

**UNIVERSIDAD AUTONOMA JUAN MISAEL SARACHO**  
**FACULTAD DE CIENCIAS Y TECNOLOGIA**  
**INGENIERIA INFORMATICA**



**Mejoramiento en la administración de la información de Alquiler de Trajes  
Típicos, para la empresa "Ballet Sin Fronteras".**

**Por:**

**Rosely Valdez Aramayo**

**Trabajo final presentado a consideración de la UNIVERSIDAD AUTONOMA  
JUAN MISAEL SARACHO, como requisito para optar el grado académico de  
Licenciatura en ingeniería Informática.**

**Tarija, Marzo 2012**

## **I Proyecto**

### **I.1 Identificación del Proyecto**

#### **I.1.1 Título del proyecto**

Mejoramiento en la administración de la información de Alquiler de Trajes típicos, para la empresa “*Ballet Sin Fronteras*”.

#### **I.1.2 Carrera**

Ing. Informática.

#### **I.1.3 Facultad**

Facultad Ciencias y Tecnología.

#### **I.1.4 Institución/Centro Cooperante**

Ballet Sin Fronteras.

#### **I.1.5 Provincia/Municipio**

Tarija.

#### **I.1.6 Duración del proyecto**

12 meses.

#### **I.1.7 Área/línea de investigación priorizada**

Desarrollo de sistemas de gestión web.

## **I.2 Personal Vinculado al Proyecto**

### **I.2.1 Director del Proyecto**

Univ. Rosely Valdez Aramayo.

### **I.2.2 Equipo de trabajo de: Empresas/Instituciones/Organizaciones participantes/cooperantes**

La empresa de alquiler de vestuarios de danzas en Gral. "Ballet Sin Fronteras"

Dirección: Calle Santa Cruz N°0896 esq. Domingo Paz.

### **I.2.3 Actividades previstas para los integrantes del equipo de investigación**

Director:

- Planificar y controlar el cronograma del proyecto.
- Asignar y gestionar recursos y prioridades a los componentes y actividades del proyecto.
- Establecer un conjunto de prácticas que aseguren la calidad e integridad del proyecto.
- Supervisar el desarrollo del proyecto.
- Elaborar el análisis y diseño del sistema.
- Elaborar la base de datos del sistema.
- Elaborar los diagramas UML que definen la arquitectura del Sistema.
- Realizar la programación del sistema informático.
- Implementar el sistema en la empresa.

### **I.3 Descripción del Proyecto**

#### **I.3.1 Resumen Ejecutivo del Proyecto**

La empresa “*Ballet Sin Fronteras*” se dedica al alquiler de trajes típicos con la misión de brindar prendas de calidad y llegar al cliente con un servicio eficiente.

El control de alquiler es de vital importancia para la administración de la empresa destinada al alquiler de prendas, se comete varios procesos erróneos que afectan negativamente a la empresa.

Los procesos para el ingreso, salidas, reportes de prendas y alquileres en la empresa, se manejan manualmente también se utiliza un software que no devuelve todos los datos necesarios para la gestión de alquiler y control de inventarios eficaz, además toda la información diaria que se maneja manualmente no es almacenada, por otra parte los datos que se tiene sobre los almacenes no se encuentran actualizados diariamente, sino que sólo se actualizan periódicamente cuando se realizan inventarios manuales.

Es por esto que es indispensable aplicar las herramientas y técnicas más avanzadas que permitan facilitar las tareas administrativas y humanas en los diferentes procesos del control de alquileres para optimizar los recursos financieros y materiales de manera que se reflejen en los resultados del negocio.

Con el presente proyecto se pretende aplicar la tecnología de Información (TIC) para optimizar todos los procesos de la empresa, es decir, registro, almacenamiento, modificación y recuperación de todo tipo de información generada por las transacciones diarias que intervienen en el control de alquileres, mediante la implementación de un sistema Informático y además de la capacitación del personal encargado de administrar y manejar el sistema.

El Sistema realizará estos procesos:

- Registro de entradas de prendas
- Registro de salida de prendas
- Control de alquiler
- Control de Garantías
- Control de Usuarios
- Control de Clientes
- Control de Proveedores
- Control de Prendas

Este sistema será para el personal una herramienta que le ayudará a ser más eficiente en sus funciones, además colaborará con el proceso de toma de decisiones de la empresa gracias a su disponibilidad de información actualizada.

La capacitación se llevará a cabo en las instalaciones de la empresa una vez implementado el Sistema Informático, mediante demostraciones y explicaciones de las utilidades del mismo, además de la transferencia del manual de usuario.

### **I.3.2 Descripción y Fundamentación del Proyecto (qué y por qué)**

La Tecnología de Información (TI) nació como soporte a las necesidades dentro de las organizaciones, entre sus aplicaciones están los sistemas de información automatizados que presentan información automatizada con características de importancia, relevancia, claridad, sencillez y oportunidad de tal forma que sea útil para las personas a quienes se les entrega. Esto se refiere a que cuando los usuarios reciben los datos, estos son preparados e introducidos a la Base de Datos de la empresa mediante una aplicación de computadora. El procesamiento de estos datos depende de los propósitos de cada aplicación de tal forma que se genere la información adecuada y se presente los respectivos reportes.

El presente proyecto tiene por finalidad optimizar los procesos de la empresa para el control de alquileres con la realización de sus componentes: un Sistema Informático y la capacitación al personal de la empresa para el uso del mismo. El software que se desarrolla, basado en la tecnología de información, es un sistema de información automatizado, utilizado para registro de las transacciones diarias y la generación de reportes, con el propósito de registrar y manejar la información necesaria para la empresa.

La empresa *“Ballet Sin Fronteras”* se dedica al alquiler de trajes típicos, actualmente los procesos para el ingreso de prendas, en la empresa se manejan manualmente, y para la salida, entradas, reportes de prendas, etc. se utiliza un software que no devuelve todos los datos necesarios para una gestión de alquiler.

Además los datos que se tienen sobre las prendas no se encuentran actualizados diariamente, sólo se actualizan periódicamente cuando se realizan inventarios manuales.

El principal problema que se presenta es la deficiencia en los procesos para el control de alquiler, éste engloba:

- Control deficiente de las entradas y salidas de prendas.
- Demora en el proceso de alquiler y cálculo de comisiones.
- Datos no actualizados.
- Falta de reportes para el control de alquiler, stock de prendas, pedidos, proveedores y clientes.

Por esta razón es una necesidad para la empresa ejecutar el presente proyecto que pretende optimizar estos procesos mediante:

- La implementación de un Sistema Informático para el control eficiente de inventarios y alquileres, que facilite las tareas a los usuarios, proporcione datos que se necesitan en el menor tiempo posible y de forma confiable, minimice los errores que ocurren cuando se trabaja de forma manual y ayude a mejorar la atención al cliente final.
- Al finalizar el sistema se procederá a realizar la capacitación a los usuarios, esta capacitación será realizada de forma individual, el capacitador usará un medio de enseñanza TIC, estas tecnologías son instrumento de trabajo, además, es imprescindible su valor en su máxima expresión como forjadoras de valores y conocimientos construidos sobre fundamentos psicopedagógicos que permitan la gradación en la asimilación de conocimientos desde la familiarización de los mismos, pasando por su reproductibilidad hasta la creatividad en las acciones para alcanzar nuevos resultados académicos que luego le servirán para desarrollar hábitos y habilidades competitivas en su profesión y asimilar otras en una diferente.

### **I.3.3 Alcances**

El sistema web gestiona la información de la empresa “Ballet Sin Fronteras”, donde tendrá un acceso el Gerente propietario y otros usuarios de la empresa, el sistema contará con controles de seguridad.

El sistema contará con un informe detallado de cada alquiler, cada registro de prendas, también dará a conocer cada actividad.

El sistema contará con un informe detallado de cada prenda, también dará a conocer la disponibilidad de prendas en cada tienda, el color la talla y cantidad de dicha prenda.

#### **I.3.4 Limitaciones**

El sistema computarizado orientado a la web sólo contempla la gestión de información el cual no está orientado a la parte contable.

El sistema no contemplará el área de marketing y publicidad para la empresa.

#### **I.3.5 Objetivos**

##### **I.3.5.1 Objetivo General**

Administración de la información de Alquiler de Trajes típicos, para la empresa “*Ballet Sin Fronteras*” mejorado.

##### **I.3.5.2 Objetivos Específicos**

- Desarrollar un sistema para el control en la administración de la información de trajes típicos para la empresa “Ballet Sin Fronteras”
- Mejorar el nivel de conocimiento en el uso de nuevas tecnologías mediante la capacitación al personal de la empresa.

#### **I.3.6 Estrategias y planes de acción**

- 1. Sistema para el control en la administración de la información de trajes típicos para la empresa “Ballet Sin Fronteras”.**



- 1.1 Recopilación de la información para el desarrollo del proyecto propuesto.
- 1.2 Análisis y diseño de la página web.
- 1.3 Diseño de la Base de Datos.
- 1.4 Programación de la página web y el software del sistema.
- 1.5 Prueba y modificaciones del sistema y la página web.
- 1.6 Diseño y documentación del manual de usuario.

## 2. Programa de Formación en el uso de las TIC's y el sistema "Ballet Sin Fronteras".

- 2.1 Definición de las estrategias para el curso de formación en recursos TIC's.
- 2.2 Desarrollar el contenido del curso de Formación.
- 2.3 Coordinación del gerente y usuarios de la empresa.
- 2.4 Preparar el material del curso de formación.
- 2.5 Formación del personal en el uso del sistema.

### I.3.7 Metodología

En este presente proyecto se contempla la realización de dos componentes: un Sistema Informático y la capacitación al personal de la empresa para el uso del mismo.

Las metodologías a utilizar se describen a continuación:

Para el desarrollo de un sistema Informático se utilizará la metodología RUP (Proceso Unificado de Rational) y UML (Lenguaje Unificado de Modelado), que es un lenguaje gráfico que utiliza diagramas ya definidos para especificar y describir métodos y procesos, y definir un sistema. En RUP se siguen cuatro fases para el desarrollo de software, al final de las cuales, y tras de una serie de iteraciones, se establece objetivos precisos a conseguir:

- **Concepción.** En esta fase se establece los requisitos de negocio que cubrirá el sistema, se obtendrá la especificación de requerimientos.
- **Elaboración.** En esta fase el problema se analiza y comprende desde el punto de vista del equipo de desarrollo. Al final de la fase se tiene definida la arquitectura y el modelo de requisitos del sistema empleado, los diagramas de casos de uso especificados en el lenguaje UML.
- **Construcción.** En esta fase se profundiza en el diseño de los componentes del sistema y de manera iterativa se van añadiendo las funcionalidades al software a medida que se construyen y prueban, permitiendo a la vez que se puedan ir incorporando cambios. Al final de esta fase se obtiene un sistema operativo y la documentación (diagrama de clases, de secuencia, modelo entidad-relación, manual de instalación, manual de usuario) para entregar a los usuarios.
- **Transición.** La fase final del RUP se ocupa del traslado del software desde los entornos de desarrollo a los entornos de producción, en los que el usuario final hará uso del sistema.

Para la capacitación, la metodología a utilizar se fundamenta en los siguientes factores.

- La estructura. El modelo básico de estructura de capacitación que se toma en cuenta es el modelo de capacitación simultánea, en el que se entrena a todo el personal simultáneamente, en varios días.
- El diseño del programa de capacitación. Se enfoca en el objetivo, lograr que el personal maneje el sistema implementado sin dificultades, la disposición del personal y los principios pedagógicos de aprendizaje, los que se toman en cuenta para esta capacitación son los de participación, repetición, relevancia, transferencia y retroalimentación.

- Grado de conocimientos sobre la TI de los empleados de la empresa. Se toma en cuenta el nivel de conocimientos de los empleados sobre la TI para impartir la capacitación básica y la capacitación del sistema.
- La implementación del programa de capacitación. Se utiliza uno de los métodos de uso más generalizado: la capacitación en el trabajo, que proporciona la ventaja de la experiencia directa sobre el sistema.
- El estilo para impartir la capacitación. Se realiza el estilo participativo, como una duración de las sesiones de una hora por día, durante cinco días según cronograma, las sesiones se dividen en módulos, uno básico sobre la TI y los demás para el uso del sistema.

Por otra parte la tecnología a utilizar se fundamenta en que el software que se desarrolla es un sistema de información automatizada o Sistema Informático, que será utilizado para el registro de las transacciones diarias y la generación de reportes para presentar información y se orienta al área de “Soporte de los procesos empresariales”, como un sistema de procesamiento de transacciones. La arquitectura que se implementa es la de Cliente/Servidor, debido a la existencia de varios usuarios que utilizarán el sistema.

Se utilizará las siguientes herramientas para el desarrollo del software:

- Se desarrolla el sistema en el lenguaje de programación Java, porque es una de las tecnologías más seguras para el desarrollo de programas, es multiplataforma, de fuente abierta y utiliza el paradigma orientado a objetos, que debido al enfoque y complejidad del problema en este proyecto nos proporciona la mejor solución al permitir crear programas modulares, visuales de fácil manejo para el usuario y facilitar el mantenimiento del software.

- La base de datos se desarrolla en **PostgreSQL** ya que es un sistema de gestión de base de datos relacional orientado a objetos de software libre, publicado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de **PostgreSQL** no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizadores comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (**PostgreSQL Global Development Group**)

### I.3.8 Bibliografía Consultada para la realización del Perfil de Proyecto

- Lenguaje Unificado de Modelado <http://es.wikipedia.org/wiki/UML>
- Proceso Unificado de Rational <http://es.wikipedia.org/wiki/RUP>
- Sistemas de Información Administrativos I. Programa . Universidad Arturo Prat, Departamento de Auditoria y Sistemas de Información Iquique – Chile.
- <http://www.unap.cl/-setcheve/siiqq/page34.html>
- Diccionario Informático
- Concepto y Características del Sistema de Gestión de Información <http://www.scribd.com/doc/12227873/Concepto-y-Caracteristicas-Del-Sistemas-de-Gestion-de-Informacion>.
- MIDEPLAN, Programas Públicos. Marco Lógico: Instrumento Para La Formulación de Programas.
- Trabajo de grado: APLICACIÓN DE LA TECNOLOGÍA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIOS Y VENTAS EN LA EMPRESA EXIMPORTTARIJA LTDA

### **I.3.9 Resultados Esperados**

Sistema Informático para el control eficiente de alquileres desarrollado, con el cual se optimizarán todos los procesos que se realizan para el control de alquiler en la empresa, la documentación y los manuales de usuario.

El Personal de la empresa capacitado en el uso del software implementado, con lo que se pretende formar al personal en el uso de las utilidades del sistema y el objetivo es lograr que el personal maneje el sistema implementado sin dificultades.

### **I.3.10 Transferencia de resultados**

#### **I.3.10.1 Medios y estrategias para la transferencia de resultados.**

Se establecen dos etapas en el proyecto, que se realizarán de manera secuencial y mediante un convenio con la empresa de “Ballet Sin Fronteras”, las mismas que se detallan a continuación:

- Diseñar un Sistema Informático para el control eficiente de alquiler: Que comprende el área de inventarios, ingreso y salida de productos en el área de alquiler, control sobre usuarios, proveedores, clientes. Una vez desarrollado este sistema será instalado en la empresa para su correspondiente utilización.
- Realizar la capacitación al personal de la empresa en el manejo del Sistema Informático: La capacitación al personal encargado del manejo y administración del Sistema Informático se realiza de forma personal, mediante ejemplos y demostraciones, en instalaciones de la empresa.

### **I.3.10.2 Grupo de beneficiarios de los resultados**

Con el presente proyecto los grupos beneficiados son los siguientes:

- La empresa “Ballet Sin Fronteras”
- Los Usuarios que harán uso del sistema.
- Los Clientes finales de la empresa.

### I.3.11 Cronograma de Actividades

**Tabla 1: Cronograma de actividades**

Nº	Actividad	Nº días	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
<b>C1. Sistema informático</b>														
1	Especificación de requerimientos	25				X								
2	Elaboración de los diagramas UML del Sistema	30				X	X							
3	Diseño de Base de Datos del Sistema	20					X							





### I.3.12 Marco lógico del Proyecto

#### I.3.12.1 Cuadro de Involucrados

GRUPO	INTERESES	PROBLEMAS PERCIBIDOS	RECURSOS Y MANDATOS
Gerencia	Aumentar utilidades de la empresa.	El control de la empresa en general tiene fallas en: <ul style="list-style-type: none"> <li>• Inventarios</li> <li>• Control de registro de alquiler y devolución ineficiente</li> <li>• Reportes Inexactos</li> </ul>	<b>R:</b> Infraestructura adecuada  <b>M:</b> Disponibilidad de usar el sistema.
Administración	Optimizar el control de los recursos económicos y humanos	<ul style="list-style-type: none"> <li>• El sistema actual no proporciona todos los datos necesarios para el control eficaz de inventarios y alquileres.</li> <li>• No todo el personal utiliza el software para realizar los procesos de control de inventario y alquiler.</li> <li>• Falta de capacidad del personal para el uso del software actual, entrega de</li> </ul>	<b>R:</b> Equipos de computación suficientes para los procesos.  <b>R:</b> Personal adecuado  <b>M:</b> Responder con resultados óptimos a la Gerencia, mejores

		productos.	precios.
Trabajadores de la empresa.	Cumplir y aumentar los objetivos de alquiler trazados en a gestión	<p>Los procesos para el control de inventario y alquiler son deficientes:</p> <p>Control de entradas y salidas de prendas es deficiente.</p> <p>Demora en el proceso de liquidación de alquiler y cálculo de comisiones.</p> <p>Demora en la realización de reportes.</p>	<p><b>R:</b> Apoyo de supervisión de ventas.</p> <p><b>R:</b> Tiempo suficiente para cumplir las entregas.</p>

**Tabla 2: Cuadro de involucrados**

### I.3.12.2 Árbol de Problemas

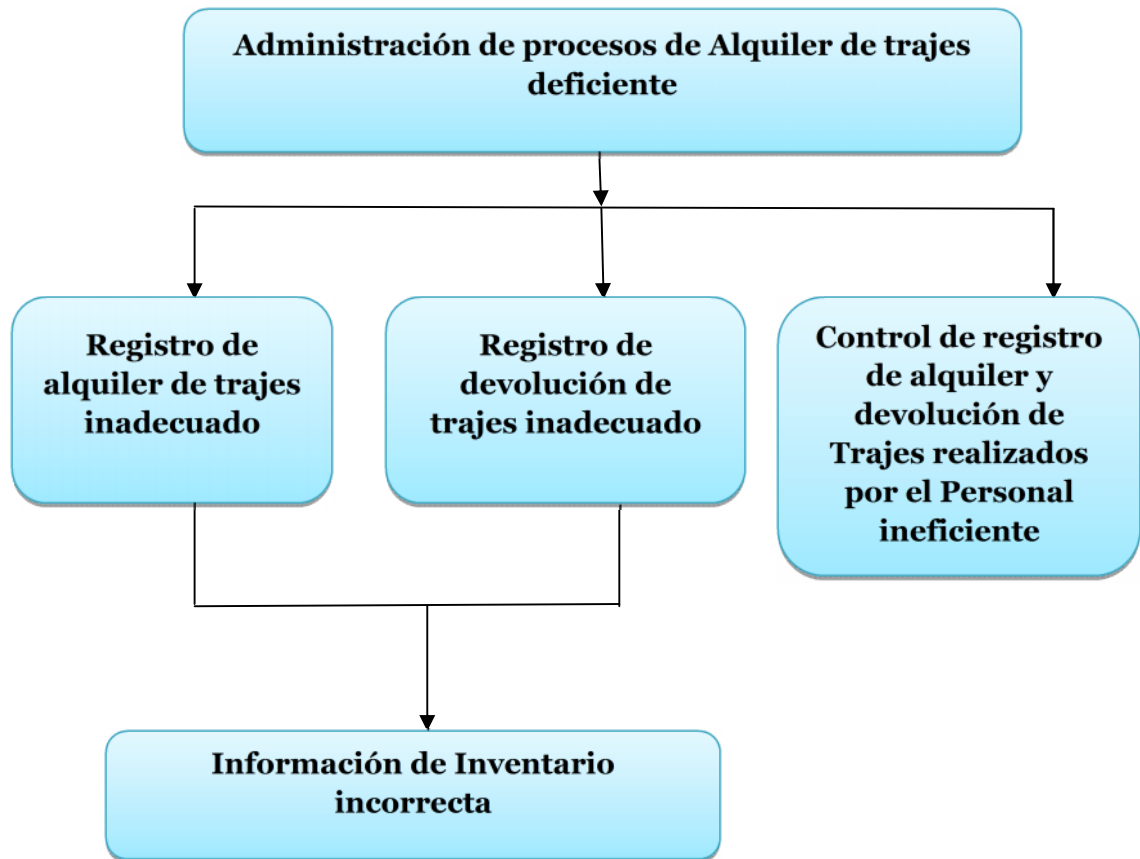
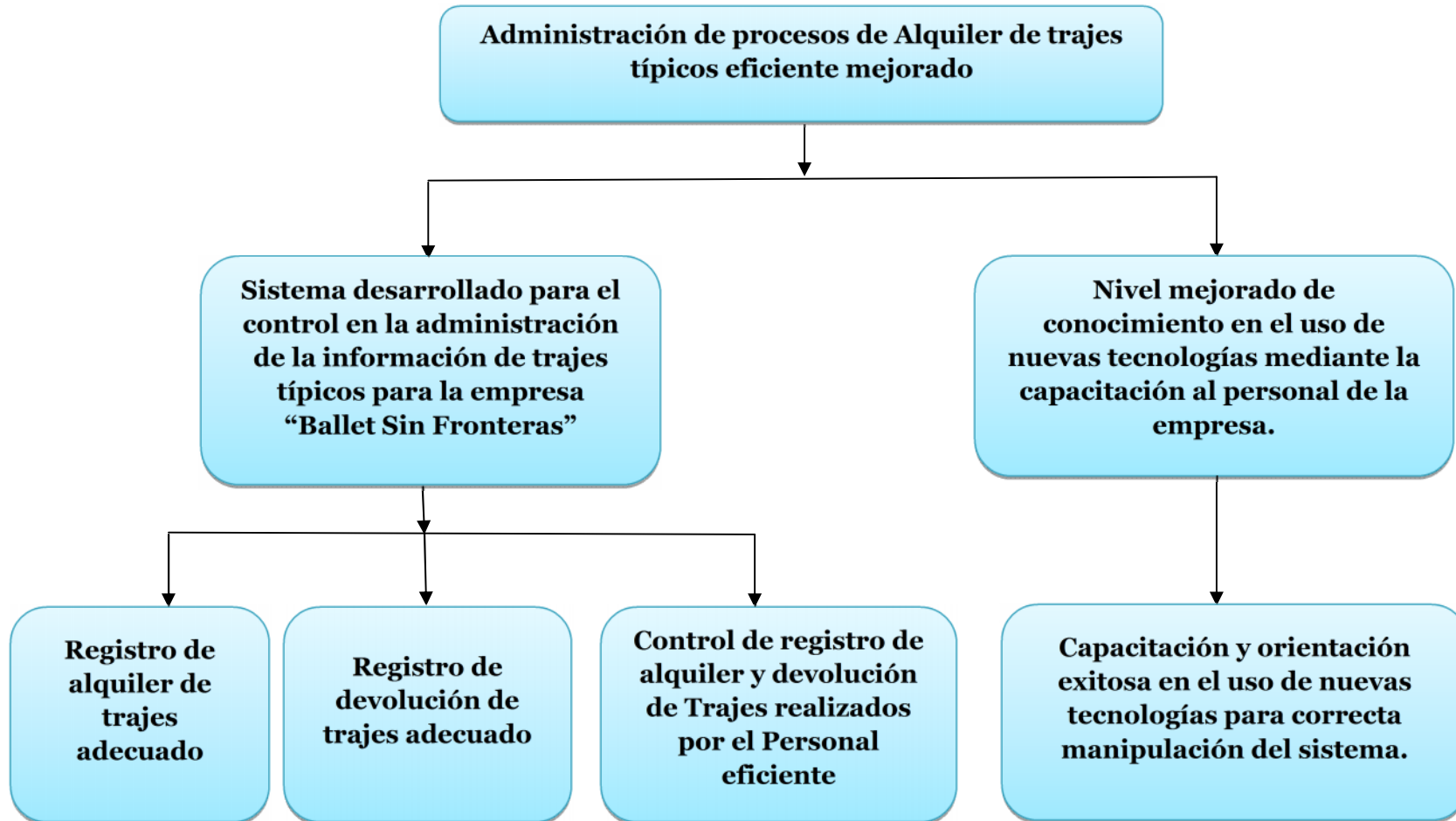


Figura 1: Árbol de Problemas

## I.3.12.3 Árbol de Objetivos figura 2



### I.3.12.4 Matriz de Marco Lógico

**Tabla 3: Matriz de Marco Lógico**

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
<p><b>Fin</b></p> <p>Optimizar utilidades y minimizar pérdidas de información de inventario y alquiler de trajes típicos de la empresa “Ballet Sin Fronteras”</p>	<p>A partir del año 2013 la información se procesa con rapidez y seguridad en un 80 % en el Departamento de Deportes. (EFICACIA)</p>	<p>Entrevistas realizadas para aprobar el proyecto.</p> <p>Informe proporcionado por la empresa expresando su conformidad con los resultados obtenidos.</p>	<p>La empresa utiliza el sistema que le proporciona el proyecto.</p>

<p><b>Objetivo General (Propósito)</b></p> <p>Mejoramiento en la administración de la información de Alquiler de Trajes típicos, para la empresa <b>“Ballet Sin Fronteras”</b>.</p>	<p>Al finalizar el proyecto los usuarios de la empresa cuentan con información oportuna de prendas y alquileres, en un 88% para el desempeño de las actividades administrativas.</p> <p>Fórmula empleada es de eficacia.</p>	<p>Informe de conformidad proporcionado por parte del Gerente propietario que exprese la satisfacción en los procesos administración de la información.</p>	<p>Presupuesto suficiente para la implementación del Sistema Informático.</p> <p>Disponibilidad de equipos para la utilización del sistema.</p>
---	--	---	---

<p><b>Objetivos Específicos (Componentes)</b></p> <ol style="list-style-type: none"> <li>1. Sistema desarrollado para el control en la administración de la información de trajes típicos para la empresa “Ballet Sin Fronteras”</li> <li>2. Nivel mejorado de conocimiento en el uso de nuevas tecnologías mediante la capacitación al personal de la empresa.</li> </ol>	<ol style="list-style-type: none"> <li>1. Al finalizar el proyecto, una vez implementado el sistema en la empresa, el 95% de los procesos administrativos son de eficaz funcionamiento.</li> <li>2. El 50% de los usuarios formaron parte del programa completo de capacitación.</li> </ol> <p>Fórmula empleada es de eficacia.</p>	<ol style="list-style-type: none"> <li>1. Certificado de conformidad de la funcionalidad del sistema proporcionado por la empresa y firmado por el personal encargado de la administración de la tienda.</li> </ol>	<ol style="list-style-type: none"> <li>1. El personal de la empresa hace uso del sistema.</li> <li>2. El personal que conforma la institución participan activamente en las capacitaciones.</li> </ol>
<p><b>Actividades:</b></p> <p><b>Componente 1:</b></p> <p>Eficiente y seguro tratamiento de la información automática en la gestión de alquileres.</p> <p><b>Componente 2:</b></p> <p>Capacitación y orientación en el uso de nuevas tecnologías para correcta</p>			<p>Disponibilidad de equipos de computación.</p> <p>Existencia y disponibilidad de recursos económicos.</p>

manipulación del sistema.			
---------------------------	--	--	--



#### I.4 Presupuesto / Justificación

Tabla 4: Presupuesto

<b>ITEM</b>	<b>RUBROS</b>	<b>Aporte Universidad</b>	<b>Otro Aporte</b>	<b>TOTAL (Bs.)</b>
<b>10000</b>	<b>SERVICIOS PERSONALES</b>			
	<b>12000 Empleados no Permanentes</b>		1500	1500
	<b>Sub total rubro</b>		1500	1500
<b>20000</b>	<b>SERVICIOS NO PERSONALES</b>			
	<b>21000. Servicios Básicos</b>			
	<b>22000. Servicios de transporte</b>		80	80
	<b>23000. Alquileres</b>			
	<b>24000. Mantenimiento y reparación</b>			
	<b>25000. Servicios Profesionales y</b>		700	700

	<b>Comerciales</b>			
	<b>Sub total rubro</b>		780	780
<b>30000</b>	<b>MATERIALES Y SUMINISTROS</b>			
	<b>31000. Alimentos y Productos Forestales</b>		150	150
	<b>32000. Productos de Papel, Cartón e Impresos</b>		500	500
	<b>33000. Textiles y Vestuario.</b>			
	<b>34000. Productos Químicos, Combustibles y Lubricantes</b>			
	<b>39000. Productos Varios.</b>		50	50
	<b>Sub total rubro</b>		700	700
<b>40000</b>	<b>ACTIVOS REALES</b>			
	<b>43000. Maquinaria y Equipo.</b>			

	<b>46000. Descripción de estudios y proyectos para inversión</b>			
	<b>49000. Otros Activos</b>			
	<b>Sub total rubro</b>			
	<b>TOTAL</b>		2980	2980
	<b>TOTAL + 40% Incentivo</b>			

**1) GRUPO 10000. SERVICIOS PERSONALES**

**a) SUB GRUPO 12000. Empleados no Permanentes**

<b>Partida</b>	<b>Personal</b>	<b>Remuneración</b>	<b>Tiempo/meses</b>	<b>Total</b>
12100	Personal Eventual	250	6	1500
	<b>Total</b>			

**Tabla 5: Presupuesto: Empleados no Permanentes**

\* Se refiere a gastos para remunerar a personas sujetas a contrato dependientes según la necesidad de cada entidad

**2) GRUPO 20000. SERVICIOS NO PERSONALES**

**b) SUB GRUPO 21000. Descripción de los gastos de servicios básicos**

<b>Partida</b>	<b>Tipo de servicio básico *</b>	<b>Costo</b>	<b>Tiempo mes</b>	<b>Costo Total</b>
21100	Comunicación			
21200	Energía Eléctrica			
21300	Agua			
21400	Servicios Telefónicos			
<b>Total</b>				

**Tabla 6: Presupuesto: Descripción de los gastos de servicios básicos**

\* Se refiere principalmente a los gastos por servicios; como: servicio de correo, radiogramas, servicio telefónico, fax, Internet.

**c) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal**

<b>Partida</b>	<b>Personal</b>	<b>Lugar</b>	<b>Nº de viajes</b>	<b>Costo unitario*</b>	<b>Costo total</b>
22100	Pasajes	Ubicación de la empresa	80	1	80
<b>Total</b>					80

**Tabla 7: Presupuesto: Descripción de los gastos de transporte**

<b>Partida</b>	<b>Personal</b>	<b>Lugar</b>	<b>Duración (días)</b>	<b>Costo unitario*</b>	<b>Costo total</b>
22200	Viáticos				
22300	Fletes y Almacenamientos				
22600	Transporte de Personal				
<b>Total</b>					
<b>Total sub grupo 22000</b>					

**Tabla 8: Presupuesto: Descripción de los gastos de viajes y transporte de personal**

\* En el caso de los viáticos, debe considerarse la escala establecida por la UAJMS.

**d) SUB GRUPO 23000. Descripción de los gastos por concepto de alquileres de equipos y maquinarias**

<b>Partida</b>	<b>Alquiler de equipo y maquinaria</b>	<b>Costo unitario</b>	<b>Tiempo mes</b>	<b>Costo total</b>
23100	Alquiler de Edificios			
23200	Alquiler de Equipos y Maquinaria			
23300	Alquiler de Tierras y Terrenos			
<b>Total</b>				

**Tabla 9: Presupuesto: Descripción de los gastos por concepto de alquileres de equipos y maquinarias**

\* Se refiere principalmente a los gastos por el uso de edificios y equipos y maquinaria en general

**e) SUB GRUPO 24000. Descripción mantenimiento y reparación**

<b>Partida</b>	<b>Mantenimiento y reparación de equipo y maquinaria</b>	<b>Costo unitario</b>	<b>Tiempo mes</b>	<b>Costo total</b>
24100	Mantenimiento y Reparación de Edificios y Equipos			
24300	Otros Gastos por Mantenimiento y Reparación			
<b>Total</b>				

**Tabla 10: Presupuesto: Descripción de los gastos por concepto de alquileres de equipos y maquinarias**

\* Se refiere principalmente a los gastos por el mantenimiento y reparación de edificios y equipos y maquinaria en general

**f) SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales**

<b>Partida</b>	<b>Tipo de servicio profesional y comercial *</b>	<b>Cantidad</b>	<b>Costo unitario</b>	<b>Tiempo mes</b>	<b>Costo total</b>
25200	Estudios e Investigaciones				
25500	Publicidad				

25600	Imprenta				
25700	Capacitación de Personal	2	350	1	700
25800	Estudios e Investigaciones Para Proyectos de Inversión				
<b>Total</b>					700

**Tabla 11: Presupuesto: Descripción de los gastos en servicios profesionales y comerciales**

\* Se refiere a gastos por servicios profesionales de asesoramiento especializado, se incluyen, estudios, investigaciones, publicidad, imprenta, fotocopias, capacitación de personal y otros ejecutados por terceros.

### 3) GRUPO 30000. MATERIALES Y SUMINISTROS

#### g) SUB GRUPO 31000. Descripción de los gastos Alimentos y Productos Agroforestales

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
31110	Refrigerios y Gastos Administrativos			
	• Refrigerio(empanadas)	• 50	• 2	• 100
	• Refrigerio(refresco))	• 5	• 10	• 50



31200	Alimento para Animales			
31300	Productos Agroforestales y Pecuarios			
<b>Total</b>				150

**Tabla 12: Presupuesto: Descripción de los gastos Alimentos y Productos Agroforestales**

\* Se refiere a la adquisición de materiales y bienes como: alimentos y productos agroforestales, alimentos y bebidas para personas (indicar el total de refrigerios), alimentos para animales, productos pecuarios.

**h) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos**

<b>Partida</b>	<b>Tipo de material *</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
32100	Papel de Escritorio Resmas de papel tamaño carta	5	70	350
32200	Productos de Artes Gráficas, Papel y Cartón			
32300	Libros y Revistas			

32400	Textos de Enseñanza	10	15	150
32500	Periódicos			
<b>Total</b>				500

**Tabla 13: Presupuesto: Descripción del gasto de Productos de Papel,  
Cartón e Impresos**

\* Se refiere a la adquisición de; papel y cartón en sus diversas formas y clases,  
impresos y publicaciones, periódicos, revistas, libros, fotocopias, etc.

**i) SUB GRUPO 33000. Descripción del gasto en textiles y vestuario**

<b>Partida</b>	<b>Productos textiles y vestuarios</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
33100	Hilados y Telas			
33200	Confecciones Textiles			
33300	Prendas de vestir			
33400	Calzados			
<b>Total</b>				

**Tabla 14: Presupuesto: Descripción del gasto en textiles y vestuario**

\* Se refiere principalmente a los gastos por vestuario uniformes, ropa de trabajo

**j) SUB GRUPO 34000. Combustibles, Productos Químicos, Farmacéuticos y Otros**

<b>Partida</b>	<b>Combustibles, Productos Químicos, Farmacéuticos y Otros</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
34110	Combustibles y Lubricantes para Consumo			
34200	Productos químicos y Farmacéuticos			
34400	Productos de Cuero y Caucho			
34500	Productos de Minerales no Metálicos y Plásticos			
34600	Productos Metálicos			
34800	Herramientas Menores			
<b>Total</b>				

**Tabla 15: Presupuesto: Combustibles, Productos Químicos, Farmacéuticos y Otros**

\* Se refiere a gastos de combustibles, químicos, productos farmacéuticos, llantas etc.

**k) SUB GRUPO 39000. Descripción del gasto en productos varios**

<b>Partida</b>	<b>Productos de cuero y caucho</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
39100	Material de Limpieza			
39400	Instrumental Menor Médico - Quirúrgico			
39500	Útiles de Escritorio y de Oficina	25	2	50
39700	Útiles y Materiales Eléctricos			
39800	Otros Repuestos y Accesorios			
<b>Total</b>				50

**Tabla 16: Presupuesto: Descripción del gasto en productos varios**

\*Se refiere principalmente a los gastos por productos de limpieza, todo lo referente al funcionamiento de la oficina en material de escritorio.

4) **GRUPO 40000. ACTIVOS REALES**1) **SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo**

<b>Partida</b>	<b>Tipos de productos</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
43100	Equipo de Oficina y Muebles			
43200	Maquinaria y Equipo de Producción			
43300	Equipos de Transporte, Tracción y Elevación			
43400	Equipo Médico y de Laboratorio			
43600	Equipo Educativo y Recreativo			
43700	Otra Maquinaria y Equipo			
<b>Total</b>				

**Tabla 17: Presupuesto: Descripción del gasto de Maquinaria y Equipo**

\* Se refiere principalmente a los gastos por muebles y enseres, equipo de oficina, comunicación, equipamiento.

**m) SUB GRUPO 46000. Descripción de estudios y proyectos para inversión**

<b>Partida</b>	<b>Productos textiles y vestuarios</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
46100	Para Construcción de Bienes de Dominio Privado			
<b>Total</b>				

**Tabla 18: Presupuesto: Descripción de estudios y proyectos para inversión**

\* Se refiere principalmente a los gastos por servicios de terceros para la realización de investigaciones y otras actividades técnico – Profesionales necesarias para la construcción y mejoramiento de bienes.

**n) SUB GRUPO 49000. Descripción del gasto de Otros Activos**

<b>Partida</b>	<b>Tipos de productos *</b>	<b>Cantidad</b>	<b>Costo/Unitario</b>	<b>Total</b>
49100	Activos Intangibles			
49200	Compra de Bienes Muebles Existentes (Usados)			
49300	Semovientes y otros Animales			

49900	Otros Activos			
<b>Total</b>				

**Tabla 19: Presupuesto: Descripción del gasto de Otros Activos**

\* Se refiere a los gastos en la compra de software, licencias.

## **II Componentes**

### **II.1 Componente 1: Sistema Informático para el control de alquiler.**

#### **II.1.1 Plan de Desarrollo del Software**

##### **II.1.1.1 Introducción**

El plan de Desarrollo es el documento que proporciona la visión global del enfoque del 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 se procederá a cumplir con las 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 que serán generados. Este documento es a su vez uno de los artefactos de RUP.

##### **II.1.1.2 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.



- Los miembros del equipo de desarrollo lo usan para entender lo que deben hacer, cuando deben hacerlo y que otras actividades dependen de ello.

### **II.1.1.3 Alcance**

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Sistema Informático para el control de inventarios y alquiler “Ballet Sin Fronteras”. 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.1.1.4 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 RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es necesario destacar que de acuerdo a la filosofía de RUP, todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, solo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de

completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

### **1) Plan de Desarrollo del Software**

Es el presente documento.

### **2) Modelo de Casos de Uso del Negocio**

Éste es un modelo de las funciones vistas desde las perspectivas de los actores externos. Permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito.

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

### **3) Modelo de Objetos del Negocio**

Éste 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. Para la representación de este modelo se utiliza Diagrama de Colaboración para mostrar actores externos, internos y las entidades que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones.

### **4) Glosario**

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

## **5) 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.

## **6) Modelo de Casos de Uso**

Presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagrama de Casos de Uso.

## **7) Especificaciones de Casos de Uso**

Para los casos de uso se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no funcionales asociados, cada especificación se sitúa con su correspondiente diagrama de Casos de Uso.

## **8) Especificaciones Adicionales**

Las especificaciones adicionales contienen los requisitos funcionales y no funcionales que serán la base para el desarrollo del sistema.

## **9) Modelo de Análisis de 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), de acuerdo al avance del proyecto.

## **10) Modelo de Datos**

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes de acuerdo con el enfoque para modelado relacional de datos, para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un perfil UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

## **11) Casos de Prueba**

Cada caso de prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba y los resultados esperados.

Estos casos de prueba son aplicados como prueba con las instrucciones para realizar la prueba y dependiendo de tipo de prueba dicho procedimiento podrá ser automatizado mediante un script de prueba.

## **12) 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.

### **13) 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. (este modelo es sólo una versión preliminar al final de la fase de la elaboración, posteriormente tiene bastante refinamiento).

### **14) Modelo de Despliegue**

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

### **15) Solicitud de Cambio**

Los cambios propuestos para los artefactos se formalizan mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se cuenta con registro de decisiones, cambios y su evaluación e impacto, y se asegura que estos sean conocidos por el equipo de desarrollo. Los cambios se establecen respecto de la última baseline (el estado del conjunto de los artefactos en un momento determinado del proyecto) establecida. En nuestro caso al final de cada iteración se establecerá una baseline que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba llevarán asociado un procedimiento de

prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

#### **16) Plan de Iteración**

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

#### **17) Evaluación de Iteración**

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

#### **18) Manual de Instalación**

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

#### **19) Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, como guías del usuario.

#### **20) 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 en desarrollo incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

### **II.1.1.5 Resumen**

El documento está organizado en las siguientes secciones:

Vista General del Proyecto proporciona una descripción del propósito, alcance y objetos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto, está representada en el capítulo I del presente documento.

Organización del proyecto describe la estructura organizacional del equipo de desarrollo, representada también en el capítulo I del documento.

Componentes del sistema muestra los dos componentes que conforman el sistema, cada uno con su respectivo marco teórico.

### **II.1.2 Tecnologías de Información**

La tecnología de la información (TI) es un término general que describe cualquier tecnología que ayuda a producir, manipular, almacenar, comunicar, y/o esparcir información.

La TI se encuentra generalmente asociada a las computadoras y las tecnologías afines aplicadas a la toma de decisiones, esta situación está cambiando la forma tradicional de hacer las cosas, las personas que trabajan en empresas privadas que dirigen al personal o que trabajan como profesional en cualquier campo utilizan la TI cotidianamente mediante el uso del internet, las tarjetas de crédito, el pago electrónico de la nómina, entre otras funciones; es por eso que la función de la TI en los procesos de la empresa como manufactura y alquileres se han expandido grandemente.

Utilizando eficientemente la tecnología de la información se pueden obtener ventajas competitivas, ya que hoy en día, los sistemas de información juegan un papel primordial en la vida de la empresa, ayudando a mejorar procesos, reducir tiempo y a centrarse en tareas que agreguen valor.

La función principal de estos sistemas es mejorar el flujo y proceso de la información para que pueda ser accesible, se obtenga a tiempo y sea de calidad.

A medida que los precios de los equipos de computación bajan, su capacidad aumenta, y se hacen más fáciles de usar, la TI se utiliza en nuevas y varias formas. Hoy en día, la mayoría de las empresas medianas y grandes utilizan la TI para administrar casi todos los aspectos del negocio, especialmente el manejo de los registros de empleados, facturación, cobranza, pagos, compras y otros.

#### **II.1.2.1 Tipos de Sistemas**

Los sistemas de información automatizada sirven para el registro de las transacciones diarias y la generación de reportes que presentan información. Sus características son relevancia, claridad, sencillez y oportunidad de tal forma que sea útil para las personas a quienes se les entrega. Se puede identificar los siguientes componentes de un sistema de información.

- Datos: Hechos aislados que por sí solos no significan mucho.
- Información: Datos modificados o combinados que poseen significado de utilidad para quien los recibe.
- Personas: Personas que hacen uso de los sistemas de información y que según su puesto los desarrollan, manipulan, mantienen y usan la información resultante de ellos.



- Actividad: Conjunto de asignaciones o pasos desarrollados para completar un proceso o función de empresa, como: Registro de pedidos de proveedores, alquiler, reservas; cada una de las cuales está asociada a una aplicación de computadora.

### **II.1.2.2 Sistemas Transaccionales**

Los sistemas transaccionales son sistemas que automatizan las tareas operativas, normalmente son el primer tipo de sistemas de información que se implementan en las organizaciones. Estos sistemas muestran intensa entrada y salida de información.

Propiedades de los sistemas transaccionales:

- Automatizan tareas operativas en una organización, permitiendo ahorrar en personal.
- Suelen dirigirse especialmente al área de alquiler, administración de recursos humanos.
- Sus cálculos y procesos suelen ser simples.
- Se suelen utilizar para grandes bases de datos.
- Los beneficios de este tipo de sistemas en una organización son rápidamente visibles.
- Estos sistemas son optimizados para almacenar grandes volúmenes de datos, pero no para analizar los mismos.

Características:

- Rapidez: deben ser capaces de responder rápidamente, en general la respuesta no debe ser mayor un par de segundos.
- Fiabilidad: deben ser altamente fiables, de lo contrario podría afectar a clientes, al negocio, a la reputación de la organización, etc. En caso de fallas,

deben tener mecanismos de recuperación y de respaldo de datos.

- Inflexibilidad: no pueden aceptar información distinta a la establecida.

### **II.1.2.3 Lenguaje de programación JAVA**

Java es un lenguaje de programación orientado a objetos desarrollados por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de puntos y memoria.

El lenguaje Java se creó con 5 objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir una ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajar en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

El diseño de java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor y amplitud de uso en distintos ámbitos de la industria de la informática.

Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. En entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PC de usuarios de los sistemas operativos más usados en el mundo. Además,

muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecuten en cualquier PC.

En las primeras versiones de la plataforma Java existían importantes limitaciones de APIs de desarrollo gráfico (AWT). Desde la aparición de la biblioteca Swing la situación mejoró substancialmente y posteriormente con la aparición de las bibliotecas como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. Sea relativamente sencillo.

#### **II.1.2.4 Base de Datos SQL**

PgAdmin es un proyecto de software libre publicado bajo la licencia de PostgreSQL. El software está disponible en fuente y el formato binario de la red de servidores espejos de PostgreSQL.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG.

#### **II.1.2.5 Rational Rose**

Rational Rose es una herramienta para “modelado visual”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software.

Rational Rose permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- Modelado del negocio
- Captura de requisitos (parcial)

- Análisis y diseño (completo)
- Implementación (como ayuda)
- Control de cambios y gestión de configuración (parte)

### **II.1.3 Aplicaciones de las TI para el control de alquiler**

Debido a los problemas en la empresa, señalados con anterioridad, es indispensable aplicar la herramienta y tecnología más avanzada que permiten facilitar las tareas administrativas y humanas en los diferentes procesos del control de alquiler para optimizar los recursos financieros y materiales de manera que se reflejen en los resultados del negocio.

