- Dependencies: 6
-- Name: maquina; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE maquina (
    codmaq integer NOT NULL,
    nombre character varying(20) NOT NULL,
    descripcion character varying(150),
    estado smallint NOT NULL
);

ALTER TABLE public.maquina OWNER TO postgres;

--
-- TOC entry 1617 (class 1259 OID 23578)
-- Dependencies: 1616 6
-- Name: maquina_codmaq_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE maquina_codmaq_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.maquina_codmaq_seq OWNER TO postgres;

--
-- TOC entry 2119 (class 0 OID 0)
-- Dependencies: 1617
-- Name: maquina_codmaq_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE maquina_codmaq_seq OWNED BY maquina.codmaq;

```

```
--
-- TOC entry 1618 (class 1259 OID 23580)
-- Dependencies: 1942 6
-- Name: marca; Type: TABLE; Schema: public; Owner: postgres;
-- Tablespace:
```

```
CREATE TABLE marca (
    codmar integer NOT NULL,
    nombre character varying NOT NULL,
    estado smallint DEFAULT 1 NOT NULL
);
```

```
ALTER TABLE public.marca OWNER TO postgres;
```

```
--
-- TOC entry 1619 (class 1259 OID 23587)
-- Dependencies: 1618 6
-- Name: marca_codmar_seq; Type: SEQUENCE; Schema: public;
-- Owner: postgres
--
```

```
CREATE SEQUENCE marca_codmar_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.marca_codmar_seq OWNER TO postgres;
```

```
--
-- TOC entry 2120 (class 0 OID 0)
-- Dependencies: 1619
-- Name: marca_codmar_seq; Type: SEQUENCE OWNED BY; Schema:
-- public; Owner: postgres
--
```

```
ALTER SEQUENCE marca_codmar_seq OWNED BY marca.codmar;
```

```
--
-- TOC entry 1620 (class 1259 OID 23589)
-- Dependencies: 6
-- Name: material; Type: TABLE; Schema: public; Owner:
-- postgres; Tablespace:
```

--

```
CREATE TABLE material (
    codmat integer NOT NULL,
    nombre character varying(150),
    estado smallint,
    descripcion character varying(200),
    codgram integer,
    codmedmat integer,
    codmar integer,
    codtipomat integer
);
```

```
ALTER TABLE public.material OWNER TO postgres;
```

--

```
-- TOC entry 1621 (class 1259 OID 23592)
-- Dependencies: 1620 6
-- Name: material_codmat_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
```

--

```
CREATE SEQUENCE material_codmat_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.material_codmat_seq OWNER TO postgres;
```

--

```
-- TOC entry 2121 (class 0 OID 0)
-- Dependencies: 1621
-- Name: material_codmat_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
```

--

```
ALTER SEQUENCE material_codmat_seq OWNED BY material.codmat;
```

--

```
-- TOC entry 1622 (class 1259 OID 23594)
-- Dependencies: 6
-- Name: medidamat; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
```

--

```
CREATE TABLE medidamat (
    codmedmat integer NOT NULL,
    nombre character varying(20),
    estado smallint,
    alto double precision,
    ancho double precision
);
```

```
ALTER TABLE public.medidamat OWNER TO postgres;
```

```
--
-- TOC entry 1623 (class 1259 OID 23597)
-- Dependencies: 1622 6
-- Name: medidamat_codmedmat_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--
```

```
CREATE SEQUENCE medidamat_codmedmat_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.medidamat_codmedmat_seq OWNER TO postgres;
```

```
--
-- TOC entry 2122 (class 0 OID 0)
-- Dependencies: 1623
-- Name: medidamat_codmedmat_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE medidamat_codmedmat_seq OWNED BY
medidamat.codmedmat;
```

```
--
-- TOC entry 1624 (class 1259 OID 23599)
-- Dependencies: 6
-- Name: medidamatmat; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--
```

```
CREATE TABLE medidamatmat (
    codmedmat integer NOT NULL,
```

```

        codmat integer NOT NULL
    );

ALTER TABLE public.medidamatmat OWNER TO postgres;

--
-- TOC entry 1625 (class 1259 OID 23602)
-- Dependencies: 6
-- Name: nota; Type: TABLE; Schema: public; Owner: postgres;
-- Tablespace:
--

CREATE TABLE nota (
    codnot integer NOT NULL,
    titulo character varying(20),
    descripcion character varying(150),
    fechaini date,
    fechafin date,
    estado smallint,
    codusu integer NOT NULL
);

ALTER TABLE public.nota OWNER TO postgres;

--
-- TOC entry 1626 (class 1259 OID 23605)
-- Dependencies: 6 1625
-- Name: nota_codnot_seq; Type: SEQUENCE; Schema: public;
-- Owner: postgres
--

CREATE SEQUENCE nota_codnot_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.nota_codnot_seq OWNER TO postgres;

--
-- TOC entry 2123 (class 0 OID 0)
-- Dependencies: 1626
-- Name: nota_codnot_seq; Type: SEQUENCE OWNED BY; Schema:
-- public; Owner: postgres
--

```



```

ALTER SEQUENCE nota_codnot_seq OWNED BY nota.codnot;

--
-- TOC entry 1655 (class 1259 OID 24061)
-- Dependencies: 6
-- Name: orden; Type: TABLE; Schema: public; Owner: postgres;
-- Tablespace:
--

CREATE TABLE orden (
    codord integer NOT NULL,
    codcli integer NOT NULL,
    codprod integer NOT NULL,
    fechaingreso date,
    fechaaprobado date,
    fechaentrega date,
    estado smallint
);

ALTER TABLE public.orden OWNER TO postgres;

--
-- TOC entry 1654 (class 1259 OID 24059)
-- Dependencies: 1655 6
-- Name: orden_codord_seq; Type: SEQUENCE; Schema: public;
-- Owner: postgres
--

CREATE SEQUENCE orden_codord_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.orden_codord_seq OWNER TO postgres;

--
-- TOC entry 2124 (class 0 OID 0)
-- Dependencies: 1654
-- Name: orden_codord_seq; Type: SEQUENCE OWNED BY; Schema:
-- public; Owner: postgres
--

ALTER SEQUENCE orden_codord_seq OWNED BY orden.codord;

```

```

--
-- TOC entry 1627 (class 1259 OID 23607)
-- Dependencies: 6
-- Name: persona; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE persona (
    codper integer NOT NULL,
    nombre character varying(50),
    appaterno character varying(50),
    apmaterno character varying(50),
    ci integer,
    nit bigint,
    fax integer,
    telefono integer,
    direccion character varying(100)
);

ALTER TABLE public.persona OWNER TO postgres;

--
-- TOC entry 1628 (class 1259 OID 23610)
-- Dependencies: 6 1627
-- Name: persona_codper_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE persona_codper_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.persona_codper_seq OWNER TO postgres;

--
-- TOC entry 2125 (class 0 OID 0)
-- Dependencies: 1628
-- Name: persona_codper_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE persona_codper_seq OWNED BY persona.codper;

```

```

--
-- TOC entry 1629 (class 1259 OID 23612)
-- Dependencies: 1948 6
-- Name: proceso; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE proceso (
    codproc integer NOT NULL,
    nombre character varying(50),
    descripcion character varying(150),
    estado smallint DEFAULT 1,
    link character varying(100)
);

ALTER TABLE public.proceso OWNER TO postgres;

--
-- TOC entry 1630 (class 1259 OID 23616)
-- Dependencies: 6 1629
-- Name: proceso_codproc_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE proceso_codproc_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.proceso_codproc_seq OWNER TO postgres;

--
-- TOC entry 2126 (class 0 OID 0)
-- Dependencies: 1630
-- Name: proceso_codproc_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE proceso_codproc_seq OWNED BY proceso.codproc;

--
-- TOC entry 1631 (class 1259 OID 23618)

```

```

-- Dependencies: 6
-- Name: producto; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE producto (
    codprod integer NOT NULL,
    codtipprod integer,
    nombre character varying(200),
    fechaingreso date,
    fechaaprobado date,
    fechaentrega date,
    frecuencia integer,
    precioneto double precision,
    precioven double precision,
    porcentaje double precision,
    numerado smallint,
    barniz smallint,
    guillotina smallint,
    emblocado smallint,
    estado smallint,
    unidades integer
);

ALTER TABLE public.producto OWNER TO postgres;

--
-- TOC entry 1632 (class 1259 OID 23621)
-- Dependencies: 6 1631
-- Name: producto_codprod_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE producto_codprod_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.producto_codprod_seq OWNER TO postgres;

--
-- TOC entry 2127 (class 0 OID 0)
-- Dependencies: 1632
-- Name: producto_codprod_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres

```

```
--
ALTER SEQUENCE producto_codprod_seq OWNED BY producto.codprod;

--
-- TOC entry 1633 (class 1259 OID 23623)
-- Dependencies: 6
-- Name: proveedor; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE proveedor (
    codprov integer NOT NULL,
    descripcion character varying(150),
    estado smallint
);

ALTER TABLE public.proveedor OWNER TO postgres;

--
-- TOC entry 1634 (class 1259 OID 23626)
-- Dependencies: 6 1633
-- Name: proveedor_codprov_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE proveedor_codprov_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.proveedor_codprov_seq OWNER TO postgres;

--
-- TOC entry 2128 (class 0 OID 0)
-- Dependencies: 1634
-- Name: proveedor_codprov_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE proveedor_codprov_seq OWNED BY
proveedor.codprov;
```