Con este presente proyecto se pretende aplicar la tecnología de Información (TI) para optimizar todos los procesos de la empresa, es decir el registro, almacenamiento, modificación y recuperación de todo tipo de información generada por las transacciones diarias que intervienen en el control de inventarios y alquiler, mediante implementación de un Sistema Informático.

Este Sistema Informático aplicado a los procesos de gestión de inventario ayuda a aumentar la eficiencia operativa del negocio, mejorando las capacidades de automatización y seguimiento de las prendas para optimizar la exactitud del inventario, controlar su existencia y cubrir de forma precisa la demanda del cliente.

Áreas que abarcará el Sistema.

- Registro de prendas que ingresan
- Registro de prendas que salen
- Control de Clientes
- Control de Alquiler
- Control de Usuarios
- Control de Proveedores

Este sistema será para el personal una herramienta que le ayudará a ser más eficiente en sus funciones y además colabora con el proceso de toma de decisiones de la empresa gracias a su disponibilidad de información actualizada.

### **II.1.3.1 Marco Metodológico**

#### **II.1.3.1.1 RUP**

##### **II.1.3.1.1.1 Concepto y Características**

RUP es una metodología que destaca tres características en el proceso del software: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental. Los casos de uso son una técnica de captura de los requisitos para pensar en términos de importancia para el usuario y no sólo en términos de funciones que se deben contemplar. Se define un Caso de Uso como fragmento de funcionalidad del sistema. En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba.

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que nos permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema y ayuda a determinar en qué orden se ve la influencia por la plataforma software, sistema operativo, gestor de base de datos, protocolos, estas restricciones constituyen requisitos no funcionales del sistema.

En RUP se establece desde el inicio una buena arquitectura que no se vea fuertemente afectada ante cambios posteriores durante la construcción y el mantenimiento. RUP tiene un proceso iterativo e incremental donde el trabajo se divide en partes más pequeñas o mini proyectos, cada uno de éstos es considerado una

iteración, de la cual se obtiene un incremento que produce un crecimiento en el producto.

El proceso iterativo e incremental consta de una secuencia de iteraciones, cada una comprende una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina, así se determina si es que hay nuevos requisitos o los que existían han cambiado.

#### **II.1.3.1.1.2 Fases**

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto.

- **Concepción**

En esta fase se establece los requisitos de negocio que cubrirá el sistema, se obtendrá la especificación de requerimientos.

- **Elaboración**

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

- **Construcción**

En esta fase se profundiza en el diseño de los componentes del sistema y de manera iterativa se va añadiendo las funcionalidades del software a medida que se construyen y prueban, permitiendo a la vez que se puedan ir incorporando cambios. Al final de esta fase se obtiene un sistema completamente operativo y la documentación (diagrama de clases, de casos de uso, de secuencia, de actividades modelo entidad relación, manual de instalación, manual de usuario) para entregar a los usuarios.

- **Transición**

La fase final del RUP se ocupa del traslado del software desde los entornos de desarrollo a los entornos de producción, en los que el usuario final hará uso de sistema.

#### **II.1.3.1.1.3 Ventajas**

Gracias a RUP se logra implementar un software que cumpla con los siguientes factores de calidad.

- **Eficiencia:** La eficiencia del software en su capacidad para hacer un buen uso de los recursos que manipula.
- **Transportabilidad:** (portabilidad): La transportabilidad es la facilidad con la que un software puede ser transportado sobre diferentes sistemas físicos o lógicos.
- **Verificabilidad:** La verificación de un software; en su capacidad para soportar los procedimientos de validación y de aceptar juegos de test o ensayo de programas.
- **Integridad:** La integridad es la capacidad de un software para sus propios componentes contra los procesos que no tengan derecho de acceso.
- **Facilidad de uso:** Un software es fácil de utilizar si se puede comunicar con él de manera cómoda.
- **Corrección:** Capacidad de los productos software de realizar exactamente las tareas definidas por su especificación.
- **Robustez:** Capacidad de los productos software de funcionar incluso en situaciones anormales.
- **Extensibilidad:** Facilidad que tienen los productos de adaptarse a cambios en su especificación. Existen dos principios fundamentales para conseguir esto: diseño simple y descentralización.
- **Reutilización:** Capacidad de los productos para ser utilizados, en su totalidad o en parte, en nuevas aplicaciones.

- Compatibilidad: Facilidad de los productos para ser combinados con otros.

### **II.1.3.1.2 Ingeniería de Requerimientos**

#### **II.1.3.1.2.1 Concepto**

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. Los requisitos en un sistema describen los servicios que ha de ofrecer el sistema y sus restricciones asociadas a su funcionamiento.

#### **II.1.3.1.2.2 Objetivos**

Los objetivos específicos de la ingeniería de requerimientos son:

- Definir el ámbito del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada en las necesidades y metas del usuario.
- Establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer.
- Proveer una base para estimar recursos y tiempo de desarrollo de sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.

#### **II.1.3.1.2.3 Requerimientos Funcionales**

Expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento).



#### **II.1.3.1.2.4 Requerimientos no Funcionales**

Son restricciones en el espacio de posibles soluciones, como ser:

- Rendimiento del sistema: fiabilidad, tiempo de respuesta, disponibilidad.
- Interfaces: dispositivos de E/S, usabilidad, interoperabilidad.
- Proceso de desarrollo: estándares, herramientas, plazo de entrega.

#### **II.1.3.1.2.5 Modelo del Negocio**

El modelo del negocio describe en detalle cómo el negocio trabaja internamente para llevar a cabo las funciones que ejecuta, puede contener: los diagramas, trabajadores, sistemas, entidades, reglas, las realizaciones, colaboraciones, entre otros elementos del negocio. Para representar los diagramas del modelo del negocio se pueden emplear diferentes diagramas de UML tales como: Diagramas de Secuencia, Diagramas de análisis del Negocio, Diagramas de Actividad, Diagramas de Estado.

### **II.1.3.1.3 UML**

#### **II.1.3.1.3.1 Concepto**

UML es un lenguaje de propósito general para el modelado orientado objetos, que visualiza, especifica, construye y documenta los artefactos de un sistema de software. Al utilizar herramientas de modelado visual se facilita la gestión de estos modelos, permitiendo ocultar y exponer detalles cuando sea necesario. El modelo visual también ayuda a mantener existencia entre los artefactos del sistema: requisitos, diseños e implementaciones. Es decir, ayuda a mejorar la capacidad para gestionar la complejidad del software. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Pretende ser tan simple como sea posible pero manteniendo la capacidad de modelar cualquier tipo de sistema

que se necesite construir. UML es lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como la encapsulación y componentes.

#### II.1.3.1.3.2 Diagramas

UML consta de varios tipos de diagramas de los cuales dependiendo el tipo del sistema a contribuir, se utilizan los necesarios. Se describe a continuación los diagramas que corresponden al caso.

- **Diagrama de Actividades:** Son similares a los diagramas de flujo de otras metodologías OO. En realidad se corresponden, caso especial de los diagramas de estado donde los estados son estado de acción (estados con una acción interna y una o más transacciones que suelen pasar al finalizar esta acción, o lo que es lo mismo, un paso en la ejecución de lo que va ser un procedimiento) y las transacciones vienen provocadas por la finalización de las acciones que tienen lugar en los estados de origen.

Siempre van unidos a una clase o a la implementación de un caso de uso o de un método (que tiene el mismo significado que en cualquier otra metodología OO).

- **Diagrama de Secuencia:** Un diagrama de secuencia representa una forma de indicar el periodo durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento. Los diagramas de secuencia permiten indicar cuál es un momento en el que se envía

o se completa un mensaje mediante el tiempo de transición, que se especifica en el diagrama.

## **II.1.4 Análisis y Diseño**

### **II.1.4.1 Requerimientos Funcionales**

- Se deberán registrar los datos de todos los usuarios que utilizarán el sistema.
- El nombre completo de un usuario debe ser único.
- Deberá registrar las salidas e ingresos de productos de almacén.
- Deberán registrar los datos de prendas y su relación con las líneas y proveedores.
- Deberán registrar los datos de los usuarios y proveedores.
- Deberá registrar los datos de los alquileres y entregas realizadas diariamente.
- Deberán reducir las cantidades del inventario cada vez que realiza la venta de un producto.

### **II.1.4.2 Requerimientos no Funcionales**

Los usuarios tienen restringido sus roles en el sistema de acuerdo a su desempeño laboral.

Los pedidos a proveedores deben ser cargados con anticipación.

El registro de prendas debe ser llenado respetando sus campos.

La carga de resultados de prendas introducidas realizadas debe ser introducida con cuidado para que se guarde correctamente en los pedidos.

## **II.1.5 Modelo de casos de uso del negocio**

### **II.1.5.1.1 Introducción**

El Modelo de Casos de Uso del Negocio es un artefacto de la disciplina Requisitos en la metodología RUP, la cual estamos implementando.

#### **II.1.5.1.2 Propósito**

- Comprender la estructura y la dinámica de la organización.
- Comprender problemas actuales e identificar posibles mejoras.
- Comprender los procesos de negocio de la organización.

#### **II.1.5.1.3 Alcance**

- Describe los procesos de negocio y los clientes.
- Identifica y define los procesos de negocio según los objetivos de la organización.
- Define un caso de uso del negocio para cada proceso del negocio (diagrama de casos de uso del negocio puede mostrar el contexto y los límites de la organización)

#### **II.1.5.1.4 Descripción de Actores del Negocio**

Al mismo tiempo que se determinan los procesos de negocio es posible identificar los agentes implicados en el proceso de Negocio.

Se identificaron los siguientes actores que interactúan con los procesos del negocio:

- Actor Gerente
- Actor Administrador(Usuarios)
- Actor Cliente

### II.1.5.1.5 Diagrama de Caso de Uso del Negocio

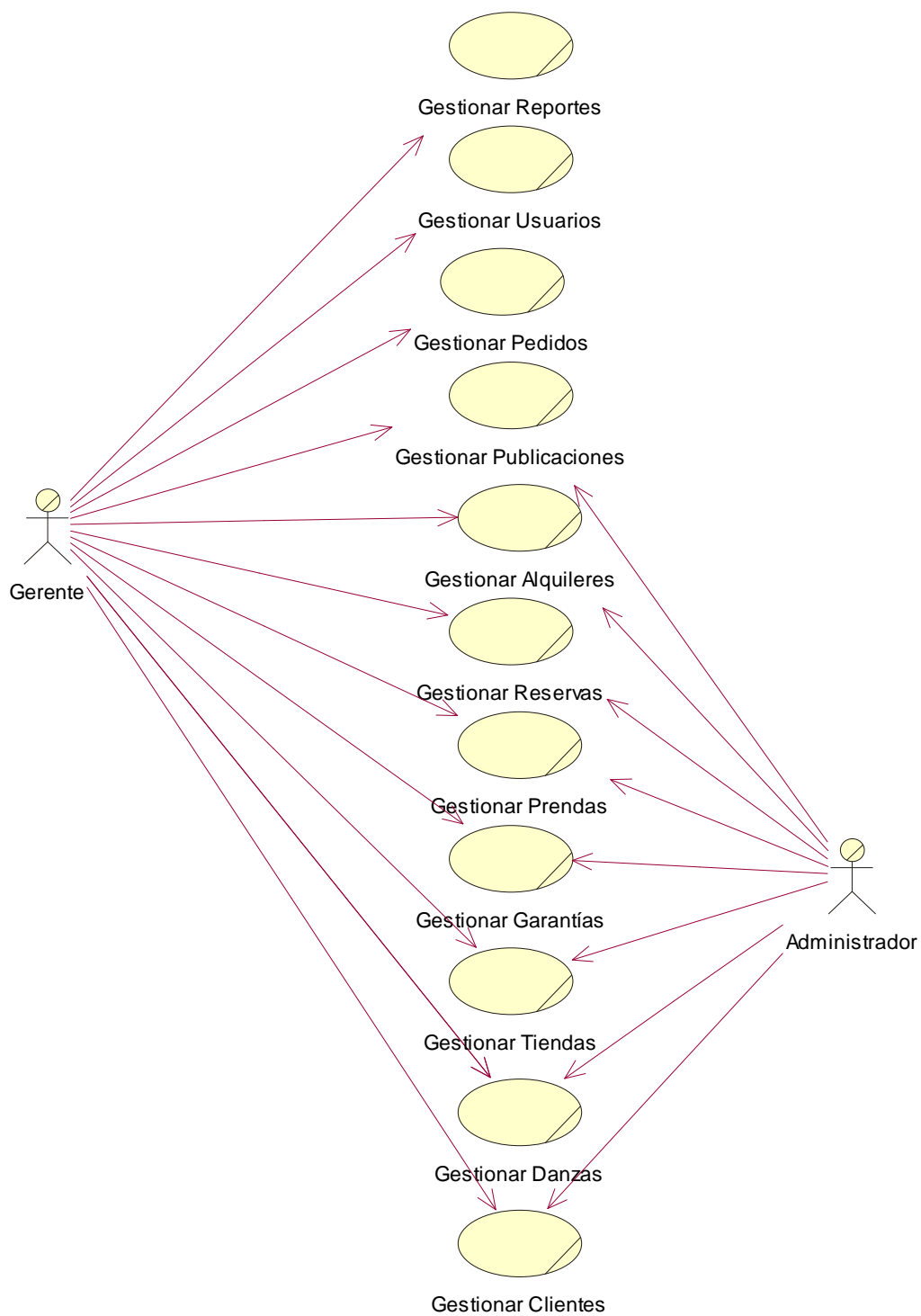


Figura 3: Diagrama de Caso de Uso del Negocio

## **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 estamos 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 Modelo de objetos del negocio

#### Registrar Cliente

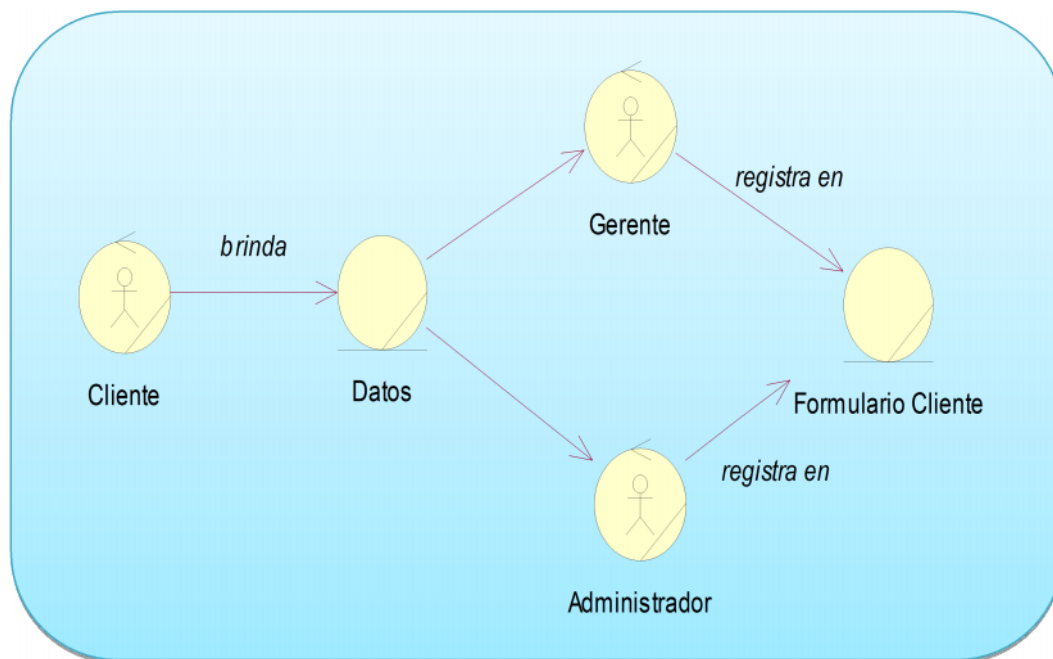
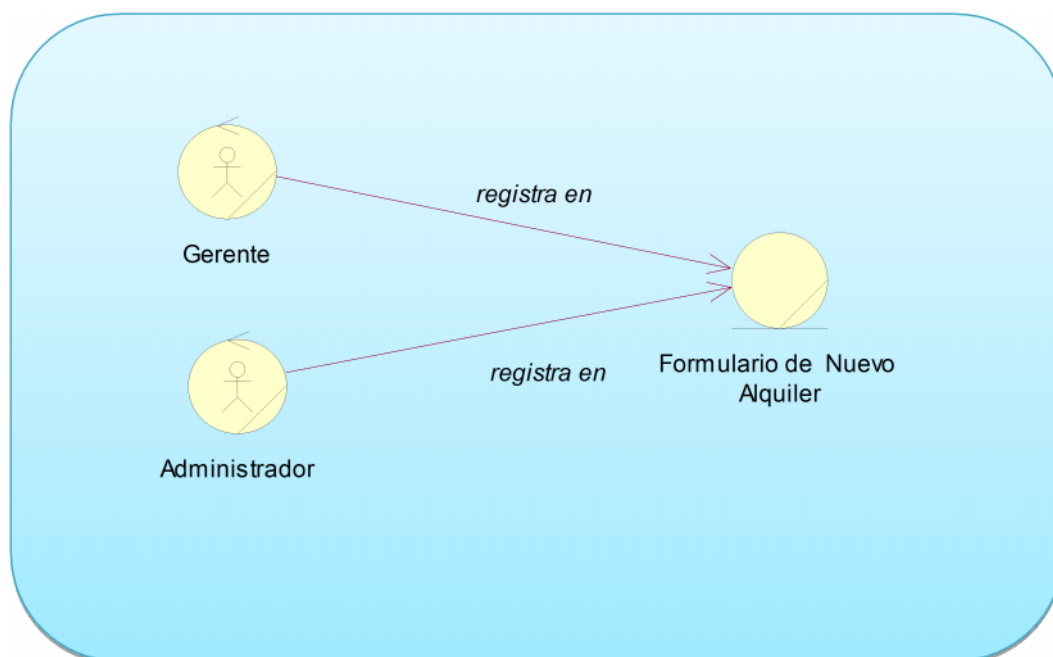


Figura 4: Registrar Cliente

#### Registrar alquiler



## Registrar Vestimenta



Figura 5: Registrar Vestimenta

## Registrar Proveedor

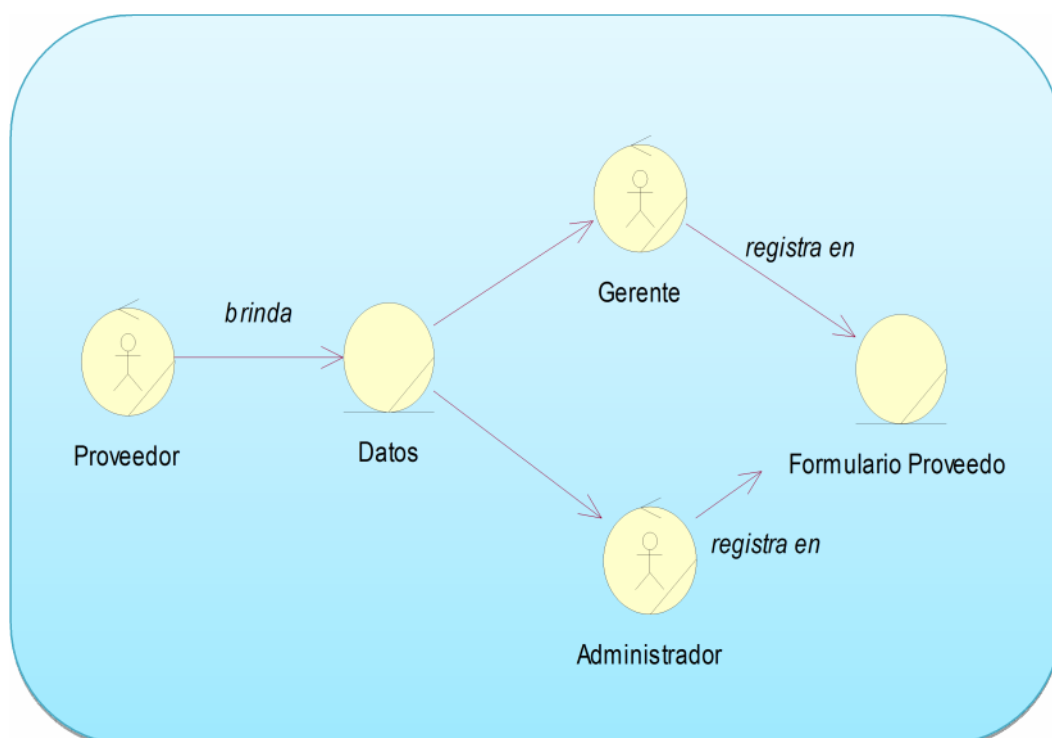


Figura 6: Registrar Proveedor



### Registrar Pedido de Proveedores



Figura 7: Registrar Proveedores

### Registrar Ingreso de Vestimentas a Tienda

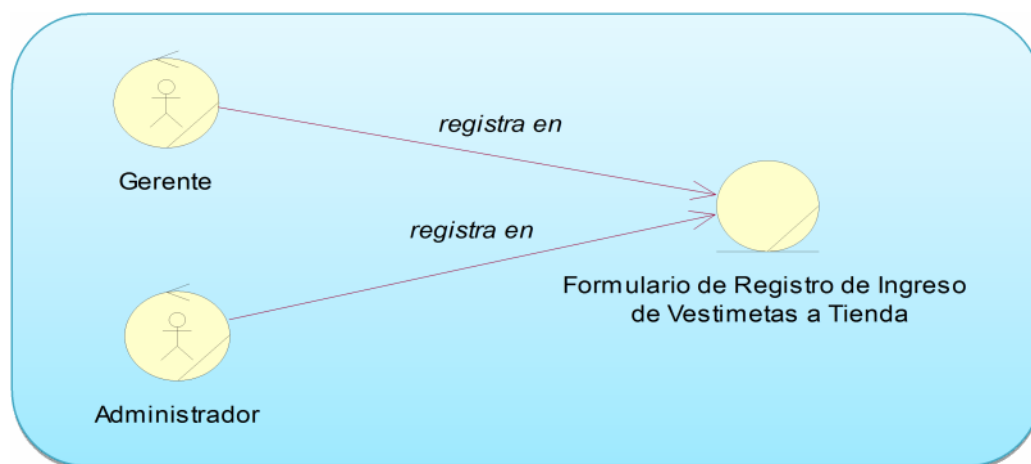
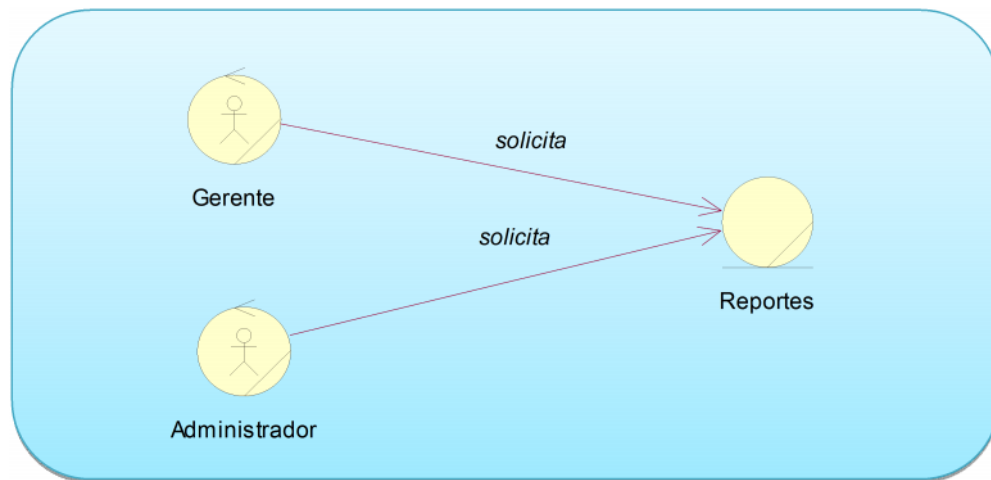


Figura 8: Registrar Ingreso de Vestimentas a Tiendas

## Reportes



**Figura 9: Reportes**

## **II.1.7 Glosario**

### **II.1.7.1 Introducción**

Este documento recoge términos manejados durante la elaboración del proyecto, consiste en un diccionario informal de datos y definiciones con los cuales se va a trabajar.

### **II.1.7.2 Propósito**

El propósito de este glosario es definir la terminología manejada en el proyecto de desarrollo de un sistema para el comercial "*Ballet Sin Fronteras*".

### **II.1.7.3 Alcance**

El alcance es que la terminología empleada en los alquileres, compras, inventarios de productos, proveedores y clientes se refleje con claridad en el documento.

### **II.1.7.4 Definiciones**

A continuación se presentan los términos manejados en el desarrollo del sistema.

- **Gerente**

Es la persona que se encarga de controlar que todas las funciones del comercial se lleven a cabo.

- **Administración**

Es la persona que se encarga de realizar los alquileres.

- **Cliente**

El cliente es una persona externa al comercial el cual puede realizar una compra.

- **Gestión de clientes**

Esta gestión de clientes lo puede hacer el administrador como también el vendedor, con el fin de obtener una referencia de los datos de los clientes del comercial.

## **II.1.8 Modelo de Casos de Uso**

### **II.1.8.1 Introducción**

El modelo de casos de uso es un artefacto de la disciplina requisitos en la metodología RUP, la cual estamos implementando.

### **II.1.8.2 Propósito**

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

### **II.1.8.3 Alcance**

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

### II.1.8.4 Diagrama de Casos de Uso General

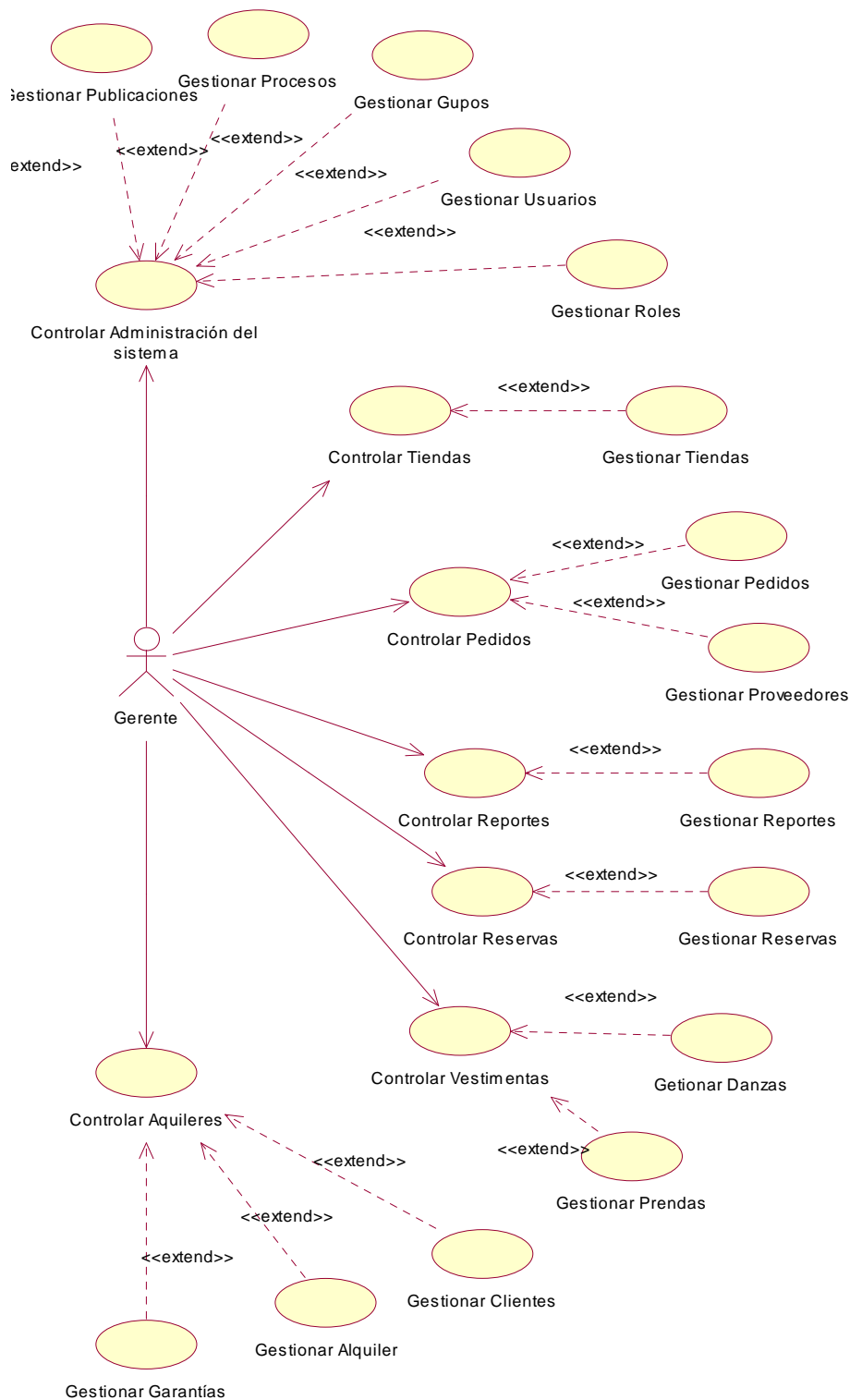
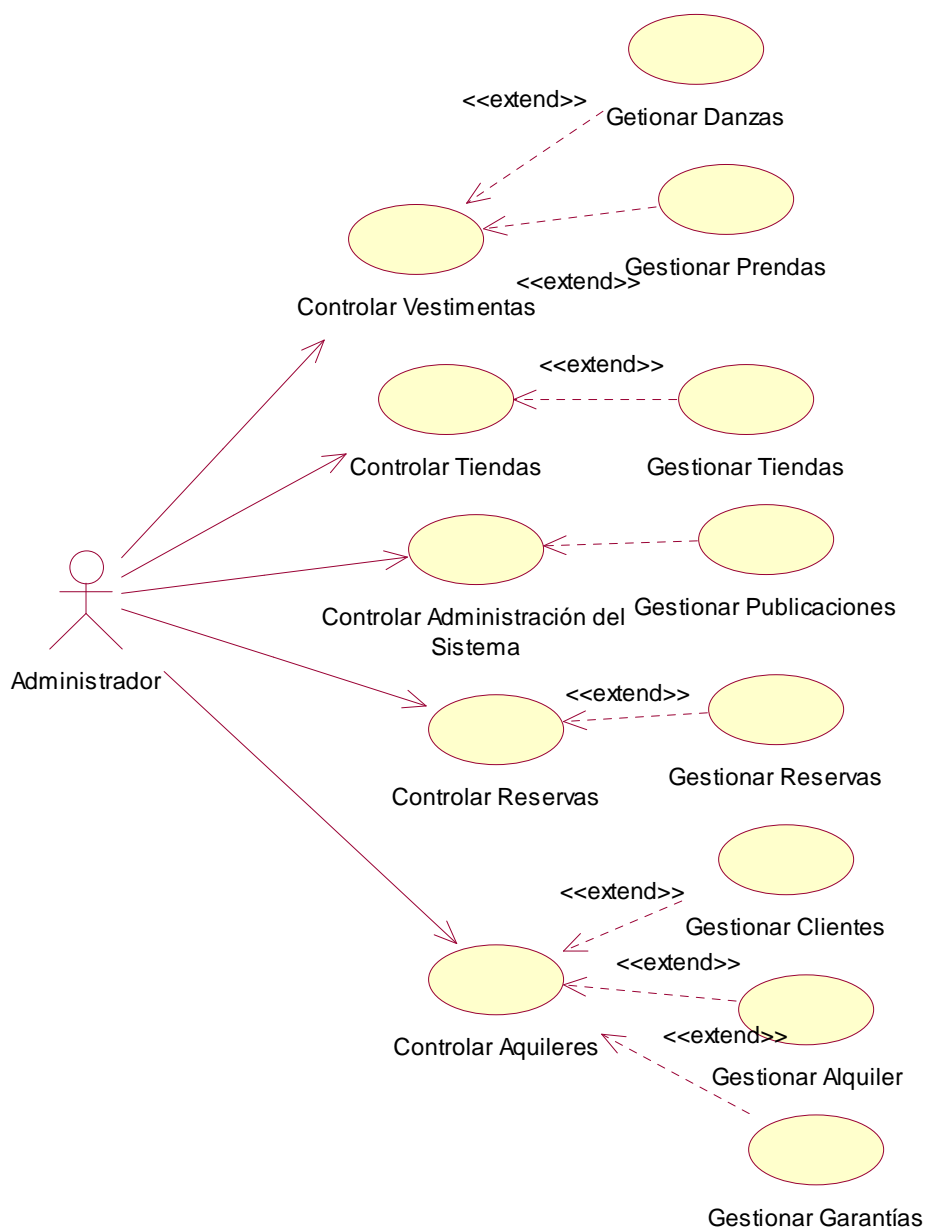


Figura 10: Diagrama de Casos de Uso: General Gerente



**Figura 11: Diagrama de Casos de Uso: General Administrador**

## **II.1.9 Especificaciones de Caso de Uso**

### **II.1.9.1 Introducción**

Las especificaciones de los Casos de Uso es una descripción detallada de los Casos de Uso del sistema.

### **II.1.9.2 Propósito**

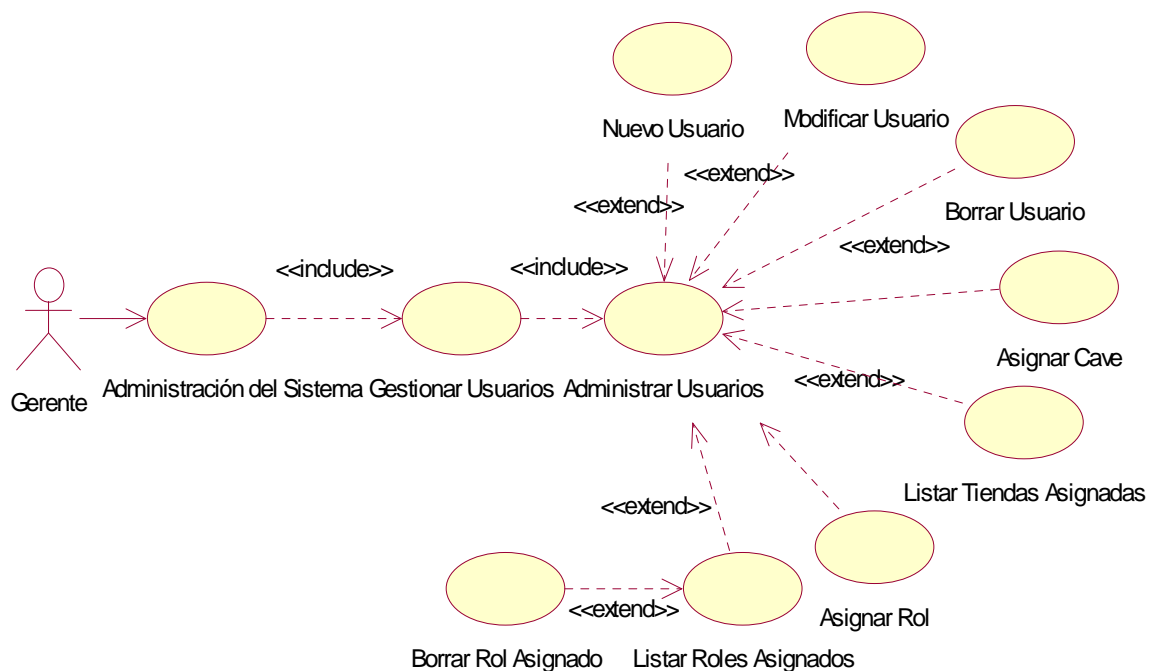
- Comprender los casos de uso del sistema.
- Describir específicamente cada caso de uso.

### **II.1.9.3 Alcance**

- Describe los procesos internos de los casos de uso.
- Detallar los flujos de cada caso de uso según lo establecido por la organización.

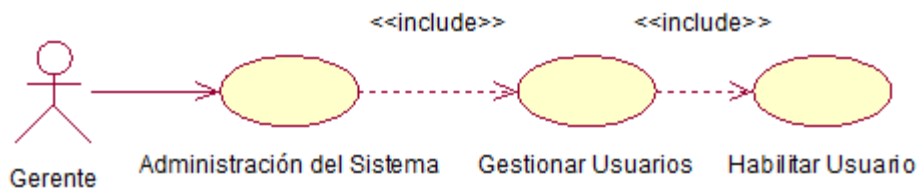
## Administración del Sistema:

### *Casos de uso: Administrar Usuarios*



**Figura 12: Diagrama de Casos de Uso: Administrar Usuarios**

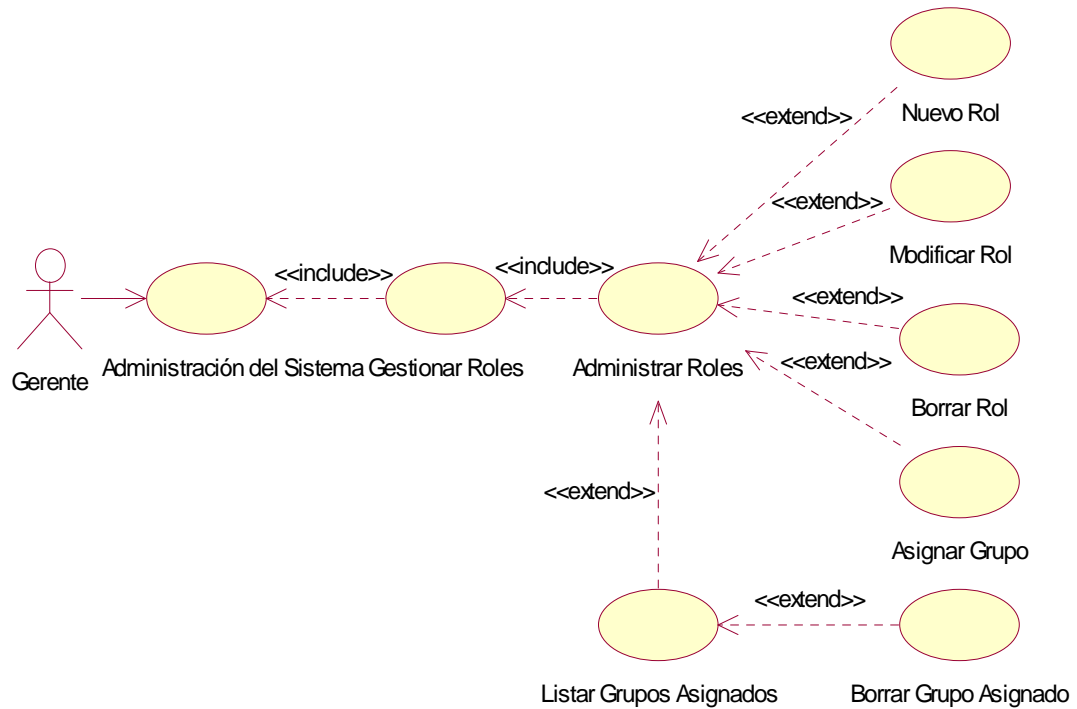
### *Casos de uso: Habilitar Usuarios*



**Figura 13: Diagrama de Casos de Uso: Habilitar Usuarios**

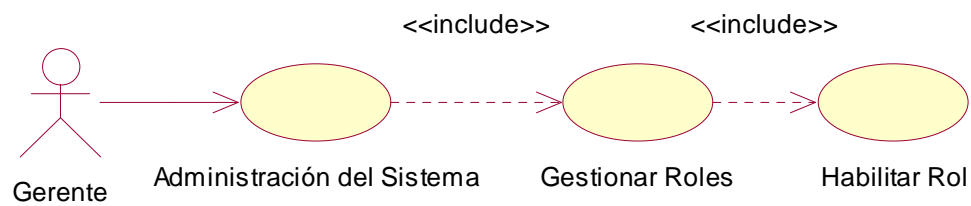


*Casos de uso: Administrar Roles*



**Figura 14: Diagrama de Casos de Uso: Administrar Roles**

*Casos de uso: Habilitar Roles*



**Figura 15: Diagrama de Casos de Uso: Habilitar Roles**

### Casos de uso: Administrar Grupos

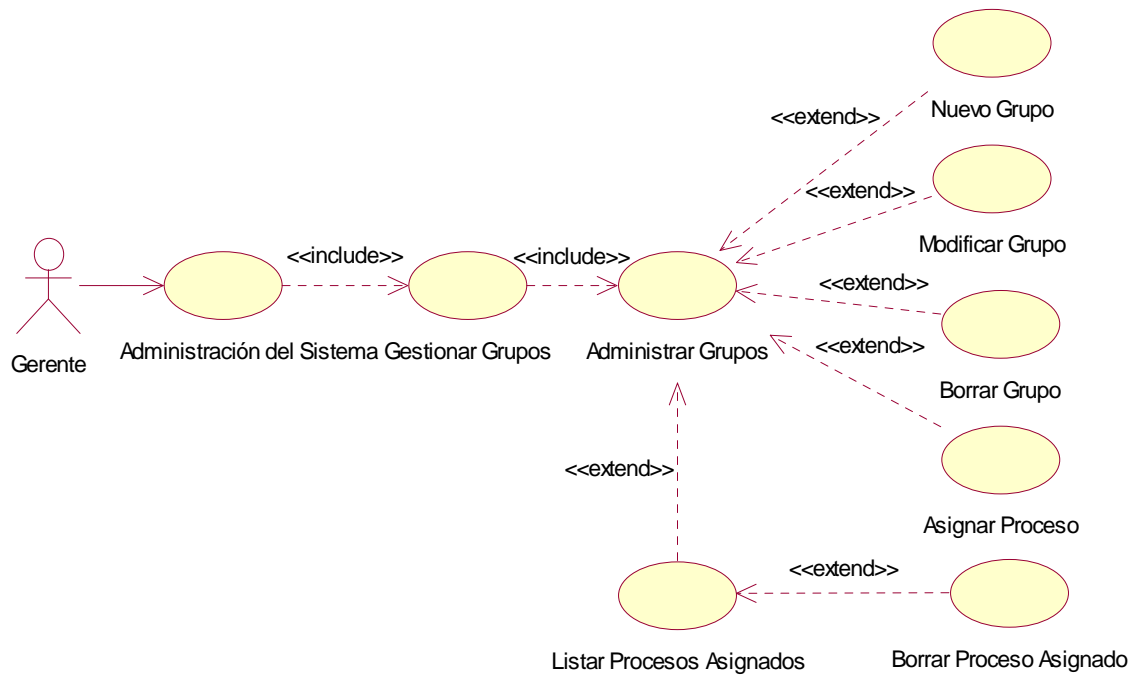


Figura 16: Diagrama de Casos de Uso: Administrar Grupos

### Casos de uso: Habilitar Grupos

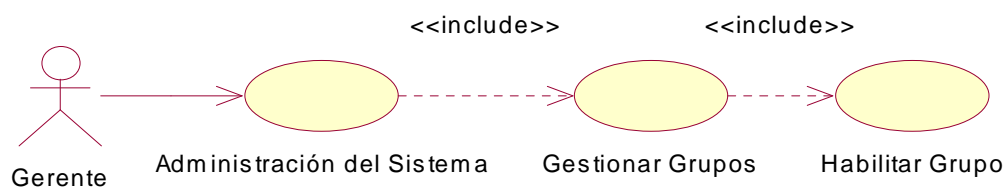
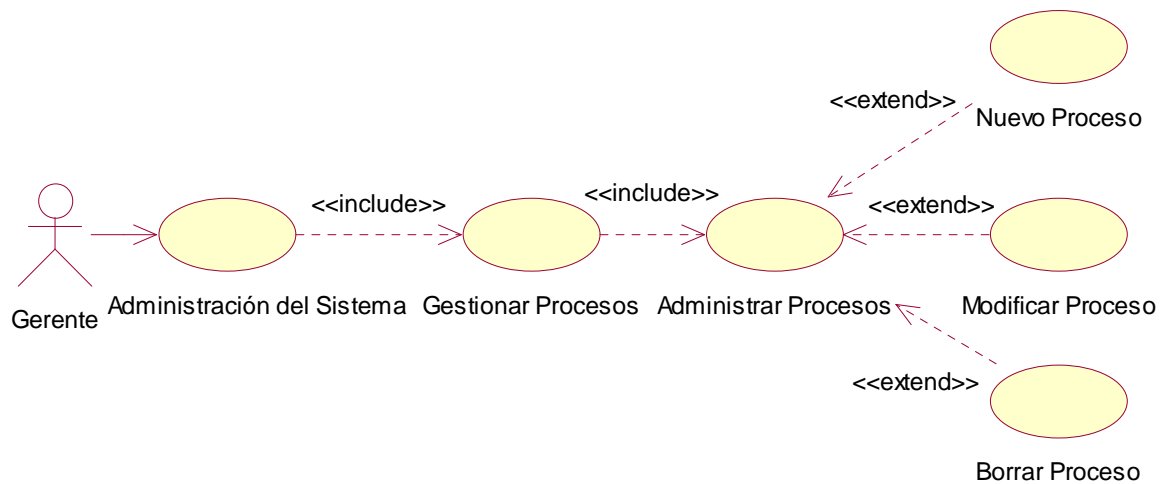


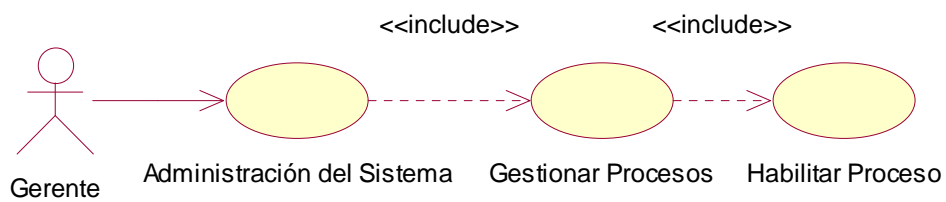
Figura 17: Diagrama de Casos de Uso: Habilitar Grupos

*Casos de uso: Administrar Procesos*



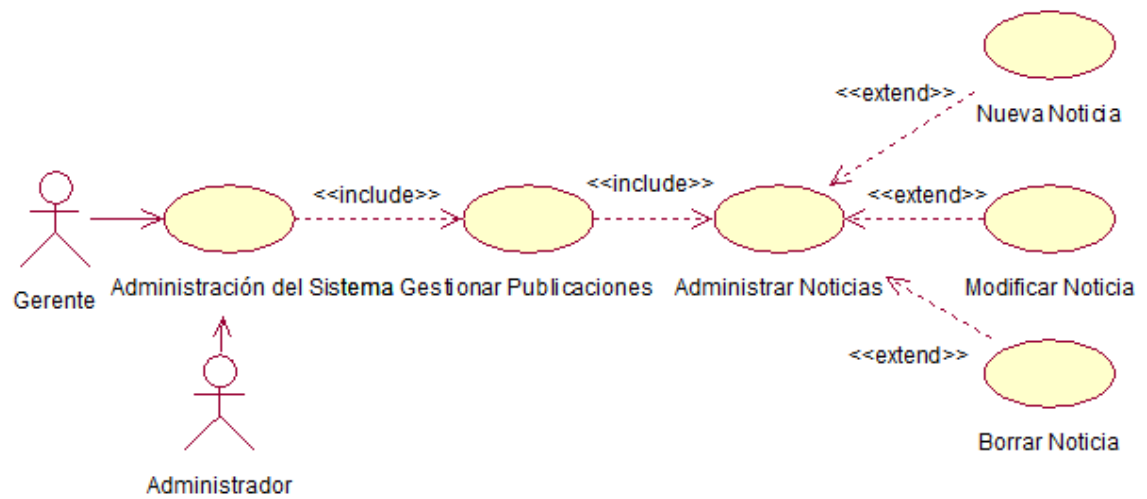
**Figura 18: Diagrama de Casos de Uso: Administrar Procesos**

*Casos de uso: Habilitar Procesos*



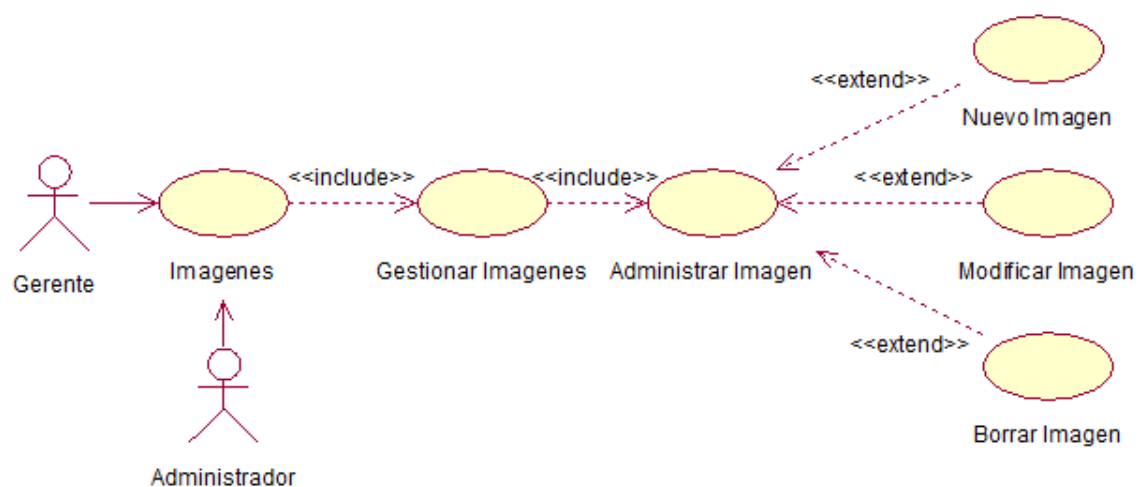
**Figura 19: Diagrama de Casos de Uso: Habilitar Procesos**

*Casos de uso: Administrar Noticias*



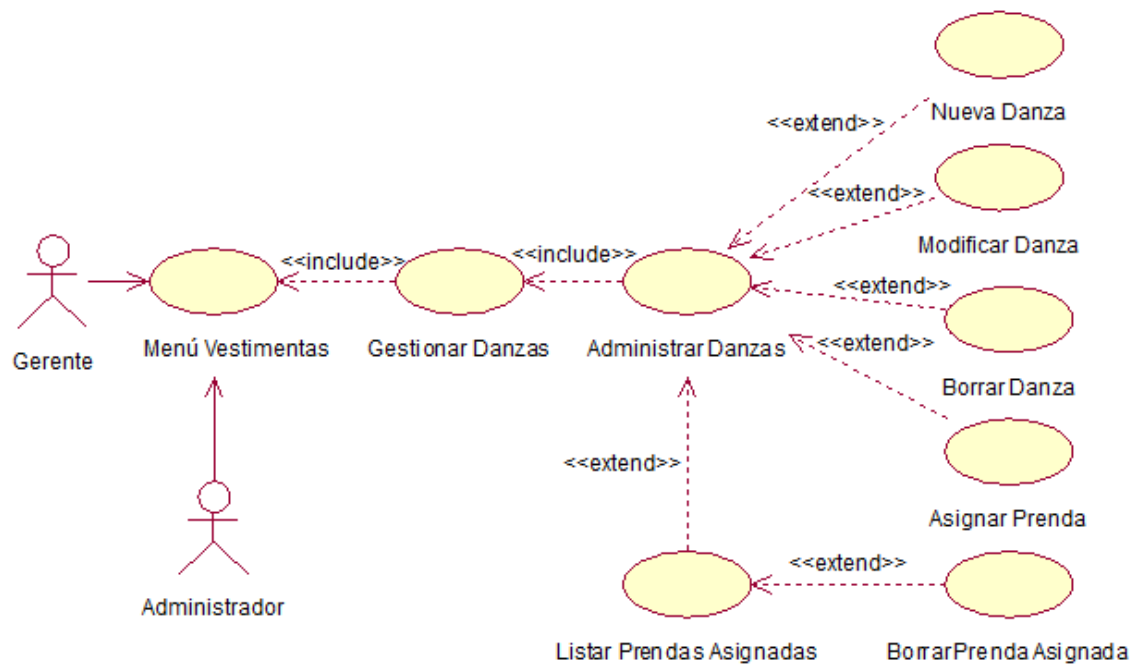
**Figura 20: Diagrama de Casos de Uso: Administrar Noticias**

*Casos de uso: Administrar Imágenes*



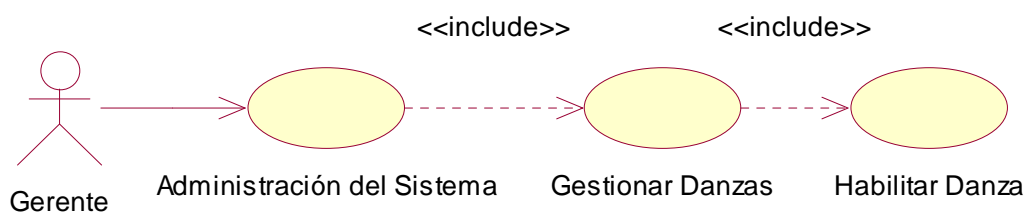
**Figura 21: Diagrama de Casos de Uso: Administrar Imágenes**

*Casos de uso: Administrar Danzas*



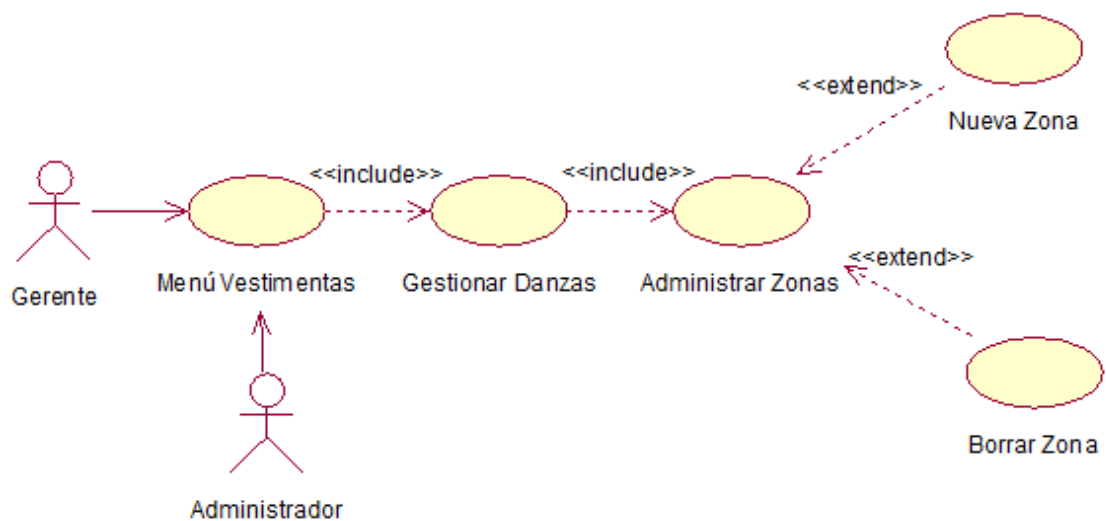
**Figura 12: Diagrama de Casos de Uso: Administrar Danzas**

*Casos de uso: Habilitar Danzas*



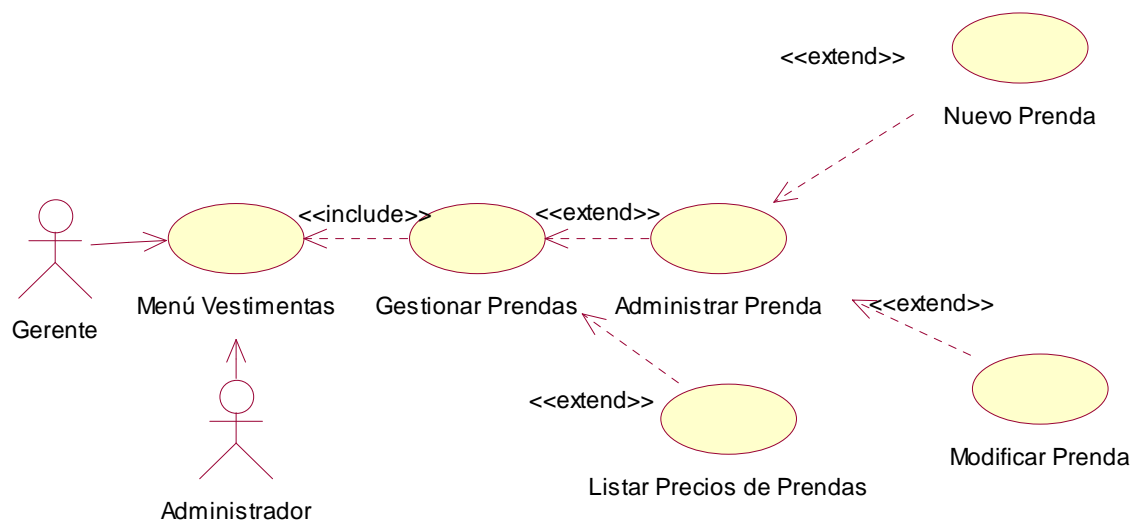
**Figura 22: Diagrama de Casos de Uso: Habilitar Danzas**

*Casos de uso: Administrar Zonas*



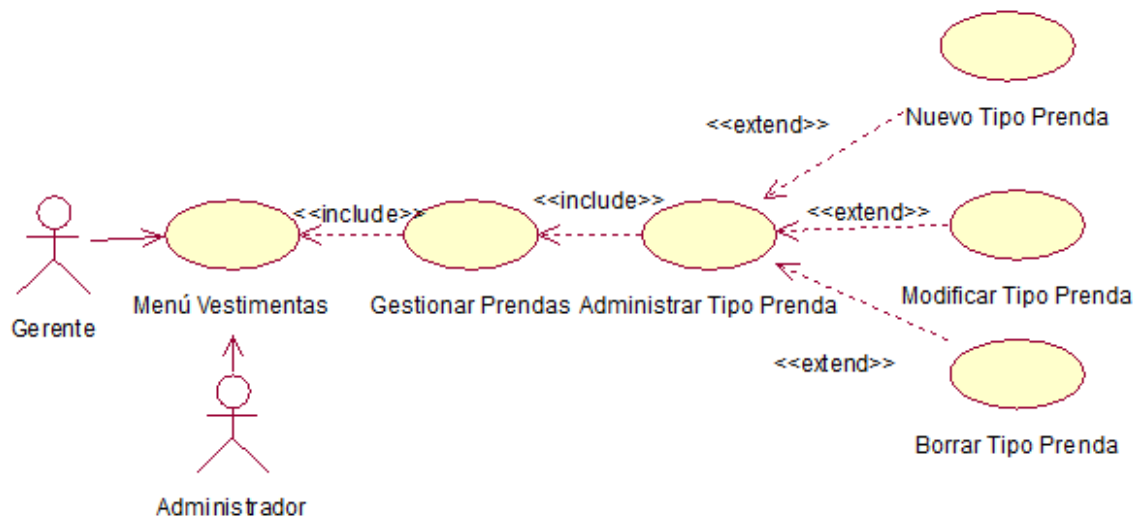
**Figura 23: Diagrama de Casos de Uso: Administrar Zonas**

*Casos de uso: Administrar Prendas*



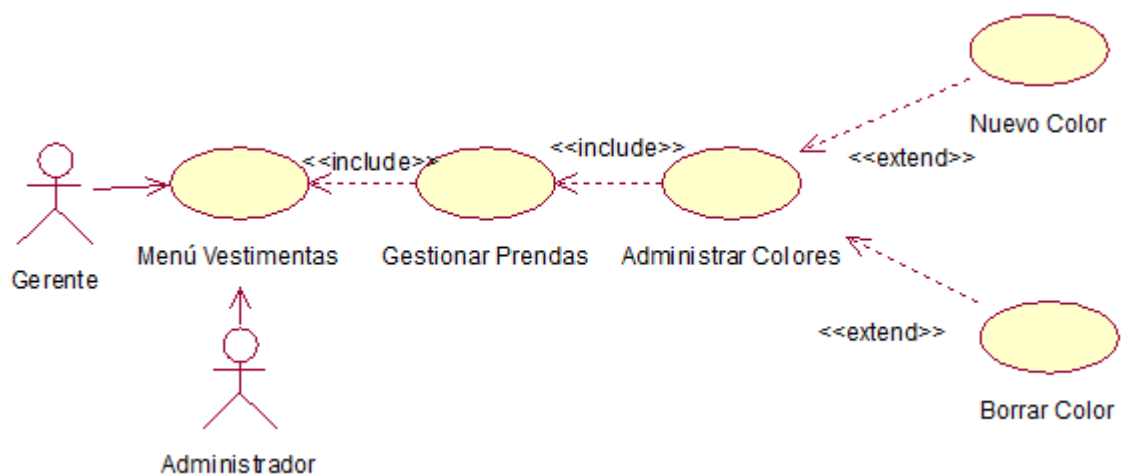
**Figura 24: Diagrama de Casos de Uso: Administrar Prendas**

*Casos de uso: Administrar Tipo Prenda*



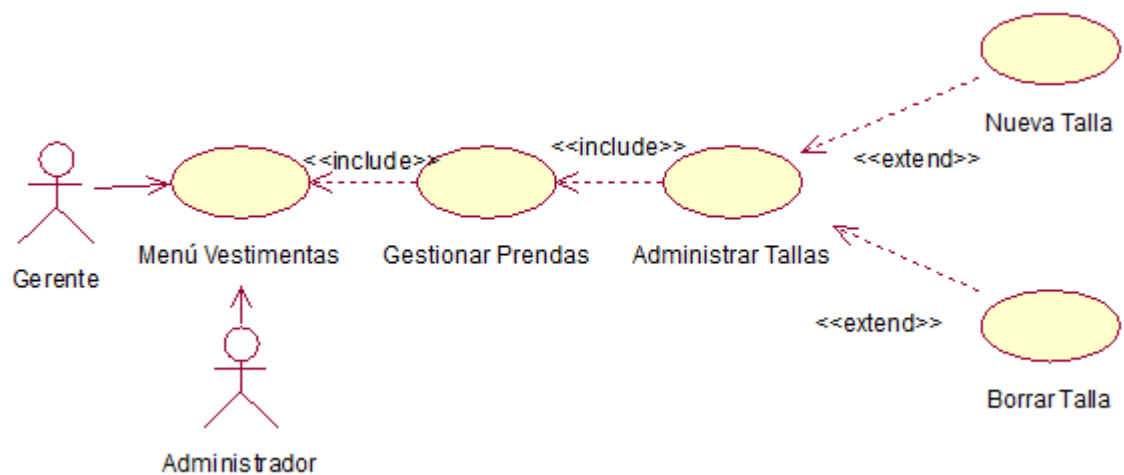
**Figura 25: Diagrama de Casos de Uso: Administrar Tipo Prenda**

*Casos de uso: Administrar Colores*



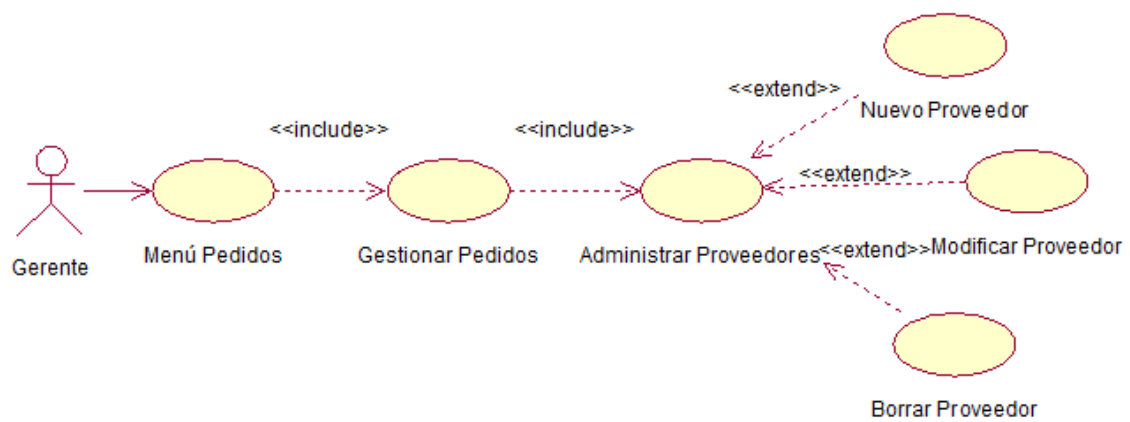
**Figura 26: Diagrama de Casos de Uso: Administrar Colores**

*Casos de uso: Administrar Tallas*



**Figura 27: Diagrama de Casos de Uso: Administrar Tallas**

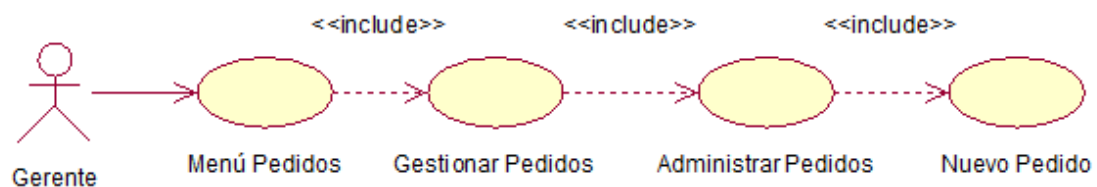
*Casos de uso: Administrar Proveedores*



**Figura 28: Diagrama de Casos de Uso: Administrar Proveedores**

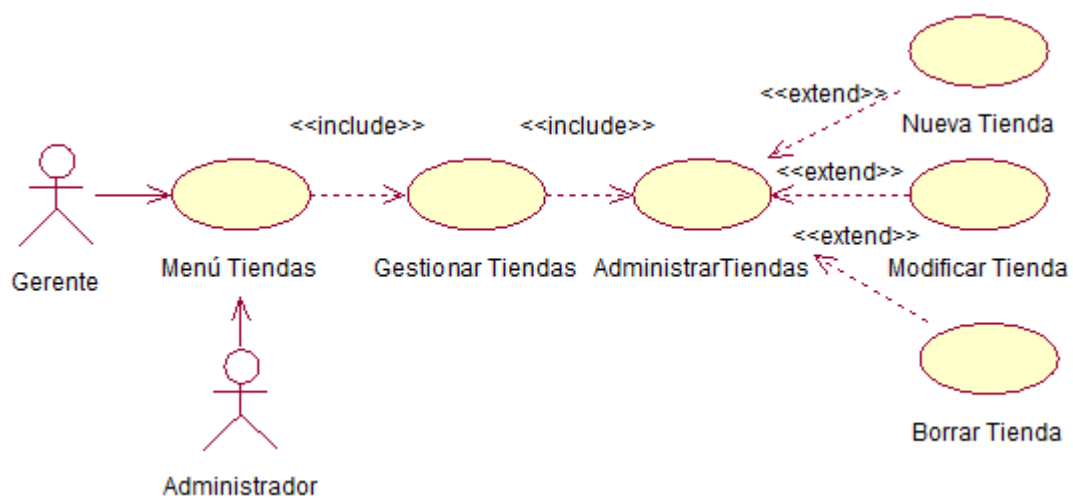


*Casos de uso: Administrar Pedidos*



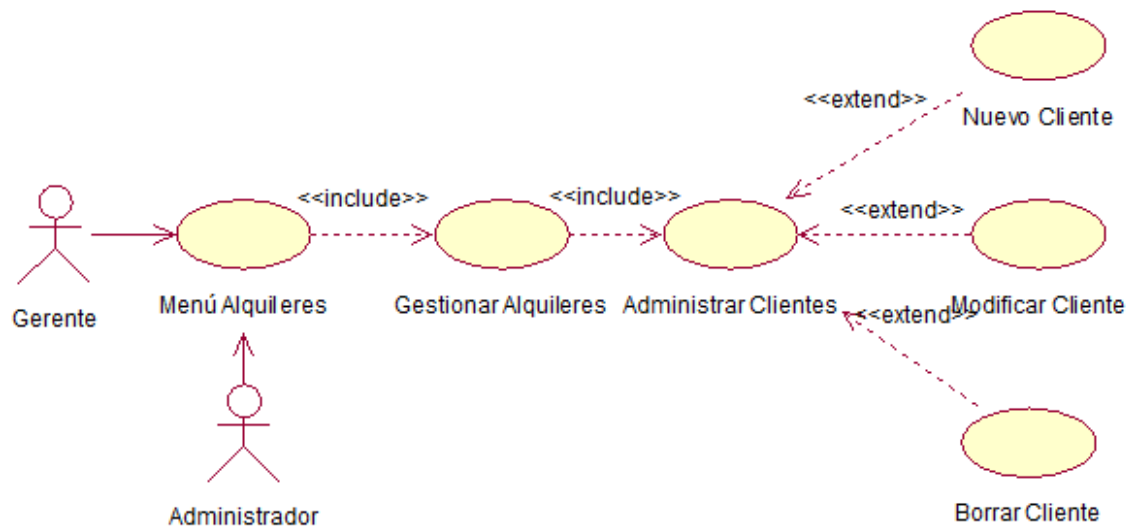
**Figura 30: Diagrama de Casos de Uso: Administrar Pedidos**

*Casos de uso: Administrar Tiendas*



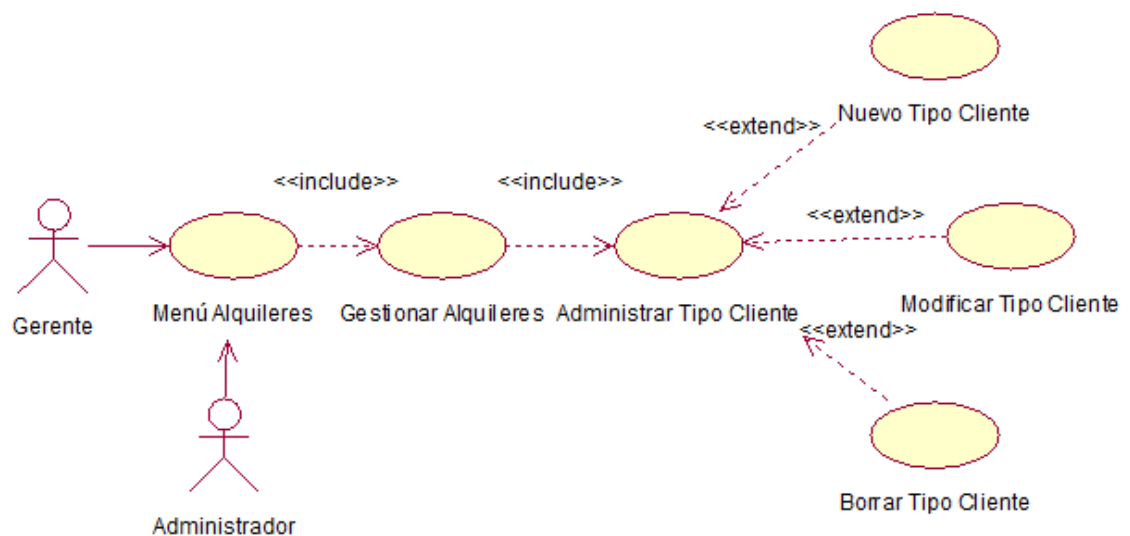
**Figura 31: Diagrama de Casos de Uso: Administrar Tiendas**

*Casos de uso: Administrar Clientes*



**Figura 32; Diagrama de Casos de Uso: Administrar Clientes**

*Casos de uso: Administrar Tipo Cliente*



**Figura 33: Diagrama de Casos de Uso: Administrar Tipo Cliente**

### Casos de uso: Administrar Garantías

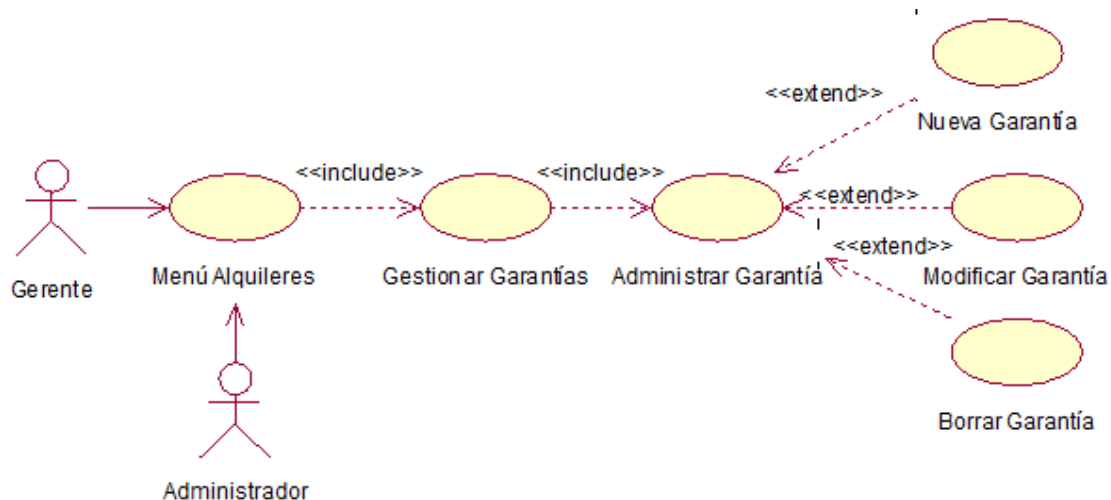


Figura 34: Diagrama de Casos de Uso: Administrar Garantías

### Casos de uso: Administrar Alquileres

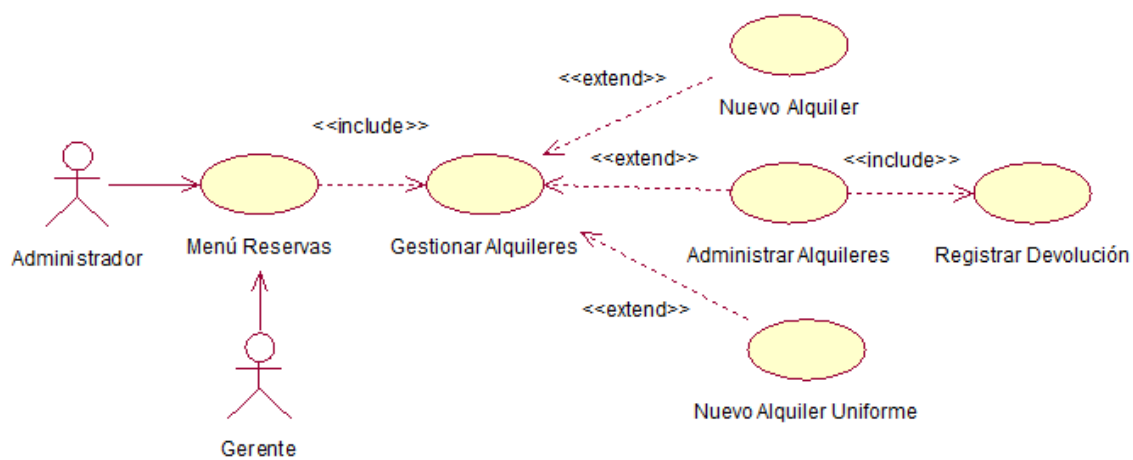
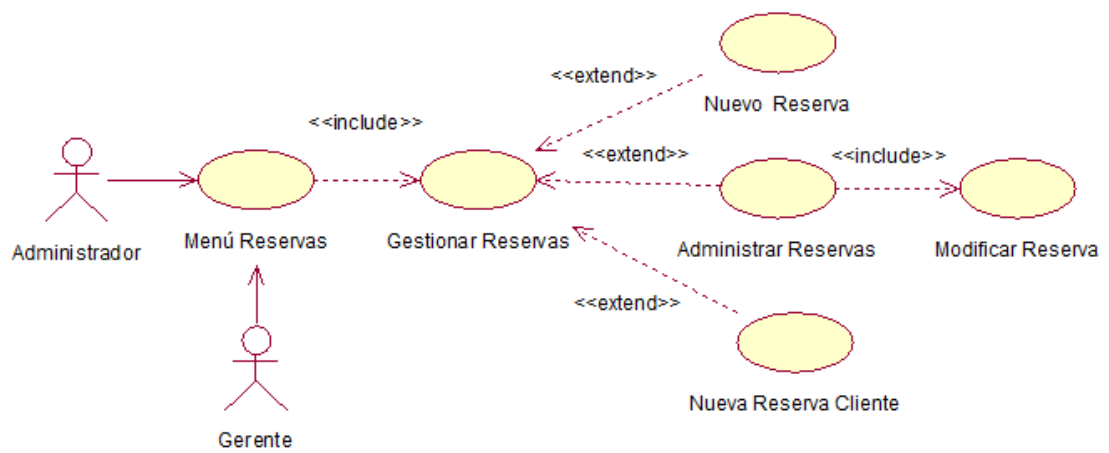


Figura 35: Diagrama de Casos de Uso: Administrar Alquileres

*Casos de uso: Administrar Reservas*



**Figura 36: Diagrama de Casos de Uso: Administrar Reservas**

### Casos de uso: Administrar Reportes

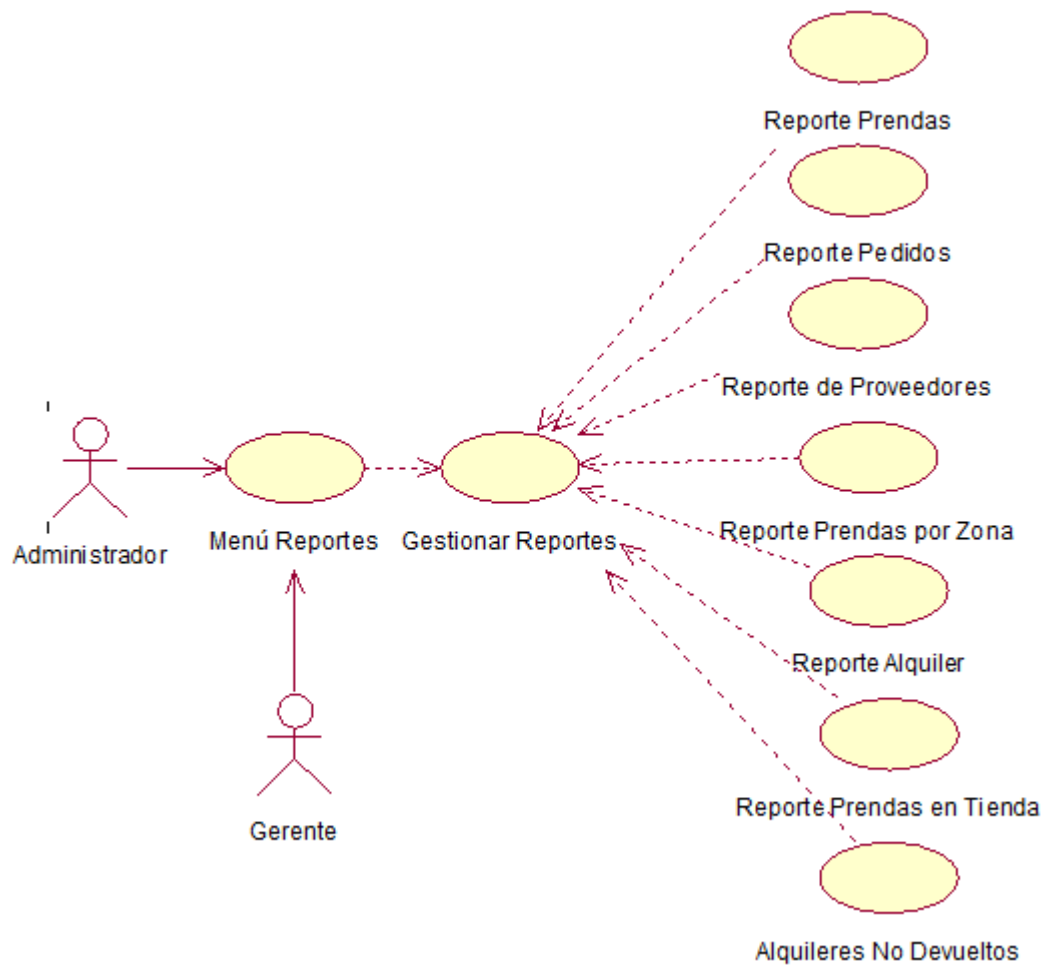


Figura 37: Diagrama de Casos de Uso: Administrar Reportes

### II.1.9.4 Especificaciones de Caso de Uso

**Actor:** Administrador

**Explicación:** En el caso Administrar usuarios, el administrador del sistema podrá tener un control de los datos de los usuarios del comercial, con la finalidad de que se pueda adicionar y modificar a dichos usuarios.

**Descripción:**

<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador selecciona administrar usuarios	
2. Se despliega la lista de usuarios registrados.	
3. El administrador selecciona una de las opciones de administrar usuarios.	4. El administrador cierra administrar usuarios.

**Tabla 20: Especificación caso de uso Administrar usuarios**

*Descripción: Nuevo Usuario*

<b>Caso de Uso:</b> Nuevo Usuario
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Usuario.
2. El sistema guarda los datos del Nuevo Usuario y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Usuario.
4. Fin del caso.

**Tabla 21: Especificación caso de uso: Nuevo Usuario***Descripción: Modificar Usuario*

<b>Caso de Uso:</b> Modificar Usuario
<b>Actor:</b> Gerente
1. El gerente selecciona el Usuario a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Usuario seleccionado.
3. El gerente Modifica datos del Usuario seleccionado.
4. El sistema guarda datos del Usuario Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Usuarios.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Usuario.
6. Fin del Caso.

**Tabla 22: Especificación caso de uso: Modificar Usuario**

**Descripción: Borrar Usuario**

<b>Caso de Uso:</b> Borrar Usuario
<b>Actor:</b> Gerente
1. El gerente selecciona el Usuario a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Usuario seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Usuarios.
4. El gerente selecciona la opción Recargar, observa la Lista de Usuarios actualizada.
5. Fin del Caso.

**Tabla 23: Especificación caso de uso: Borrar Usuario****Descripción: Habilitar Usuario**

<b>Caso de Uso:</b> Habilitar Usuario
<b>Actor:</b> Gerente
1. El gerente selecciona el Usuario a Habilitar y presiona el Botón Habilitar de la Lista de Usuarios No Habilitados.
2. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Usuarios No Habilitados.
3. El gerente selecciona Administrar Usuarios, observa la Lista de Usuarios actualizada con el Usuario Habilitado.
4. Fin del Caso.

**Tabla 24: Especificación caso de uso: Habilitar Usuario**



**Descripción: Asignar Rol a Usuario**

<b>Caso de Uso:</b> Asignar Rol a Usuario
<b>Actor:</b> Gerente
1. El gerente selecciona el Usuario a asignar.
2. El sistema devuelve una pantalla para Asignar datos del Rol.
3. El gerente Asigna datos del Rol.
4. El sistema guarda los datos del Rol a asignar y devuelve un mensaje de confirmación.
5. Fin del Caso.

**Tabla 25: Especificación caso de uso: Asignar Rol a Usuario****Descripción: Borrar Rol Asignado**

<b>Caso de Uso:</b> Borrar Rol Asignado
<b>Actor:</b> Gerente
1. El gerente selecciona el Usuario.
2. El sistema devuelve una pantalla con la lista de roles asignados a ese usuario.
3. El gerente selecciona el rol asignado a borrar.
4. El sistema actualiza los datos, devuelve un mensaje de confirmación.
5. Fin del Caso.

**Tabla 26: Especificación caso de uso: Borrar Rol Asignado**

*Descripción: Asignar Clave a Usuario*

<b>Caso de Uso:</b> Asignar Clave a Usuario
<b>Actor:</b> Administrador
1. El gerente selecciona el Usuario a asignar.
2. El sistema devuelve una pantalla para Asignar Clave.
3. El gerente introduce login y clave deseada.
4. El sistema guarda la clave y usuario y devuelve un mensaje de confirmación.
5. Fin del Caso.

**Tabla 27: Especificación caso de uso: Asignar Clave a Usuario***Descripción: Nuevo Rol*

<b>Caso de Uso:</b> Nuevo Rol
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Rol.
2. El sistema guarda los datos del Nuevo Rol y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Rol.
4. Fin del caso.

**Tabla 28: Especificación caso de uso: Nuevo Rol**

**Descripción: Modificar Rol**

<b>Caso de Uso:</b> Modificar Rol
<b>Actor:</b> Gerente
1. El gerente selecciona el Rol a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Rol seleccionado.
3. El gerente Modifica datos del Rol seleccionado.
4. El sistema guarda datos del Rol Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Rol.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Rol.
6. Fin del Caso.

**Tabla 29: Especificación caso de uso: Modificar Rol****Descripción: Borrar Rol**

<b>Caso de Uso:</b> Borrar Rol
<b>Actor:</b> Gerente
1. El gerente selecciona el Rol a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Rol seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Roles.
4. El gerente selecciona la opción Recargar, observa la Lista de Roles actualizada.
5. Fin del Caso.

**Tabla 30: Especificación caso de uso: Borrar Rol**

**Descripción: Habilitar Rol**

<b>Caso de Uso:</b> Habilitar Rol
<b>Actor:</b> Gerente
1. El gerente selecciona el Usuario a Habilitar y presiona el Botón Habilitar de la Lista de Roles No Habilitados.
2. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Roles No Habilitados.
3. El gerente selecciona Administrar Roles, observa la Lista de Roles actualizada con el Rol Habilitado.
4. Fin del Caso.

**Tabla 31: Especificación caso de uso: Habilitar Rol****Descripción: Asignar Grupo a Rol**

<b>Caso de Uso:</b> Asignar Grupo a Rol
<b>Actor:</b> Gerente
1. El gerente selecciona el Grupo a asignar.
2. El sistema devuelve una pantalla para Asignar datos del Grupo.
3. El gerente Asigna datos del Grupo.
4. El sistema guarda los datos del Grupo a asignar y devuelve un mensaje de confirmación.
5. Fin del Caso.

**Tabla 32: Especificación caso de uso: Asignar Grupo a Rol**

*Descripción: Nuevo Grupo*

<b>Caso de Uso:</b> Nuevo Grupo
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Grupo.
2. El sistema guarda los datos del Nuevo Grupo y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Grupo.
4. Fin del caso.

**Tabla 33: Especificación caso de uso: Nuevo Grupo***Descripción: Modificar Grupo*

<b>Caso de Uso:</b> Modificar Grupos
<b>Actor:</b> Gerente
1. El gerente selecciona el Grupo a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Grupo seleccionado.
3. El gerente Modifica datos del Grupo seleccionado.
4. El sistema guarda datos del Grupo Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Grupos.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Grupo.
6. Fin del Caso.

**Tabla 34: Especificación caso de uso: Modificar Grupo**

**Descripción: Borrar Grupo**

<b>Caso de Uso:</b> Borrar Grupo
<b>Actor:</b> Gerente
1. El gerente selecciona el Grupo a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Grupo seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Grupo.
4. El gerente selecciona la opción Recargar, observa la Lista de Grupos actualizada.
5. Fin del Caso.

**Tabla 35: Especificación caso de uso: Borrar Grupo****Descripción: Habilitar Grupo**

<b>Caso de Uso:</b> Habilitar Grupo
<b>Actor:</b> Gerente
1. El gerente selecciona el Grupo a Habilitar y presiona el Botón Habilitar de la Lista de Grupos No Habilitados.
2. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Grupos No Habilitados.
3. El gerente selecciona Administrar Grupos, observa la Lista de Grupos actualizada con el Grupo Habilitado.
4. Fin del Caso.

**Tabla 36: Especificación caso de uso: Habilitar Grupo**

*Descripción: Asignar Proceso a Grupo*

<b>Caso de Uso:</b> Asignar Proceso a Grupo
<b>Actor:</b> Administrador
1. El gerente selecciona el Proceso a asignar.
2. El sistema devuelve una pantalla para Asignar datos del Proceso.
3. El gerente Asigna datos del Proceso.
4. El sistema guarda los datos del Proceso a asignar y devuelve un mensaje de confirmación.
5. Fin del Caso.

**Tabla 37: Especificación caso de uso: Asignar Proceso a Grupo***Descripción: Nuevo Proceso*

<b>Caso de Uso:</b> Nuevo Proceso
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Proceso.
2. El sistema guarda los datos del Nuevo Proceso y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Proceso.
4. Fin del caso

**Tabla 38: Especificación caso de uso: Nuevo Proceso**

**Descripción: Modificar Proceso**

<b>Caso de Uso:</b> Modificar Proceso
<b>Actor:</b> Gerente
1. El gerente selecciona el Proceso a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Proceso seleccionado.
3. El gerente Modifica datos del Proceso seleccionado.
4. El sistema guarda datos del Proceso Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Proceso.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Proceso.
6. Fin del Caso.

**Tabla 39: Especificación caso de uso: Modificar Proceso****Descripción: Borrar Proceso**

<b>Caso de Uso:</b> Borrar Proceso
<b>Actor:</b> Gerente
1. El gerente selecciona el Proceso a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Proceso seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Proceso.
4. El gerente selecciona la opción Recargar, observa la Lista de Procesos actualizada.
5. Fin del Caso.

**Tabla 40: Especificación caso de uso: Borrar Proceso**



*Descripción: Nueva danza*

<b>Caso de Uso:</b> Nueva Danza
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Danza.
2. El sistema guarda los datos de la Nueva Danza y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Danza.
4. Fin del caso.

**Tabla 41: Especificación caso de uso: Nueva danza***Descripción: Modificar Danza*

<b>Caso de Uso:</b> Modificar Danza
<b>Actor:</b> Gerente
1. El gerente selecciona la Danza a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Danza seleccionada.
3. El gerente Modifica datos de la Danza seleccionada.
4. El sistema guarda datos de la Danza Modificada y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Danzas.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados de la Danza.
6. Fin del Caso.

**Tabla 42: Especificación caso de uso: Nueva danza**

**Descripción: Borrar Danza**

<b>Caso de Uso:</b> Borrar Danza
<b>Actor:</b> Gerente
1. El gerente selecciona la Danza a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Danza seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Danzas.
4. El gerente selecciona la opción Recargar, observa la Lista de Danzas actualizada.
5. Fin del Caso.

**Tabla 43: Especificación caso de uso: Borrar Danza****Descripción: Habilitar Danza**

<b>Caso de Uso:</b> Habilitar Danza
<b>Actor:</b> Gerente
1. El gerente selecciona la Danza a Habilitar y presiona el Botón Habilitar de la Lista de Danzas No Habilitados.
2. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Danza No Habilitados.
3. El gerente selecciona Administrar Danza, observa la Lista de Danza actualizada con la Danza Habilitada.
4. Fin del Caso.

**Tabla 44: Especificación caso de uso: Habilitar Danza**

*Descripción: Asignar Prenda a Danza*

<b>Caso de Uso:</b> Asignar Prenda a Danza
<b>Actor:</b> Administrador
1. El gerente selecciona la Prenda a asignar.
2. El sistema devuelve una pantalla para Asignar datos de la Prenda.
3. El gerente Asigna datos de la Prenda.
4. El sistema guarda los datos de la Prenda a asignar y devuelve un mensaje de confirmación.
5. Fin del Caso.

**Tabla 45: Especificación caso de uso: Asignar Prenda a Danza***Descripción: Nueva Zona*

<b>Caso de Uso:</b> Nueva Zona
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Zona.
2. El sistema guarda los datos de la Nueva Zona y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Zona.
4. Fin del caso.

**Tabla 46: Especificación caso de uso: Nueva Zona**

**Descripción: Borrar Zona**

<b>Caso de Uso:</b> Borrar Zona
<b>Actor:</b> Gerente
1. El gerente selecciona la Zona a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Zona seleccionada.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Zonas.
4. El gerente selecciona la opción Recargar, observa la Lista de Zonas actualizada.
5. Fin del Caso.

**Tabla 47: Especificación caso de uso: Borrar Zona****Descripción: Nueva Prenda**

<b>Caso de Uso:</b> Nueva Prenda
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Prenda.
2. El sistema guarda los datos de la Nueva Prenda y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Prenda.
4. Fin del caso.

**Tabla 48: Especificación caso de uso: Nueva Prenda**

***Descripción: Modificar Prenda***

<b>Caso de Uso:</b> Modificar Prenda
<b>Actor:</b> Gerente
1. El gerente selecciona la Prenda a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Prenda seleccionada.
3. El gerente Modifica datos de la Prenda seleccionada.
4. El sistema guarda datos de la Prenda Modificada y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Prendas.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados de la Prenda.
6. Fin del Caso.

**Tabla 49: Especificación caso de uso: Modificar Prenda*****Descripción: Nuevo Tipo de Prenda***

<b>Caso de Uso:</b> Nuevo Tipo de Prenda
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Tipo de Prenda.
2. El sistema guarda los datos del Nuevo Tipo de Prenda y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Tipo de Prenda.
4. Fin del caso.

**Tabla 50: Especificación caso de uso: Nuevo Tipo de Prenda**

**Descripción: Modificar Tipo de Prenda**

<b>Caso de Uso:</b> Modificar Tipo de Prenda
<b>Actor:</b> Gerente
1. El gerente selecciona el Tipo de Prenda a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Tipo de Prenda seleccionado.
3. El gerente Modifica datos del Tipo de Prenda seleccionado.
4. El sistema guarda datos del Tipo de Prenda Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tipo de Prenda.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Tipo de Prenda.
6. Fin del caso

**Tabla 51: Especificación caso de uso: Modificar Tipo de Prenda****Descripción: Borrar Tipo de Prenda**

<b>Caso de Uso:</b> Borrar Tipo de Prenda
<b>Actor:</b> Gerente
1. El gerente selecciona el Tipo de Prenda a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Tipo de Prenda seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tipo de Prendas.
4. El gerente selecciona la opción Recargar, observa la Lista de Tipo Prendas actualizada.
5. Fin del Caso.