```
--
-- TOC entry 1635 (class 1259 OID 23628)
-- Dependencies: 6
-- Name: proveedorempresa; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--
```

```
CREATE TABLE proveedorempresa (
    codprov integer NOT NULL,
    codemp integer NOT NULL
);
```

```
ALTER TABLE public.proveedorempresa OWNER TO postgres;
```

```
--
-- TOC entry 1636 (class 1259 OID 23631)
-- Dependencies: 6
-- Name: proveedorpersona; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--
```

```
CREATE TABLE proveedorpersona (
    codprov integer NOT NULL,
    codper integer NOT NULL
);
```

```
ALTER TABLE public.proveedorpersona OWNER TO postgres;
```

```
--
-- TOC entry 1637 (class 1259 OID 23634)
-- Dependencies: 1952 6
-- Name: rol; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE rol (
    codrol integer NOT NULL,
    nombre character varying(50),
    descripcion character varying(150),
    estado smallint DEFAULT 1
);
```

```
ALTER TABLE public.rol OWNER TO postgres;
```

```
--
-- TOC entry 1638 (class 1259 OID 23638)
```

```

-- Dependencies: 6 1637
-- Name: rol_codrol_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE rol_codrol_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.rol_codrol_seq OWNER TO postgres;

--
-- TOC entry 2129 (class 0 OID 0)
-- Dependencies: 1638
-- Name: rol_codrol_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE rol_codrol_seq OWNED BY rol.codrol;

--
-- TOC entry 1639 (class 1259 OID 23640)
-- Dependencies: 6
-- Name: rolproc; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE rolproc (
    codrol integer NOT NULL,
    codproc integer NOT NULL,
    estado smallint
);

ALTER TABLE public.rolproc OWNER TO postgres;

--
-- TOC entry 1640 (class 1259 OID 23643)
-- Dependencies: 6
-- Name: tipoespecificacion; Type: TABLE; Schema: public;
Owner: postgres; Tablespace:
--

CREATE TABLE tipoespecificacion (

```

```

        codtipoesp integer NOT NULL,
        nombre character varying(20),
        estadoe smallint
    );

ALTER TABLE public.tipoespecificacion OWNER TO postgres;

--
-- TOC entry 1641 (class 1259 OID 23646)
-- Dependencies: 6 1640
-- Name: tipoespecificacion_codtipoesp_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE tipoespecificacion_codtipoesp_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.tipoespecificacion_codtipoesp_seq OWNER TO
postgres;

--
-- TOC entry 2130 (class 0 OID 0)
-- Dependencies: 1641
-- Name: tipoespecificacion_codtipoesp_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE tipoespecificacion_codtipoesp_seq OWNED BY
tipoespecificacion.codtipoesp;

--
-- TOC entry 1642 (class 1259 OID 23648)
-- Dependencies: 6
-- Name: tipomat; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE tipomat (
    codtipomat integer NOT NULL,
    codcatmat integer NOT NULL,
    nombre character varying(150),
    descripcion character varying(200),

```



```

        estado smallint
    );

```

```

ALTER TABLE public.tipomat OWNER TO postgres;

```

```

--
-- TOC entry 1643 (class 1259 OID 23651)
-- Dependencies: 6 1642
-- Name: tipomat_codtipomat_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

```

```

CREATE SEQUENCE tipomat_codtipomat_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

```

```

ALTER TABLE public.tipomat_codtipomat_seq OWNER TO postgres;

```

```

--
-- TOC entry 2131 (class 0 OID 0)
-- Dependencies: 1643
-- Name: tipomat_codtipomat_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

```

```

ALTER SEQUENCE tipomat_codtipomat_seq OWNED BY
tipomat.codtipomat;

```

```

--
-- TOC entry 1644 (class 1259 OID 23653)
-- Dependencies: 6
-- Name: tipomatmat; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

```

```

CREATE TABLE tipomatmat (
    codtipomat integer NOT NULL,
    codmat integer NOT NULL
);

```

```

ALTER TABLE public.tipomatmat OWNER TO postgres;

```

```

--
-- TOC entry 1645 (class 1259 OID 23656)
-- Dependencies: 6
-- Name: tipoproducto; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE tipoproducto (
    codtipprod integer NOT NULL,
    nombre character varying(20) NOT NULL,
    estado smallint NOT NULL,
    descripcion character varying
);

ALTER TABLE public.tipoproducto OWNER TO postgres;

--
-- TOC entry 1646 (class 1259 OID 23662)
-- Dependencies: 1645 6
-- Name: tipoproducto_codtipprod_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE tipoproducto_codtipprod_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.tipoproducto_codtipprod_seq OWNER TO
postgres;

--
-- TOC entry 2132 (class 0 OID 0)
-- Dependencies: 1646
-- Name: tipoproducto_codtipprod_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE tipoproducto_codtipprod_seq OWNED BY
tipoproducto.codtipprod;

--
-- TOC entry 1647 (class 1259 OID 23664)
-- Dependencies: 6

```

```
-- Name: usuario; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--
```

```
CREATE TABLE usuario (
    codusu integer NOT NULL,
    clave character varying(20) NOT NULL,
    fantiguedad date,
    estado smallint NOT NULL,
    ulogin character varying(20) NOT NULL,
    codper integer NOT NULL
);
```

```
ALTER TABLE public.usuario OWNER TO postgres;
```

```
--
-- TOC entry 1648 (class 1259 OID 23667)
-- Dependencies: 1647 6
-- Name: usuario_codusu_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--
```

```
CREATE SEQUENCE usuario_codusu_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.usuario_codusu_seq OWNER TO postgres;
```

```
--
-- TOC entry 2133 (class 0 OID 0)
-- Dependencies: 1648
-- Name: usuario_codusu_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--
```

```
ALTER SEQUENCE usuario_codusu_seq OWNED BY usuario.codusu;
```

```
--
-- TOC entry 1649 (class 1259 OID 23669)
-- Dependencies: 6
-- Name: usurol; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE usuro1 (
    codusu integer NOT NULL,
    codrol integer NOT NULL,
    estado smallint
);
```

```
ALTER TABLE public.usuro1 OWNER TO postgres;
```

```
--
-- TOC entry 1650 (class 1259 OID 23672)
-- Dependencies: 6
-- Name: venta; Type: TABLE; Schema: public; Owner: postgres;
-- Tablespace:
```

```
CREATE TABLE venta (
    codven integer NOT NULL,
    codusu integer NOT NULL,
    codcli integer NOT NULL,
    fecha date NOT NULL,
    cttotal double precision NOT NULL
);
```

```
ALTER TABLE public.venta OWNER TO postgres;
```

```
--
-- TOC entry 1651 (class 1259 OID 23675)
-- Dependencies: 1650 6
-- Name: venta_codven_seq; Type: SEQUENCE; Schema: public;
-- Owner: postgres
--
```

```
CREATE SEQUENCE venta_codven_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.venta_codven_seq OWNER TO postgres;
```

```
--
-- TOC entry 2134 (class 0 OID 0)
-- Dependencies: 1651
```

```
-- Name: venta_codven_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--
```

```
ALTER SEQUENCE venta_codven_seq OWNED BY venta.codven;
```

```
--
-- TOC entry 1933 (class 2604 OID 23677)
-- Dependencies: 1595 1594
-- Name: codalm; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE almacen ALTER COLUMN codalm SET DEFAULT
nextval('almacen_codalm_seq'::regclass);
```

```
--
-- TOC entry 1934 (class 2604 OID 23678)
-- Dependencies: 1597 1596
-- Name: codcatmat; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE categoriamat ALTER COLUMN codcatmat SET DEFAULT
nextval('categoriamat_codcatmat_seq'::regclass);
```

```
--
-- TOC entry 1935 (class 2604 OID 23679)
-- Dependencies: 1599 1598
-- Name: codcli; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE cliente ALTER COLUMN codcli SET DEFAULT
nextval('cliente_codcli_seq'::regclass);
```

```
--
-- TOC entry 1936 (class 2604 OID 23681)
-- Dependencies: 1603 1602
-- Name: codcom; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE compra ALTER COLUMN codcom SET DEFAULT
nextval('compra_codcom_seq'::regclass);
```

```
--
-- TOC entry 1937 (class 2604 OID 23682)
-- Dependencies: 1610 1609
-- Name: codemp; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE empresa ALTER COLUMN codemp SET DEFAULT
nextval('empresa_codemp_seq'::regclass);
```

```
--
-- TOC entry 1959 (class 2604 OID 24079)
-- Dependencies: 1653 1652 1653
-- Name: codesp; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE especificacionprod ALTER COLUMN codesp SET DEFAULT
nextval('especificacionprod_codesp_seq'::regclass);
```

```
--
-- TOC entry 1938 (class 2604 OID 23683)
-- Dependencies: 1612 1611
-- Name: codfoto; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE fotosprod ALTER COLUMN codfoto SET DEFAULT
nextval('fotosprod_codfoto_seq'::regclass);
```

```
--
-- TOC entry 1940 (class 2604 OID 23684)
-- Dependencies: 1614 1613
-- Name: codgram; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE gramaje ALTER COLUMN codgram SET DEFAULT
nextval('gramaje_codgram_seq'::regclass);
```

```
--
-- TOC entry 1941 (class 2604 OID 23685)
-- Dependencies: 1617 1616
```

```
-- Name: codmaq; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE maquina ALTER COLUMN codmaq SET DEFAULT
nextval('maquina_codmaq_seq'::regclass);
```