**Tabla 52: Especificación caso de uso: Borrar Tipo de Prenda**

**Descripción: Nuevo Color**

<b>Caso de Uso:</b> Nuevo Color
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Color.
2. El sistema guarda los datos del Nuevo Color y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Color.
4. Fin del caso.

**Tabla 53: Especificación caso de uso: Nuevo Color****Descripción: Borrar Color**

<b>Caso de Uso:</b> Borrar Color
<b>Actor:</b> Gerente
1. El gerente selecciona el Color a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Color seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Colores.
4. El gerente selecciona la opción Recargar, observa la Lista de Colores actualizada.
5. Fin del Caso.

**Tabla 54: Especificación caso de uso: Borrar Color**

**Descripción: Nueva Talla**

<b>Caso de Uso:</b> Nueva Talla
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Talla.
2. El sistema guarda los datos de la Nueva Talla y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Talla.
4. Fin del caso.

**Tabla 55: Especificación caso de uso: Nueva Talla****Descripción: Borrar Talla**

<b>Caso de Uso:</b> Borrar Talla
<b>Actor:</b> Gerente
1. El gerente selecciona la Talla a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Talla seleccionada.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tallas.
4. El gerente selecciona la opción Recargar, observa la Lista de Tallas actualizada.
5. Fin del Caso.

**Tabla 56: Especificación caso de uso: Borrar Talla**



***Descripción: Modificar Precio***

<b>Caso de Uso:</b> Modificar Precio
<b>Actor:</b> Gerente
1. El gerente selecciona el Precio a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Precio seleccionado.
3. El gerente Modifica datos del Precio seleccionado.
4. El sistema guarda datos del Precio Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Precios.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Precio.
6. Fin del Caso.

**Tabla 57: Especificación caso de uso: Modificar Precio*****Descripción: Nuevo Proveedor***

<b>Caso de Uso:</b> Nuevo Proveedor
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Proveedor.
2. El sistema guarda los datos del Nuevo Proveedor y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Proveedor.
4. Fin del caso.

**Tabla 58: Especificación caso de uso: Nuevo Proveedor**

**Descripción: Modificar Proveedor**

<b>Caso de Uso:</b> Modificar Proveedor
<b>Actor:</b> Gerente
1. El gerente selecciona el Proveedor a Modificar y presiona el Botón Modificar.
2. El sistema devuelve una pantalla para Modificar datos del Proveedor seleccionado.
3. El gerente Modifica datos del Proveedor seleccionado.
4. El sistema guarda datos del Proveedor Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Proveedor.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Proveedor.
6. Fin del Caso.

**Tabla 59: Especificación caso de uso: Modificar Proveedor****Descripción: Borrar Proveedor**

<b>Caso de Uso:</b> Borrar Proveedor
<b>Actor:</b> Gerente
1. El gerente selecciona el Proveedor a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Proveedor seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Proveedores.
4. El gerente selecciona la opción Recargar, observa la Lista de Proveedores actualizada.
5. Fin del Caso.

**Tabla 60: Especificación caso de uso: Borrar Proveedor**

***Descripción: Nuevo Pedido***

<b>Caso de Uso:</b> Nuevo Pedido
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Pedido.
2. El sistema guarda los datos del Nuevo Pedido y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Pedido.
4. Fin del caso.

**Tabla 61: Especificación caso de uso: Nuevo Pedido*****Descripción: Nueva Tienda***

<b>Caso de Uso:</b> Nueva Tienda
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Tienda.
2. El sistema guarda los datos de la Nueva Tienda y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Tienda.
4. Fin del caso.

**Tabla 62: Especificación caso de uso: Nueva Tienda**

**Descripción: Modificar Tienda**

<b>Caso de Uso:</b> Modificar Tienda
<b>Actor:</b> Gerente
1. El gerente selecciona la Tienda a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Tienda seleccionada.
3. El gerente Modifica datos de la Tienda seleccionada.
4. El sistema guarda datos de la Tienda Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tiendas.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Tiendas.
6. Fin del Caso.

**Tabla 63: Especificación caso de uso: Modificar Tienda****Descripción: Borrar Tienda**

<b>Caso de Uso:</b> Borrar Tienda
<b>Actor:</b> Gerente
1. El gerente selecciona la Tienda a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Tienda seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tiendas.
4. El gerente selecciona la opción Recargar, observa la Lista de Tiendas actualizada.
5. Fin del Caso.

**Tabla 64: Especificación caso de uso: Borrar Tienda**

***Descripción: Nueva Noticia***

<b>Caso de Uso:</b> Nueva Noticia
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Noticia.
2. El sistema guarda los datos de la Nueva Noticia y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Noticia.
4. Fin del caso.

**Tabla 65: Especificación caso de uso: Nueva Noticia*****Descripción: Modificar Noticia***

<b>Caso de Uso:</b> Modificar Noticia
<b>Actor:</b> Gerente
1. El gerente selecciona la Noticia a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Noticia seleccionada.
3. El gerente Modifica datos de la Noticia seleccionada.
4. El sistema guarda datos de la Noticia Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Noticia.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados de la Noticia.
6. Fin del Caso.

**Tabla 66: Especificación caso de uso: Modificar Noticia**

**Descripción: Borrar Noticia**

<b>Caso de Uso:</b> Borrar Noticia
<b>Actor:</b> Gerente
1. El gerente selecciona la Noticia a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Noticia seleccionada.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Noticias.
4. El gerente selecciona la opción Recargar, observa la Lista de Noticias actualizada.
5. Fin del Caso.

**Tabla 67: Especificación caso de uso: Borrar Noticia****Descripción: Nueva Imagen**

<b>Caso de Uso:</b> Nueva Imagen
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Imagen.
2. El sistema guarda los datos de la Nueva Imagen y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Imagen.
4. Fin del caso.

**Tabla 68: Especificación caso de uso: Nueva Imagen**

**Descripción: Modificar Imagen**

<b>Caso de Uso:</b> Modificar Imagen
<b>Actor:</b> Gerente
1. El gerente selecciona la Imagen a Modificar, presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Imagen seleccionada.
3. El gerente Modifica datos de la Imagen seleccionada.
4. El sistema guarda datos de la Imagen Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Imagen.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados de la Imagen.
6. Fin del Caso.

**Tabla 69: Especificación caso de uso: Modificar Imagen****Descripción: Borrar Imagen**

<b>Caso de Uso:</b> Borrar Imagen
<b>Actor:</b> Gerente
1. El gerente selecciona la Imagen a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Imagen seleccionada.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Imágenes.
4. El gerente selecciona la opción Recargar, observa la Lista de Imágenes actualizada.
5. Fin del Caso.

**Tabla 70: Especificación caso de uso: Borrar Imagen**

*Descripción: Nuevo Cliente*

<b>Caso de Uso:</b> Nuevo Cliente
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Cliente.
2. El sistema guarda los datos del Nuevo Cliente y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Cliente
4. Fin del caso.

**Tabla 71: Especificación caso de uso: Nuevo Cliente***Descripción: Modificar Cliente*

<b>Caso de Uso:</b> Modificar Cliente
<b>Actor:</b> Gerente
1. El gerente selecciona el Cliente a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Nuevo Cliente seleccionado.
3. El gerente Modifica datos del Cliente seleccionado.
4. El sistema guarda datos del Cliente Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Clientes.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Cliente.
6. Fin del Caso.

**Tabla 72: Especificación caso de uso: Modificar Cliente**



*Descripción: Borrar Cliente*

<b>Caso de Uso:</b> Borrar Cliente
<b>Actor:</b> Gerente
1. El gerente selecciona el Cliente a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Cliente seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Clientes.
4. El gerente selecciona la opción Recargar, observa la Lista de Clientes actualizada.
5. Fin del Caso.

**Tabla 73: Especificación caso de uso: Borrar Cliente***Descripción: Nuevo Tipo Cliente*

<b>Caso de Uso:</b> Nuevo Tipo Cliente
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Tipo de Cliente.
2. El sistema guarda los datos del Nuevo Tipo de Cliente y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Tipo de Cliente.
4. Fin del caso.

**Tabla 74: Especificación caso de uso: Nuevo Tipo Cliente**

*Descripción: Modificar Tipo de Cliente*

<b>Caso de Uso:</b> Modificar Tipo Cliente
<b>Actor:</b> Gerente
1. El gerente selecciona el Tipo Cliente a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos del Tipo de Cliente seleccionado.
3. El gerente Modifica datos del Tipo de Cliente seleccionado.
4. El sistema guarda datos del Tipo de Cliente Modificado y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tipo de Cliente.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados del Tipo de Cliente.
6. Fin del Caso.

**Tabla 75: Especificación caso de uso: Modificar Tipo de Cliente***Descripción: Borrar Tipo de Cliente*

<b>Caso de Uso:</b> Borrar Tipo de Cliente
<b>Actor:</b> Gerente
1. El gerente selecciona el Tipo de Cliente a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos del Tipo de Cliente seleccionado.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Tipo de Cliente.
4. El gerente selecciona la opción Recargar, observa la Lista de Tipo Cliente actualizada.
5. Fin del Caso.

**Tabla 76: Especificación caso de uso: Borrar Tipo de Cliente**

*Descripción: Nueva Garantía*

<b>Caso de Uso:</b> Nueva Garantía
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Garantía.
2. El sistema guarda los datos de la Nueva Garantía y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Garantía.
4. Fin del caso.

**Tabla 77: Especificación caso de uso: Nueva Garantía***Descripción: Modificar Garantía*

<b>Caso de Uso:</b> Modificar Garantía
<b>Actor:</b> Gerente
1. El gerente selecciona la Garantía a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Garantía seleccionada.
3. El gerente Modifica datos de la Garantía seleccionada.
4. El sistema guarda datos de la Garantía Modificada y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Garantía.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados de la garantía.
6. Fin del Caso.

**Tabla 78: Especificación caso de uso: Modificar Garantía**

**Descripción: Borrar Garantía**

<b>Caso de Uso:</b> Borrar Garantía
<b>Actor:</b> Gerente
1. El gerente selecciona la Garantía a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Garantía seleccionada.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Garantía.
4. El gerente selecciona la opción Recargar, observa la Lista de Garantías actualizada.
5. Fin del Caso.

**Tabla 79: Especificación caso de uso: Borrar Garantía****Descripción: Nuevo Alquiler**

<b>Caso de Uso:</b> Nuevo Alquiler
<b>Actor:</b> Gerente
1. El gerente ingresa los datos del Nuevo Alquiler.
2. El sistema guarda los datos del Nuevo Alquiler y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza el Nuevo Alquiler
4. Fin del caso.

**Tabla 80: Especificación caso de uso: Nuevo Alquiler**

**Descripción: Nueva Reserva**

<b>Caso de Uso:</b> Nueva Reserva
<b>Actor:</b> Gerente
1. El gerente ingresa los datos de la Nueva Reserva.
2. El sistema guarda los datos de la Nueva Reserva y devuelve un mensaje de confirmación.
3. El gerente observa la lista actualizada donde se visualiza la Nueva Reserva
4. Fin del caso.

**Tabla 81: Especificación caso de uso: Nueva Reserva****Descripción: Modificar Reservas**

<b>Caso de Uso:</b> Modificar Reservas
<b>Actor:</b> Gerente
1. El gerente selecciona la Reserva a Modificar y presiona el Botón modificar.
2. El sistema devuelve una pantalla para Modificar datos de la Reserva seleccionada.
3. El gerente Modifica datos de la Reserva seleccionada.
4. El sistema guarda datos de la Reserva Modificada y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Reserva.
5. El gerente selecciona la opción Recargar, observa la lista actualizada donde se visualiza los datos Modificados de la Reserva.
6. Fin del Caso.

**Tabla 82: Especificación caso de uso: Modificar Reservas**

**Descripción: Borrar Reserva**

<b>Caso de Uso:</b> Borrar Reserva
<b>Actor:</b> Gerente
1. El gerente selecciona la Reserva a Borrar y presiona el Botón Borrar.
2. El sistema devuelve una pantalla de datos de la Reserva seleccionada.
3. El sistema actualiza la información y devuelve un mensaje de confirmación, vuelve a la pantalla Administrar Reserva.
4. El gerente selecciona la opción Recargar, observa la Lista de Reservas actualizada.
5. Fin del Caso.

**Tabla 83: Especificación caso de uso: Borrar Reserva**

## II.1.10 Modelado Diagrama 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 actividades se utilizan para modelar los aspectos dinámicos de un sistema, esto implica modelar los pasos secuenciales de un proceso.

### II.1.10.2 Propósito

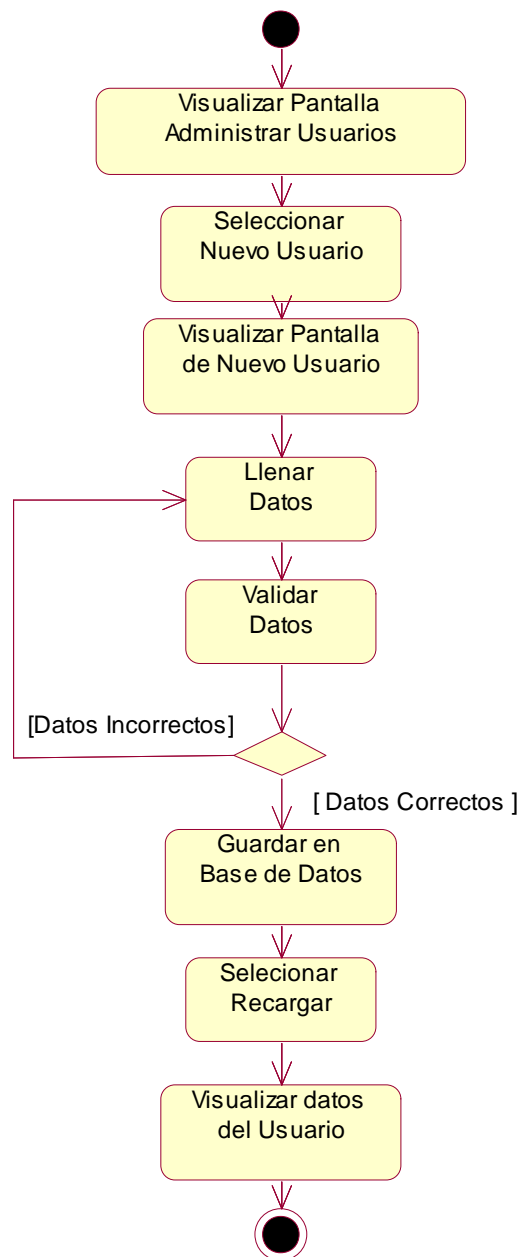
- Comprender la estructura y la dinámica del sistema deseado para la organización.
- Identificar posibles mejoras.

### II.1.10.3 Alcance

- Describir los procesos del sistema y los clientes.
- Identificar y definir los **procesos de los casos de uso** según los **objetivos** de la organización.
- Definir un **diagrama de actividad** para cada caso de uso del sistema.

#### II.1.10.4 Diagrama de Actividades

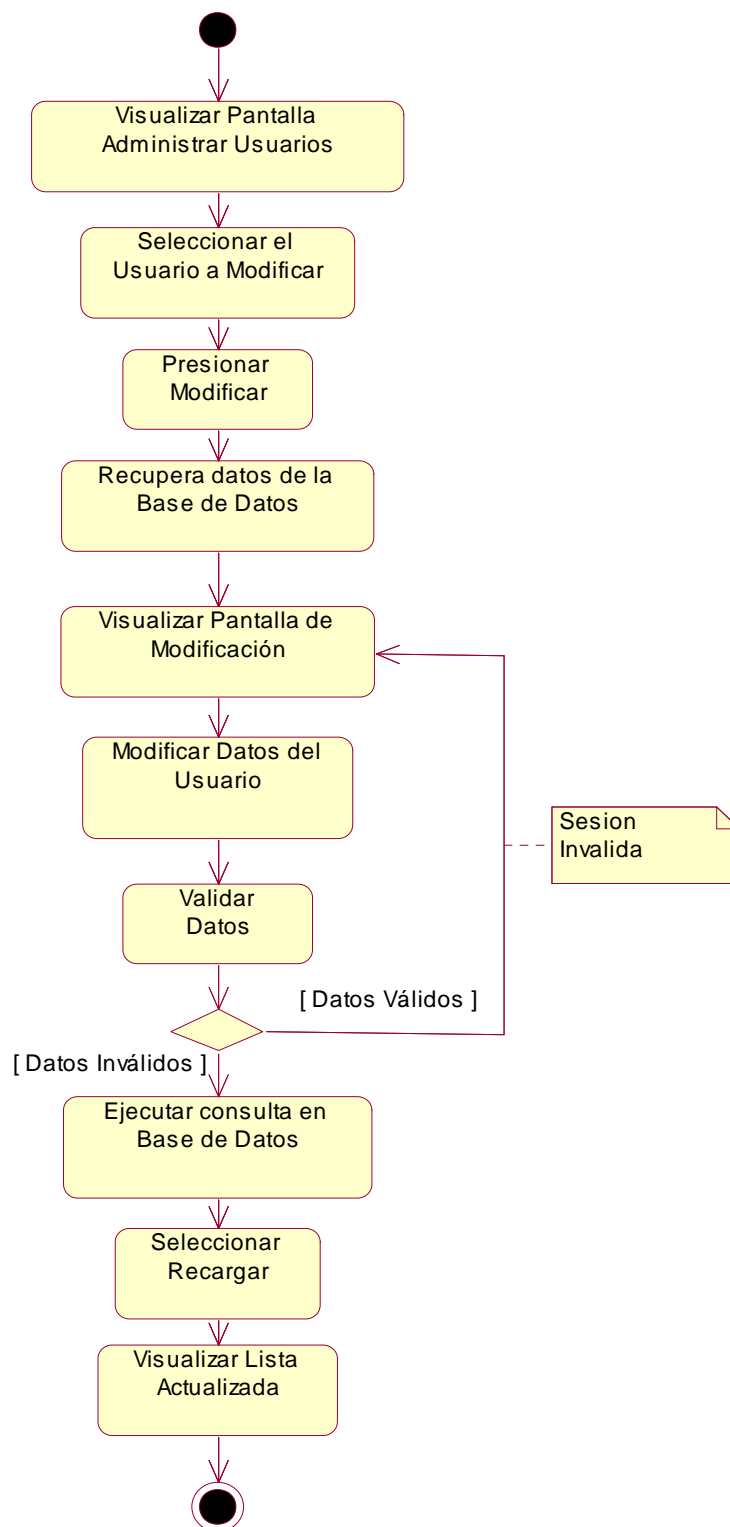
##### *Diagrama de Actividad: Nuevo Usuario*



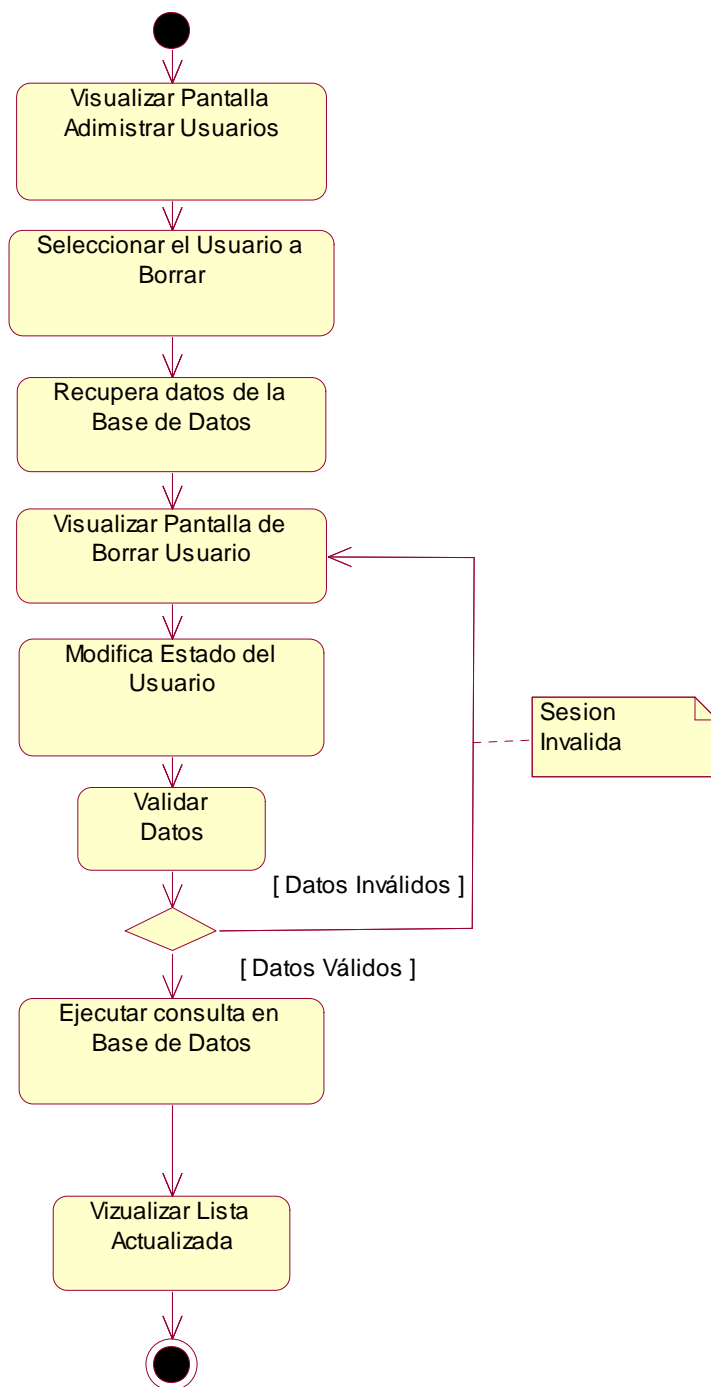
**Figura 38: Diagrama de Actividad: Nuevo Usuario**



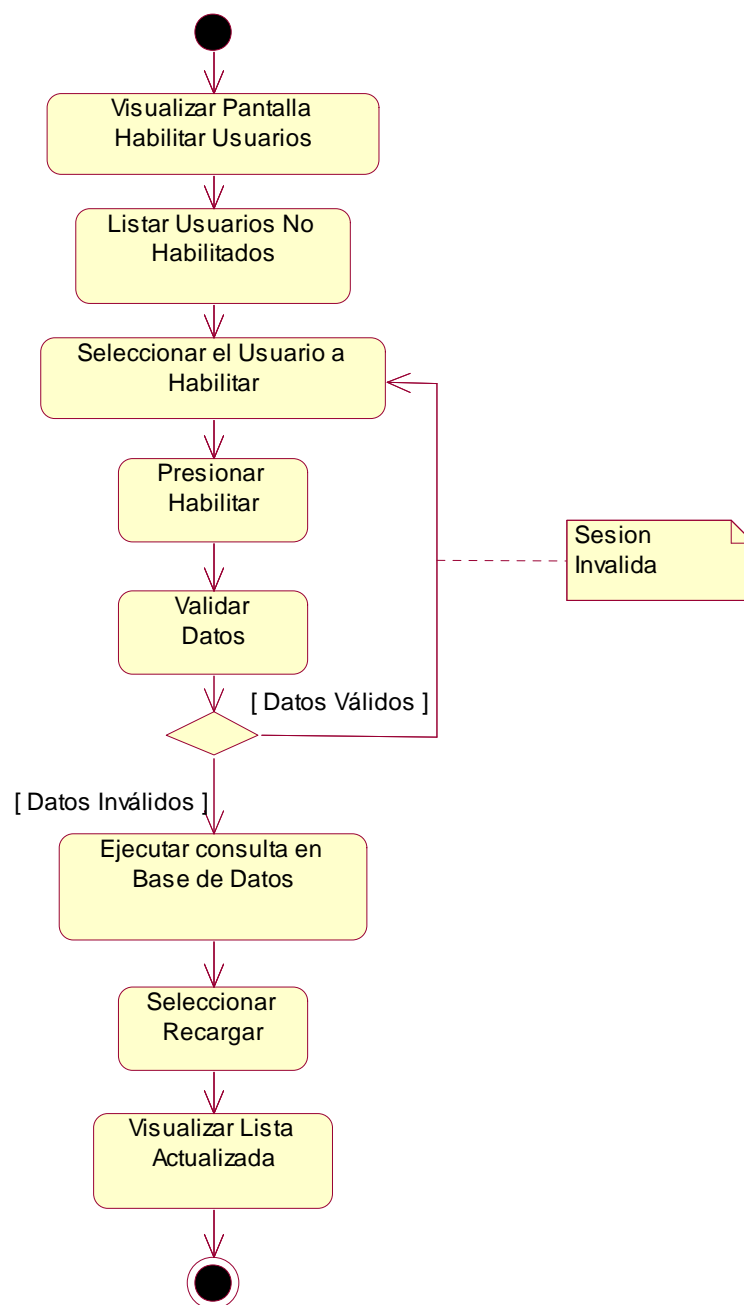
*Diagrama de Actividad: Modificar Usuario*



**Figura 39: Diagrama de Actividad: Modificar Usuario**

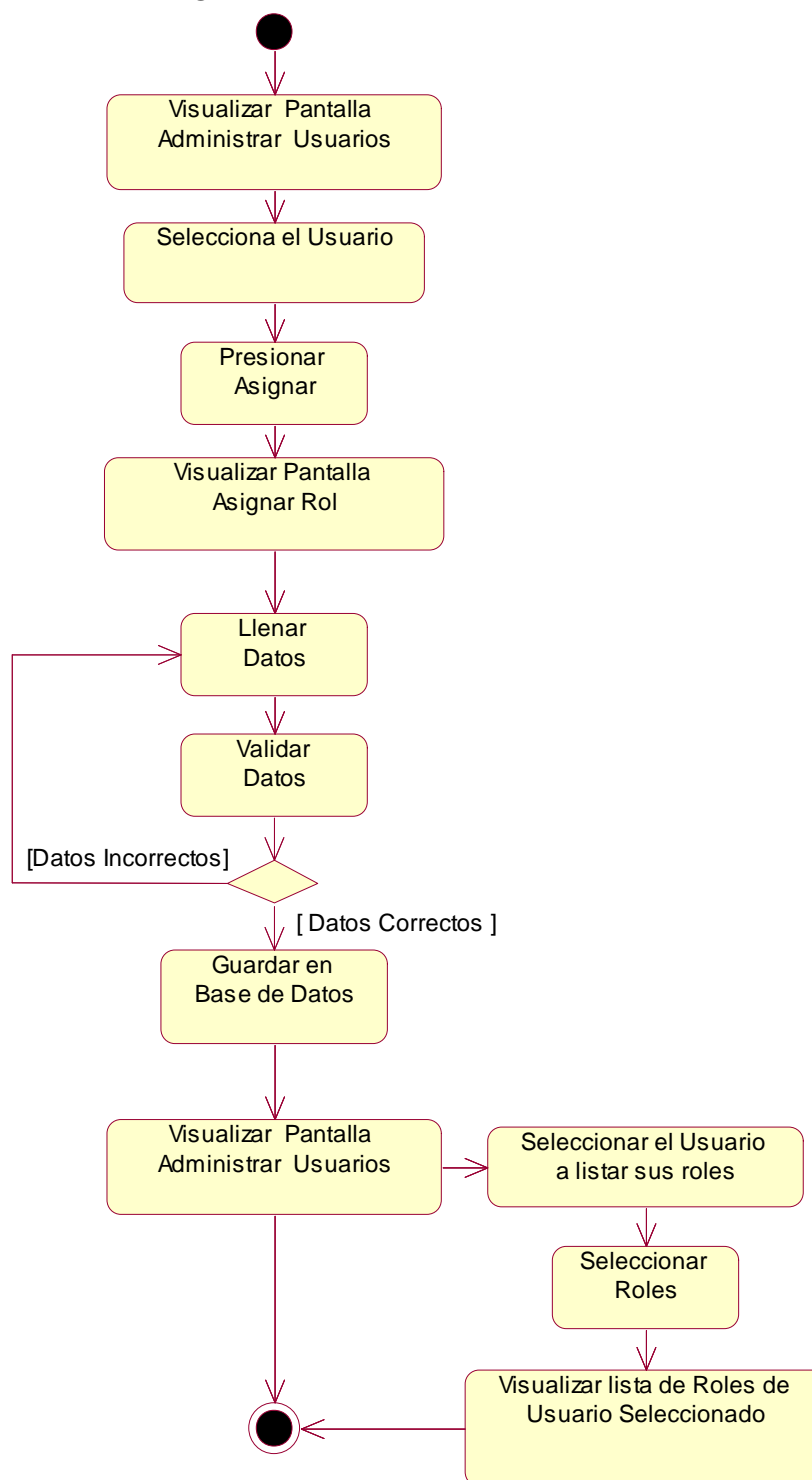
*Diagrama de Actividad: Borrar Usuario***Figura 40: Diagrama de Actividad: Borrar Usuario**

*Diagrama de Actividad: Habilitar Usuario*



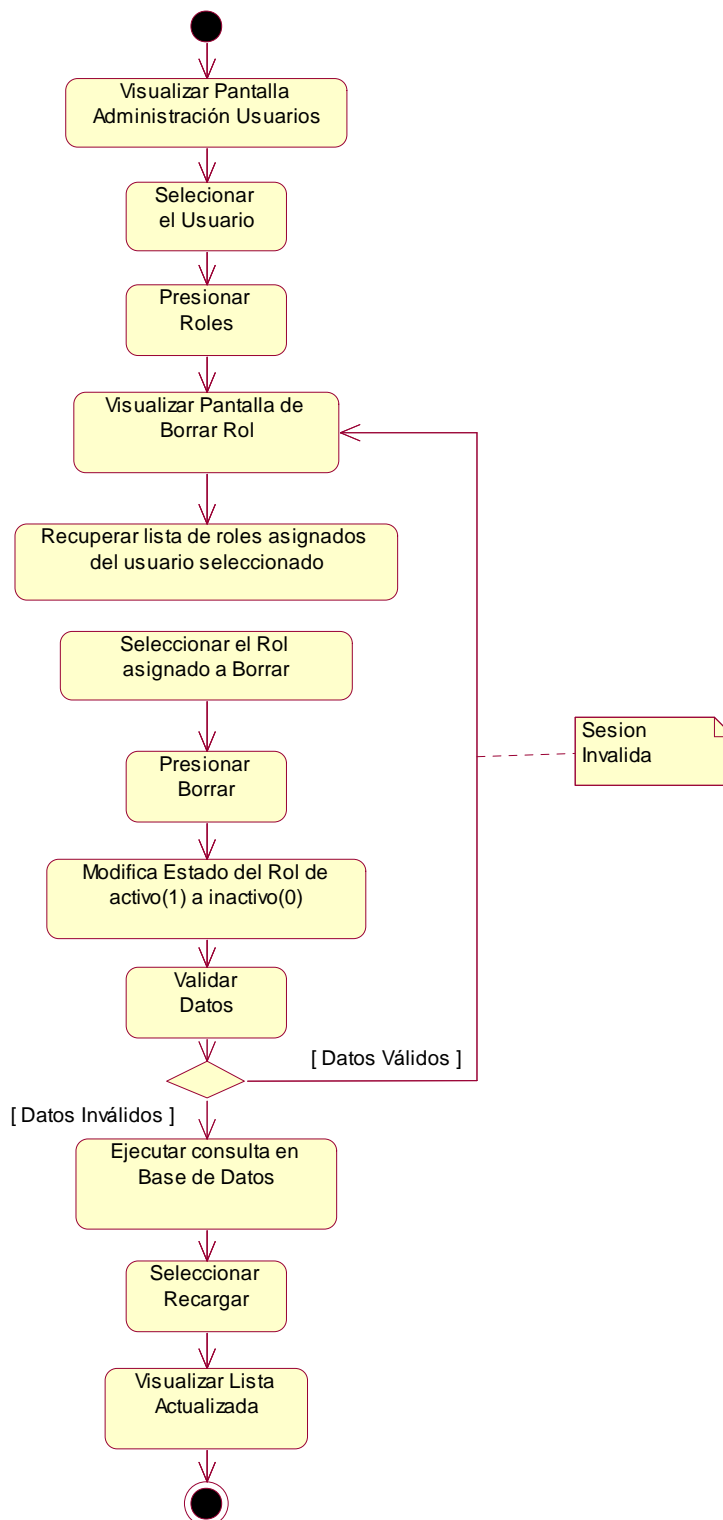
**Figura 41: Habilitar Usuario**

*Diagrama de Actividad: Asignar Rol a Usuario*

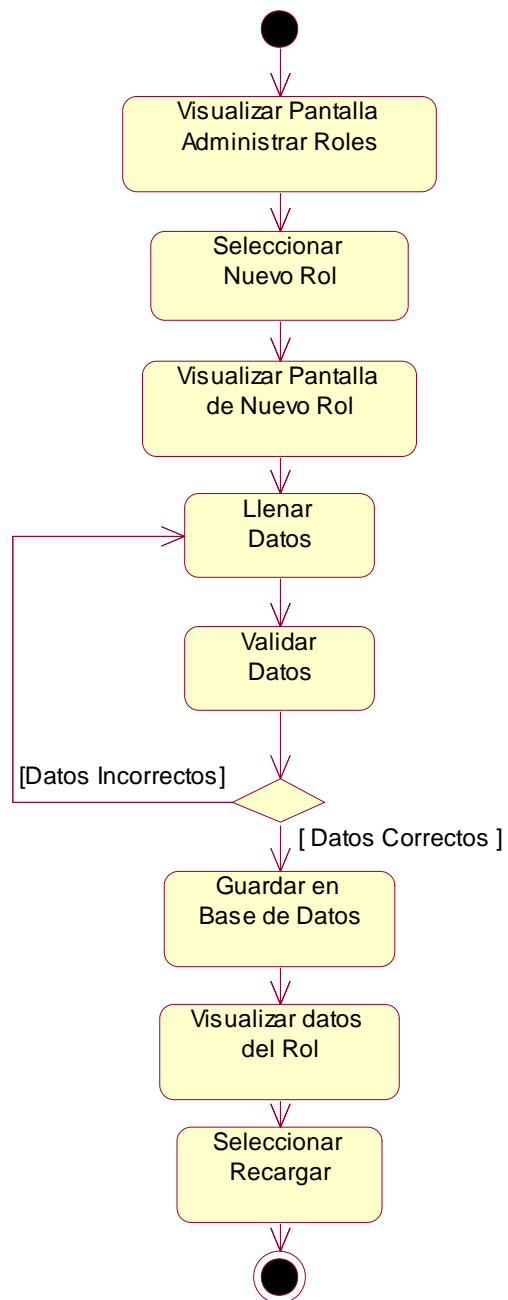


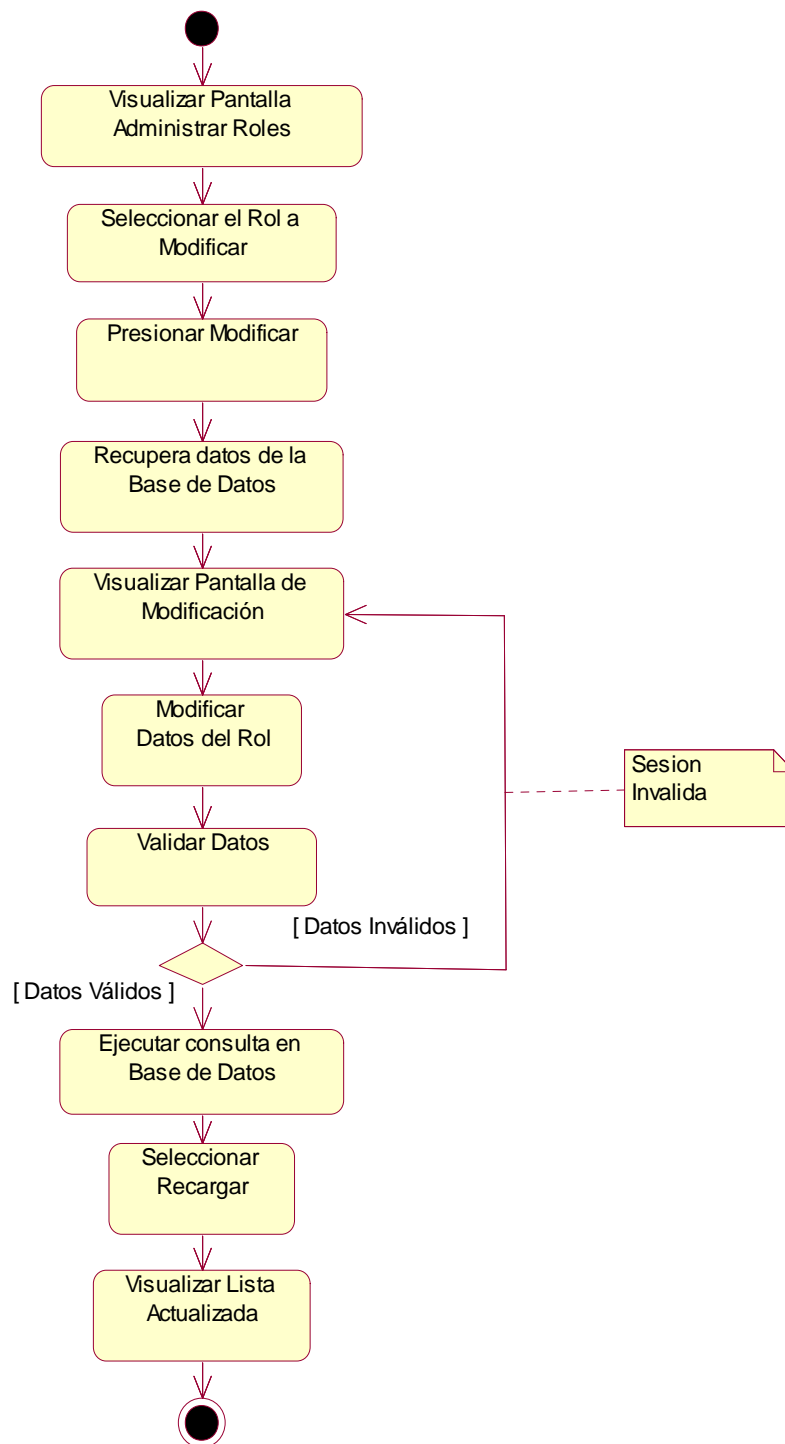
**Figura 42: Diagrama de Actividad: Asignar Rol a Usuario**

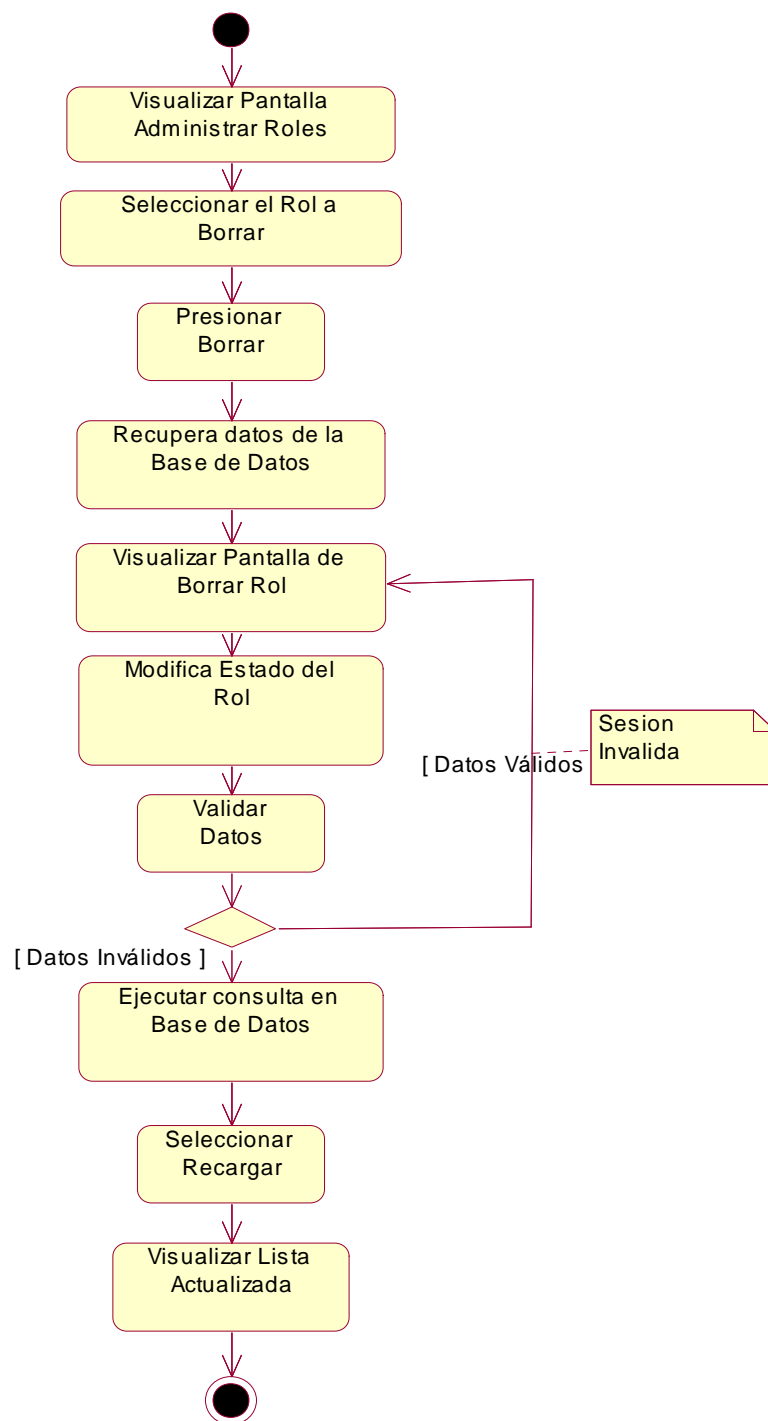
**Diagrama de Actividad: Borrar Rol Asignado**



**Figura 43: Diagrama de Actividad: Borrar Rol Asignado**

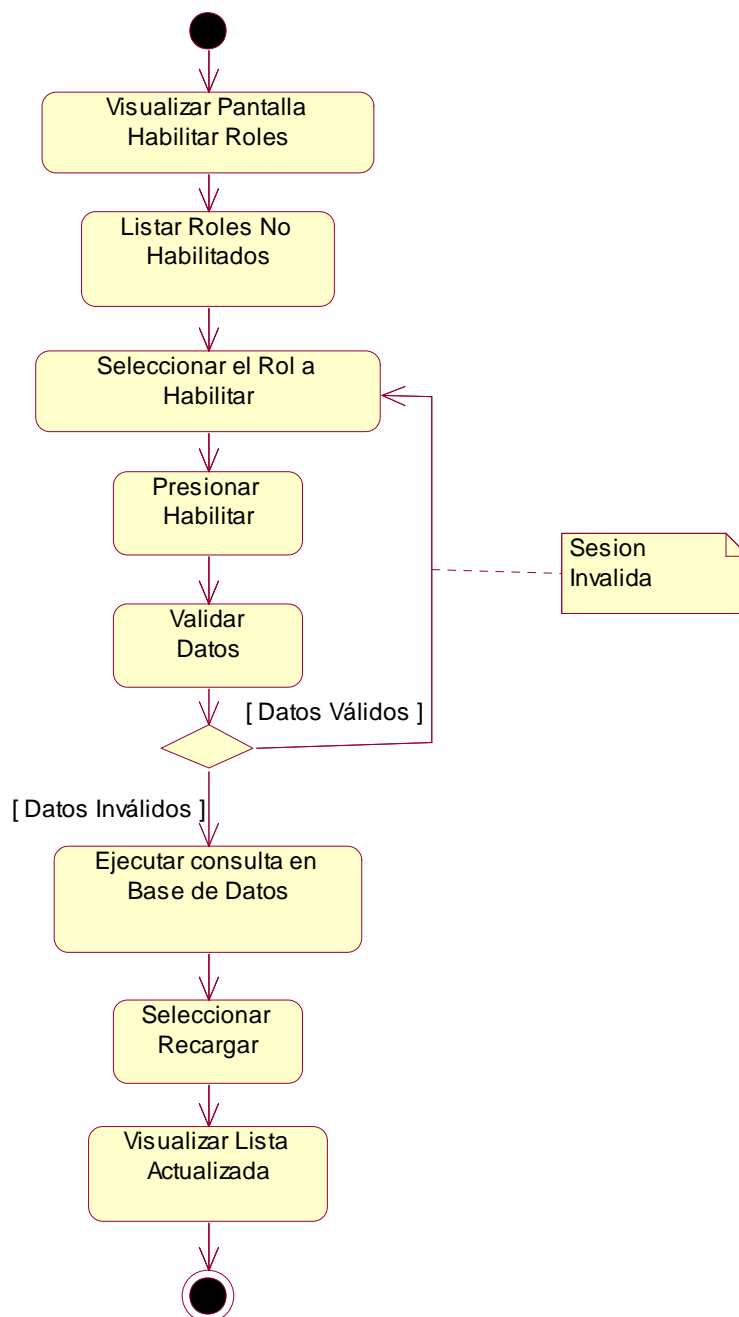
*Diagrama de Actividad: Nuevo Rol***Figura 44: Diagrama de Actividad: Nuevo Rol**

*Diagrama de Actividad: Modificar Rol***Figura 45: Diagrama de Actividad: Modificar Rol**

**Diagrama de Actividad: Borrar Rol****Figura 46: Diagrama de Actividad: Borrar Rol**

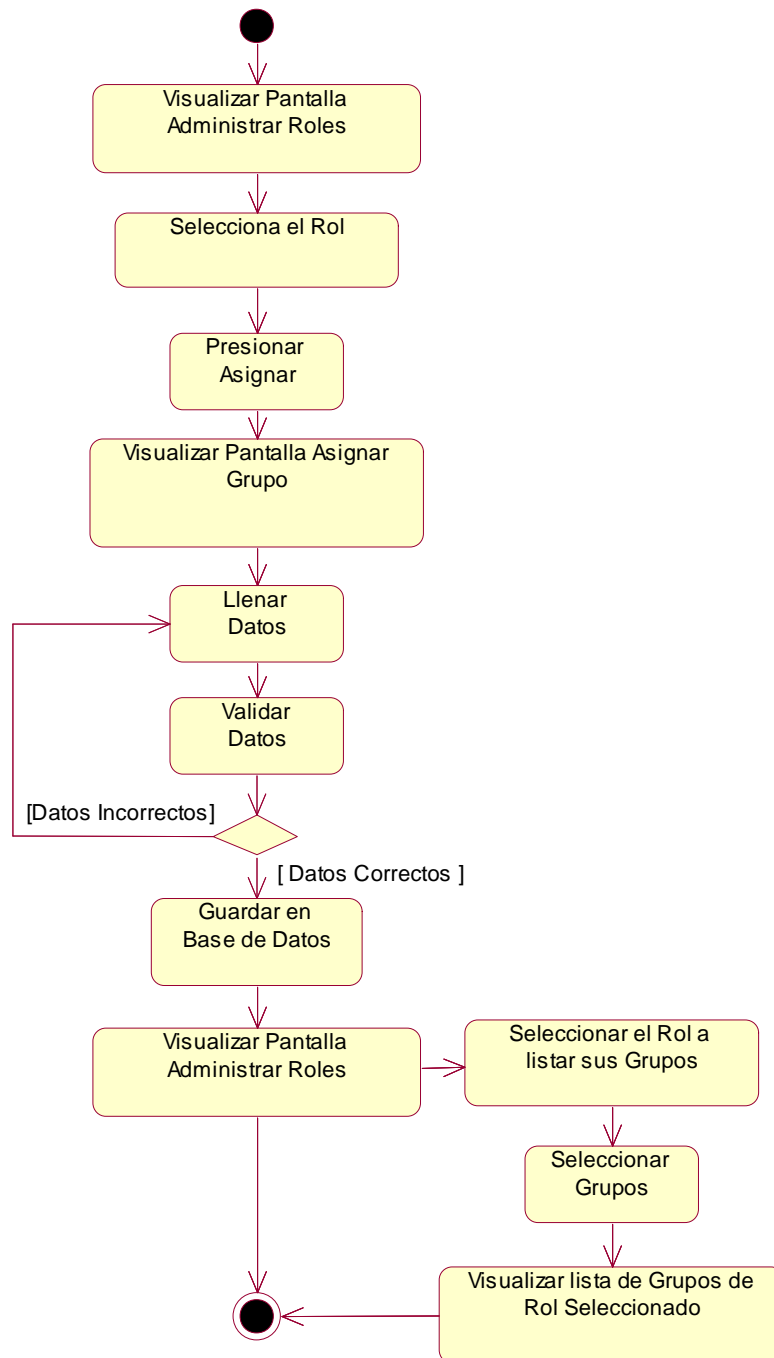


*Diagrama de Actividad: Habilitar Rol*



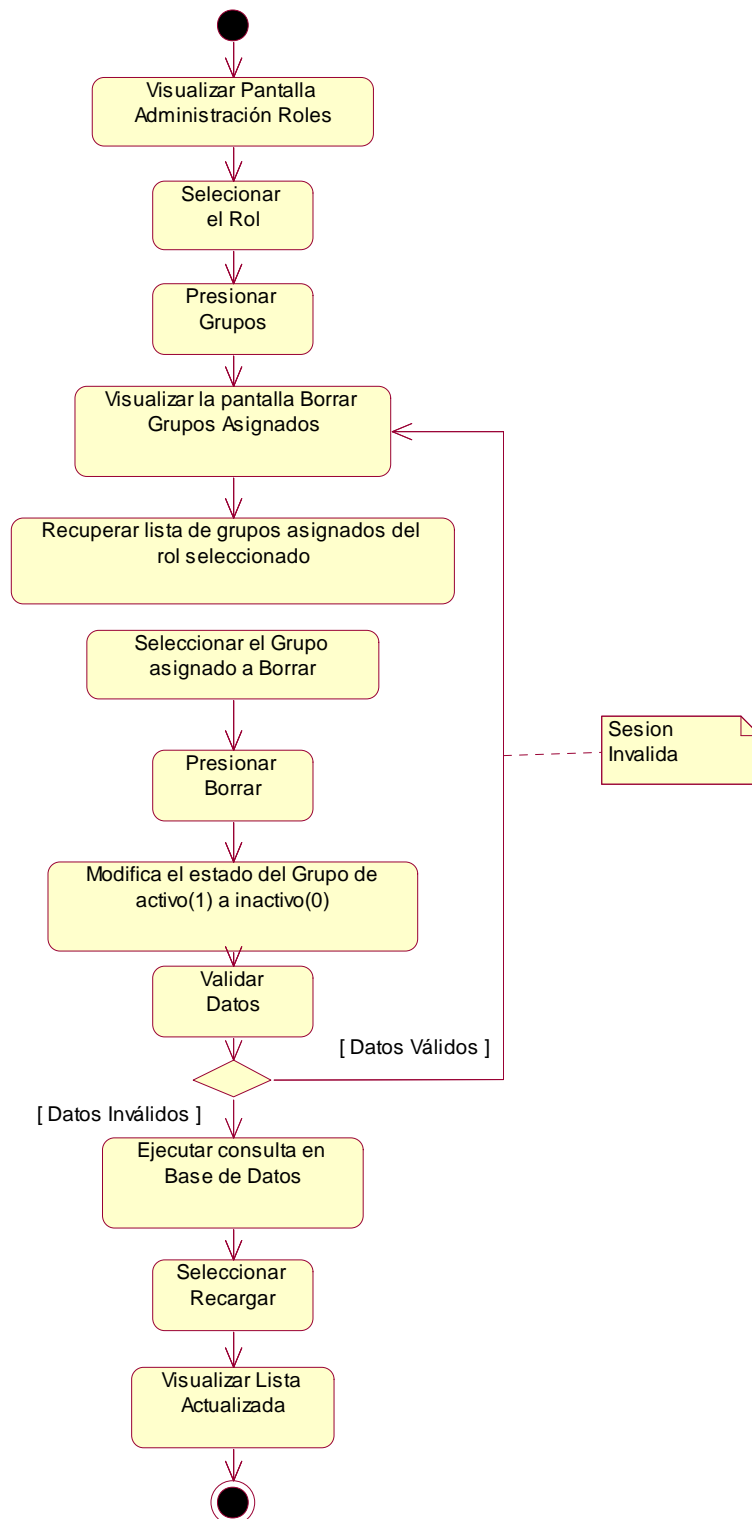
**Figura 47: Diagrama de Actividad: Habilitar Rol**

*Diagrama de Actividad: Asignar Grupo a Rol*

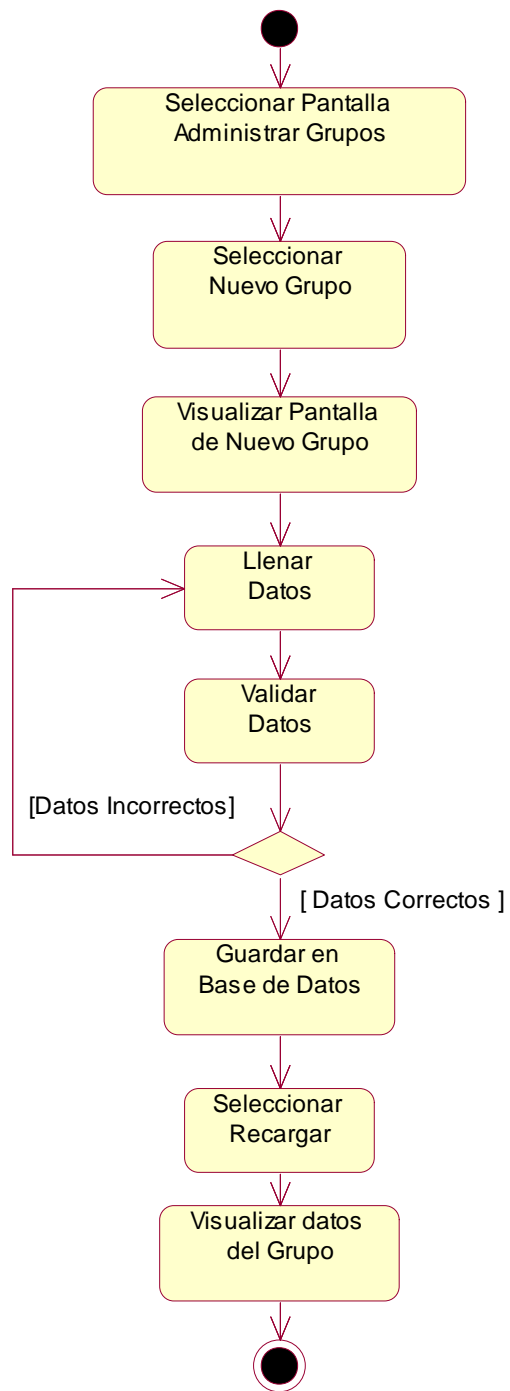


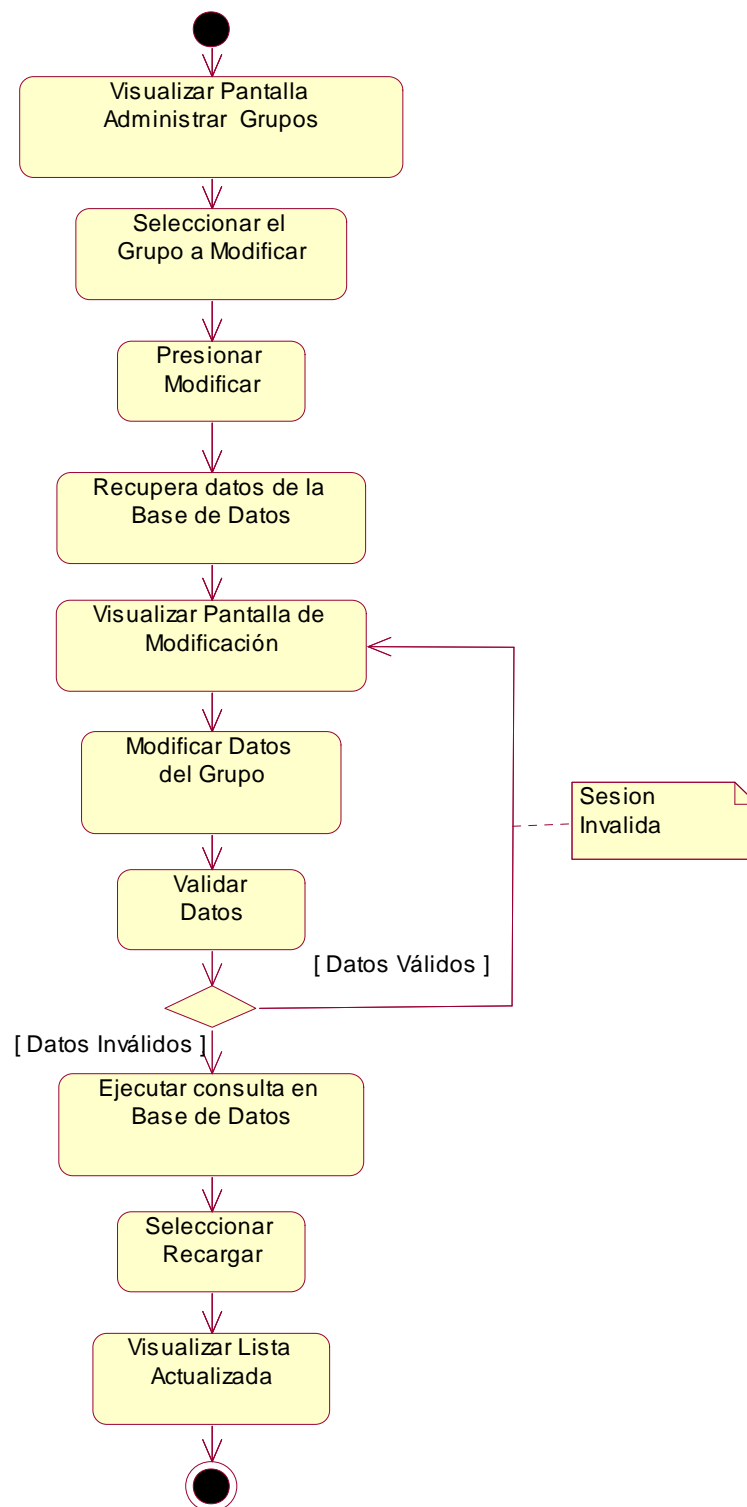
**Figura 48: Diagrama de Actividad: Asignar Grupo a Rol**

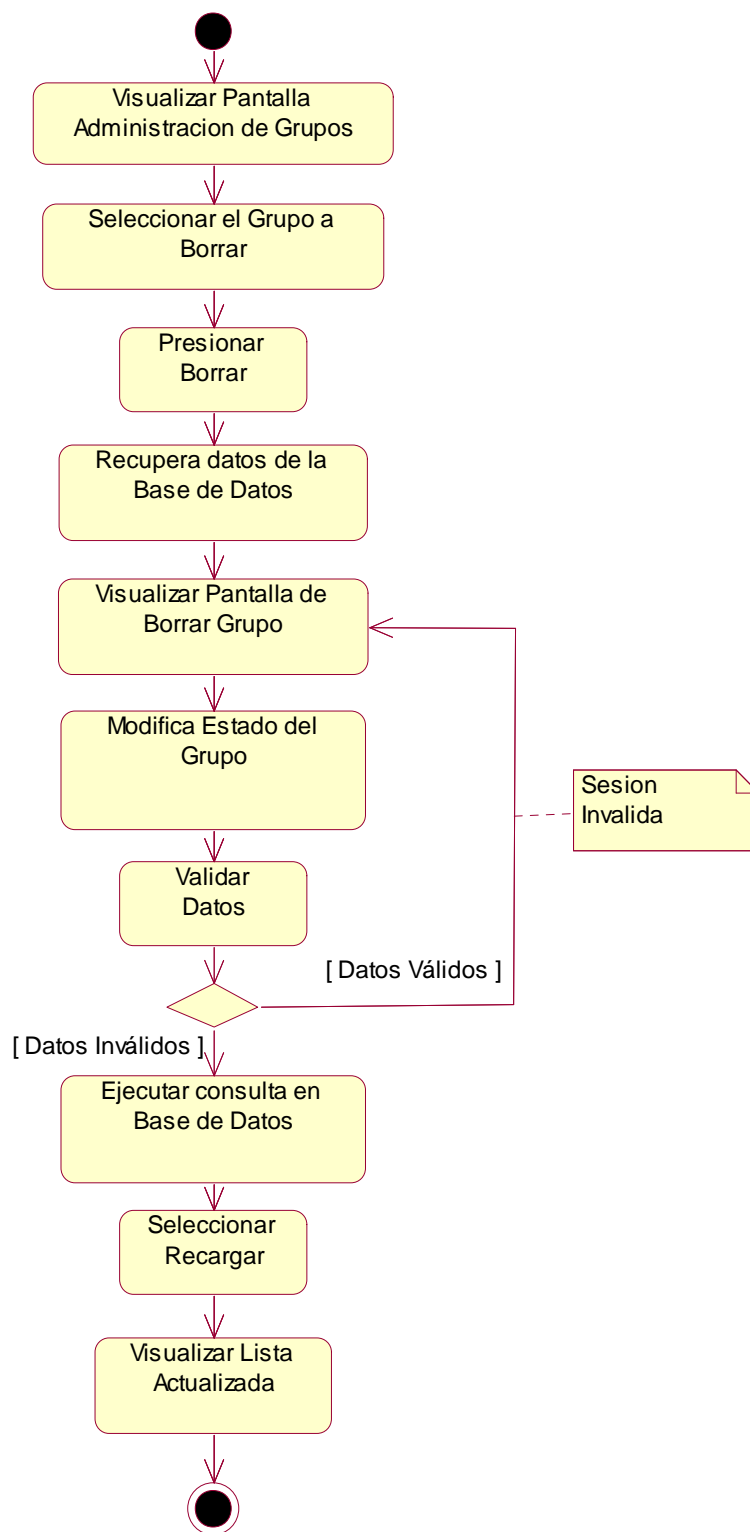
*Diagrama de Actividad: Borrar Grupos Asignados*



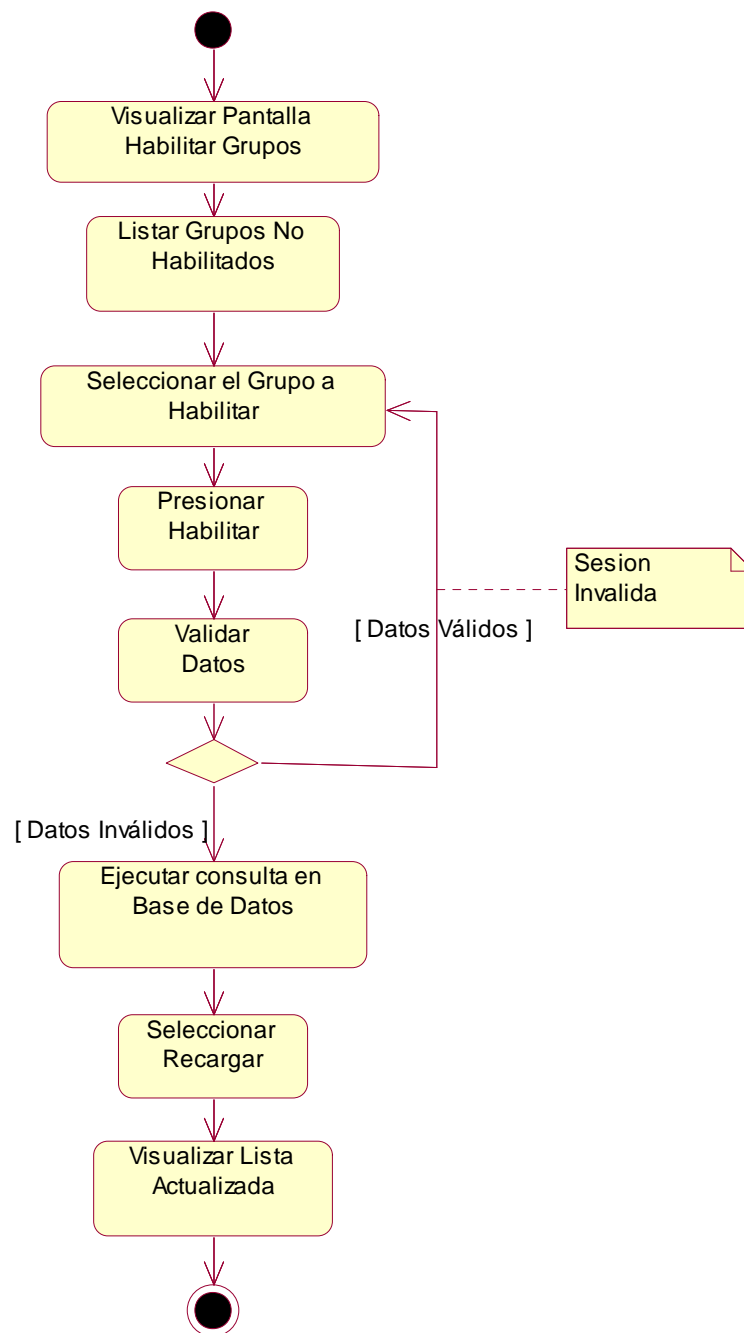
**Figura 49: Diagrama de Actividad: Borrar Grupos Asignados**

*Diagrama de Actividad: Nuevo Grupo***Figura 50: Diagrama de Actividad: Nuevo Grupo**

**Diagrama de Actividad: Modificar Grupo****Figura 51: Diagrama de Actividad: Modificar Grupo**

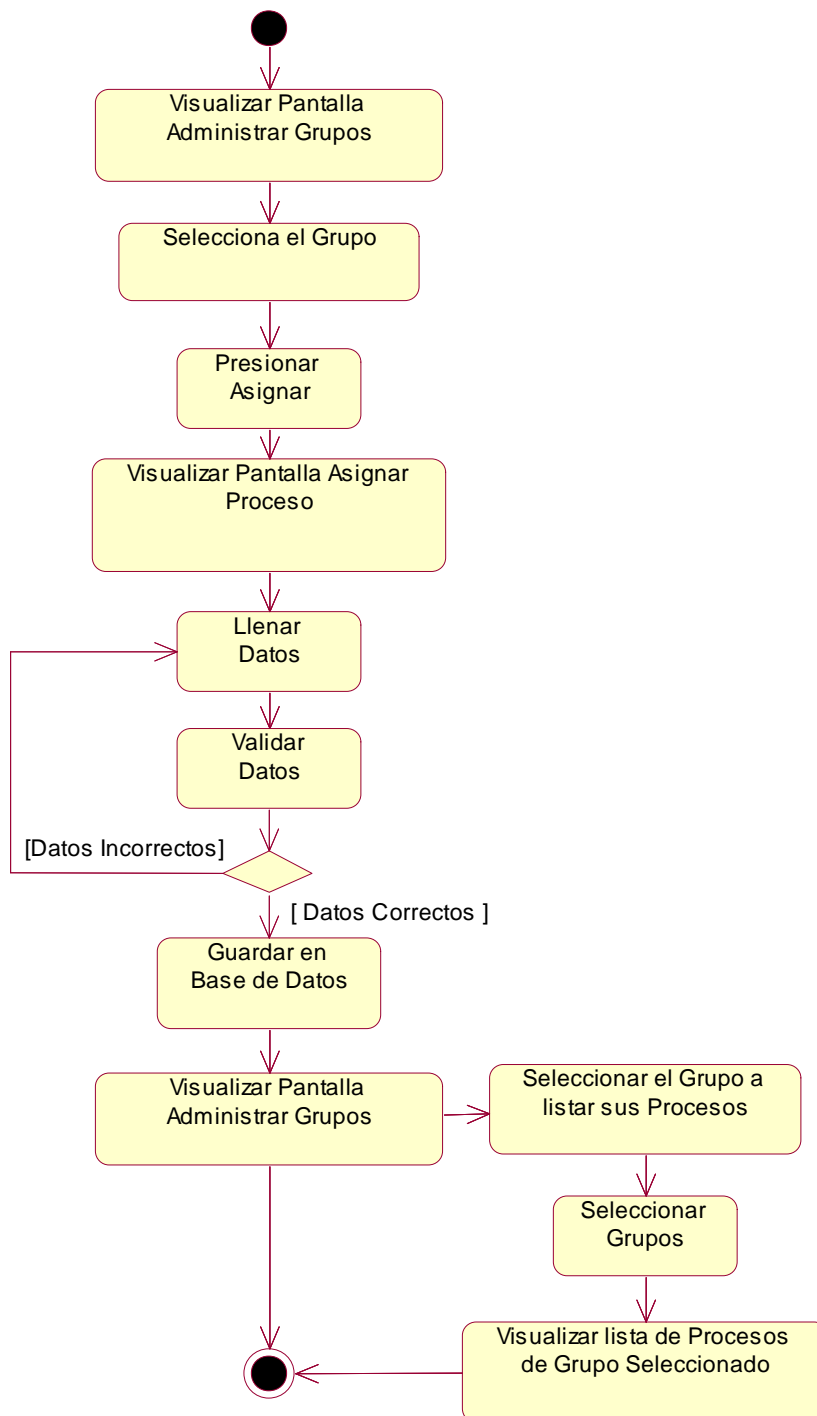
**Diagrama de Actividad: Borrar Grupo****Figura 52: Diagrama de Actividad: Borrar Grupo**

*Diagrama de Actividad: Habilitar Grupos*



**Figura 53: Diagrama de Actividad: Habilitar Grupos**

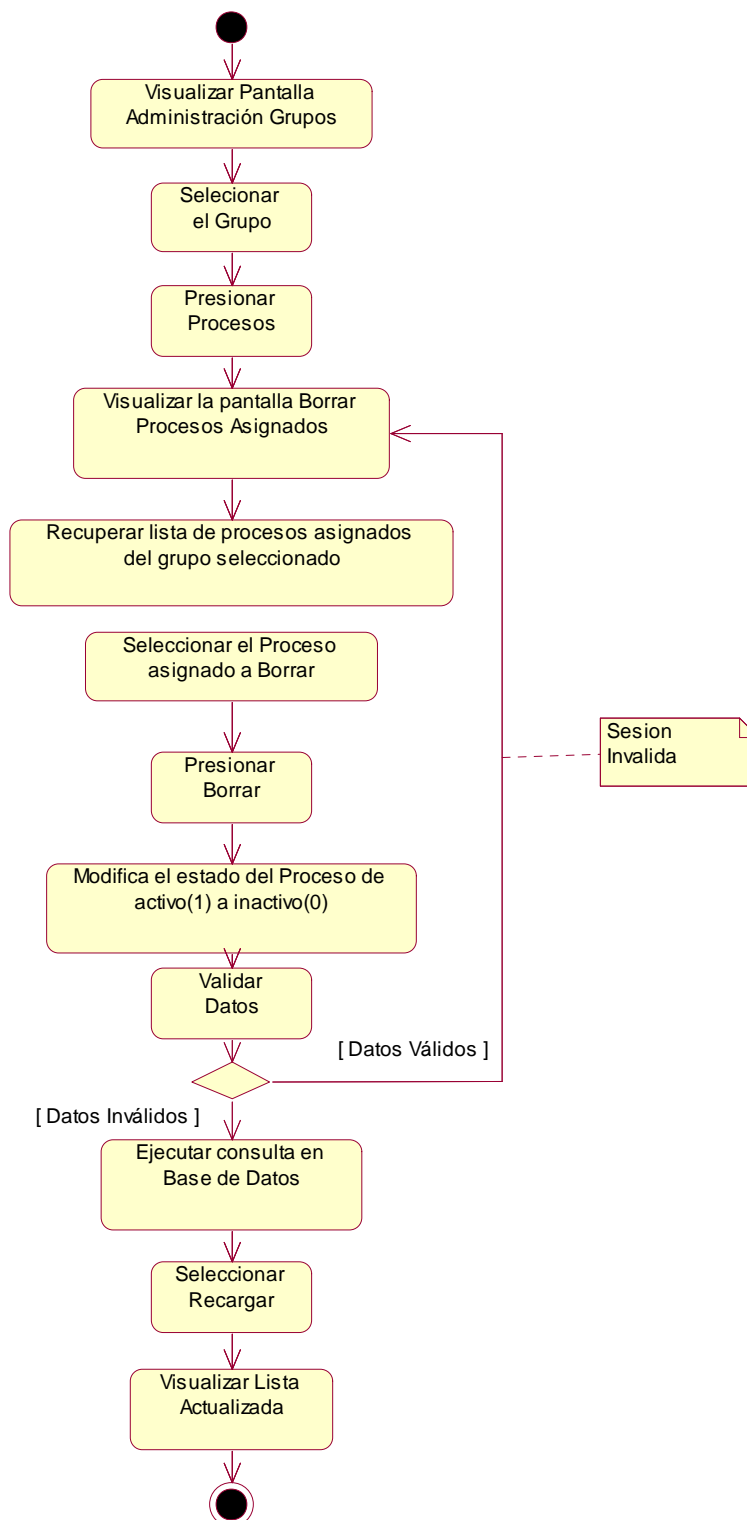
*Diagrama de Actividad: Asignar Proceso a Grupo*



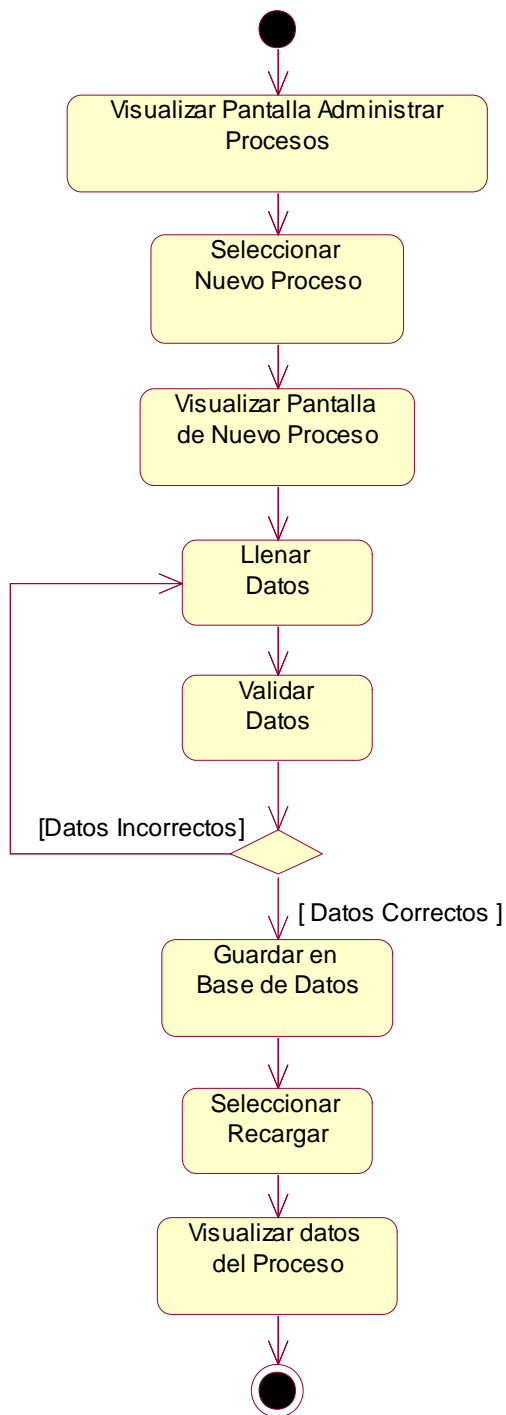
**Figura 54: Diagrama de Actividad: Asignar Proceso a Grupo**



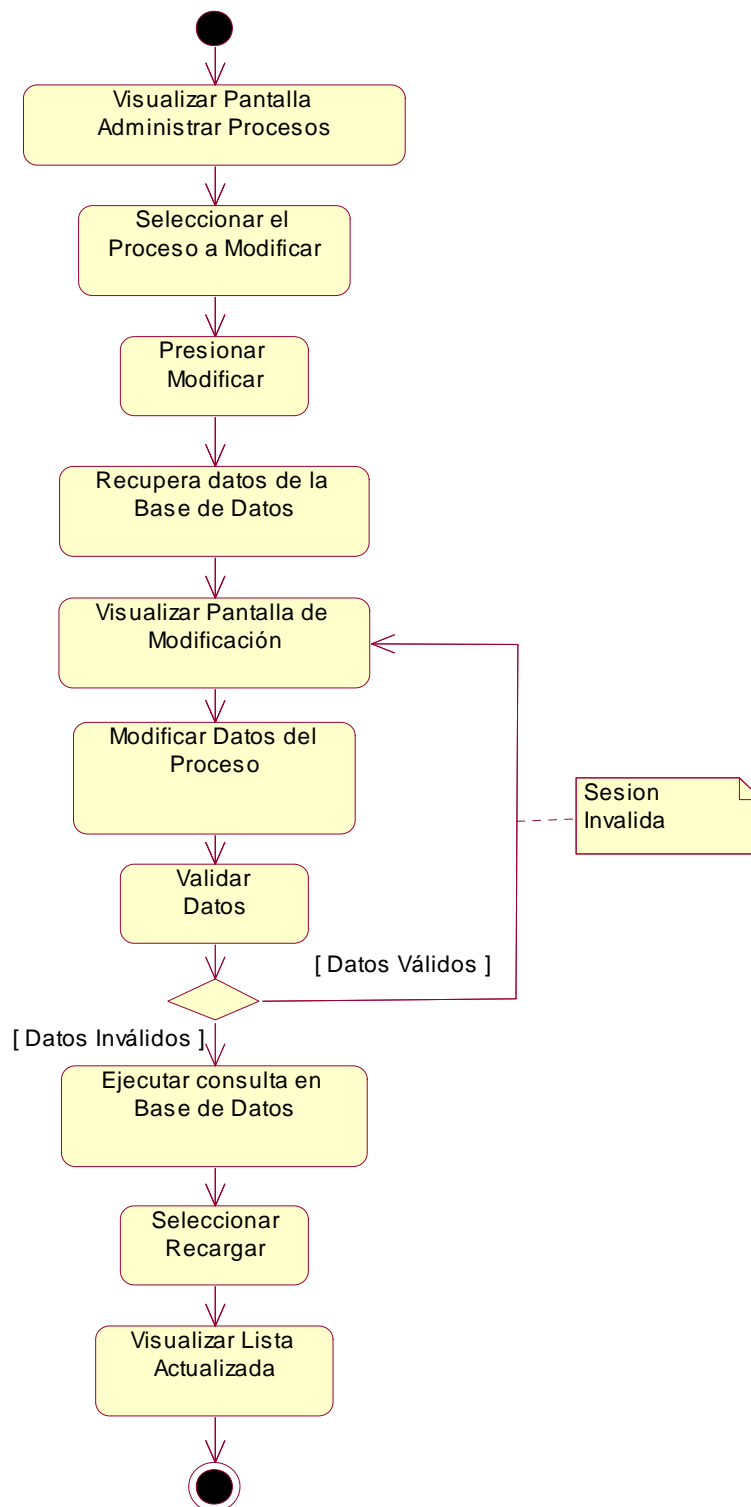
**Diagrama de Actividad: Borrar Procesos Asignados**



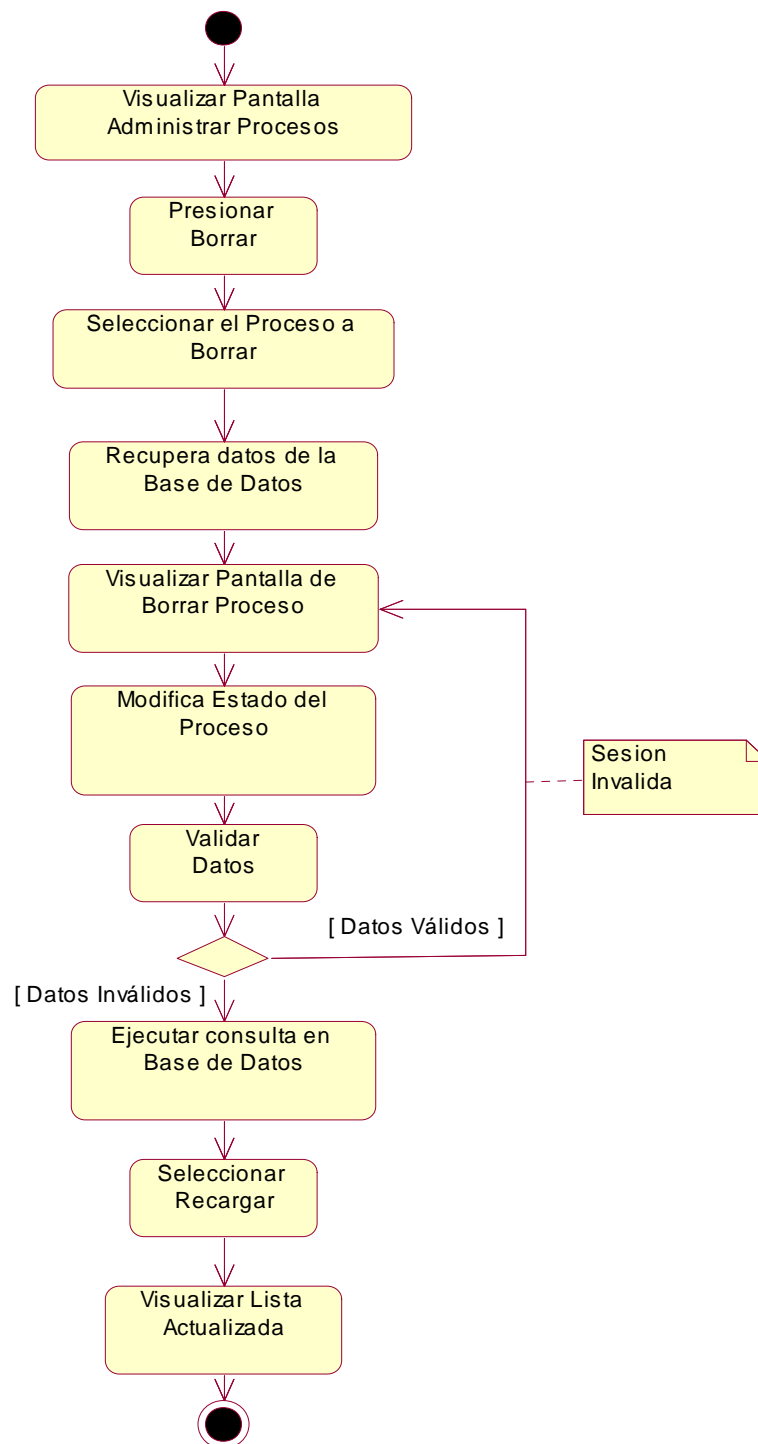
**Figura 55: Diagrama de Actividad: Borrar Procesos Asignados**

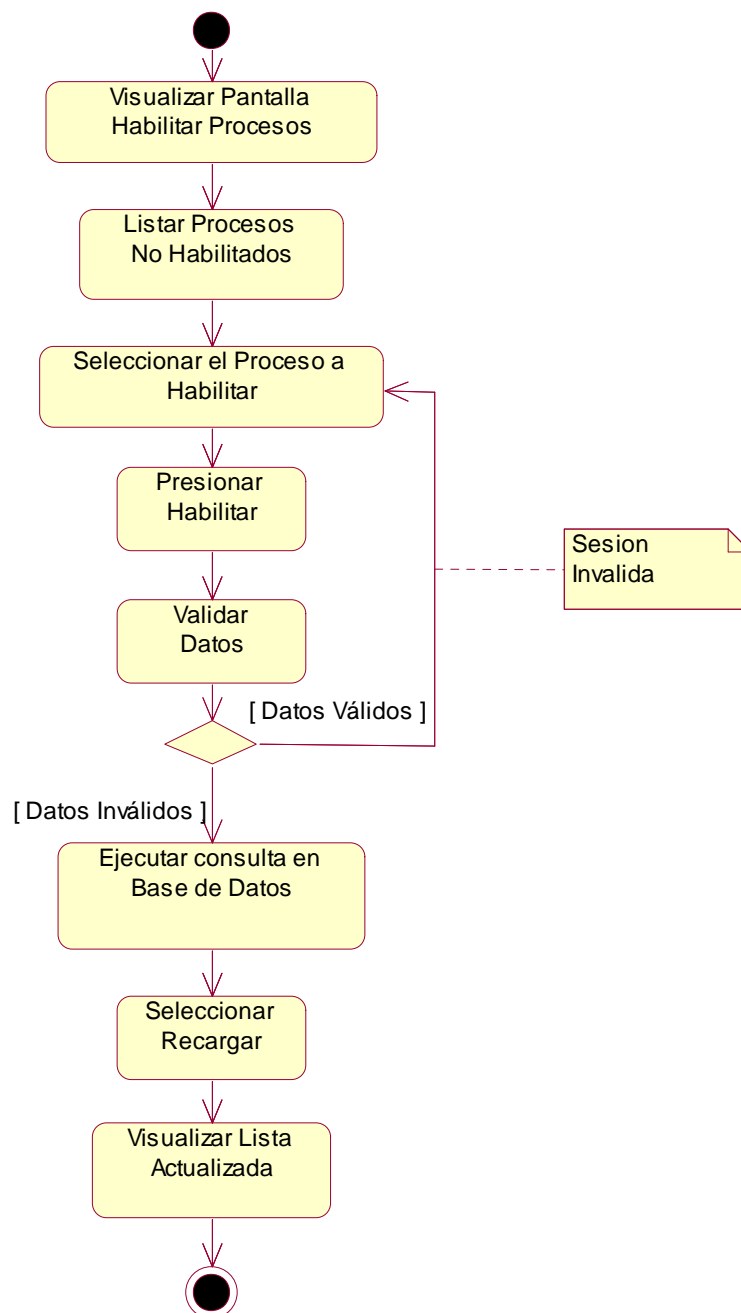
*Diagrama de Actividad: Nuevo Proceso***Figura 56: Diagrama de Actividad: Nuevo Proceso**

*Diagrama de Actividad: Modificar Proceso*

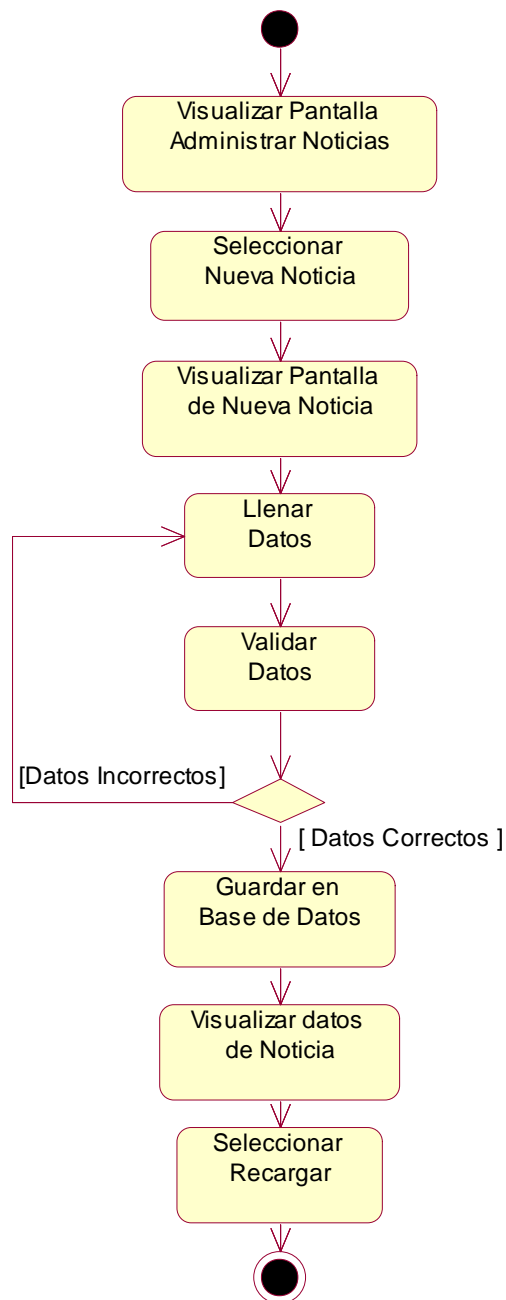


**Figura 57: Diagrama de Actividad: Modificar Proceso**

*Diagrama de Actividad: Borrar Proceso***Figura 58: Diagrama de Actividad: Borrar Proceso**

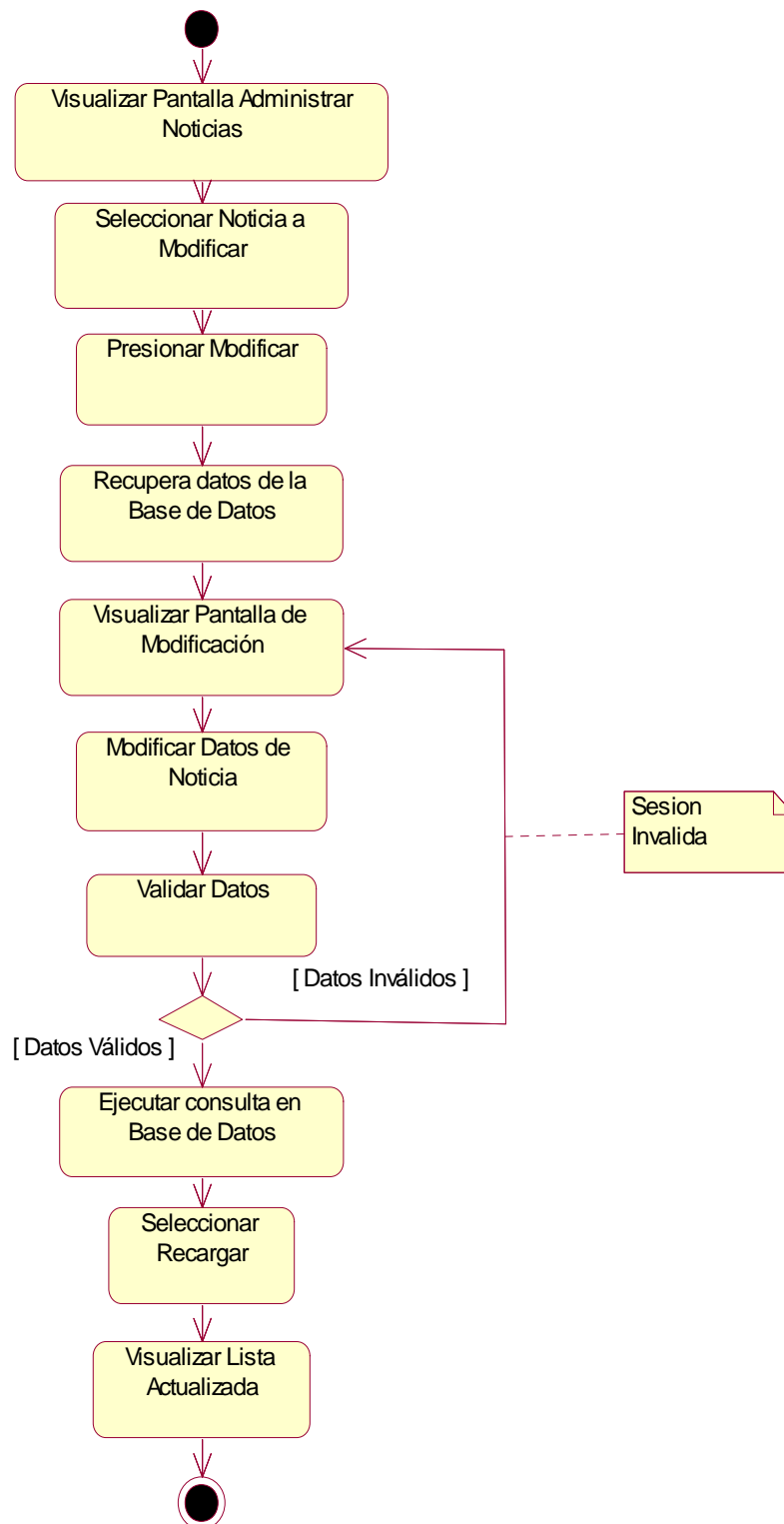
*Diagrama de Actividad: Habilitar Proceso***Figura 59: Diagrama de Actividad: Habilitar Proceso**