```
--
-- TOC entry 1943 (class 2604 OID 23686)
-- Dependencies: 1619 1618
-- Name: codmar; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE marca ALTER COLUMN codmar SET DEFAULT
nextval('marca_codmar_seq'::regclass);
```

```
--
-- TOC entry 1944 (class 2604 OID 23687)
-- Dependencies: 1621 1620
-- Name: codmat; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE material ALTER COLUMN codmat SET DEFAULT
nextval('material_codmat_seq'::regclass);
```

```
--
-- TOC entry 1945 (class 2604 OID 23688)
-- Dependencies: 1623 1622
-- Name: codmedmat; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE medidamat ALTER COLUMN codmedmat SET DEFAULT
nextval('medidamat_codmedmat_seq'::regclass);
```

```
--
-- TOC entry 1946 (class 2604 OID 23689)
-- Dependencies: 1626 1625
-- Name: codnot; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE nota ALTER COLUMN codnot SET DEFAULT
nextval('nota_codnot_seq'::regclass);
```

```
--
```

```
-- TOC entry 1960 (class 2604 OID 24064)
-- Dependencies: 1655 1654 1655
-- Name: codord; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE orden ALTER COLUMN codord SET DEFAULT
nextval('orden_codord_seq'::regclass);
```

```
--
```

```
-- TOC entry 1947 (class 2604 OID 23690)
-- Dependencies: 1628 1627
-- Name: codper; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE persona ALTER COLUMN codper SET DEFAULT
nextval('persona_codper_seq'::regclass);
```

```
--
```

```
-- TOC entry 1949 (class 2604 OID 23691)
-- Dependencies: 1630 1629
-- Name: codproc; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE proceso ALTER COLUMN codproc SET DEFAULT
nextval('proceso_codproc_seq'::regclass);
```

```
--
```

```
-- TOC entry 1950 (class 2604 OID 23692)
-- Dependencies: 1632 1631
-- Name: codprod; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE producto ALTER COLUMN codprod SET DEFAULT
nextval('producto_codprod_seq'::regclass);
```

```
--
```



```
-- TOC entry 1951 (class 2604 OID 23693)
-- Dependencies: 1634 1633
-- Name: codprov; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE proveedor ALTER COLUMN codprov SET DEFAULT
nextval('proveedor_codprov_seq'::regclass);
```

```
--
-- TOC entry 1953 (class 2604 OID 23694)
-- Dependencies: 1638 1637
-- Name: codrol; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE rol ALTER COLUMN codrol SET DEFAULT
nextval('rol_codrol_seq'::regclass);
```

```
--
-- TOC entry 1954 (class 2604 OID 23695)
-- Dependencies: 1641 1640
-- Name: codtipoesp; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE tipospecificacion ALTER COLUMN codtipoesp SET
DEFAULT
nextval('tipospecificacion_codtipoesp_seq'::regclass);
```

```
--
-- TOC entry 1955 (class 2604 OID 23696)
-- Dependencies: 1643 1642
-- Name: codtipomat; Type: DEFAULT; Schema: public; Owner:
postgres
--
```

```
ALTER TABLE tipomat ALTER COLUMN codtipomat SET DEFAULT
nextval('tipomat_codtipomat_seq'::regclass);
```

```
--
-- TOC entry 1956 (class 2604 OID 23697)
-- Dependencies: 1646 1645
-- Name: codtipprod; Type: DEFAULT; Schema: public; Owner:
postgres
```

```
--

ALTER TABLE tipoproducto ALTER COLUMN codtipprod SET DEFAULT
nextval('tipoproducto_codtipprod_seq'::regclass);

--
-- TOC entry 1957 (class 2604 OID 23698)
-- Dependencies: 1648 1647
-- Name: codusu; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE usuario ALTER COLUMN codusu SET DEFAULT
nextval('usuario_codusu_seq'::regclass);

--
-- TOC entry 1958 (class 2604 OID 23699)
-- Dependencies: 1651 1650
-- Name: codven; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE venta ALTER COLUMN codven SET DEFAULT
nextval('venta_codven_seq'::regclass);

--
-- TOC entry 1962 (class 2606 OID 23701)
-- Dependencies: 1594 1594
-- Name: almacen_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY almacen
    ADD CONSTRAINT almacen_pkey PRIMARY KEY (codalm);

--
-- TOC entry 1964 (class 2606 OID 23703)
-- Dependencies: 1596 1596
-- Name: categoriamat_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY categoriamat
    ADD CONSTRAINT categoriamat_pkey PRIMARY KEY (codcatmat);
```

```

--
-- TOC entry 1966 (class 2606 OID 23705)
-- Dependencies: 1598 1598
-- Name: cliente_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY cliente
    ADD CONSTRAINT cliente_pkey PRIMARY KEY (codcli);

--
-- TOC entry 1968 (class 2606 OID 23707)
-- Dependencies: 1600 1600
-- Name: clienteempresa_codcli_key; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY clienteempresa
    ADD CONSTRAINT clienteempresa_codcli_key UNIQUE (codcli);

--
-- TOC entry 1970 (class 2606 OID 23709)
-- Dependencies: 1600 1600
-- Name: clienteempresa_codemp_key; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY clienteempresa
    ADD CONSTRAINT clienteempresa_codemp_key UNIQUE (codemp);

--
-- TOC entry 1972 (class 2606 OID 23711)
-- Dependencies: 1600 1600 1600
-- Name: clienteempresa_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY clienteempresa
    ADD CONSTRAINT clienteempresa_pkey PRIMARY KEY (codcli,
codemp);

--
-- TOC entry 1974 (class 2606 OID 23713)
-- Dependencies: 1601 1601

```

```
-- Name: clientepersona_codcli_key; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY clientepersona
    ADD CONSTRAINT clientepersona_codcli_key UNIQUE (codcli);
```

```
--
-- TOC entry 1976 (class 2606 OID 23715)
-- Dependencies: 1601 1601
-- Name: clientepersona_codper_key; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY clientepersona
    ADD CONSTRAINT clientepersona_codper_key UNIQUE (codper);
```

```
--
-- TOC entry 1978 (class 2606 OID 23717)
-- Dependencies: 1601 1601 1601
-- Name: clientepersona_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY clientepersona
    ADD CONSTRAINT clientepersona_pkey PRIMARY KEY (codper,
codcli);
```

```
--
-- TOC entry 1980 (class 2606 OID 23721)
-- Dependencies: 1602 1602
-- Name: compra_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--
```

```
ALTER TABLE ONLY compra
    ADD CONSTRAINT compra_pkey PRIMARY KEY (codcom);
```

```
--
-- TOC entry 1982 (class 2606 OID 23723)
-- Dependencies: 1604 1604
-- Name: datos_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--
```

```

ALTER TABLE ONLY datos
    ADD CONSTRAINT datos_pkey PRIMARY KEY (codusu);

--
-- TOC entry 1984 (class 2606 OID 23725)
-- Dependencies: 1606 1606 1606
-- Name: detallecompra_pkey; Type: CONSTRAINT; Schema: public;
-- Owner: postgres; Tablespace:
--

ALTER TABLE ONLY detallecompra
    ADD CONSTRAINT detallecompra_pkey PRIMARY KEY (codcom,
codmat);

--
-- TOC entry 1986 (class 2606 OID 24041)
-- Dependencies: 1607 1607 1607
-- Name: detalleproducto_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY detalleproducto
    ADD CONSTRAINT detalleproducto_pkey PRIMARY KEY (codprod,
codesp);

--
-- TOC entry 1988 (class 2606 OID 23729)
-- Dependencies: 1608 1608 1608
-- Name: detalleventa_pkey; Type: CONSTRAINT; Schema: public;
-- Owner: postgres; Tablespace:
--

ALTER TABLE ONLY detalleventa
    ADD CONSTRAINT detalleventa_pkey PRIMARY KEY (codven,
codprod);

--
-- TOC entry 1990 (class 2606 OID 25373)
-- Dependencies: 1609 1609
-- Name: empresa_nit_key; Type: CONSTRAINT; Schema: public;
-- Owner: postgres; Tablespace:
--

ALTER TABLE ONLY empresa
    ADD CONSTRAINT empresa_nit_key UNIQUE (nit);

```

```

--
-- TOC entry 1992 (class 2606 OID 23733)
-- Dependencies: 1609 1609
-- Name: empresa_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY empresa
    ADD CONSTRAINT empresa_pkey PRIMARY KEY (codemp);

--
-- TOC entry 2057 (class 2606 OID 24078)
-- Dependencies: 1653 1653
-- Name: especificacionprod_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY especificacionprod
    ADD CONSTRAINT especificacionprod_pkey PRIMARY KEY
(codesp);

--
-- TOC entry 1994 (class 2606 OID 23737)
-- Dependencies: 1611 1611
-- Name: fotosprod_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY fotosprod
    ADD CONSTRAINT fotosprod_pkey PRIMARY KEY (codfoto);

--
-- TOC entry 1996 (class 2606 OID 23739)
-- Dependencies: 1613 1613
-- Name: gramaje_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY gramaje
    ADD CONSTRAINT gramaje_pkey PRIMARY KEY (codgram);

--
-- TOC entry 1998 (class 2606 OID 23741)

```

```

-- Dependencies: 1615 1615
-- Name: gramajemat_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY gramajemat
    ADD CONSTRAINT gramajemat_pkey PRIMARY KEY (codgram);

--

-- TOC entry 2000 (class 2606 OID 23743)
-- Dependencies: 1616 1616
-- Name: maquina_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY maquina
    ADD CONSTRAINT maquina_pkey PRIMARY KEY (codmaq);

--

-- TOC entry 2002 (class 2606 OID 23745)
-- Dependencies: 1618 1618
-- Name: marca_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY marca
    ADD CONSTRAINT marca_pkey PRIMARY KEY (codmar);

--

-- TOC entry 2004 (class 2606 OID 23747)
-- Dependencies: 1620 1620
-- Name: material_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY material
    ADD CONSTRAINT material_pkey PRIMARY KEY (codmat);

--

-- TOC entry 2006 (class 2606 OID 23749)
-- Dependencies: 1622 1622
-- Name: medidamat_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

```

```

ALTER TABLE ONLY medidamat
    ADD CONSTRAINT medidamat_pkey PRIMARY KEY (codmedmat);

--
-- TOC entry 2008 (class 2606 OID 23751)
-- Dependencies: 1624 1624 1624
-- Name: medidamatmat_pkey; Type: CONSTRAINT; Schema: public;
-- Owner: postgres; Tablespace:
--

ALTER TABLE ONLY medidamatmat
    ADD CONSTRAINT medidamatmat_pkey PRIMARY KEY (codmedmat,
codmat);

--
-- TOC entry 2010 (class 2606 OID 23753)
-- Dependencies: 1625 1625
-- Name: nota_pkey; Type: CONSTRAINT; Schema: public; Owner:
-- postgres; Tablespace:
--

ALTER TABLE ONLY nota
    ADD CONSTRAINT nota_pkey PRIMARY KEY (codnot);

--
-- TOC entry 2059 (class 2606 OID 24066)
-- Dependencies: 1655 1655
-- Name: orden_pkey; Type: CONSTRAINT; Schema: public; Owner:
-- postgres; Tablespace:
--

ALTER TABLE ONLY orden
    ADD CONSTRAINT orden_pkey PRIMARY KEY (codord);

--
-- TOC entry 2012 (class 2606 OID 25383)
-- Dependencies: 1627 1627 1627
-- Name: persona_ci_key; Type: CONSTRAINT; Schema: public;
-- Owner: postgres; Tablespace:
--

ALTER TABLE ONLY persona
    ADD CONSTRAINT persona_ci_key UNIQUE (ci, nit);

```



```

--
-- TOC entry 2014 (class 2606 OID 23757)
-- Dependencies: 1627 1627
-- Name: persona_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY persona
    ADD CONSTRAINT persona_pkey PRIMARY KEY (codper);

--
-- TOC entry 2016 (class 2606 OID 23759)
-- Dependencies: 1629 1629
-- Name: proceso_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY proceso
    ADD CONSTRAINT proceso_pkey PRIMARY KEY (codproc);

--
-- TOC entry 2018 (class 2606 OID 23761)
-- Dependencies: 1631 1631
-- Name: producto_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY producto
    ADD CONSTRAINT producto_pkey PRIMARY KEY (codprod);

--
-- TOC entry 2020 (class 2606 OID 23763)
-- Dependencies: 1633 1633
-- Name: proveedor_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY proveedor
    ADD CONSTRAINT proveedor_pkey PRIMARY KEY (codprov);

--
-- TOC entry 2022 (class 2606 OID 23765)
-- Dependencies: 1635 1635
-- Name: proveedorempresa_codemp_key; Type: CONSTRAINT;
Schema: public; Owner: postgres; Tablespace:

```

--

```
ALTER TABLE ONLY proveedorempresa
    ADD CONSTRAINT proveedorempresa_codemp_key UNIQUE
(codemp);
```

--

```
-- TOC entry 2024 (class 2606 OID 23767)
-- Dependencies: 1635 1635
-- Name: proveedorempresa_codprov_key; Type: CONSTRAINT;
Schema: public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY proveedorempresa
    ADD CONSTRAINT proveedorempresa_codprov_key UNIQUE
(codprov);
```

--

```
-- TOC entry 2026 (class 2606 OID 23769)
-- Dependencies: 1635 1635 1635
-- Name: proveedorempresa_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY proveedorempresa
    ADD CONSTRAINT proveedorempresa_pkey PRIMARY KEY (codprov,
codemp);
```

--

```
-- TOC entry 2028 (class 2606 OID 23771)
-- Dependencies: 1636 1636
-- Name: proveedorpersona_codper_key; Type: CONSTRAINT;
Schema: public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY proveedorpersona
    ADD CONSTRAINT proveedorpersona_codper_key UNIQUE
(codper);
```

--

```
-- TOC entry 2030 (class 2606 OID 23773)
-- Dependencies: 1636 1636
-- Name: proveedorpersona_codprov_key; Type: CONSTRAINT;
Schema: public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY proveedorpersona
    ADD CONSTRAINT proveedorpersona_codprov_key UNIQUE
(codprov);
```

```
--
-- TOC entry 2032 (class 2606 OID 23775)
-- Dependencies: 1636 1636 1636
-- Name: proveedorpersona_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY proveedorpersona
    ADD CONSTRAINT proveedorpersona_pkey PRIMARY KEY (codprov,
codper);
```

```
--
-- TOC entry 2034 (class 2606 OID 23777)
-- Dependencies: 1637 1637
-- Name: rol_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--
```

```
ALTER TABLE ONLY rol
    ADD CONSTRAINT rol_pkey PRIMARY KEY (codrol);
```

```
--
-- TOC entry 2036 (class 2606 OID 23779)
-- Dependencies: 1639 1639 1639
-- Name: rolproc_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY rolproc
    ADD CONSTRAINT rolproc_pkey PRIMARY KEY (codrol, codproc);
```

```
--
-- TOC entry 2038 (class 2606 OID 23781)
-- Dependencies: 1640 1640
-- Name: tipospecificacion_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY tipospecificacion
```

```

        ADD CONSTRAINT tipoespecificacion_pkey PRIMARY KEY
(codtipoesp);

```

```

--
-- TOC entry 2040 (class 2606 OID 23783)
-- Dependencies: 1642 1642
-- Name: tipomat_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

```

```

ALTER TABLE ONLY tipomat
    ADD CONSTRAINT tipomat_pkey PRIMARY KEY (codtipomat);

```

```

--
-- TOC entry 2042 (class 2606 OID 23785)
-- Dependencies: 1644 1644 1644
-- Name: tipomatmat_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

```

```

ALTER TABLE ONLY tipomatmat
    ADD CONSTRAINT tipomatmat_pkey PRIMARY KEY (codtipomat,
codmat);

```

```

--
-- TOC entry 2044 (class 2606 OID 23787)
-- Dependencies: 1645 1645
-- Name: tipoproducto_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

```

```

ALTER TABLE ONLY tipoproducto
    ADD CONSTRAINT tipoproducto_pkey PRIMARY KEY (codtipprod);

```

```

--
-- TOC entry 2047 (class 2606 OID 24654)
-- Dependencies: 1647 1647
-- Name: usuario_codper_key; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

```

```

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_codper_key UNIQUE (codper);

```

```

--
-- TOC entry 2049 (class 2606 OID 23789)
-- Dependencies: 1647 1647
-- Name: usuario_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_pkey PRIMARY KEY (codusu);

--
-- TOC entry 2053 (class 2606 OID 23791)
-- Dependencies: 1649 1649 1649
-- Name: usurol_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY usurol
    ADD CONSTRAINT usurol_pkey PRIMARY KEY (codusu, codrol);

--
-- TOC entry 2055 (class 2606 OID 23793)
-- Dependencies: 1650 1650
-- Name: venta_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY venta
    ADD CONSTRAINT venta_pkey PRIMARY KEY (codven);

--
-- TOC entry 2045 (class 1259 OID 23794)
-- Dependencies: 1647
-- Name: idx_usuario1; Type: INDEX; Schema: public; Owner:
postgres; Tablespace:
--

CREATE UNIQUE INDEX idx_usuario1 ON usuario USING btree
(codusu);

--
-- TOC entry 2050 (class 1259 OID 23795)
-- Dependencies: 1649
-- Name: idx_usurol1; Type: INDEX; Schema: public; Owner:
postgres; Tablespace:

```

```
--

CREATE INDEX idx_usurol1 ON usurol USING btree (codusu);

--
-- TOC entry 2051 (class 1259 OID 23796)
-- Dependencies: 1649
-- Name: idx_usurol2; Type: INDEX; Schema: public; Owner:
postgres; Tablespace:
--

CREATE INDEX idx_usurol2 ON usurol USING btree (codrol);

--
-- TOC entry 2060 (class 2606 OID 23797)
-- Dependencies: 1594 1620 2003
-- Name: almacen_codmat_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY almacen
    ADD CONSTRAINT almacen_codmat_fkey FOREIGN KEY (codmat)
REFERENCES material(codmat);

--
-- TOC entry 2061 (class 2606 OID 23802)
-- Dependencies: 1965 1598 1600
-- Name: clienteempresa_codcli_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY clienteempresa
    ADD CONSTRAINT clienteempresa_codcli_fkey FOREIGN KEY
(codcli) REFERENCES cliente(codcli);

--
-- TOC entry 2062 (class 2606 OID 23807)
-- Dependencies: 1600 1991 1609
-- Name: clienteempresa_codemp_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY clienteempresa
    ADD CONSTRAINT clienteempresa_codemp_fkey FOREIGN KEY
(codemp) REFERENCES empresa(codemp);
```