*Diagrama de Actividad: Habilitar Noticias*

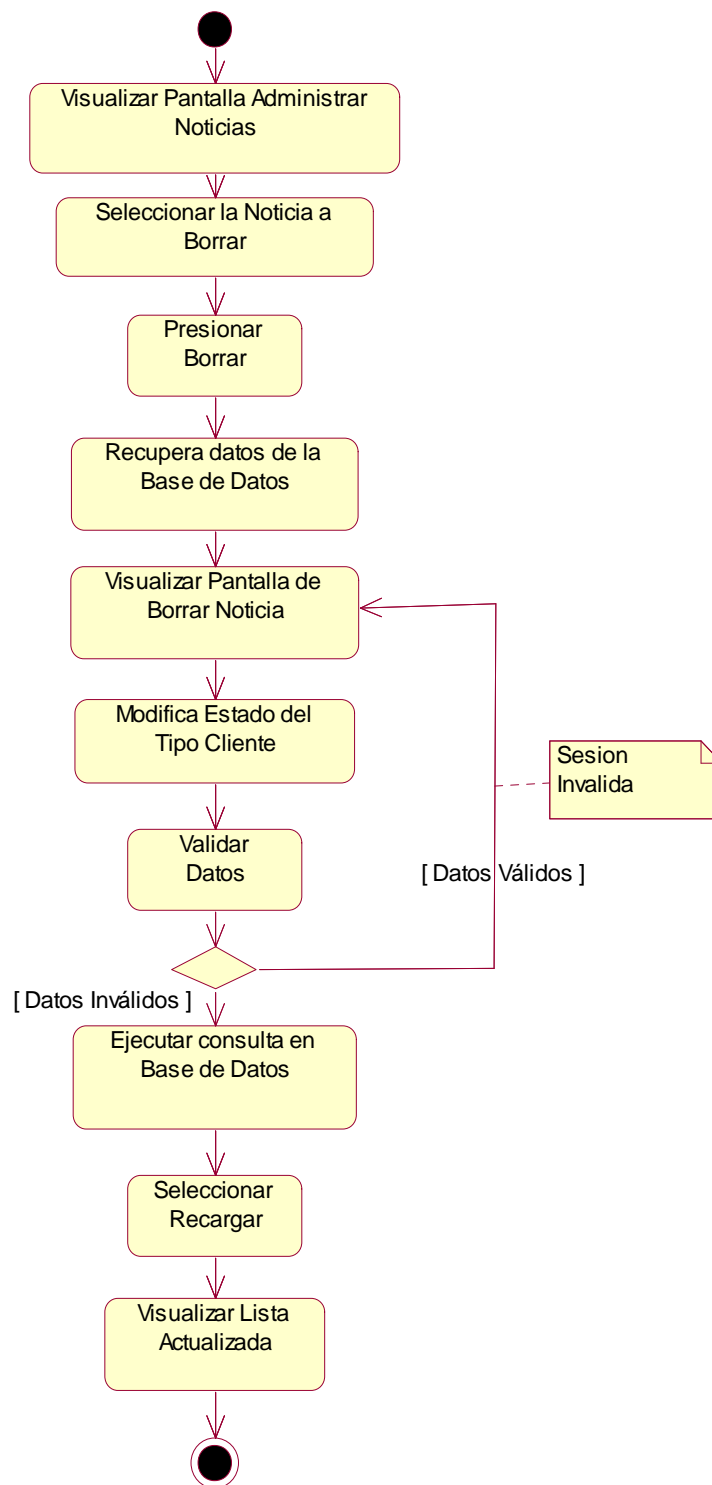


**Figura 60: Diagrama de Actividad: Habilitar Noticias**

*Diagrama de Actividad: Modificar Noticias*

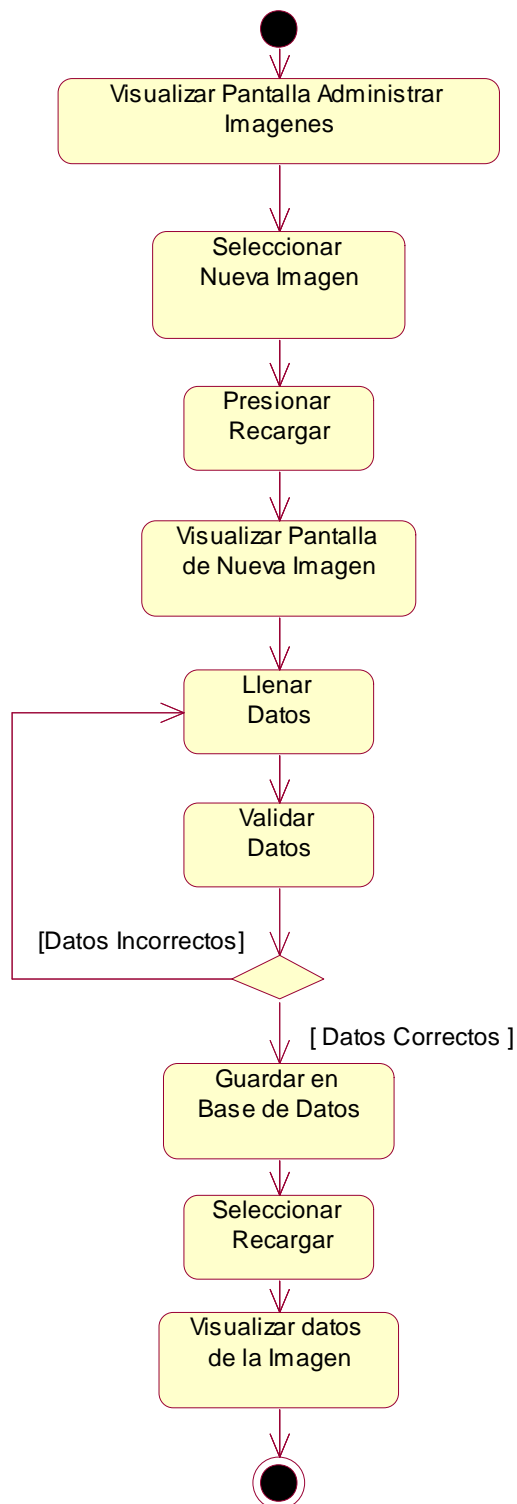


**Figura 61: Diagrama de Actividad: Modificar Noticias**

*Diagrama de Actividad: Borrar Noticias***Figura 62: Diagrama de Actividad: Borrar Noticias**

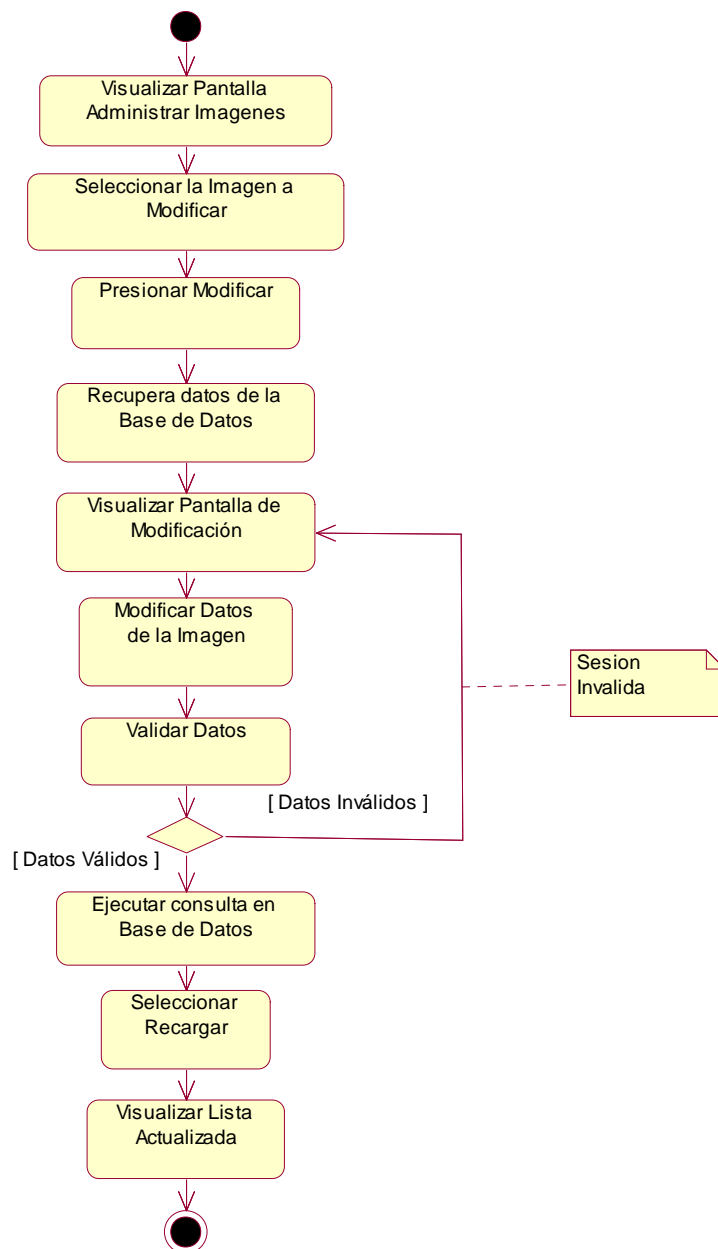


*Diagrama de Actividad: Nueva Imagen*

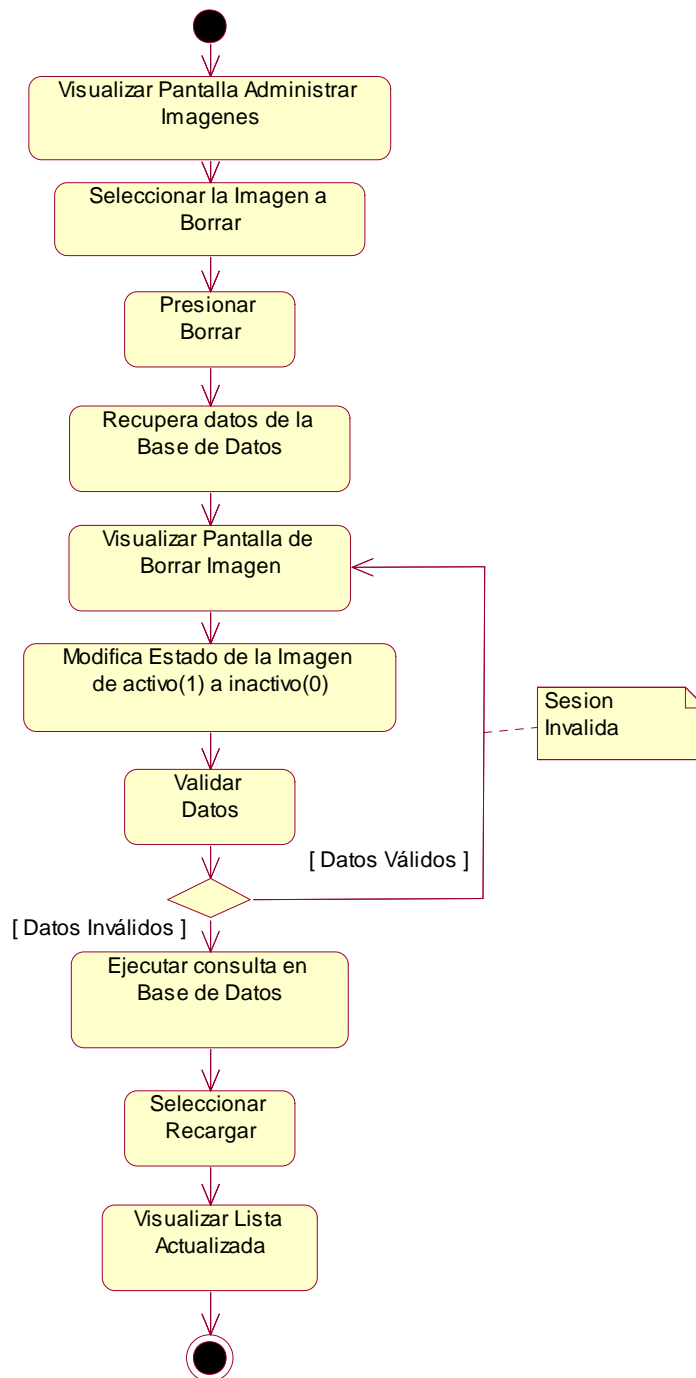


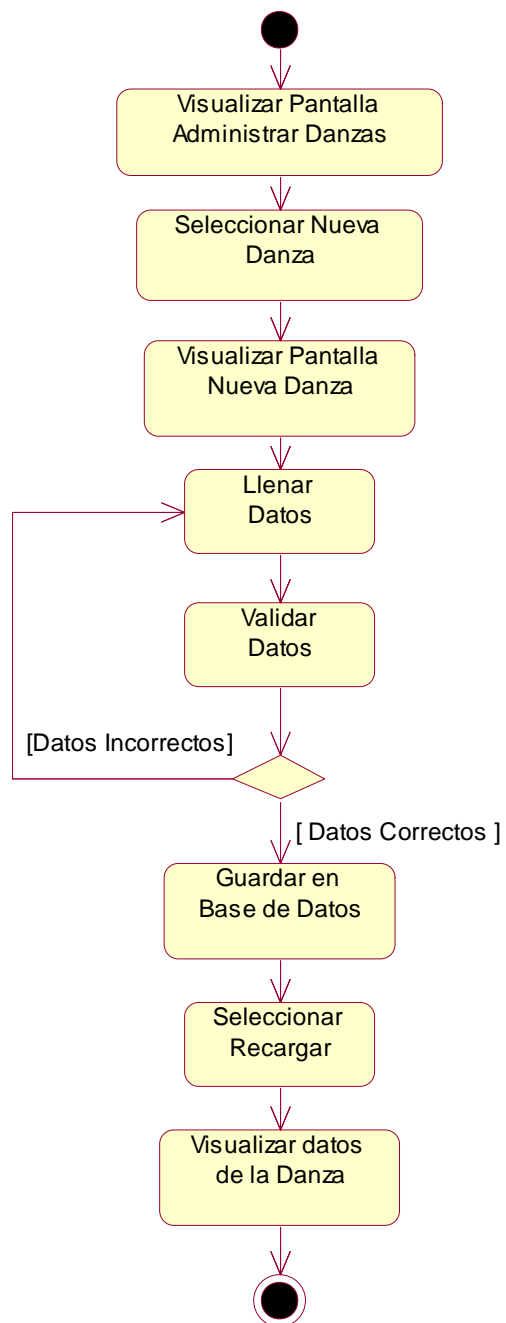
**Figura 63: Diagrama de Actividad: Nueva Imagen**

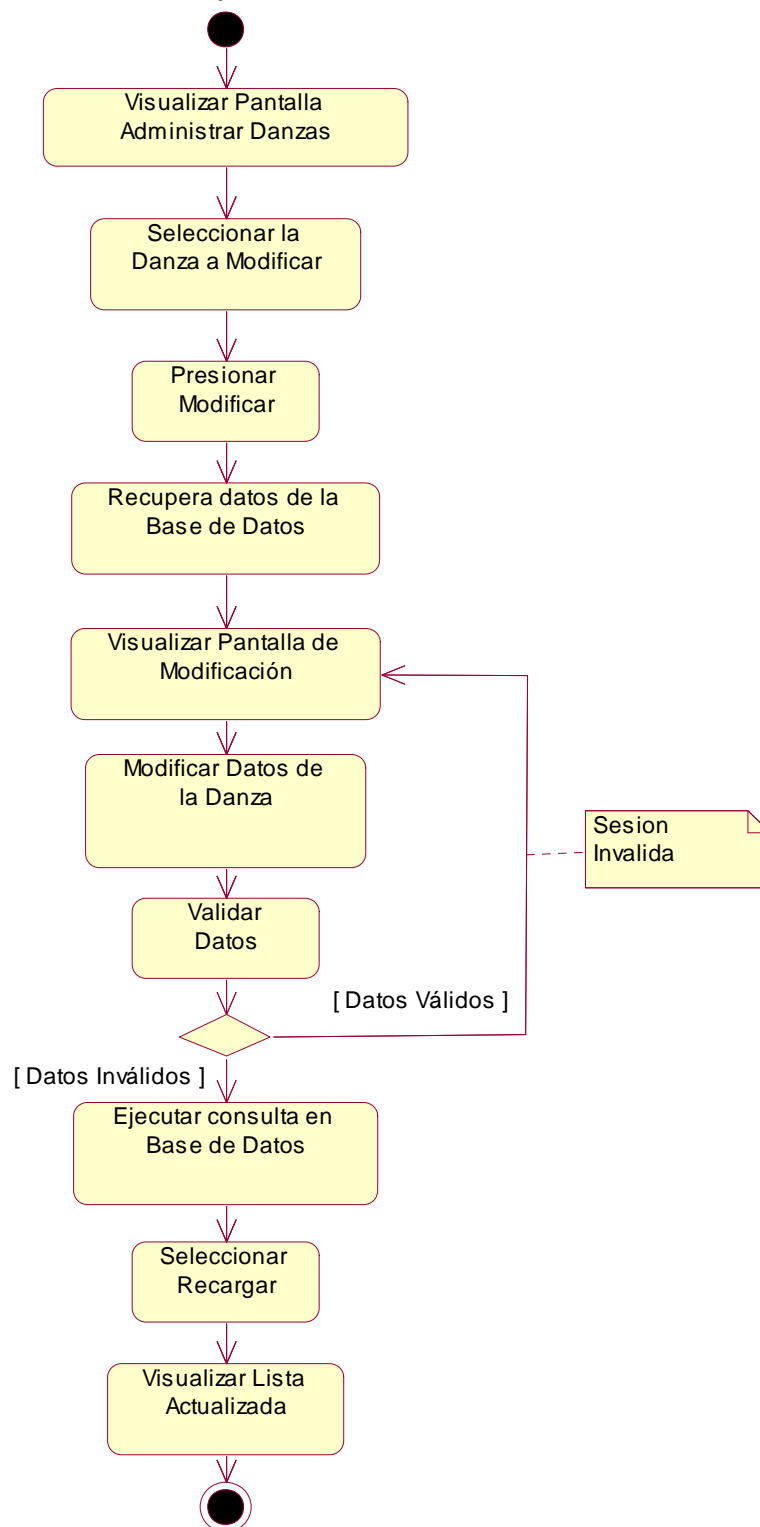
*Diagrama de Actividad: Modificar Imagen*

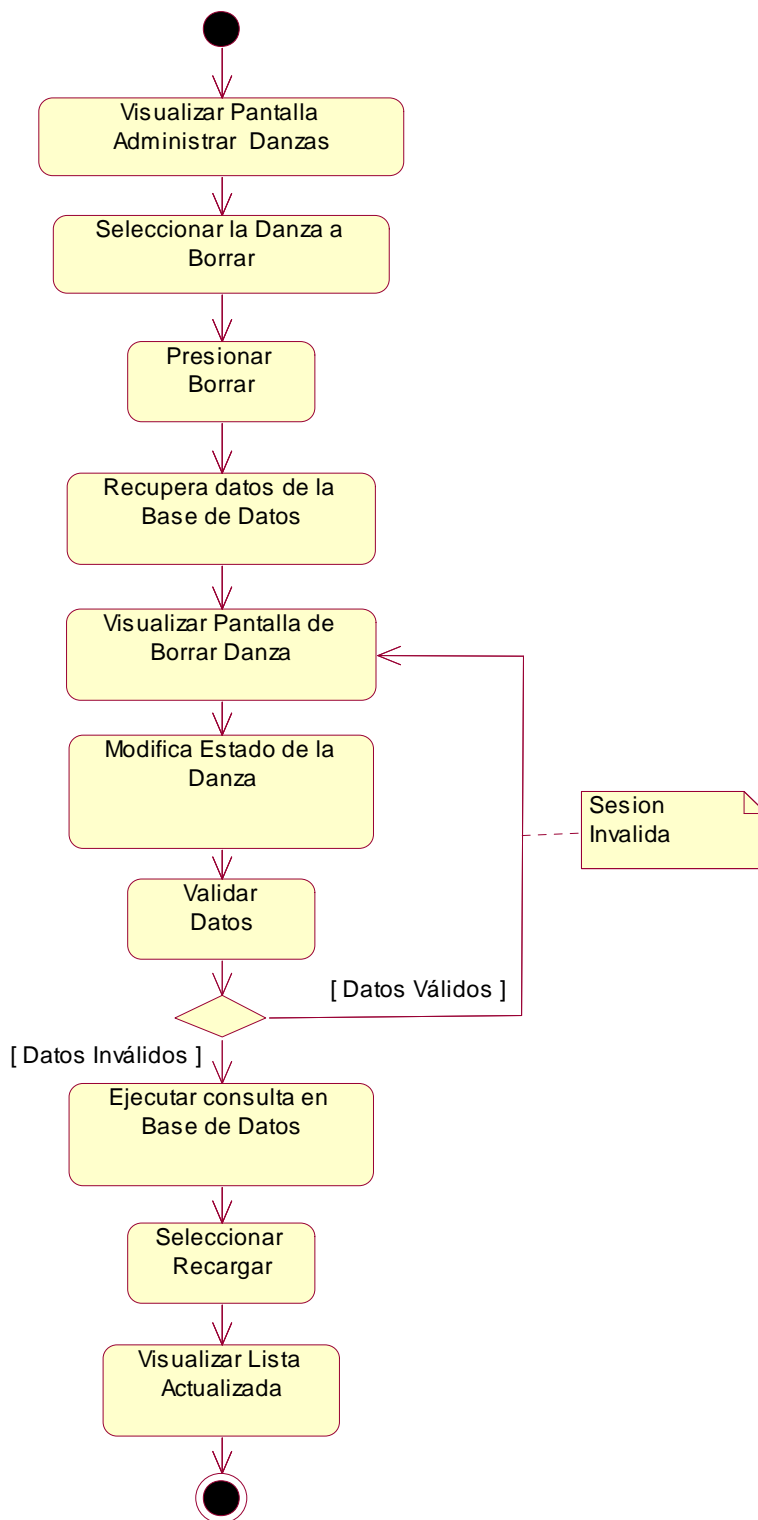


**Figura 64: Diagrama de Actividad: Modificar Imagen**

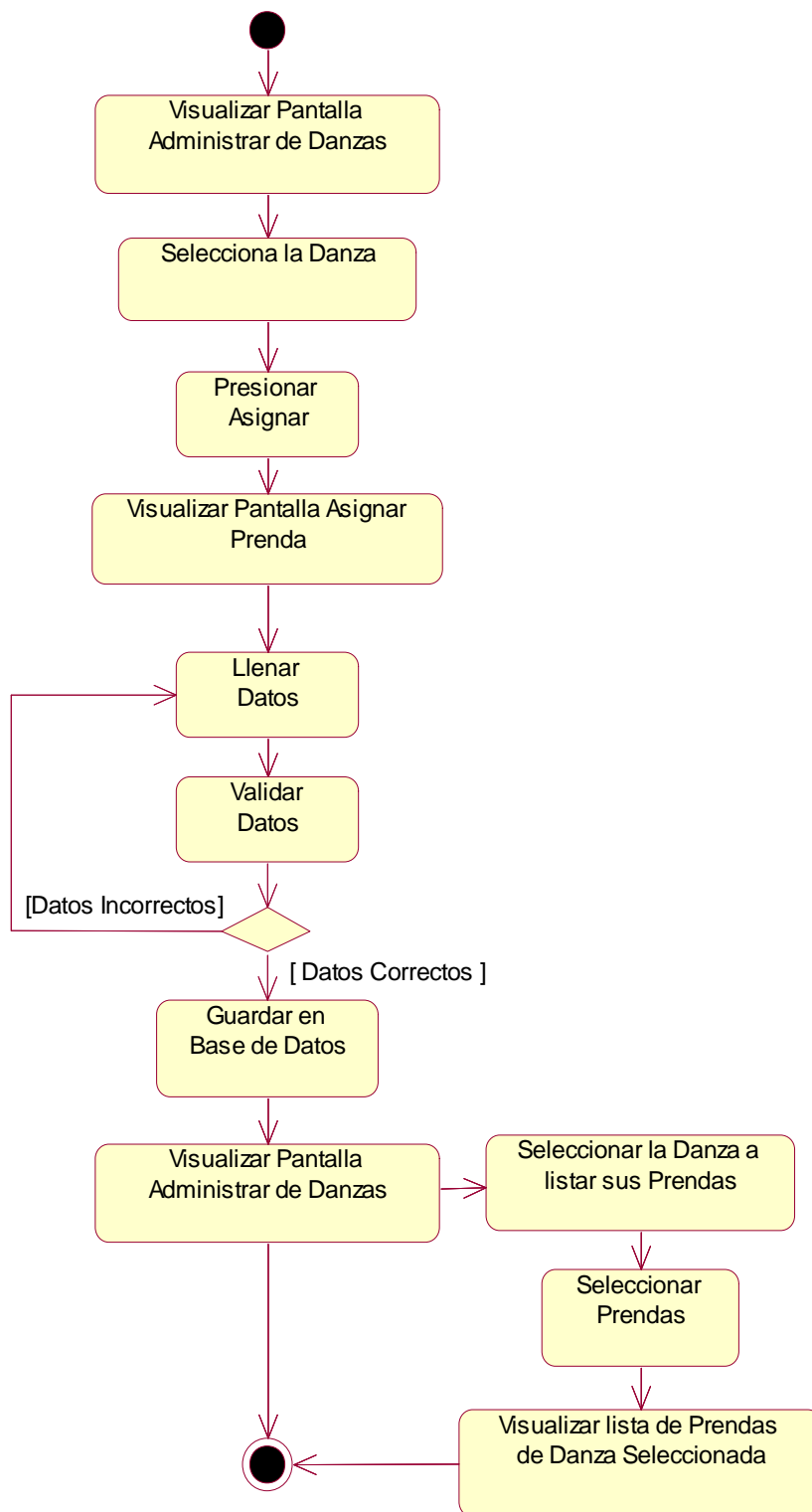
**Diagrama de Actividad: Borrar Imagen****Figura 65: Diagrama de Actividad: Borrar Imagen**

*Diagrama de Actividad: Nueva Danza***Figura 66: Diagrama de Actividad: Nueva Danza**

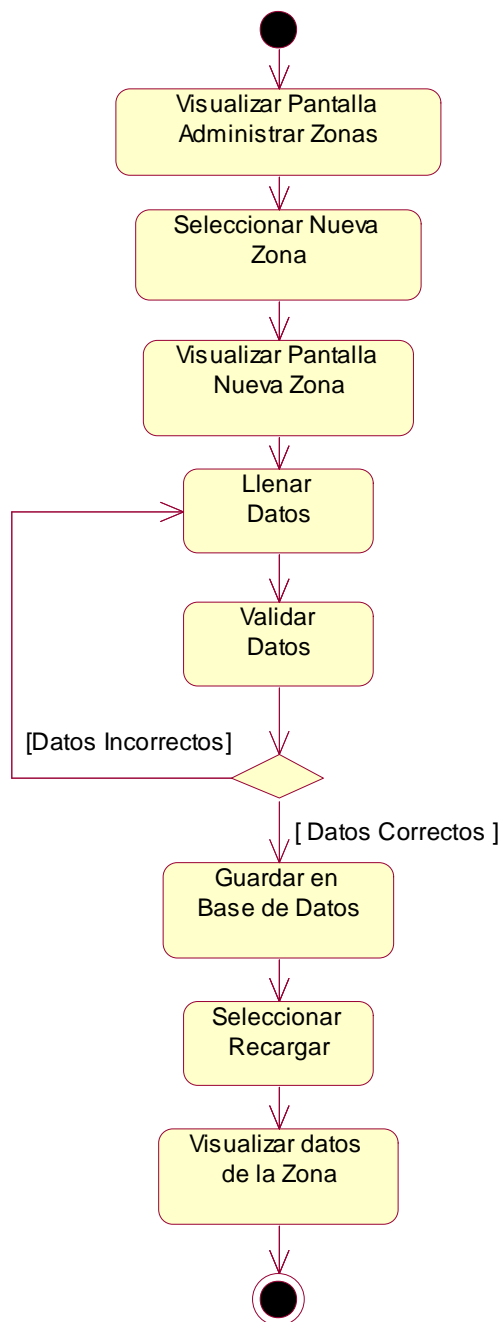
**Diagrama de Actividad: Modificar Danza****Figura 67: Diagrama de Actividad: Modificar Danza**

**Diagrama de Actividad: Borrar Danza****Figura 68: Diagrama de Actividad: Borrar Danza**

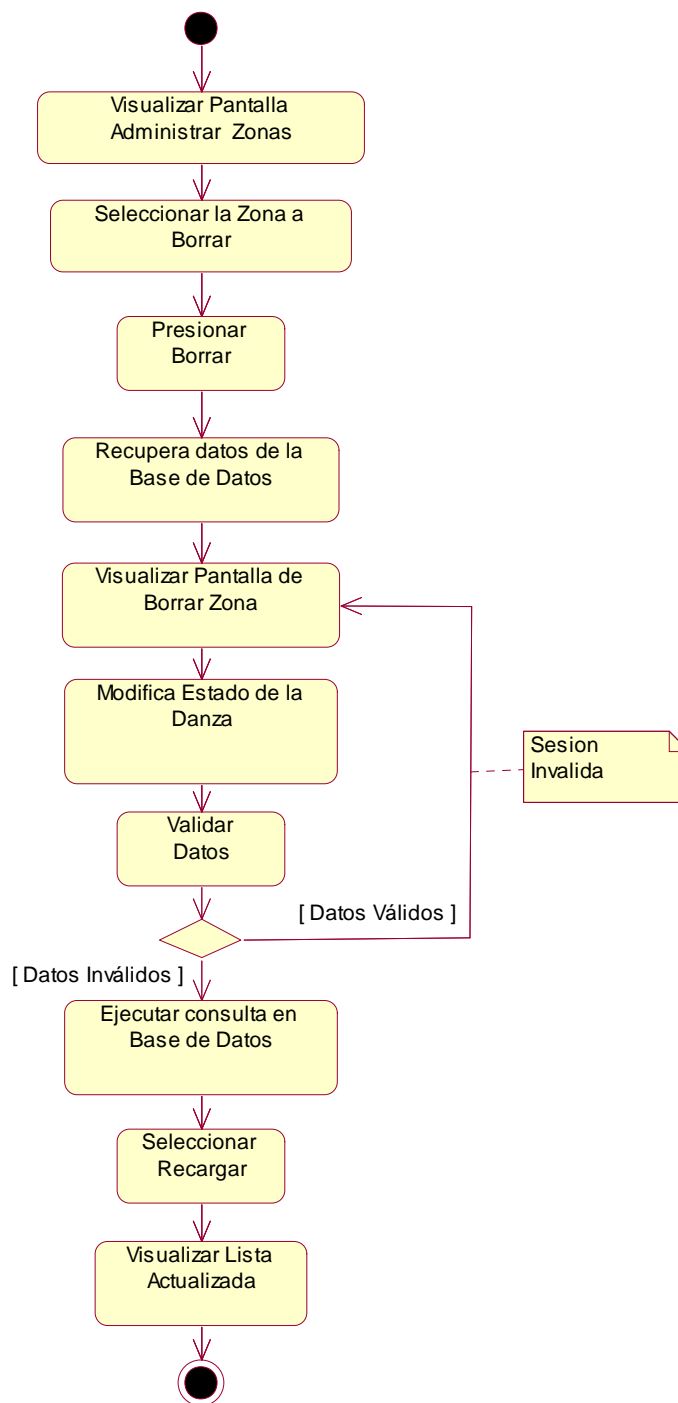
*Diagrama de Actividad: Asignar Prenda a Danza*

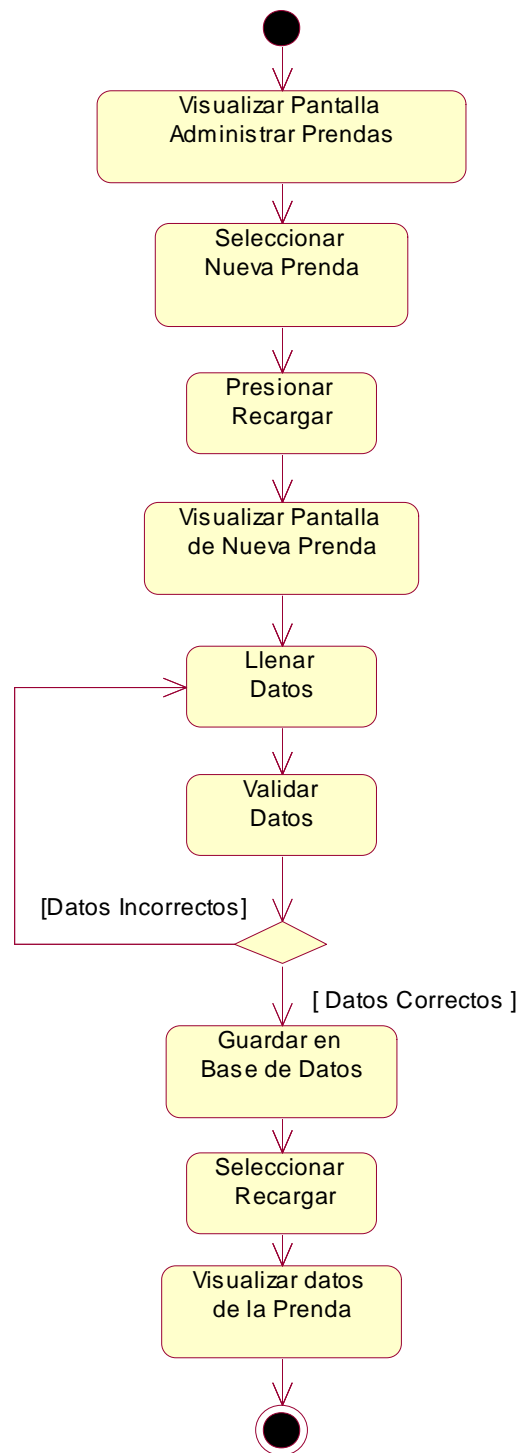


**Figura 69: Diagrama de Actividad: Asignar Prenda a Danza**

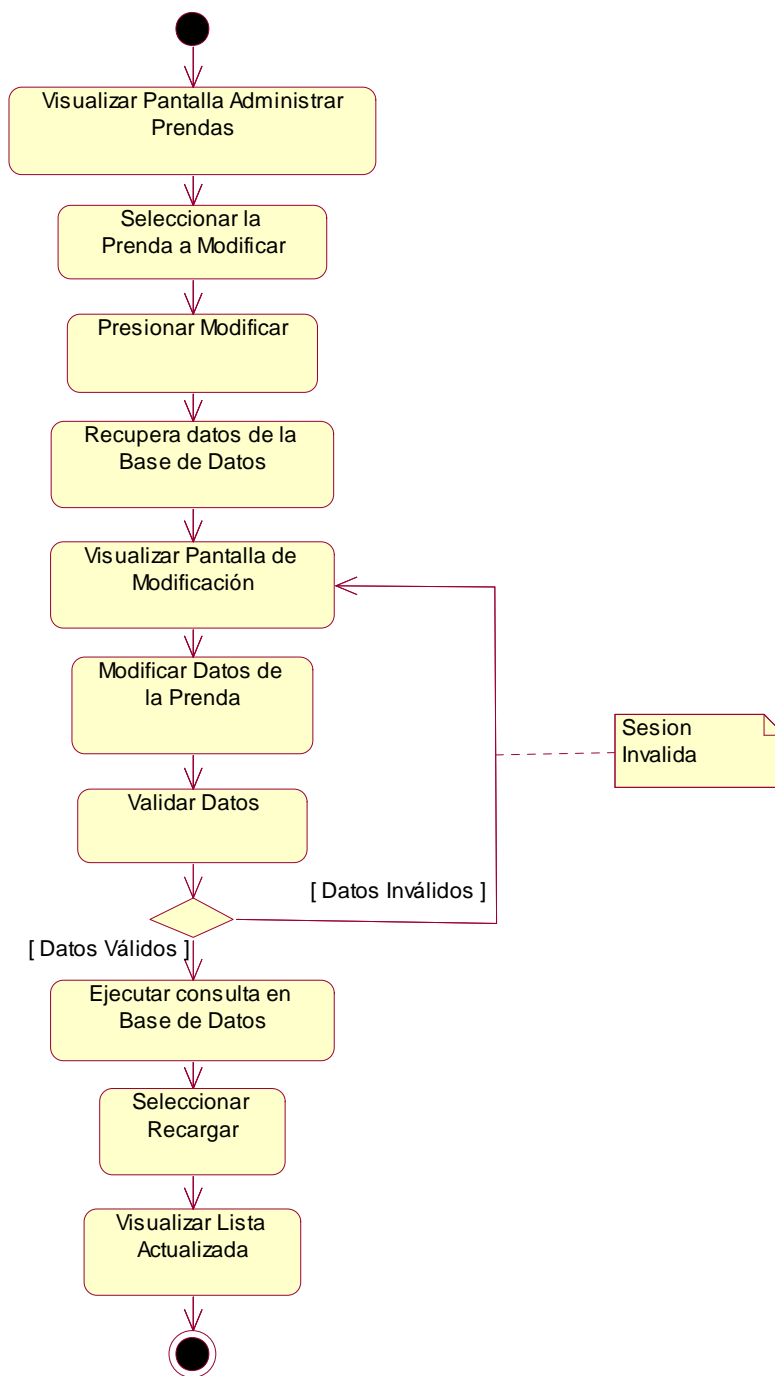
*Diagrama de Actividad: Nueva Zona***Figura 70: Diagrama de Actividad: Nueva Zona**



**Diagrama de Actividad: Borrar Zona****Figura 71: Diagrama de Actividad: Borrar Zona**

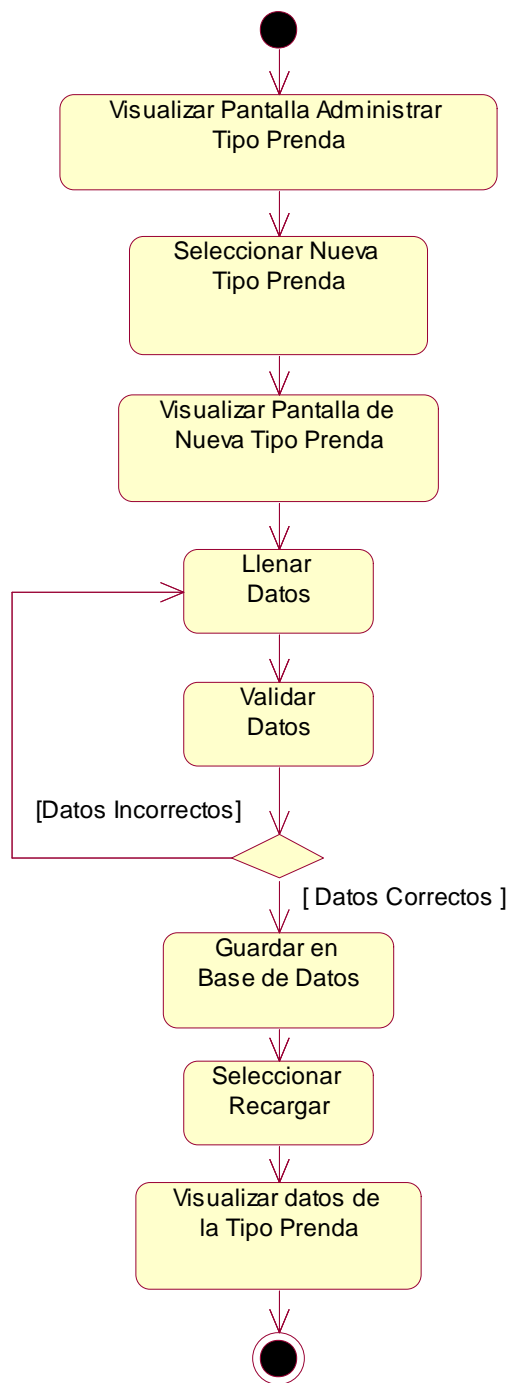
*Diagrama de Actividad: Nueva Prenda***Figura 72: Diagrama de Actividad: Nueva Prenda**

*Diagrama de Actividad: Modificar Prenda*

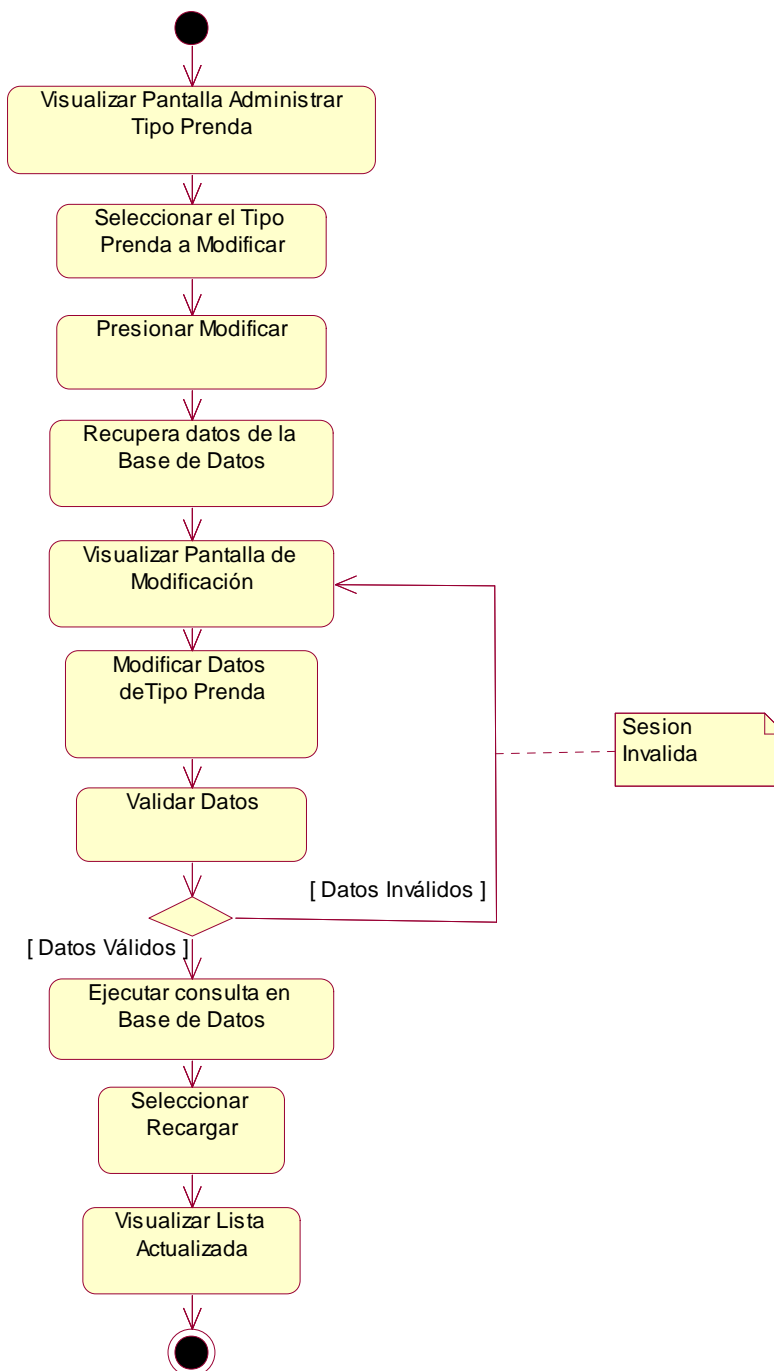


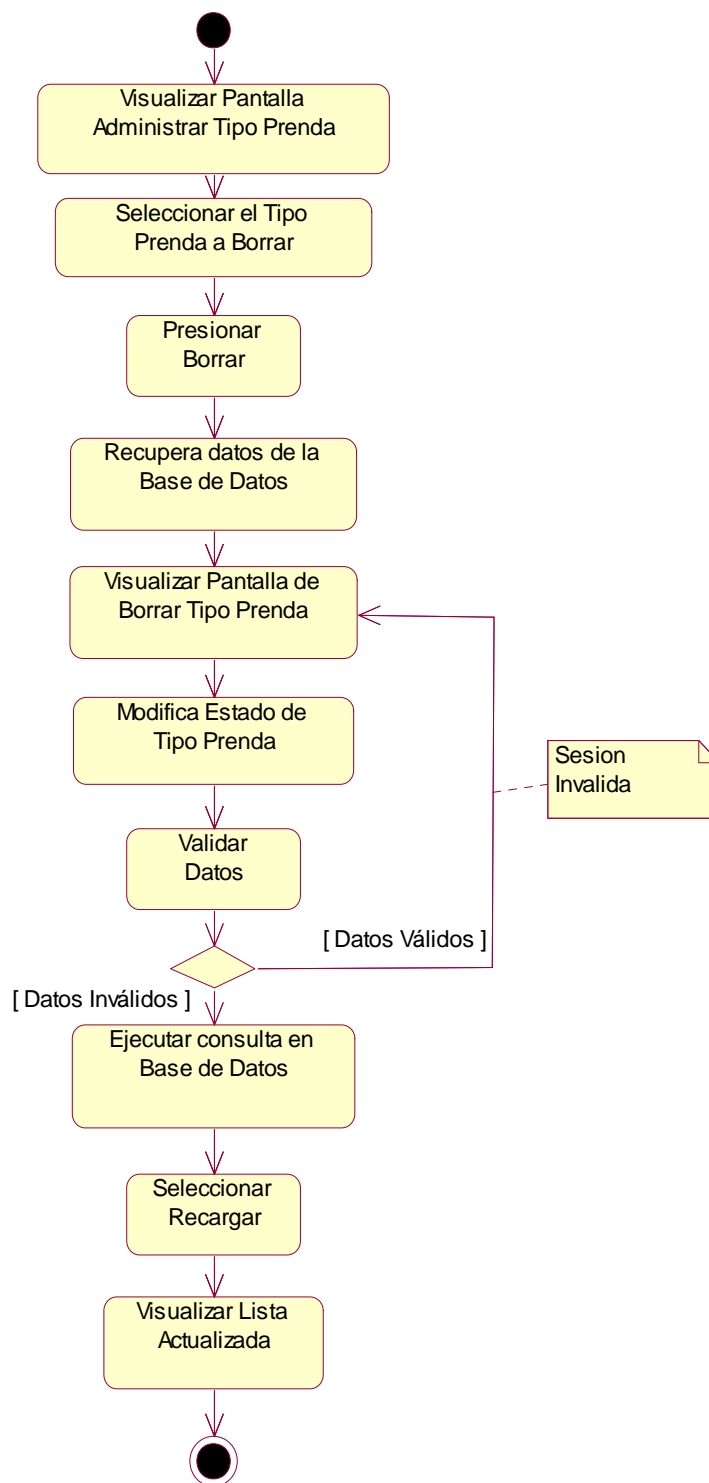
**Figura 73: Diagrama de Actividad: Modificar Prenda**