```

--
-- TOC entry 2063 (class 2606 OID 23812)
-- Dependencies: 1965 1598 1601
-- Name: clientepersona_codcli_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY clientepersona
    ADD CONSTRAINT clientepersona_codcli_fkey FOREIGN KEY
(codcli) REFERENCES cliente(codcli);

--
-- TOC entry 2064 (class 2606 OID 23817)
-- Dependencies: 1601 1627 2013
-- Name: clientepersona_codper_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY clientepersona
    ADD CONSTRAINT clientepersona_codper_fkey FOREIGN KEY
(codper) REFERENCES persona(codper);

--
-- TOC entry 2065 (class 2606 OID 23822)
-- Dependencies: 1633 1602 2019
-- Name: compra_codprov_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY compra
    ADD CONSTRAINT compra_codprov_fkey FOREIGN KEY (codprov)
REFERENCES proveedor(codprov);

--
-- TOC entry 2066 (class 2606 OID 23827)
-- Dependencies: 2048 1647 1602
-- Name: compra_codusu_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY compra
    ADD CONSTRAINT compra_codusu_fkey FOREIGN KEY (codusu)
REFERENCES usuario(codusu);

--

```

```

-- TOC entry 2067 (class 2606 OID 23832)
-- Dependencies: 1647 1604 2048
-- Name: datos_codusu_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY datos
    ADD CONSTRAINT datos_codusu_fkey FOREIGN KEY (codusu)
REFERENCES usuario(codusu);

--

-- TOC entry 2068 (class 2606 OID 23837)
-- Dependencies: 1602 1979 1606
-- Name: detallecompra_codcom_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY detallecompra
    ADD CONSTRAINT detallecompra_codcom_fkey FOREIGN KEY
(codcom) REFERENCES compra(codcom);

--

-- TOC entry 2069 (class 2606 OID 23842)
-- Dependencies: 1606 1620 2003
-- Name: detallecompra_codmat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY detallecompra
    ADD CONSTRAINT detallecompra_codmat_fkey FOREIGN KEY
(codmat) REFERENCES material(codmat);

--

-- TOC entry 2070 (class 2606 OID 23847)
-- Dependencies: 2003 1607 1620
-- Name: detalleproducto_codmat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY detalleproducto
    ADD CONSTRAINT detalleproducto_codmat_fkey FOREIGN KEY
(codmat) REFERENCES material(codmat);

--

-- TOC entry 2071 (class 2606 OID 23852)

```



```
-- Dependencies: 1631 1607 2017
-- Name: detalleproducto_codprod_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY detalleproducto
    ADD CONSTRAINT detalleproducto_codprod_fkey FOREIGN KEY
(codprod) REFERENCES producto(codprod);
```

```
--
-- TOC entry 2072 (class 2606 OID 23857)
-- Dependencies: 1631 1608 2017
-- Name: detalleventa_codprod_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY detalleventa
    ADD CONSTRAINT detalleventa_codprod_fkey FOREIGN KEY
(codprod) REFERENCES producto(codprod);
```

```
--
-- TOC entry 2073 (class 2606 OID 23862)
-- Dependencies: 2054 1608 1650
-- Name: detalleventa_codven_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY detalleventa
    ADD CONSTRAINT detalleventa_codven_fkey FOREIGN KEY
(codven) REFERENCES venta(codven);
```

```
--
-- TOC entry 2099 (class 2606 OID 24030)
-- Dependencies: 1999 1616 1653
-- Name: especificacionprod_codmaq_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY especificacionprod
    ADD CONSTRAINT especificacionprod_codmaq_fkey FOREIGN KEY
(codmaq) REFERENCES maquina(codmaq);
```

```
--
-- TOC entry 2102 (class 2606 OID 24054)
-- Dependencies: 1653 1620 2003
```

```

-- Name: especificacionprod_codmat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY especificacionprod
    ADD CONSTRAINT especificacionprod_codmat_fkey FOREIGN KEY
(codmat) REFERENCES material(codmat);

--
-- TOC entry 2101 (class 2606 OID 24049)
-- Dependencies: 2017 1653 1631
-- Name: especificacionprod_codprod_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY especificacionprod
    ADD CONSTRAINT especificacionprod_codprod_fkey FOREIGN KEY
(codprod) REFERENCES producto(codprod);

--
-- TOC entry 2100 (class 2606 OID 24035)
-- Dependencies: 2037 1640 1653
-- Name: especificacionprod_codtipoesp_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY especificacionprod
    ADD CONSTRAINT especificacionprod_codtipoesp_fkey FOREIGN
KEY (codtipoesp) REFERENCES tipospecificacion(codtipoesp);

--
-- TOC entry 2074 (class 2606 OID 23882)
-- Dependencies: 2043 1645 1611
-- Name: fotosprod_codtipprod_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY fotosprod
    ADD CONSTRAINT fotosprod_codtipprod_fkey FOREIGN KEY
(codtipprod) REFERENCES tipoproducto(codtipprod);

--
-- TOC entry 2075 (class 2606 OID 23887)
-- Dependencies: 1613 1615 1995

```

```
-- Name: gramajemat_codgram_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY gramajemat
    ADD CONSTRAINT gramajemat_codgram_fkey FOREIGN KEY
(codgram) REFERENCES gramaje(codgram);
```

```
--
-- TOC entry 2076 (class 2606 OID 23892)
-- Dependencies: 1615 1620 2003
-- Name: gramajemat_codmat_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY gramajemat
    ADD CONSTRAINT gramajemat_codmat_fkey FOREIGN KEY (codmat)
REFERENCES material(codmat);
```

```
--
-- TOC entry 2077 (class 2606 OID 23897)
-- Dependencies: 1995 1620 1613
-- Name: material_codgram_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY material
    ADD CONSTRAINT material_codgram_fkey FOREIGN KEY (codgram)
REFERENCES gramaje(codgram);
```

```
--
-- TOC entry 2078 (class 2606 OID 23902)
-- Dependencies: 2001 1618 1620
-- Name: material_codmar_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY material
    ADD CONSTRAINT material_codmar_fkey FOREIGN KEY (codmar)
REFERENCES marca(codmar);
```

```
--
-- TOC entry 2079 (class 2606 OID 23907)
-- Dependencies: 1622 1620 2005
```

```
-- Name: material_codmedmat_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY material
    ADD CONSTRAINT material_codmedmat_fkey FOREIGN KEY
(codmedmat) REFERENCES medidamat(codmedmat);
```

```
--
-- TOC entry 2080 (class 2606 OID 23912)
-- Dependencies: 1620 1642 2039
-- Name: material_codtipomat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY material
    ADD CONSTRAINT material_codtipomat_fkey FOREIGN KEY
(codtipomat) REFERENCES tipomat(codtipomat);
```

```
--
-- TOC entry 2081 (class 2606 OID 23917)
-- Dependencies: 1620 2003 1624
-- Name: medidamatmat_codmat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY medidamatmat
    ADD CONSTRAINT medidamatmat_codmat_fkey FOREIGN KEY
(codmat) REFERENCES material(codmat);
```

```
--
-- TOC entry 2082 (class 2606 OID 23922)
-- Dependencies: 1624 1622 2005
-- Name: medidamatmat_codmedmat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY medidamatmat
    ADD CONSTRAINT medidamatmat_codmedmat_fkey FOREIGN KEY
(codmedmat) REFERENCES medidamat(codmedmat);
```

```
--
-- TOC entry 2083 (class 2606 OID 23927)
-- Dependencies: 2048 1625 1647
```

```
-- Name: nota_codusu_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY nota
    ADD CONSTRAINT nota_codusu_fkey FOREIGN KEY (codusu)
REFERENCES usuario(codusu);
```

```
--
-- TOC entry 2103 (class 2606 OID 24067)
-- Dependencies: 1598 1655 1965
-- Name: orden_codcli_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY orden
    ADD CONSTRAINT orden_codcli_fkey FOREIGN KEY (codcli)
REFERENCES cliente(codcli);
```

```
--
-- TOC entry 2104 (class 2606 OID 24072)
-- Dependencies: 1631 2017 1655
-- Name: orden_codprod_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY orden
    ADD CONSTRAINT orden_codprod_fkey FOREIGN KEY (codprod)
REFERENCES producto(codprod);
```

```
--
-- TOC entry 2084 (class 2606 OID 23932)
-- Dependencies: 1631 2043 1645
-- Name: producto_codtipprod_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY producto
    ADD CONSTRAINT producto_codtipprod_fkey FOREIGN KEY
(codtipprod) REFERENCES tipoproducto(codtipprod);
```

```
--
-- TOC entry 2085 (class 2606 OID 23937)
-- Dependencies: 1609 1635 1991
```

```

-- Name: proveedorempresa_codemp_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY proveedorempresa
    ADD CONSTRAINT proveedorempresa_codemp_fkey FOREIGN KEY
(codemp) REFERENCES empresa(codemp);

--

-- TOC entry 2086 (class 2606 OID 23942)
-- Dependencies: 2019 1633 1635
-- Name: proveedorempresa_codprov_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY proveedorempresa
    ADD CONSTRAINT proveedorempresa_codprov_fkey FOREIGN KEY
(codprov) REFERENCES proveedor(codprov);

--

-- TOC entry 2087 (class 2606 OID 23947)
-- Dependencies: 2013 1627 1636
-- Name: proveedorpersona_codper_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY proveedorpersona
    ADD CONSTRAINT proveedorpersona_codper_fkey FOREIGN KEY
(codper) REFERENCES persona(codper);