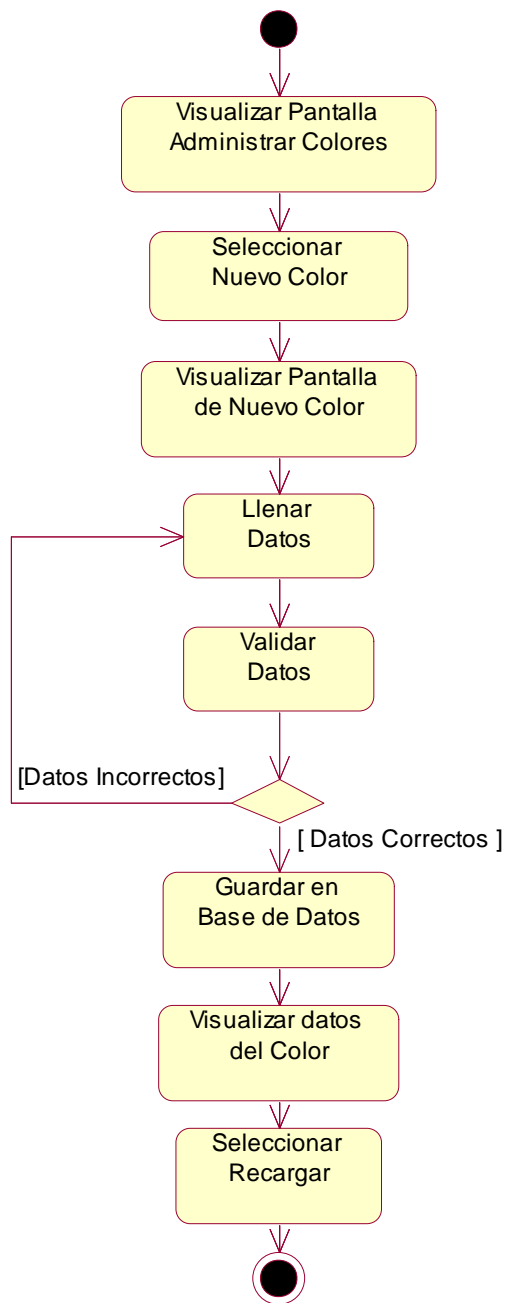
*Diagrama de Actividad: Nuevo Tipo Prenda*

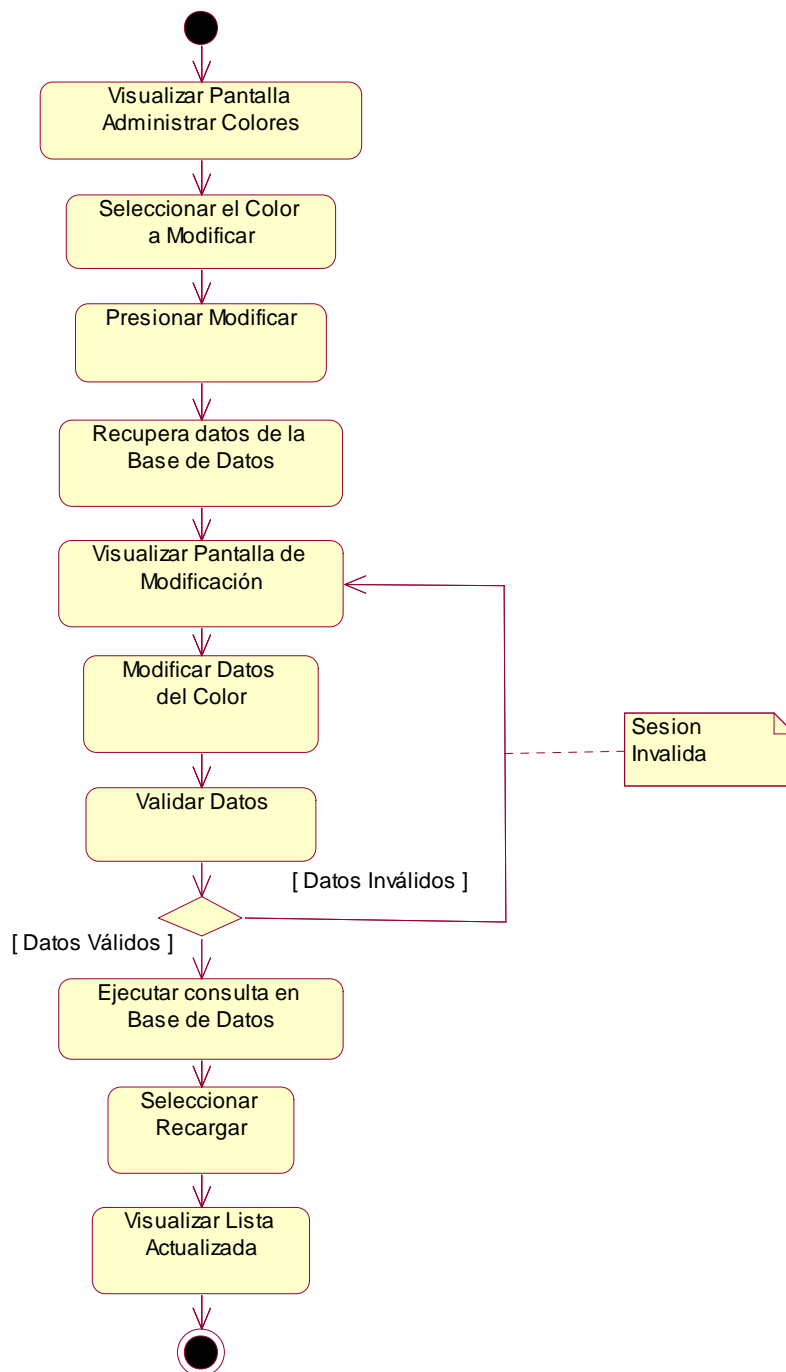


**Figura 74: Diagrama de Actividad: Nuevo Tipo Prenda**

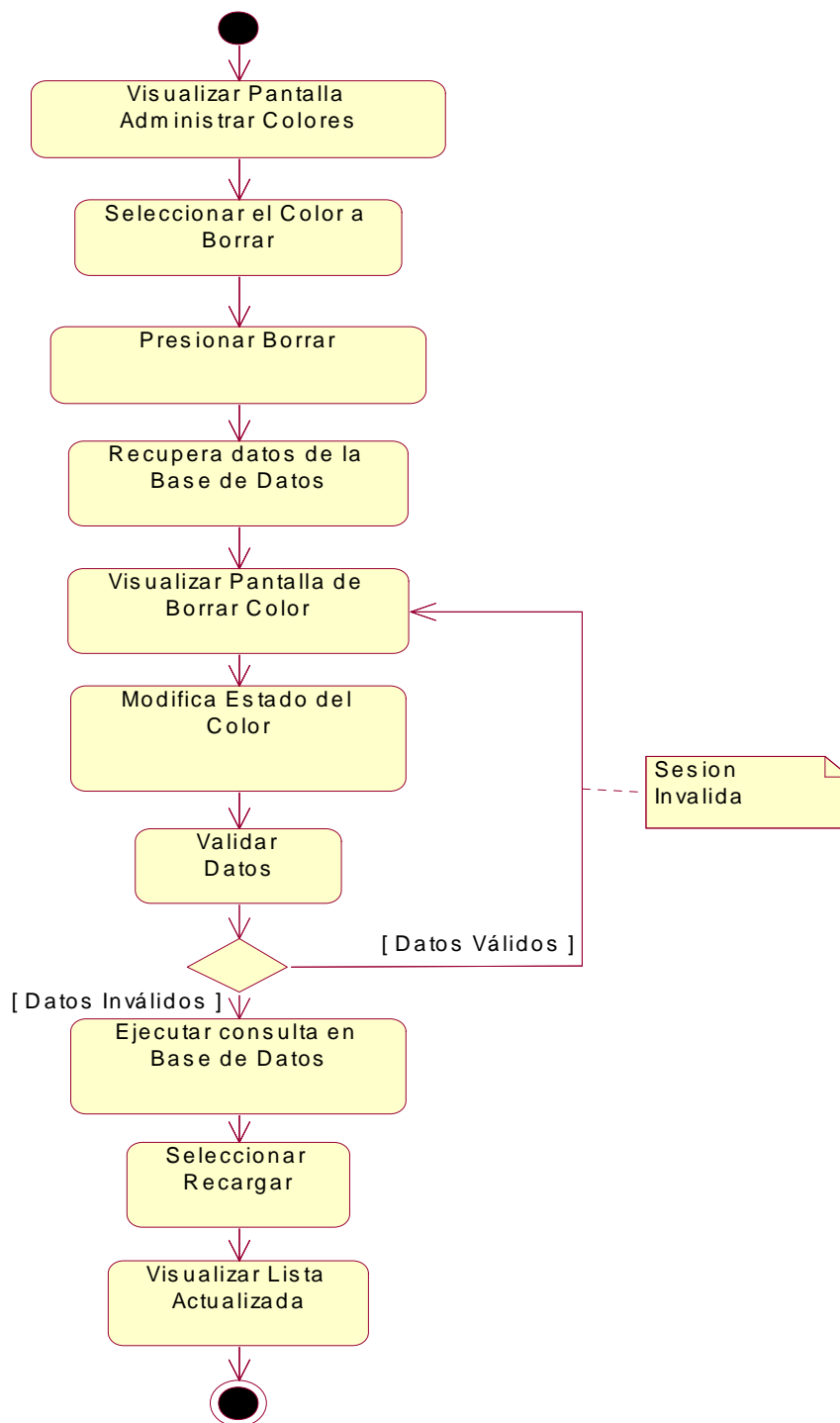
*Diagrama de Actividad: Modificar Tipo Prenda***Figura 75: Diagrama de Actividad: Modificar Tipo Prenda**

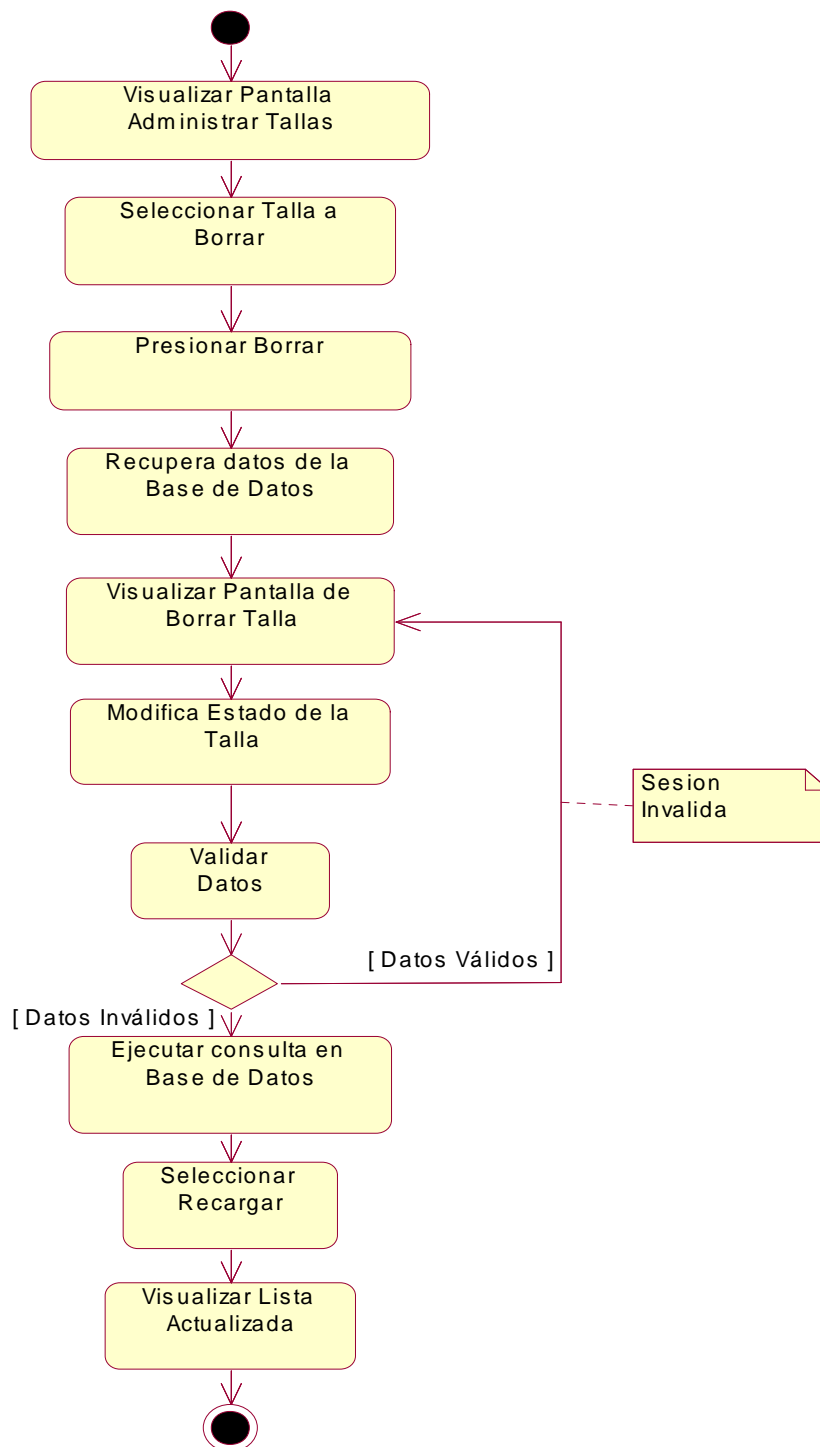
**Diagrama de Actividad: Borrar Tipo Prenda****Figura 76: Diagrama de Actividad: Borrar Tipo Prenda**

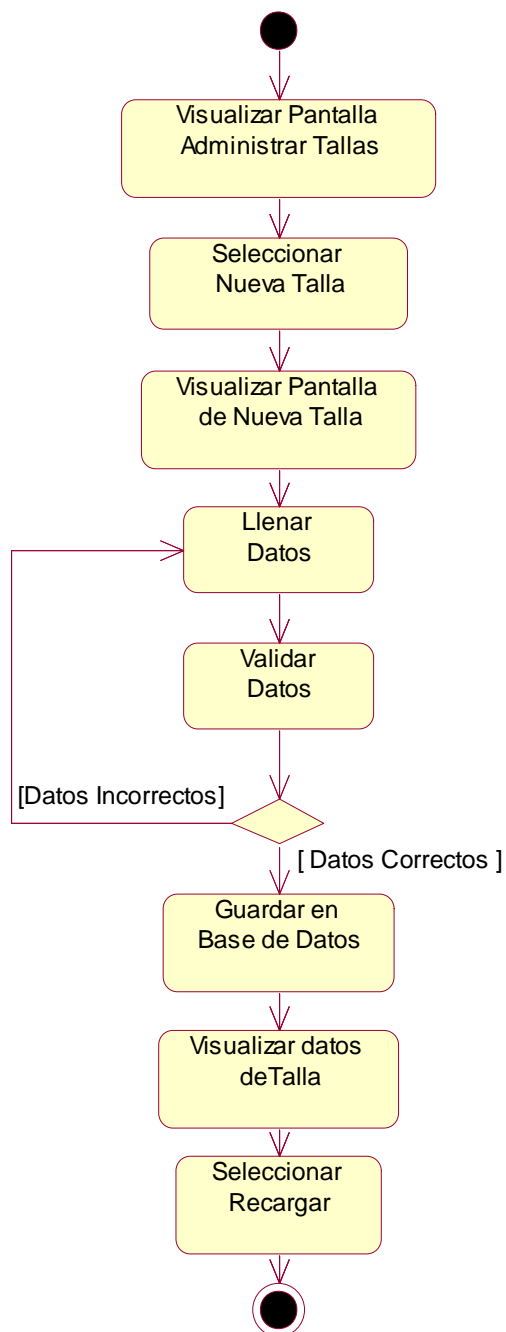
*Diagrama de Actividad: Nuevo Color***Figura 77: Diagrama de Actividad: Nuevo Color**

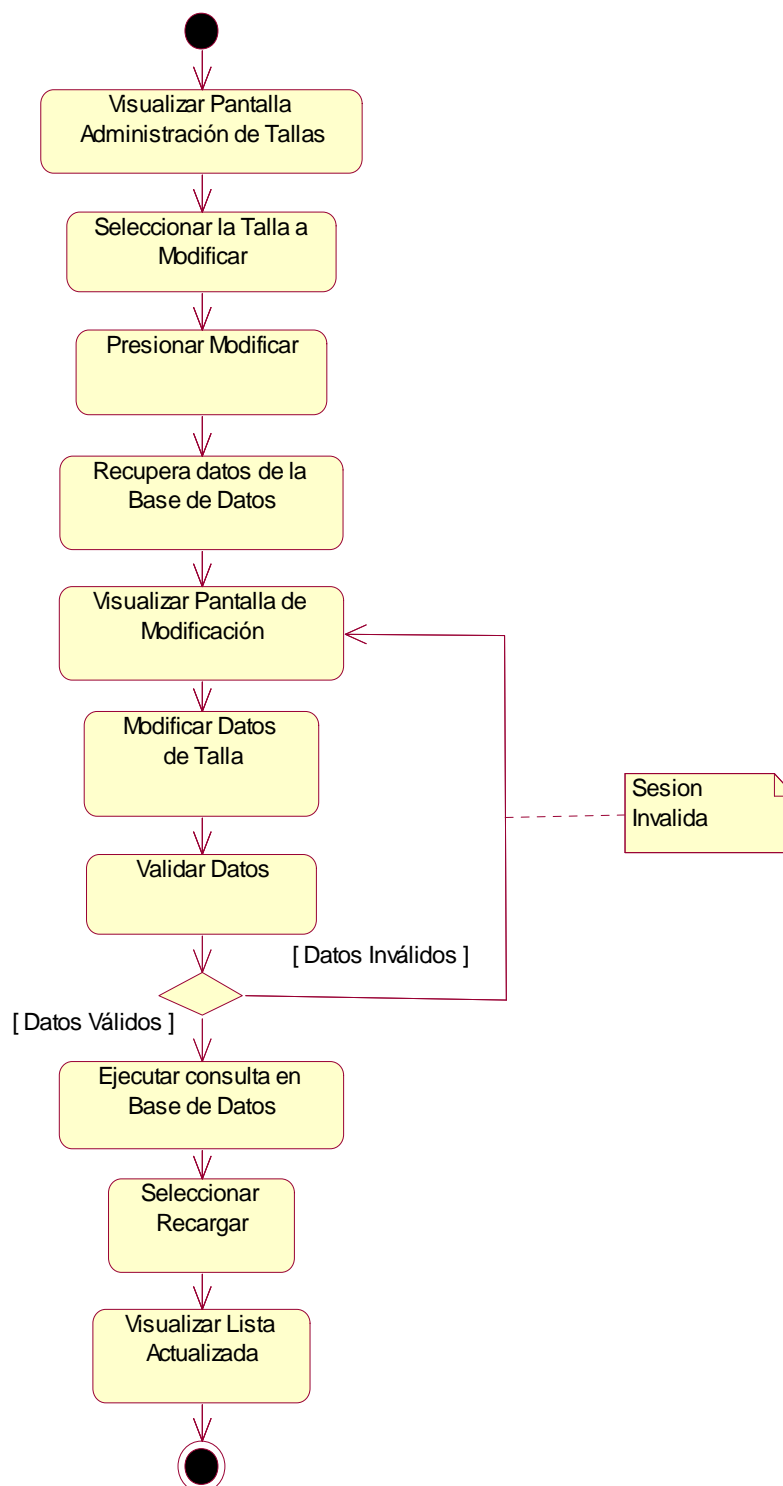
*Diagrama de Actividad: Modificar Color***Figura 78: Diagrama de Actividad: Modificar Color**



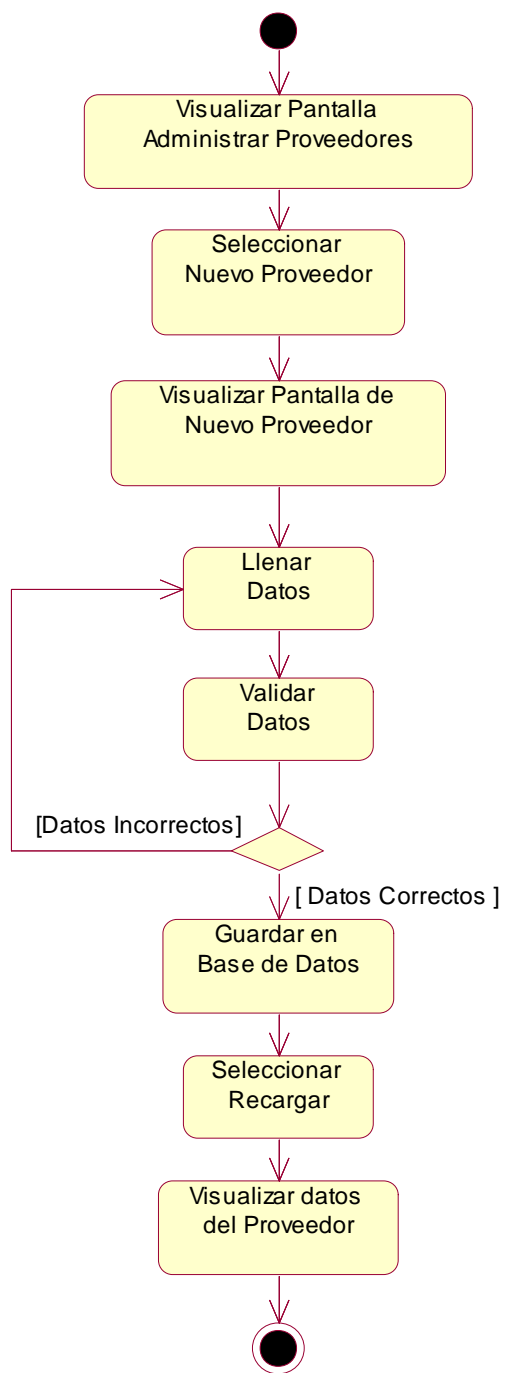
**Diagrama de Actividad: Borrar Color****Figura 79: Diagrama de Actividad: Borrar Color**

**Diagrama de Actividad: Borrar Talla****Figura 80: Diagrama de Actividad: Borrar Talla**

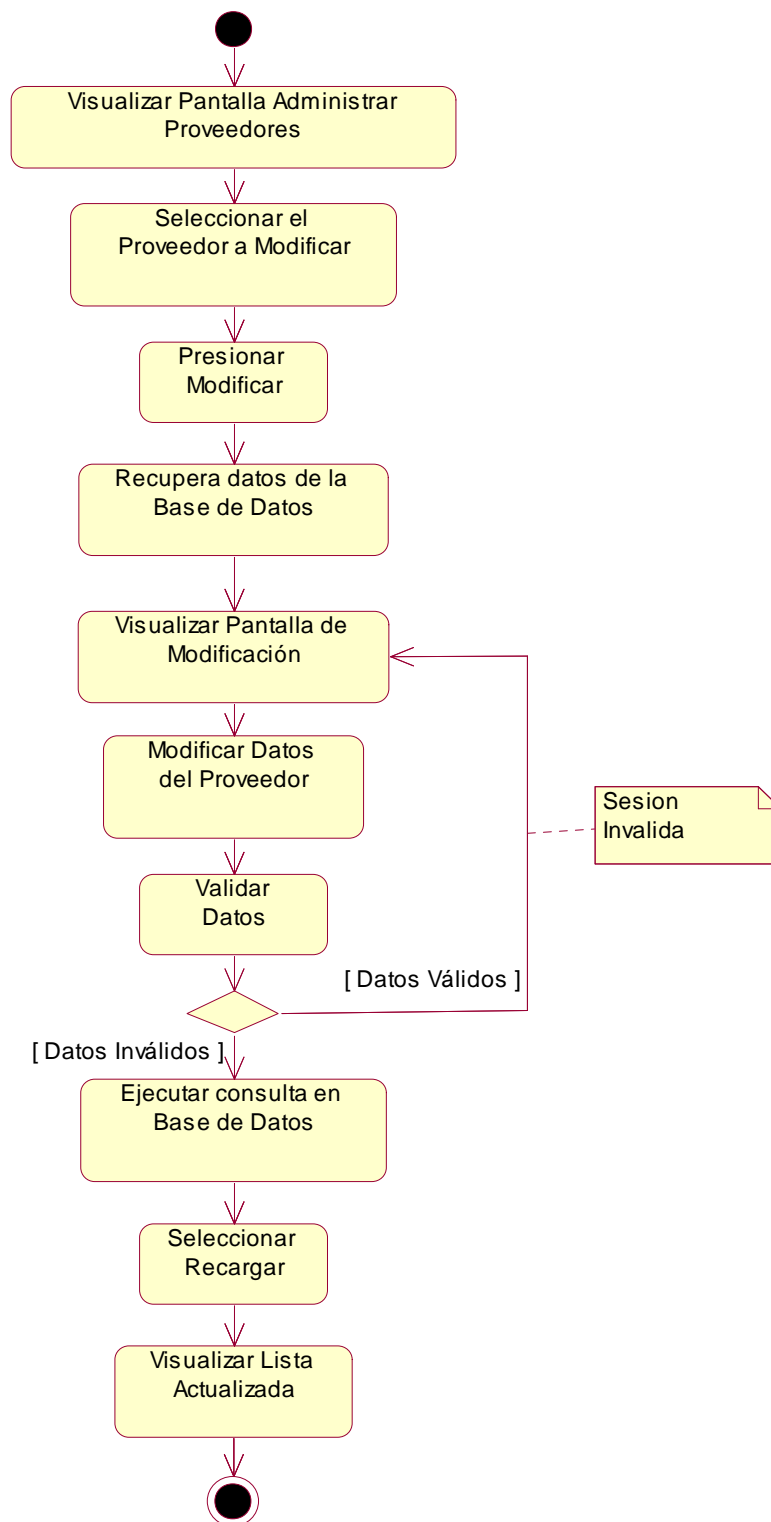
*Diagrama de Actividad: Nueva Talla***Figura 81: Diagrama de Actividad: Nueva Talla**

*Diagrama de Actividad: Modificar Talla***Figura 82: Diagrama de Actividad: Modificar Talla**

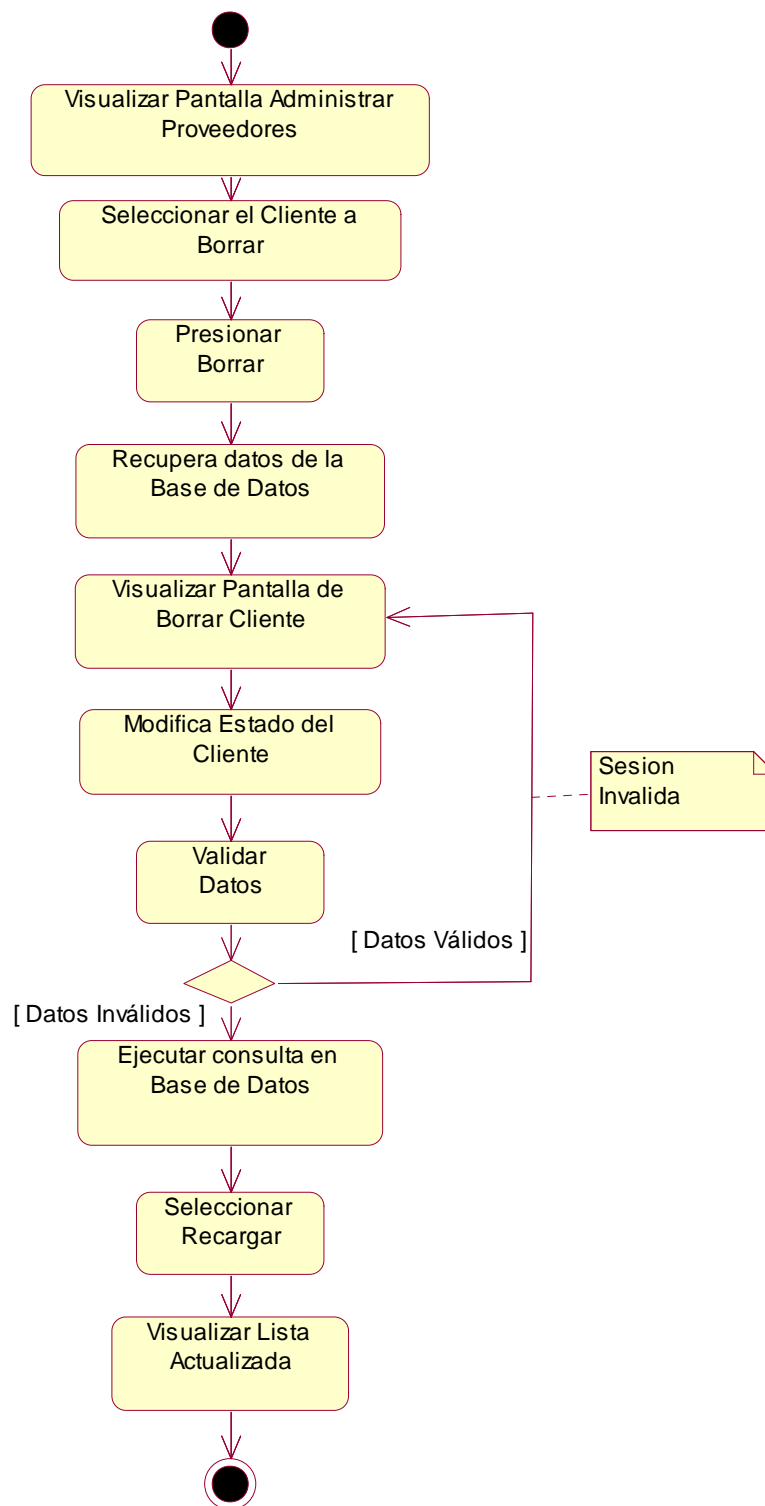
*Diagrama de Actividad: Nuevo Proveedor*



**Figura 83: Diagrama de Actividad: Nuevo Proveedor**

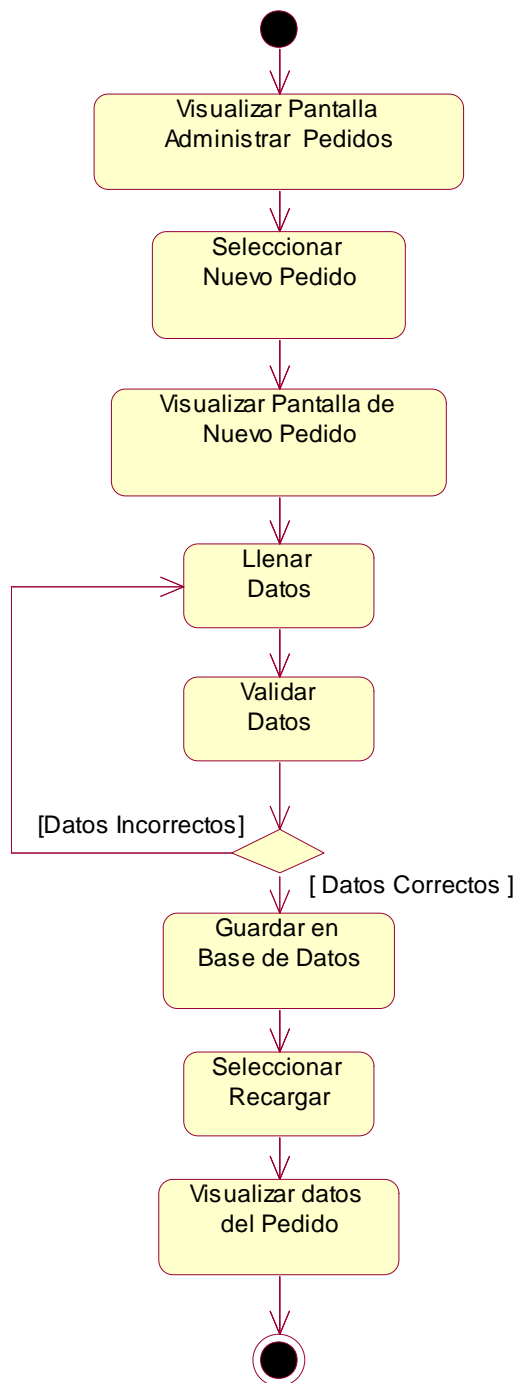
**Diagrama de Actividad: Modificar Proveedor****Figura 84: Diagrama de Actividad: Modificar Proveedor**

*Diagrama de Actividad: Borrar Proveedor*



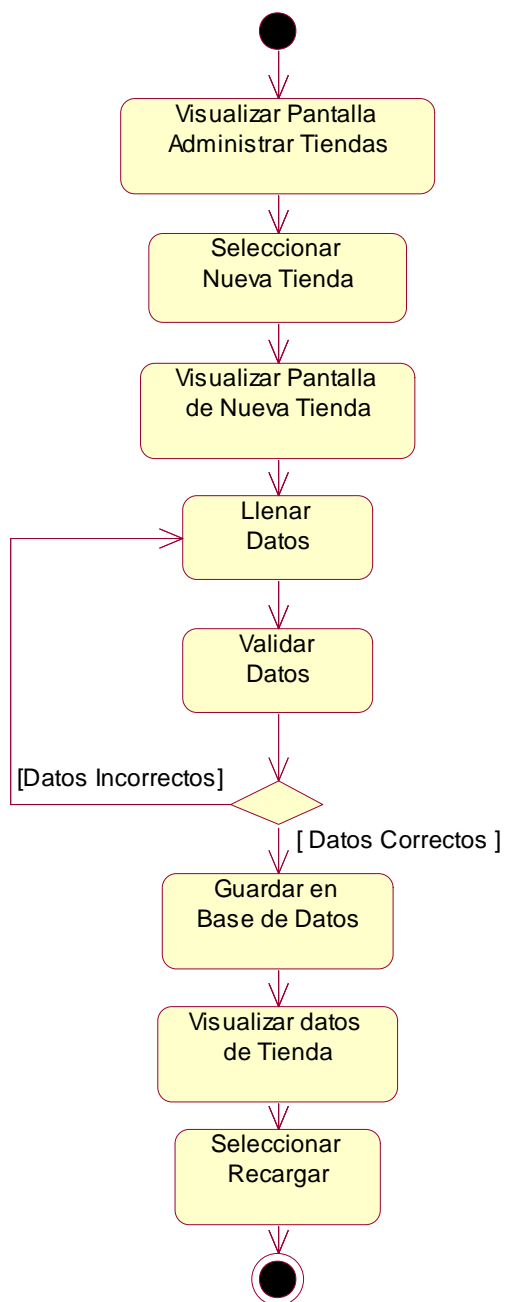
**Figura 85: Diagrama de Actividad: Borrar Proveedor**

*Diagrama de Actividad: Nuevo Pedido*

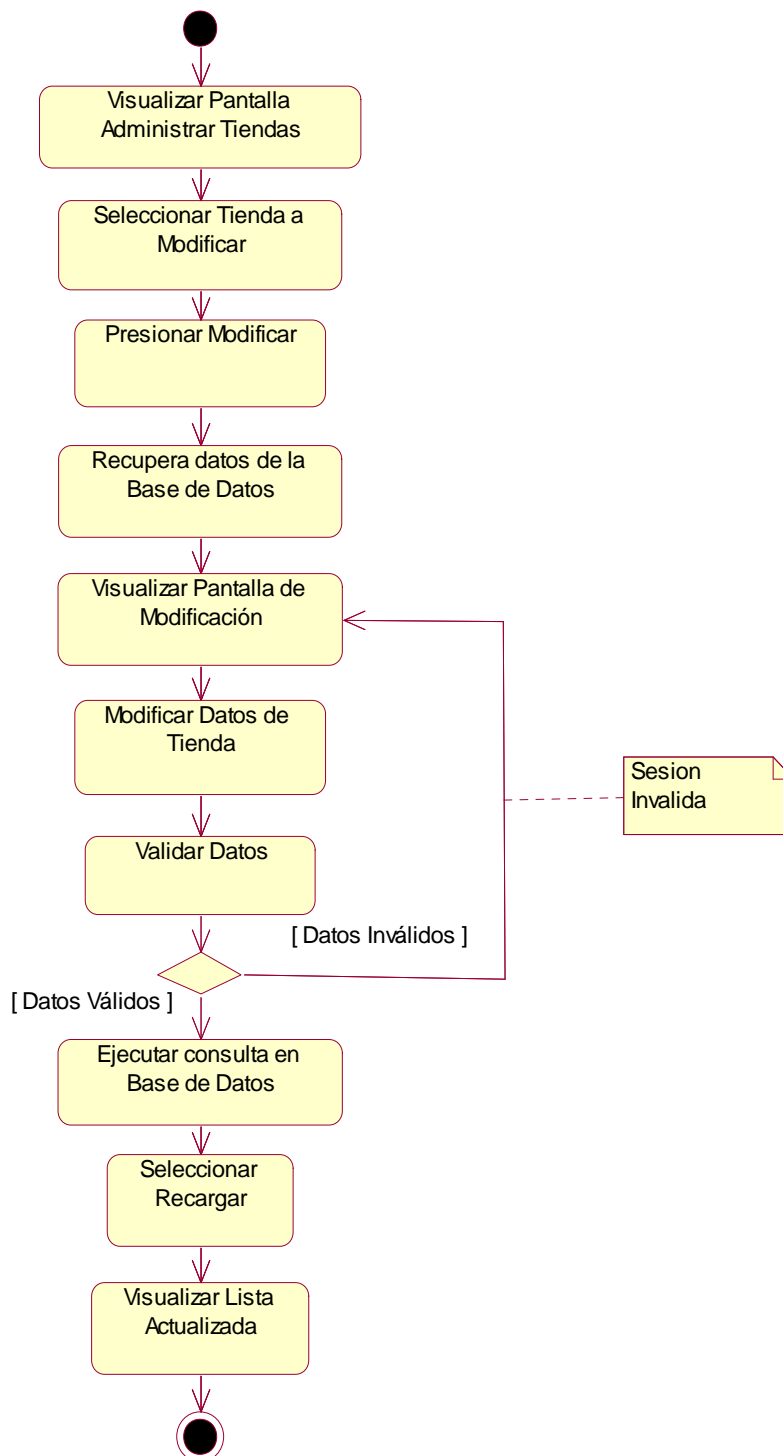


**Figura 86: Diagrama de Actividad: Nuevo Pedido**

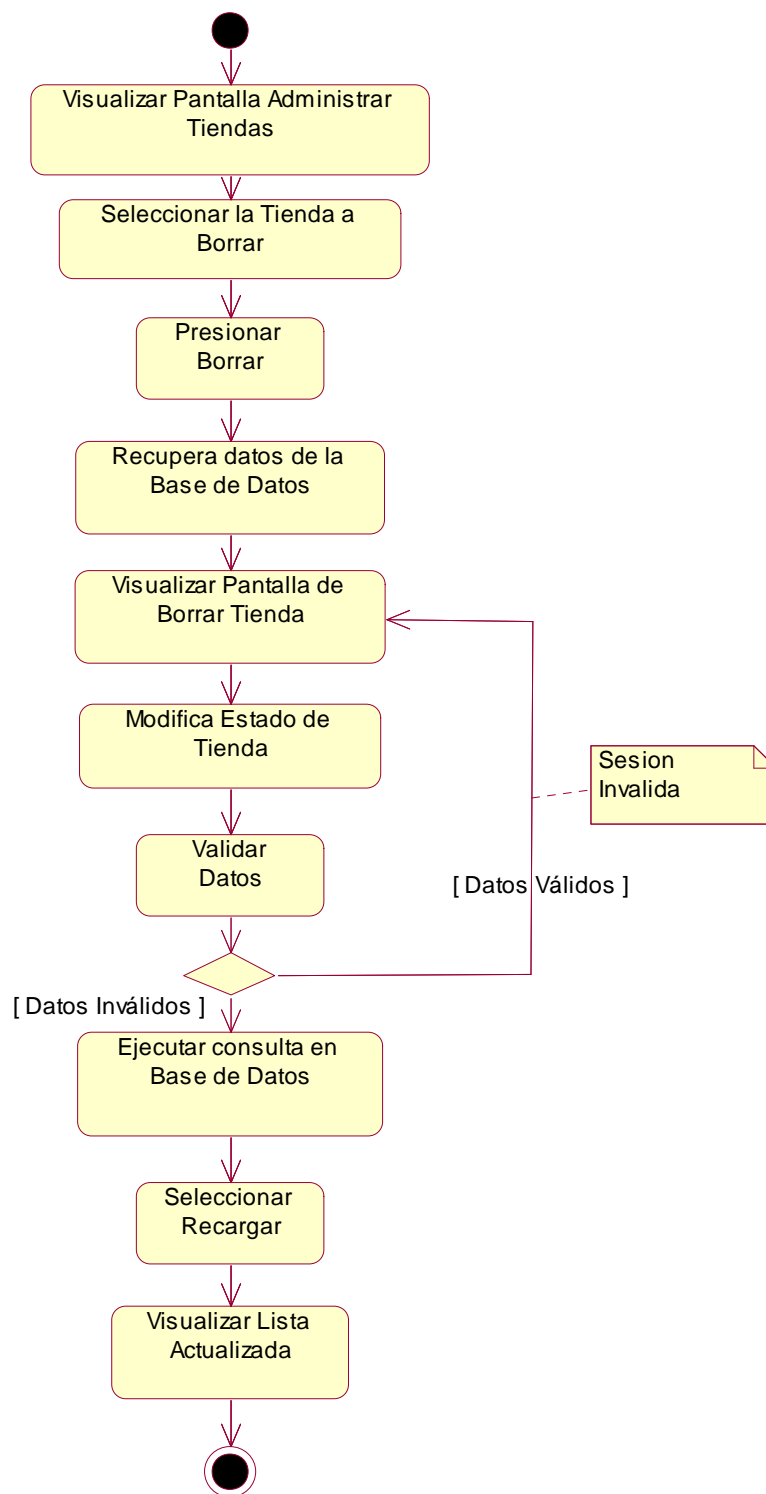


*Diagrama de Actividad: Nueva Tienda***Figura 87: Diagrama de Actividad: Nueva Tienda**

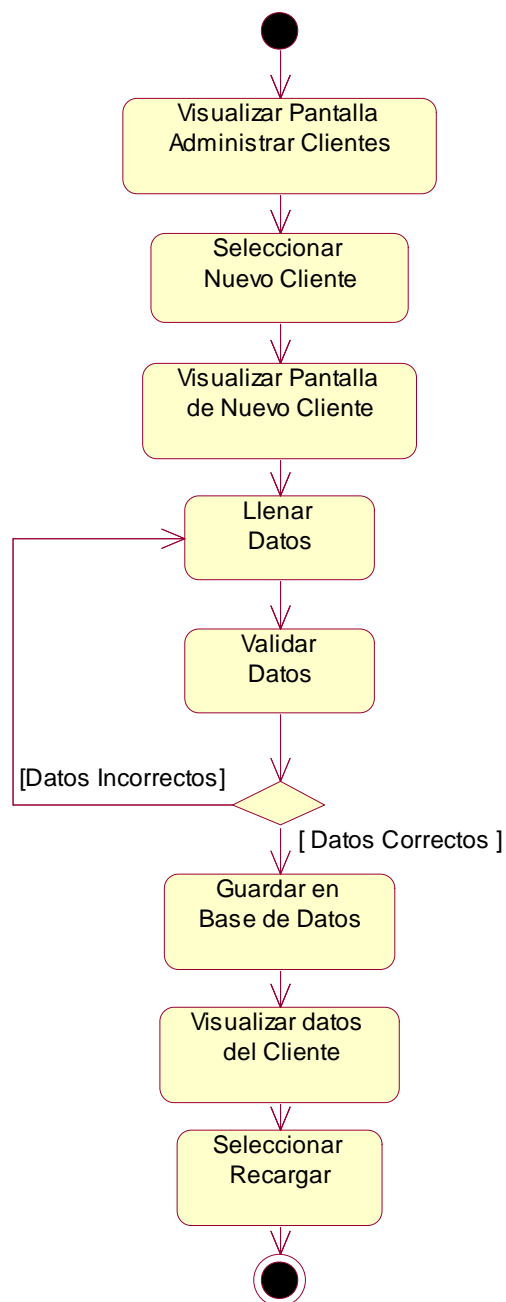
*Diagrama de Actividad: Modificar Tiendas*



**Figura 88: Diagrama de Actividad: Modificar Tiendas**

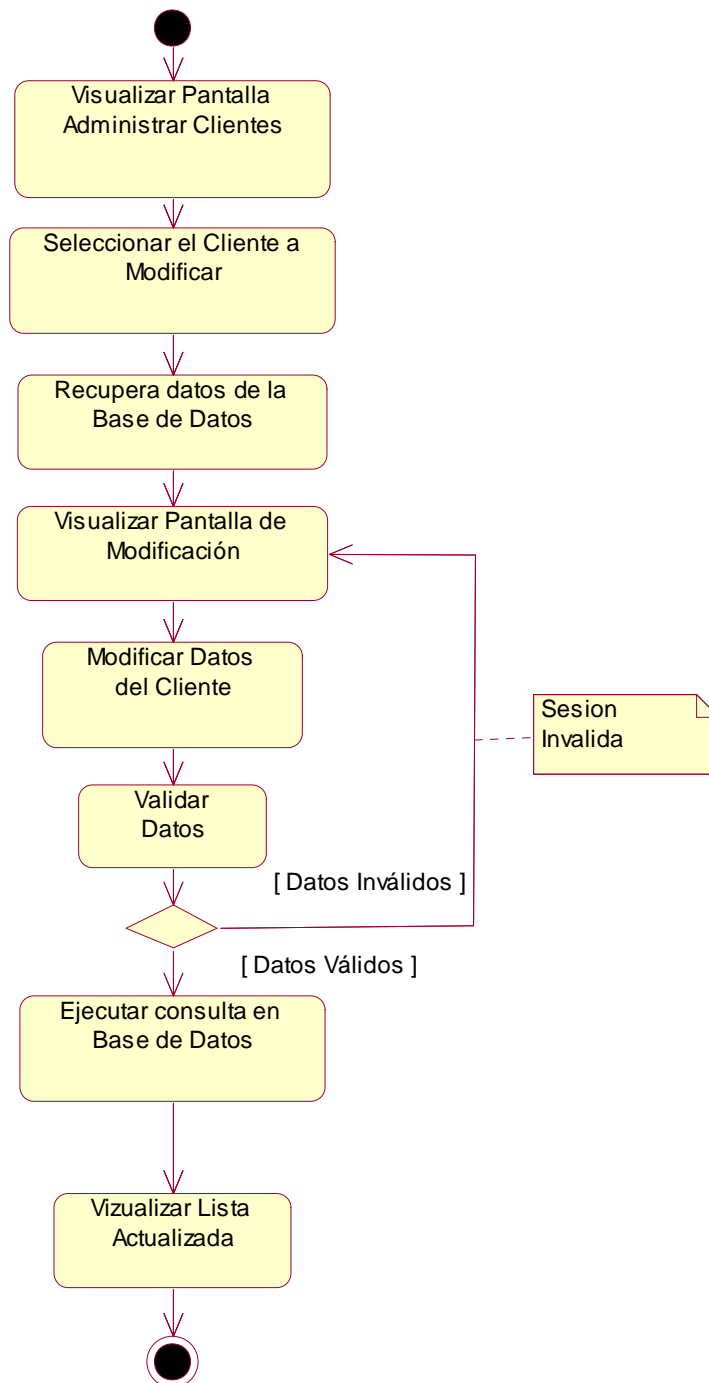
**Diagrama de Actividad: Borrar Tienda****Figura 89: Diagrama de Actividad: Borrar Tienda**