--

-- TOC entry 2088 (class 2606 OID 23952)
-- Dependencies: 2019 1633 1636
-- Name: proveedorpersona_codprov_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY proveedorpersona
    ADD CONSTRAINT proveedorpersona_codprov_fkey FOREIGN KEY
(codprov) REFERENCES proveedor(codprov);

--

-- TOC entry 2089 (class 2606 OID 23957)
-- Dependencies: 1629 2015 1639

```

```
-- Name: rolproc_codproc_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY rolproc
    ADD CONSTRAINT rolproc_codproc_fkey FOREIGN KEY (codproc)
REFERENCES proceso(codproc);
```

```
--
-- TOC entry 2090 (class 2606 OID 23962)
-- Dependencies: 1639 2033 1637
-- Name: rolproc_codrol_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY rolproc
    ADD CONSTRAINT rolproc_codrol_fkey FOREIGN KEY (codrol)
REFERENCES rol(codrol);
```

```
--
-- TOC entry 2091 (class 2606 OID 23967)
-- Dependencies: 1596 1642 1963
-- Name: tipomat_codcatmat_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY tipomat
    ADD CONSTRAINT tipomat_codcatmat_fkey FOREIGN KEY
(codcatmat) REFERENCES categoriamat(codcatmat);
```

```
--
-- TOC entry 2092 (class 2606 OID 23972)
-- Dependencies: 1644 1620 2003
-- Name: tipomatmat_codmat_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY tipomatmat
    ADD CONSTRAINT tipomatmat_codmat_fkey FOREIGN KEY (codmat)
REFERENCES material(codmat);
```

```
--
-- TOC entry 2093 (class 2606 OID 23977)
-- Dependencies: 1644 1642 2039
```

```

-- Name: tipomatmat_codtipomat_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY tipomatmat
    ADD CONSTRAINT tipomatmat_codtipomat_fkey FOREIGN KEY
(codtipomat) REFERENCES tipomat(codtipomat);

--
-- TOC entry 2094 (class 2606 OID 23982)
-- Dependencies: 1647 2013 1627
-- Name: usuario_codper_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_codper_fkey FOREIGN KEY (codper)
REFERENCES persona(codper);

--
-- TOC entry 2095 (class 2606 OID 23987)
-- Dependencies: 2033 1649 1637
-- Name: usurol_codrol_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY usurol
    ADD CONSTRAINT usurol_codrol_fkey FOREIGN KEY (codrol)
REFERENCES rol(codrol);

--
-- TOC entry 2096 (class 2606 OID 23992)
-- Dependencies: 2048 1649 1647
-- Name: usurol_codusu_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY usurol
    ADD CONSTRAINT usurol_codusu_fkey FOREIGN KEY (codusu)
REFERENCES usuario(codusu);

--
-- TOC entry 2097 (class 2606 OID 23997)
-- Dependencies: 1965 1598 1650

```



```

-- Name: venta_codcli_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY venta
    ADD CONSTRAINT venta_codcli_fkey FOREIGN KEY (codcli)
REFERENCES cliente(codcli);

--

-- TOC entry 2098 (class 2606 OID 24002)
-- Dependencies: 1650 1647 2048
-- Name: venta_codusu_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY venta
    ADD CONSTRAINT venta_codusu_fkey FOREIGN KEY (codusu)
REFERENCES usuario(codusu);

--

-- TOC entry 2109 (class 0 OID 0)
-- Dependencies: 6
-- Name: public; Type: ACL; Schema: -; Owner: postgres
--

REVOKE ALL ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON SCHEMA public FROM postgres;
GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO PUBLIC;

--

--
-- PostgreSQL database dump complete
--

```

II.1.13. MODELADO DE IMPLEMENTACIÓN

II.1.13.1. Introducción

El modelo de Implementación describe la relación que existe entre los paquetes y clases del modelo de diseño y los componentes físicos.

- El diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes.
- El diagrama de despliegue modela el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

II.1.13.2. Diagrama de Componentes

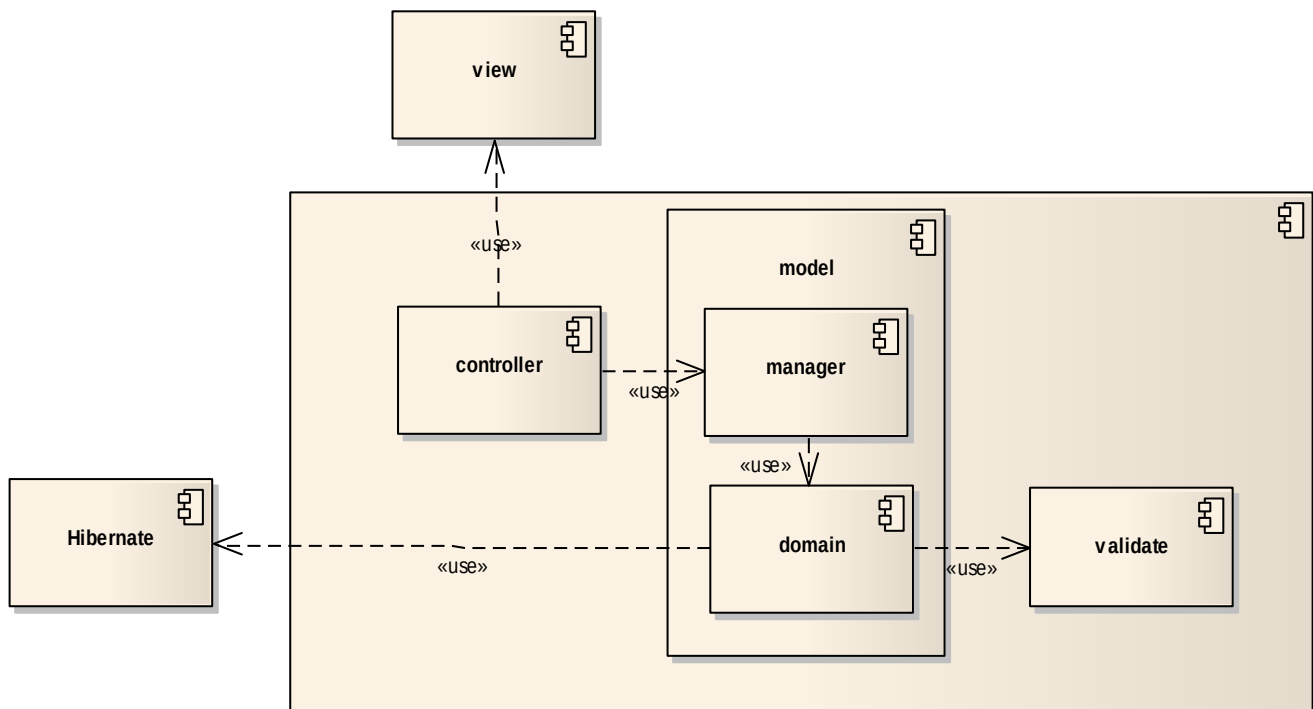


Figura 104. Diagrama de Componentes

II.1.13.3. Diagrama de despliegue

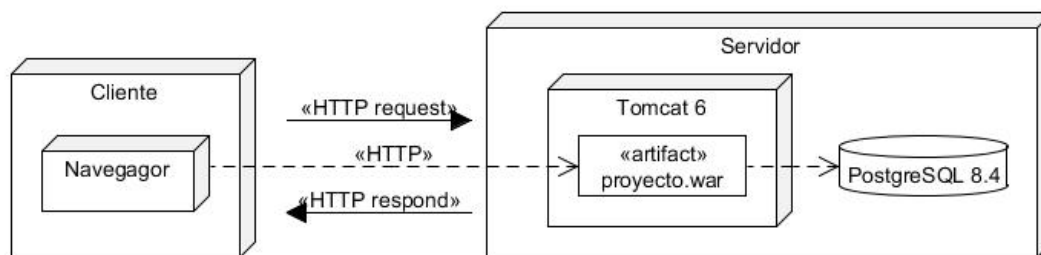


Figura 105. Diagrama de Despliegue

II.1.14. PROTOTIPO DE INTERFAZ DE USUARIO

II.1.14.1. Introducción

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

II.1.14.2. Propósito

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

II.1.14.3. Principio de diseño

Los principios de diseño que en general se pueden observar son:

Uniformidad.-Siempre que sea posible, la interfaz debe ser uniforme en el sentido de que las operaciones comparables se activen de la misma forma.

Este principio se puede observar al añadir, modificar o cambiar de estado de varias pantallas (Administrar Usuarios, Administrar Roles, Administrar Notas, etc.), algunas pantallas tienen ciertas diferencias que dada la posibilidad, fueron minimizadas.

Mínima sorpresa.-El comportamiento del sistema no debe provocar sorpresa a los usuarios.

La utilización de este principio implica que el sistema se comporte de manera similar ante situaciones similares, esto para evitar que el usuario se sorprenda y confunda ocasionándole posibles malestares.

II.1.14.4. Alcance

Mostrar los Prototipos de Pantallas, sujeto a modificaciones posteriores.

II.1.14.5. Prototipo de pantallas

Pantalla de Inicio

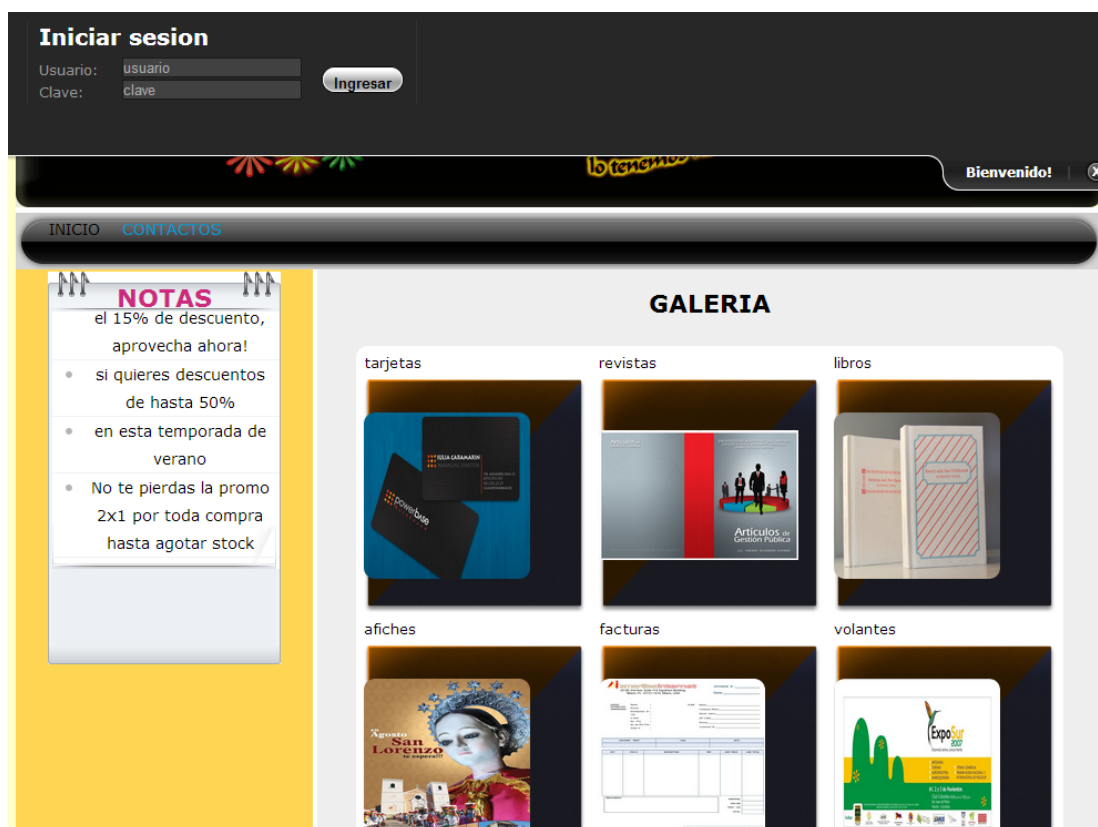


Figura 106. Pantalla Inicio. Iniciar Sesión

Pantalla Menú

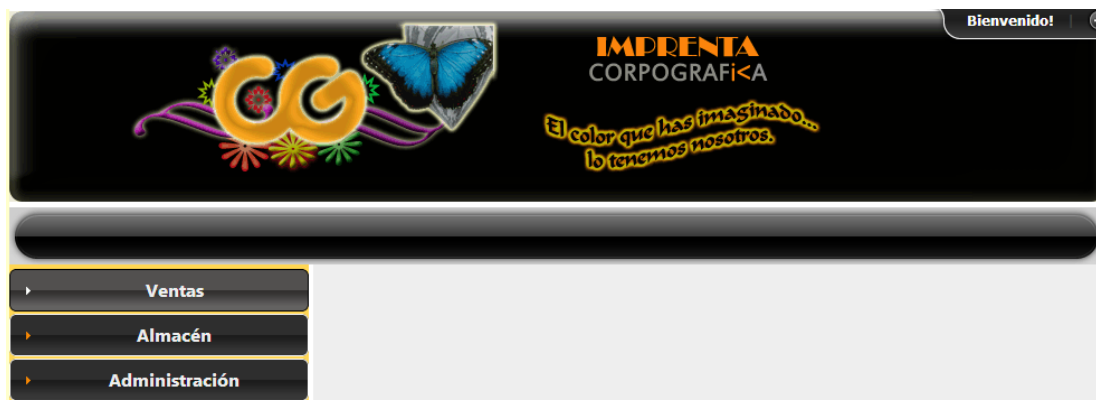


Figura 107. Pantalla Menú

Pantalla Administrar Roles



Figura 108. Pantalla Administrar Roles

Pantalla Nuevo Rol

NUEVO ROL

FORMULARIO	
*Código	0
*Nombre	
Descripción	
Estado	☐ Activo ☐ Inactivo

Procesos

13 Administrar categoria - materiales	☐
3 Administrar clientes	☐
5 Administrar compras	☐
22 Administrar cotizaciones	☐
15 Administrar foto - productos	☐
16 Administrar gramajes	☐
9 Administrar marca	☐
12 Administrar materiales	☐
18 Administrar medida - materiales	☐

Guardar Cancelar

Figura 109. Pantalla Nuevo Rol

Pantalla Modificar Rol

MODIFICAR ROL

FORMULARIO	
*Código	1
*Nombre	administrador
Descripción	rrrrr
Estado	☒ Activo ☐ Inactivo

Procesos

13 Administrar categoria - materiales	☒
3 Administrar clientes	☒
5 Administrar compras	☐
22 Administrar cotizaciones	☐
15 Administrar foto - productos	☒
16 Administrar gramajes	☒
9 Administrar marca	☒
12 Administrar materiales	☒
18 Administrar medida - materiales	☒

Guardar Cancelar

Figura 110. Pantalla Modificar Rol

Pantalla Administrar Procesos

ADMINISTRAR PROCESOS

BuscarFilas

5 ▼

☒ Activos☐ Inactivos☐ Todos

Nuevo Proceso

CODIGO	NOMBRE	DESCRIPCION	MODIFICAR	ESTADO
5	Administrar compras			
3	Administrar clientes			
15	Administrar foto - productos			
14	Administrar notas			
8	Administrar procesos			

12345

Figura 111. Pantalla Administrar Procesos

Pantalla Nuevo Proceso

NUEVO PROCESO

FORMULARIO	
* CÃ³digo	0
* Nombre	
* DescripciÃ³n	
* Link	
* Estado	☐ Activo ☐ Inactivo

Figura 112. Pantalla Nuevo Proceso

Pantalla Modificar Proceso

MODIFICAR PROCESO

FORMULARIO	
* CÃ³digo	7
* Nombre	Administrar roles
DescripciÃ³n	
* Link	AdmRolesView.html
Estado	☒ Activo ☐ Inactivo

Figura 113. Pantalla Modificar Proceso

Pantalla Administrar Usuarios

ADMINISTRAR USUARIOS

Buscar
Filas

5
▼

☒ Activos
 ☐ Inactivos
 ☐ Todos

Nuevo Usuario

CODIGO	CI	NOMBRE	VER	MODIFICAR	ESTADO
1	1111111	Javier Miranda Gonzales			
2	2222222	Ana Vargas Vargas			
3	3333333	Melisa Jimenez Jimenez			
35	7890	Juan Orozco			
25	333513451	Eduardo Santamarina Mealla			

1

Figura 114. Pantalla Administrar Usuarios

Pantalla Nuevo Usuario

NUEVO USUARIO

FORMULARIO	
* CÃ³digo	0
*Nombre	
*Apellido Paterno	
*Apellido Materno	
CI	
NIT	
Fax	
TelÃ©fono	
Direccion	
*Login	
*Clave	
*Repetir Clave	sssr
Estado	☐ Activo ☐ Inactivo

Roles

1 administrador

☐

3 gerente

☐

Guardar

Cancelar

Figura 115. Pantalla Nuevo Usuario

Pantalla Modificar Usuario

MODIFICAR USUARIO

FORMULARIO	
* CÃ³digo	1
*Nombre	Javier
*Apellido Paterno	Miranda
*Apellido Materno	Gonzales
CI	71589658
NIT	1234567890
Fax	4662845
TelÃ©fono	4662845
Direccion	calle Aroma
*Login	***
*Clave	***
*Repetir Clave	***
Estado	☒ Activo ☐ Inactivo

Roles

1 administrador ☒

3 gerente ☒

Guardar

Cancelar

Figura 116. Pantalla Modificar Usuario

Pantalla Administrar Reportes

The screenshot displays a web interface for managing reports. It features two distinct sections, each enclosed in a rounded rectangle with a yellow border. The top section is titled 'Reporte Ventas' and the bottom section is titled 'Reporte Compras'. Both sections contain the instruction 'Ingrese el intervalo de fechas para realizar la consulta'. Below this instruction, each section has two text input fields labeled 'Desde:' and 'Hasta:', both with a placeholder 'dd/MM/yyyy'. To the right of these fields is a yellow button labeled 'Generar Reporte'.

Reporte Ventas

Ingrese el intervalo de fechas para realizar la consulta

Desde: Hasta: Generar Reporte

Reporte Compras

Ingrese el intervalo de fechas para realizar la consulta

Desde: Hasta: Generar Reporte

Figura 117. Pantalla Administrar Reportes

I.1.1. Medios de Verificación

Ver Anexos Componente I

II.2.COMPONENTE II “Capacitación del sistema web implementado”.

Introducción

El objetivo de este componente es capacitar a los usuarios en el uso del sistema, según el nivel de los mismos empleando métodos y medios de enseñanza – aprendizaje adecuados.

El propósito del proyecto es: la capacitación en el uso del sistema informático al personal afectado por el proyecto, se convierte en un componente fundamental para el logro del mismo.

El componente capacitación, se encamina hacia el siguiente objetivo: usar adecuadamente el sistema informático, por el personal de la imprenta, explotando las fortalezas del mismo.

La Capacitación será personalizada dada la corta duración de la misma.

Propósito

El propósito del componente es que al finalizar el curso de capacitación el personal de la empresa sea apto para manejar el sistema Web sin dificultades.

Objetivo General

Socializar el sistema computarizado orientado a la web y de la TICs con los miembros que conforman la empresa.

Material a Impartir

Manual de Usuario

Metodología

La capacitación se la realizara mediante cursos dictados por el responsable del proyecto en los ambientes de la imprenta

Para la capacitación se emplearán manuales de usuarios que contiene toda la información detallada sobre el manejo del sitio.

Proceso de enseñanza aprendizaje y metodologías de capacitación

El proceso de enseñanza

Proceso enseñanza-aprendizaje

La base fundamental de todo proceso de enseñanza-aprendizaje se halla representada por un reflejo condicionado, es decir, por la relación asociada que existe entre la respuesta y el estímulo que la provoca. El sujeto que enseña es el encargado de provocar dicho estímulo, con el fin de obtener la respuesta en el individuo que aprende. Esta teoría da lugar a la formulación del principio de la motivación, principio básico de todo proceso de enseñanza que consiste en estimular a un sujeto para que éste ponga en actividad sus facultades, en el estudio de la motivación comprende el de los factores orgánicos de toda conducta, así como el de las condiciones que lo determinan.

Concepto de Enseñanza

Es el proceso mediante el cual se comunican o transmiten conocimientos especiales o generales sobre una materia. Este concepto es más restringido que el de educación, ya que ésta tiene por objeto la formación integral de la persona humana, mientras que la enseñanza se limita a transmitir, por medios diversos, determinados conocimientos. En este sentido la educación comprende la enseñanza propiamente dicha.

Concepto de Aprendizaje

El aprendizaje es un proceso que lleva a cabo el sujeto que aprende cuando interactúa con el objeto y lo relaciona con sus experiencias previas, aprovechando su capacidad de conocer para reestructurar sus esquemas mentales, enriqueciéndolos con la incorporación de un nuevo material que pasa a formar parte del sujeto que conoce.

El objeto es aprendido de modo diferente por cada sujeto, porque las experiencias y las capacidades de cada individuo presentan características únicas.

El aprendizaje no se agota en el proceso mental, pues abarca también la adquisición de destrezas, hábitos y habilidades, así como actitudes y valoraciones que acompañan el proceso y que ocurren en los tres ámbitos: el personal, el educativo formal y el social. El personal abarca el lenguaje, la reflexión y el pensamiento, que hacen del individuo un ser distinto a los demás.

Métodos de Enseñanza-Aprendizaje

Existen tres métodos

Método quiere decir “camino para llegar al fin”. Conducir el pensamiento o las acciones para alcanzar un fin, existen varios métodos aplicados a la educación:

☐ **Métodos de Investigación**

Son los que buscan acrecentar o profundizar nuestros conocimientos.

☐ **Métodos de Organización**

Destinados únicamente a establecer normas de disciplina para la conducta, a fin de ejecutar bien una tarea.

☐ **Métodos de Transmisión**

Transmiten conocimientos, actitudes o ideales. Son los intermediarios entre el profesor y el alumno.

Técnicas de enseñanza

Hay muchas técnicas para hacer llegar nuestro conocimiento y lograr un aprendizaje apropiado:

☐ **Técnica expositiva**

Consiste en la exposición oral, por parte del profesor; ésta debe estimular la participación del alumno en los trabajos de la clase, requiere una buena motivación para atraer la atención de los educandos. Esta técnica favorece el desenvolvimiento del autodomínio, y el lenguaje.

☐ **Técnica del dictado**

Consiste en que el profesor hable pausadamente en tanto los alumnos van tomando nota de lo que él dice.

Esto constituye una marcada pérdida de tiempo, ya que mientras el alumno escribe no puede reflexionar sobre lo que registra en sus notas.

☐ **Técnica biográfica**

Consiste en exponer los hechos o problemas a través del relato de las vidas que participan en ellos o que contribuyen para su estudio. Es más común en la historia, filosofía y la literatura.

☐ **Técnica de la argumentación**

Forma de interrogatorio destinada a comprobar lo que el alumno debería saber. Requiere fundamentalmente de la participación del alumno.

☐ **Técnica del diálogo**

El gran objetivo del diálogo es el de orientar al alumno para que reflexione, piense y se convenza que puede investigar valiéndose del razonamiento.

□ **Técnica de la discusión**

Exige el máximo de participación de los alumnos en la elaboración de conceptos y en la elaboración misma de la clase.

Consiste en la discusión de un tema, por parte de los alumnos, bajo la dirección del profesor y requiere preparación anticipada.

□ **Técnica de Experiencia Estructurada**

Es una técnica en el cual los participantes realizan una serie de actividades previamente diseñadas, cuyo propósito es destacar los principales elementos tema o aspecto del programa. Es importante destacar que hay una gran confusión entre la experiencia estructurada y las llamadas “Dinámicas de grupo”, conviene aclarar que la dinámica grupal existe en todo momento como consecuencia del comportamiento de las personas y de su interacción en el grupo, con independencia de la técnica que se emplee.

Procedimientos

El instructor diseña o selecciona la experiencia apropiada para enfatizar el tema.

El instructor prepara los materiales o instrumentos necesarios para la experiencia.

El instructor explica al grupo la mecánica de la experiencia estructurada.

El instructor conduce al grupo a lo largo de la experiencia.

Al finalizar la experiencia, solicita al grupo los comentarios y reflexiones sobre el tema.

El grupo destaca lo aprendido en la experiencia.

El instructor apoya el aprendizaje del grupo con la exposición de alguna teoría relacionada con la experiencia.

Aplicaciones

Para destacar el valor de la experiencia en el aprendizaje

Para facilitar la comprensión de los temas polémicos a partir de la vivencia de los participantes.

Para demostrar que el aprendizaje es agradable.

Para facilitar la manifestación y comprensión de emociones y sentimientos, en una estructura que proteja a las personas.

La aplicación de esta técnica se ha desvirtuado a utilizarla sin propósitos claros, utilizando como un simple juego en el cual ni la conducción ni la reflexión son adecuados.

Proceso de capacitación

La capacitación es un proceso de corta duración mediante la cual se ofrece un espacio de discusión y aprendizaje en la que los participantes podrán reflexionar sobre sus conocimientos y experiencia, sistematizar y organizar las herramientas que han aplicado.

III. Conclusiones y Recomendaciones

III.1. Conclusiones

Una vez finalizado el proyecto, se cubrió el propósito del mejoramiento de la gestión de ventas de la imprenta CORPOGRÁFIKA.

- Se desarrolló un sistema informático con plataforma web para la gestión de ventas que permite realizar los procesos requeridos por el
- En la capacitación se han utilizado los manuales y demostraciones del sistema la misma que hicieron más fácil el aprendizaje del usuario.
- A pesar de que la metodología utilizada para el desarrollo del software es bastante conocida y muy completa, pude observar que es muy pesada realizarla de forma individual y sin un completo dominio de la misma, pero tiene la ventaja de que permite la documentación del proyecto, la cual puede ser utilizada para futuras modificaciones de los componentes del proyecto.
- Se debe realizar prever los riesgos y supuestos de forma más explícitas y tener planes de apoyo para evitar retrasos en el tiempo de desarrollo del proyecto.
- La aplicación de las Tecnologías de la Información es indispensable para la optimización de los procesos en cualquier organización.

III.2. Recomendaciones

Para la implementación y sostenibilidad del proyecto es necesario contar con un sistema de red bien estructurado con los equipos clientes necesarios y un servidor.

La metodología (RUP) utilizada en el desarrollo del proyecto es bastante conocida y utilizada en la actualidad, existe gran cantidad de información en español de la misma, y es una metodología muy completa en el desarrollo de un proyecto, pero así también puede resultar bastante pesada para un solo individuo y aún más si no se tiene dominio de la misma, por lo tanto recomendaría el uso de otra metodología más ligera (XP, FDD) para futuros proyectos.

La parte de modelados de la documentación del presente proyecto, fue desarrollada con el uso de herramientas CASE, las ventajas del uso de estas herramientas, son que permiten realizar los diagramas del proyecto de forma más rápida, además que permiten facilidad a la hora de modificarlos, sin embargo, como desventaja, hacer uso de estas herramientas supone un previo aprendizaje de las mismas, y esto implica inversión de tiempo.

Para la conversión del sistema manual al nuevo sistema se puede recomendar los siguientes pasos:

- Instalación y configuración de una red, ya sea cableada o wireless, con los equipos clientes necesarios y un servidor.
- Instalación de los programas necesarios en el servidor como en los equipos clientes.
- Creación de la Base de Datos.
- Carga de la Base de Datos inicial.
- Realizar pruebas al sistema ingresando datos inválidos y utilizando el sistema bajo diversas circunstancias, como la utilización de otros programas al mismo tiempo, que varios clientes realicen operaciones al mismo tiempo.
- Realizar comparaciones en cuanto a las cotizaciones y ventas realizadas manualmente, verificando si el sistema arroja los mismos resultados que el esperado manualmente.

El personal afectado por el proyecto debería tener mínimamente un nivel de conocimiento básico en el uso de equipos de computación.

Se recomienda incorporar las TIC en el área contable y de inventarios de la empresa.