*Diagrama de Actividad: Nuevo Cliente*

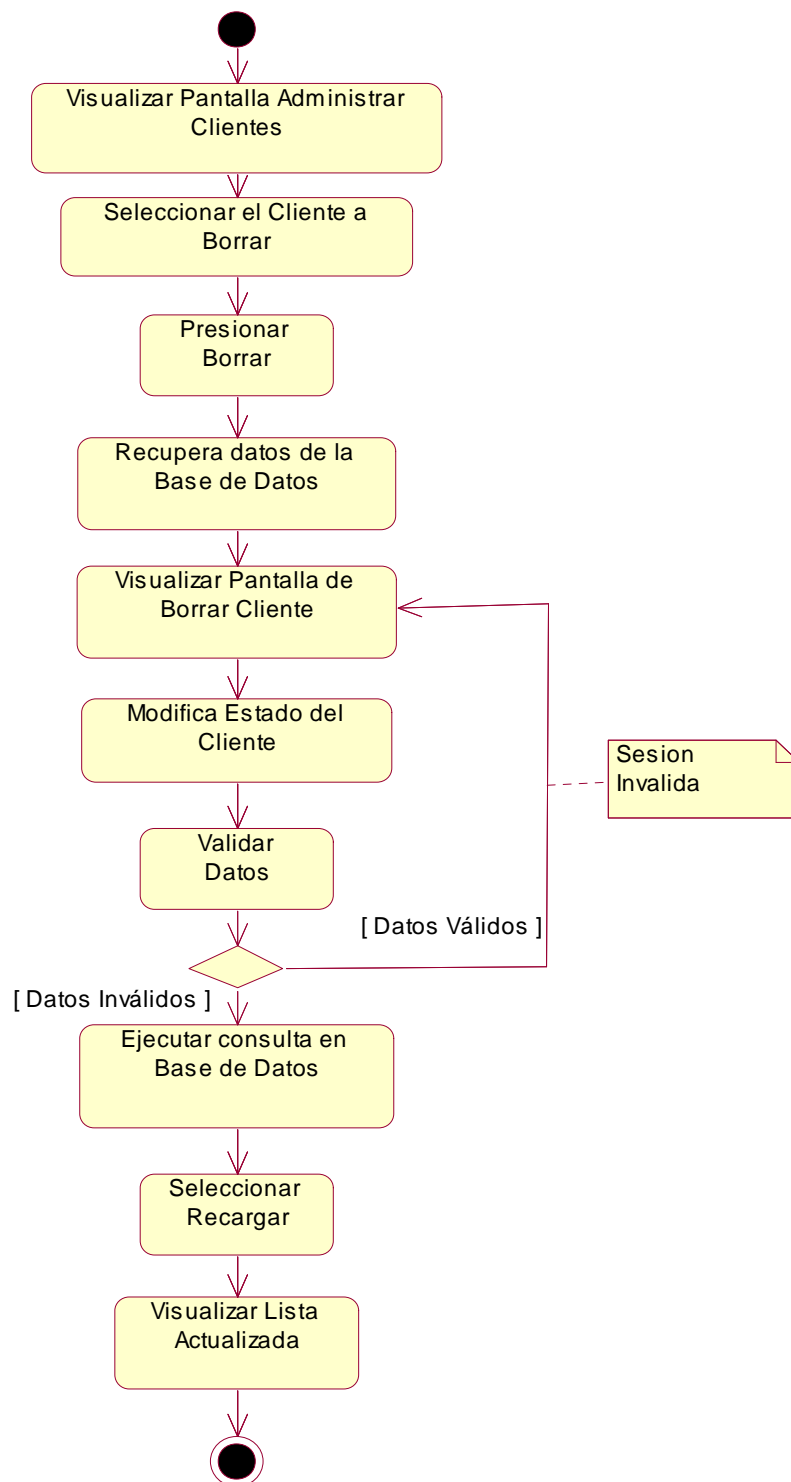


**Figura 90: Diagrama de Actividad: Nuevo Cliente**

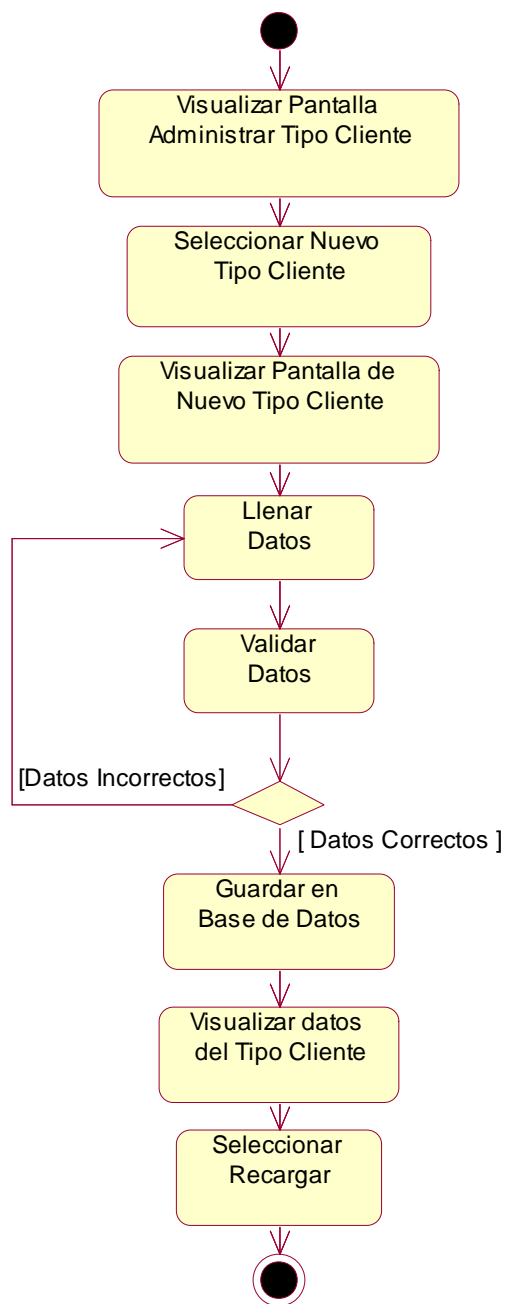
*Diagrama de Actividad: Modificar Cliente*



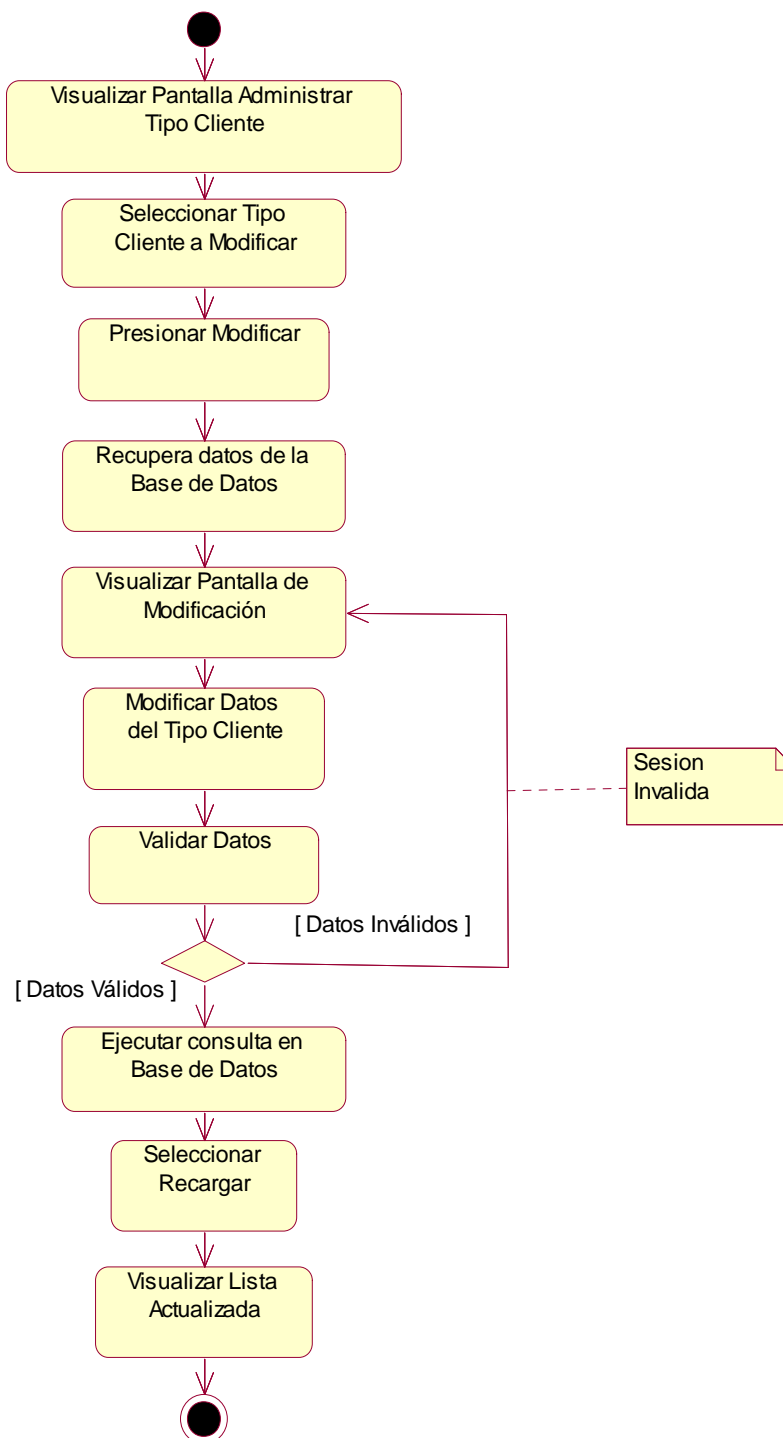
**Figura 91: Diagrama de Actividad: Modificar Cliente**

**Diagrama de Actividad: Borrar Cliente****Figura 92: Diagrama de Actividad: Borrar Cliente**

*Diagrama de Actividad: Nuevo Tipo Cliente*

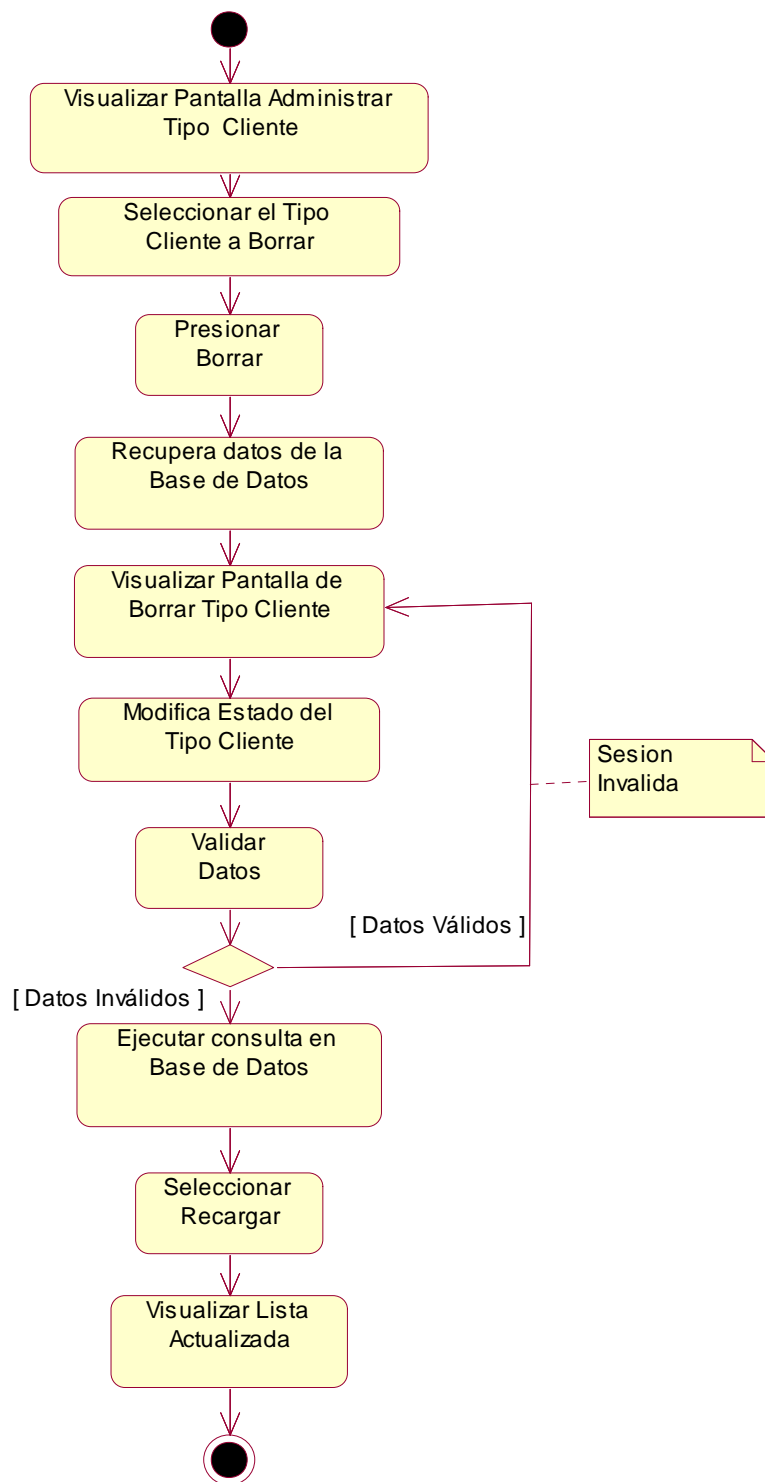


**Figura 93: Diagrama de Actividad: Nuevo Tipo Cliente**

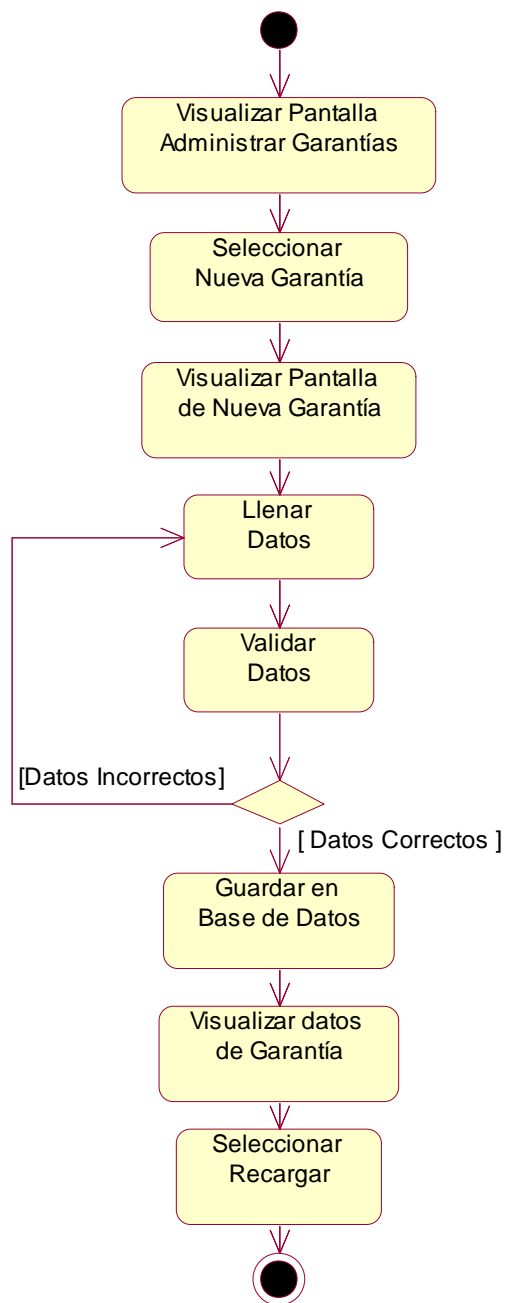
*Diagrama de Actividad: Modificar Tipo Cliente***Figura 94: Diagrama de Actividad: Modificar Tipo Cliente**

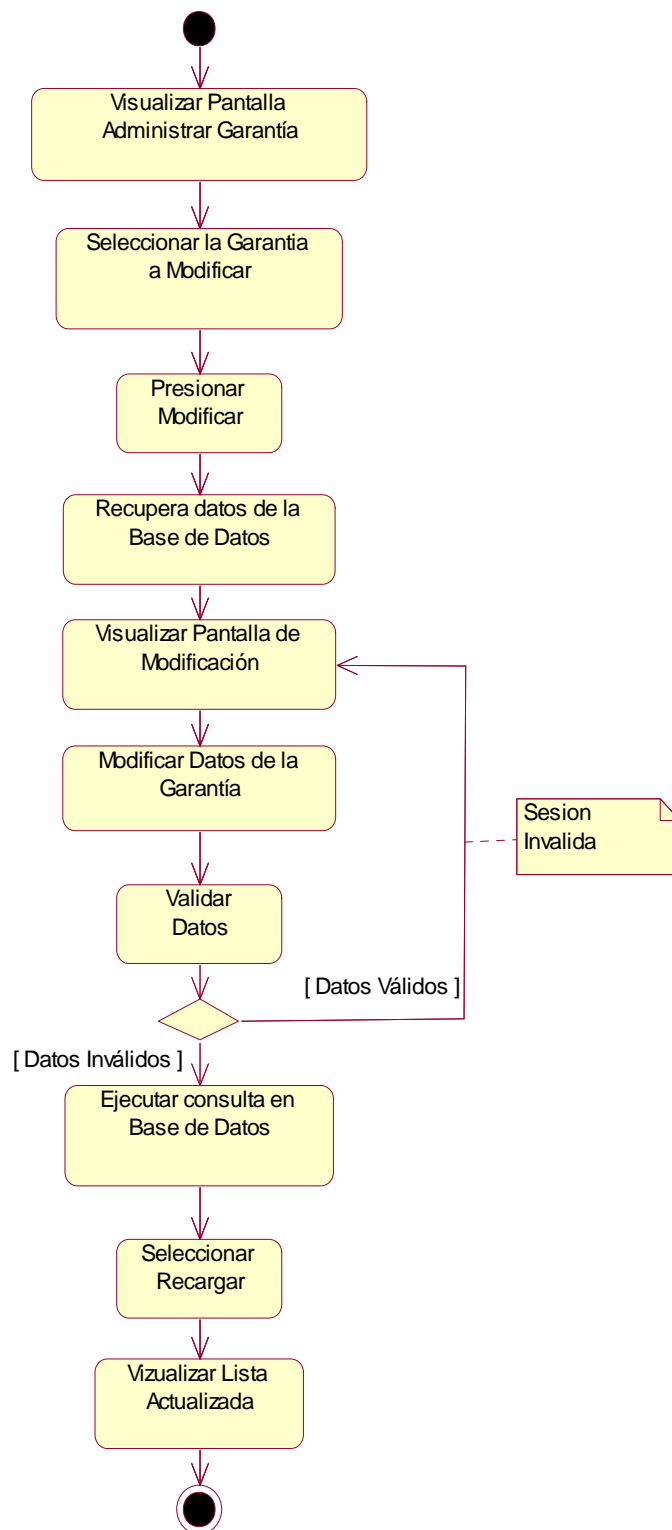


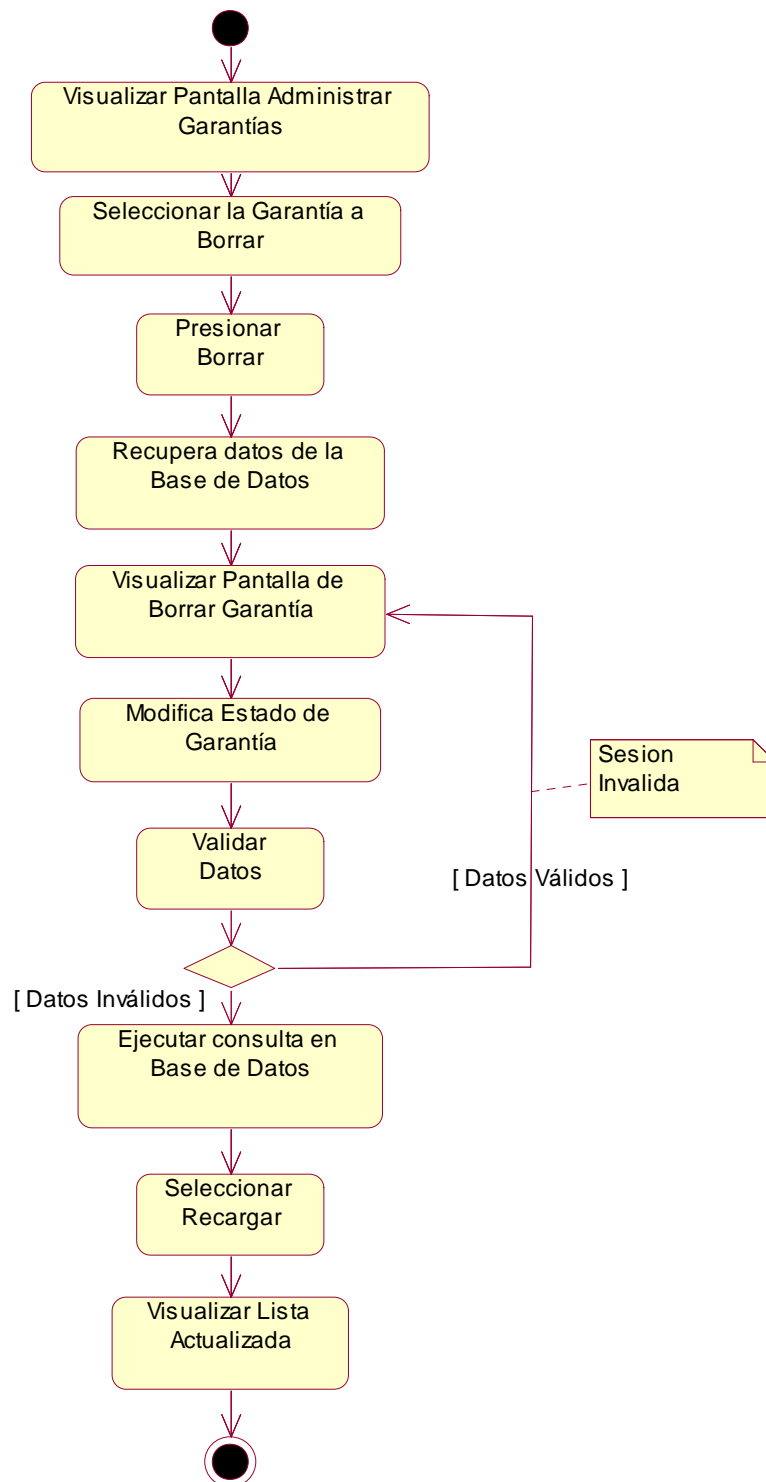
*Diagrama de Actividad: Borrar Tipo Cliente*

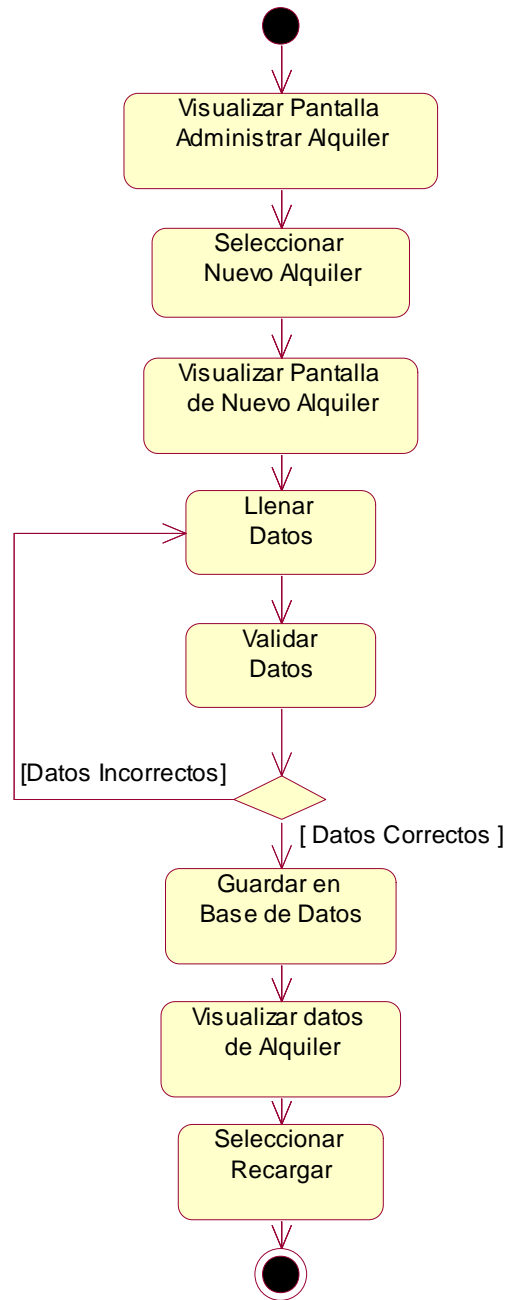


**Figura 95: Diagrama de Actividad: Borrar Tipo Cliente**

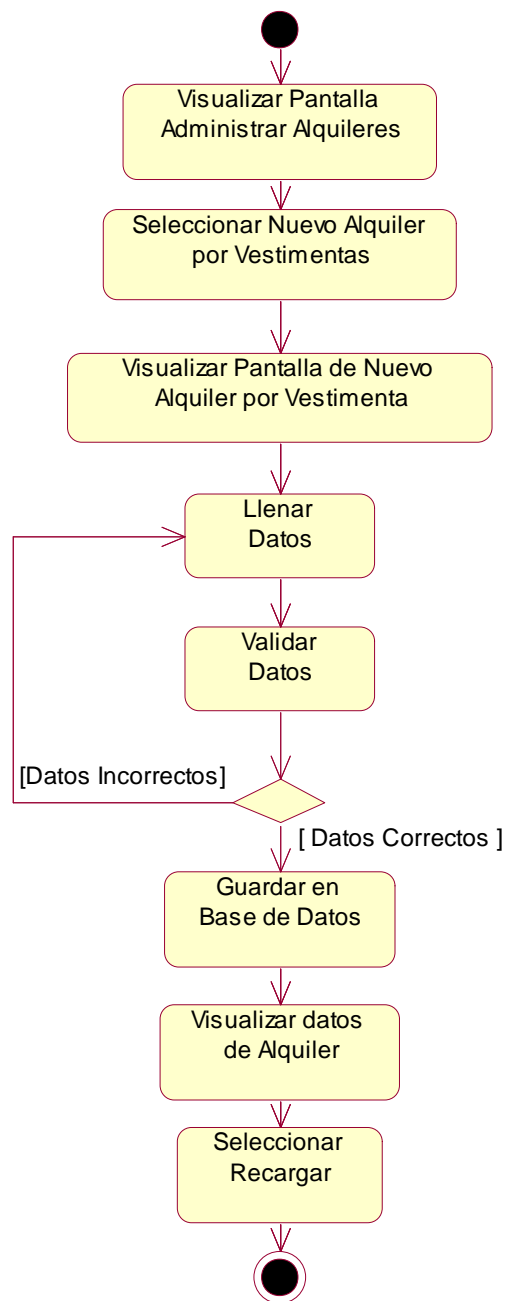
*Diagrama de Actividad: Nueva Garantía***Figura 96: Diagrama de Actividad: Nueva Garantía**

**Diagrama de Actividad: Modificar Garantía****Figura 97: Diagrama de Actividad: Modificar Garantía**

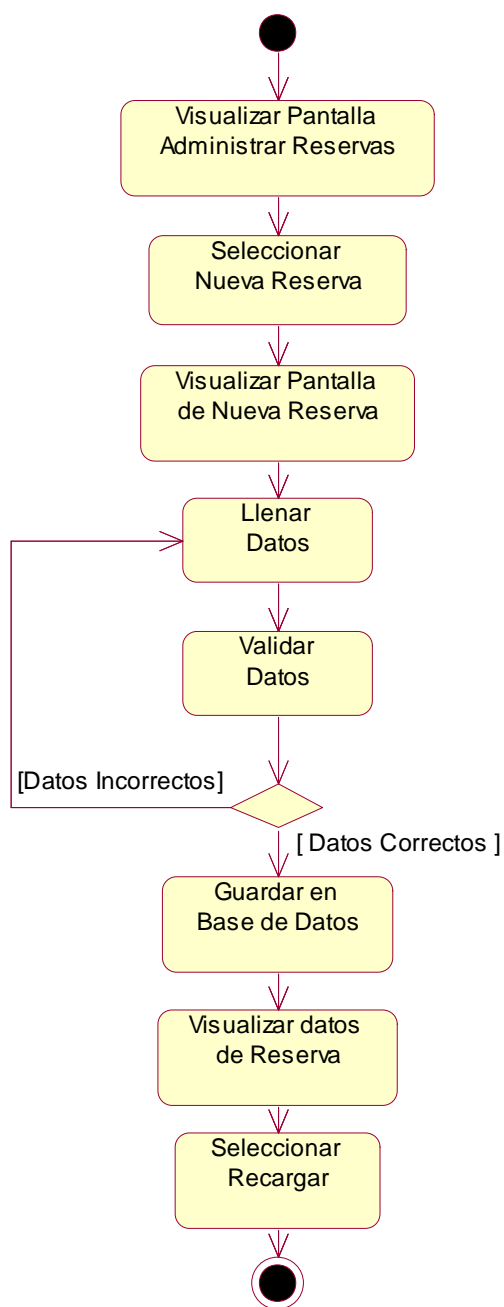
**Diagrama de Actividad: Borrar Garantía****Figura 98: Diagrama de Actividad: Borrar Garantía**

*Diagrama de Actividad: Nuevo Alquiler***Figura 99: Diagrama de Actividad: Nuevo Alquiler**

*Diagrama de Actividad: Nuevo Alquiler Uniforme*



**Figura 100: Diagrama de Actividad: Nuevo Alquiler Uniforme**

*Diagrama de Actividad: Nueva Reserva***Figura 101: Diagrama de Actividad: Nueva Reserva**

## II.1.11 Modelado de Diagrama de Interacción

### II.1.11.1 Introducción

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

### II.1.11.2 Propósito

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

### II.1.11.3 Alcance

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



## II.1.12 Diagrama de Secuencias

### Diagrama de Secuencia: Ingresar al Sistema

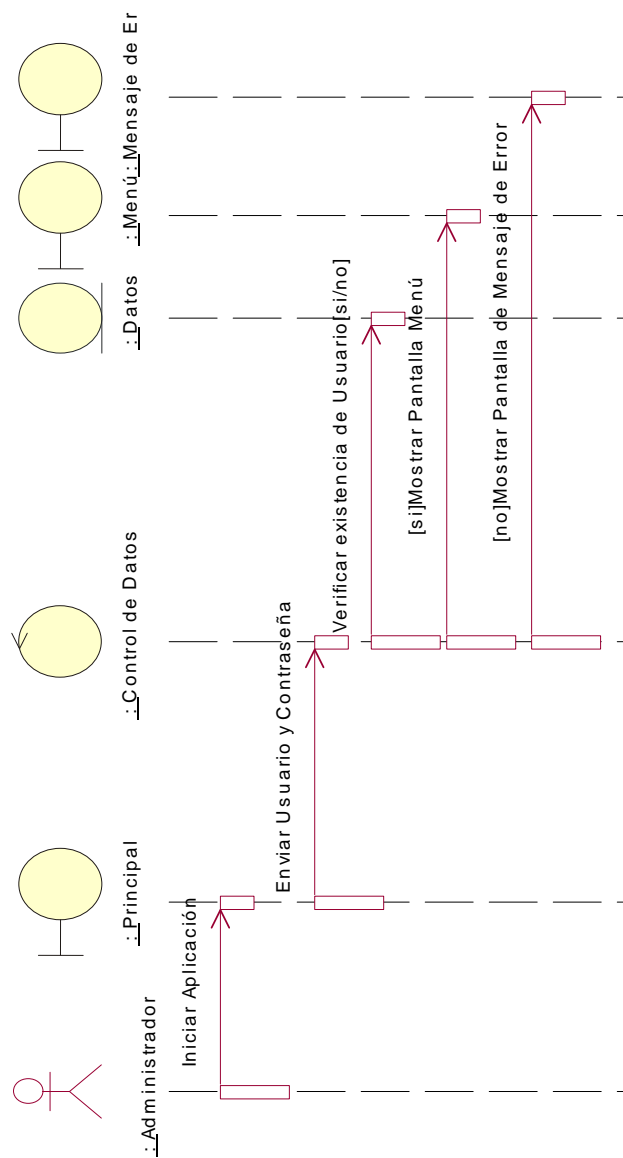
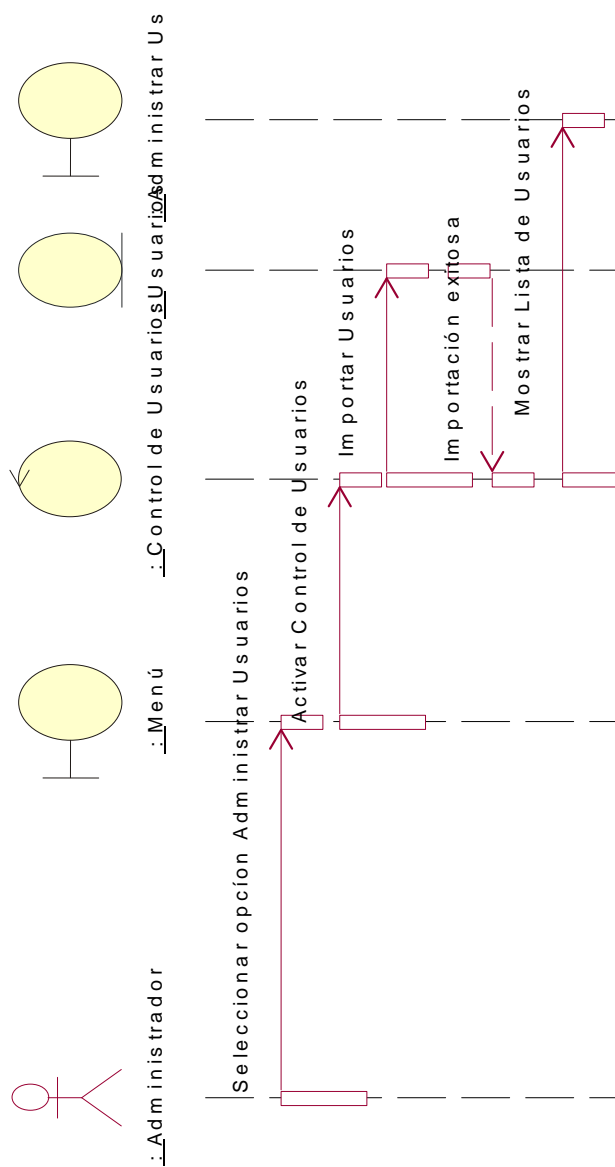


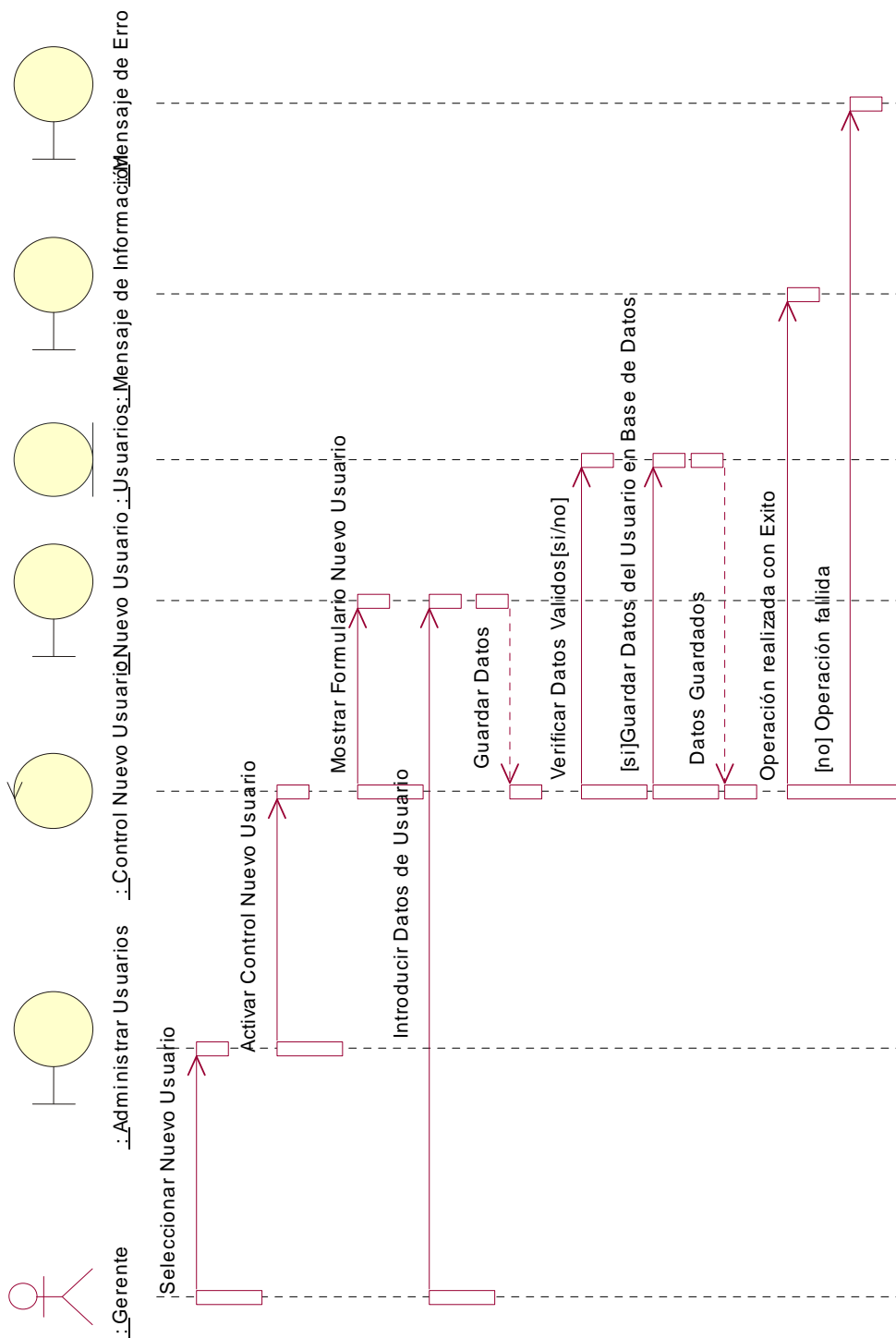
Figura 102: Diagrama de Secuencia: Ingresar al Sistema

*Diagrama de Secuencia: Administrar Usuarios*



**Figura 1103: Diagrama de Secuencia: Administrar Usuarios**

**Diagrama de Secuencia: Nuevo Usuario**



**Figura 1204: Diagrama de Secuencia: Nuevo Usuario**

### Diagrama de Secuencia: Modificar Usuario

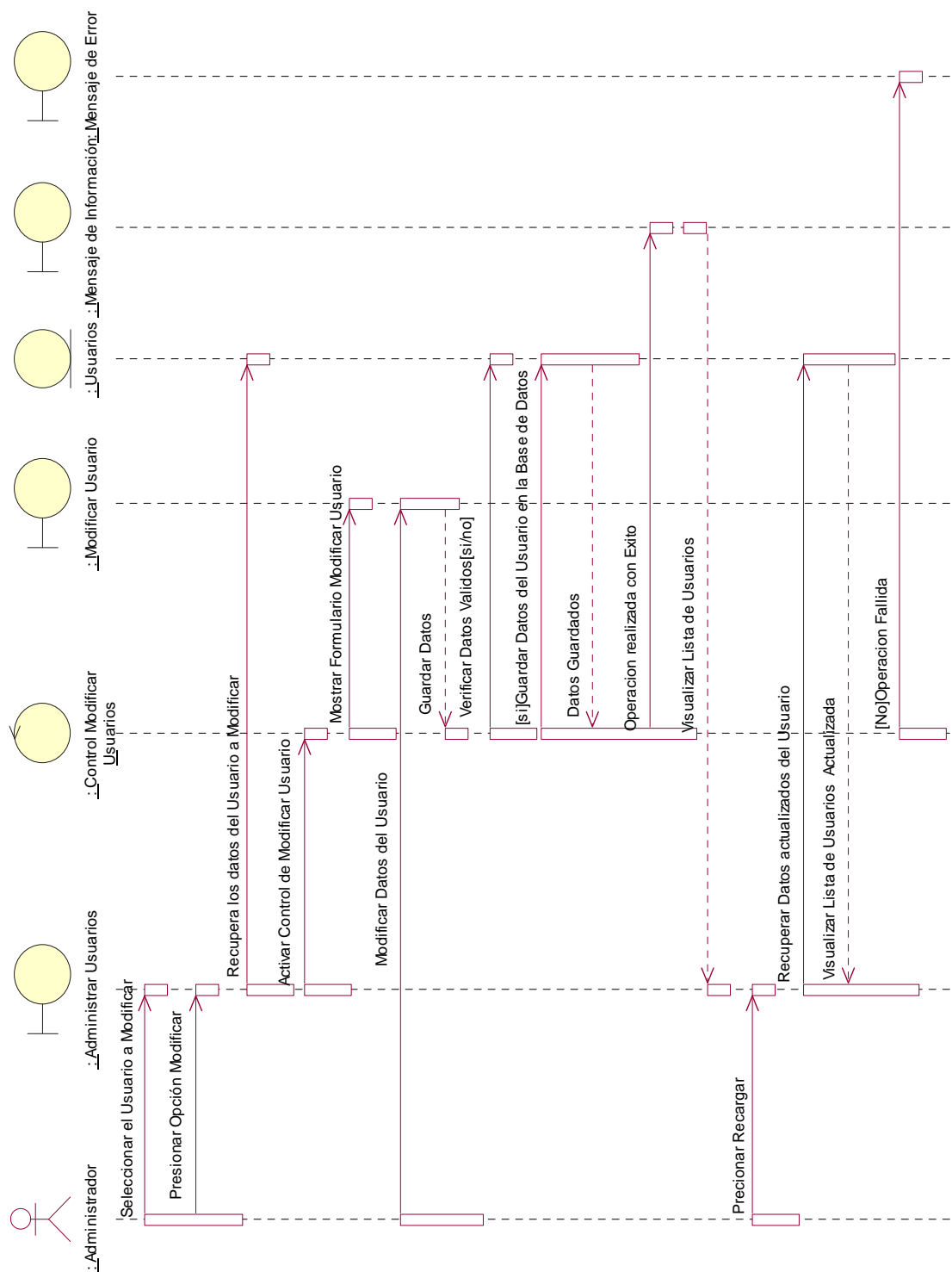
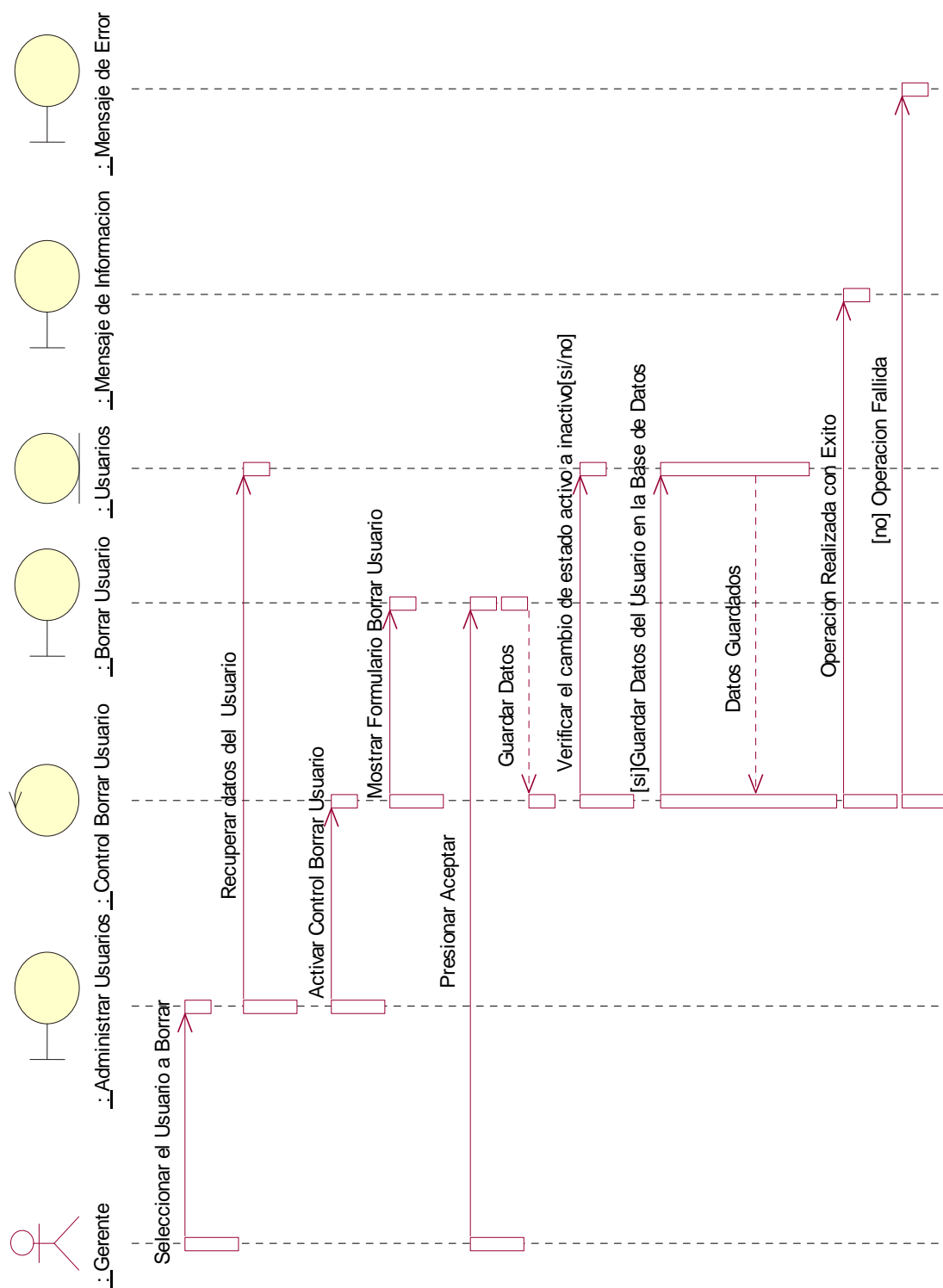


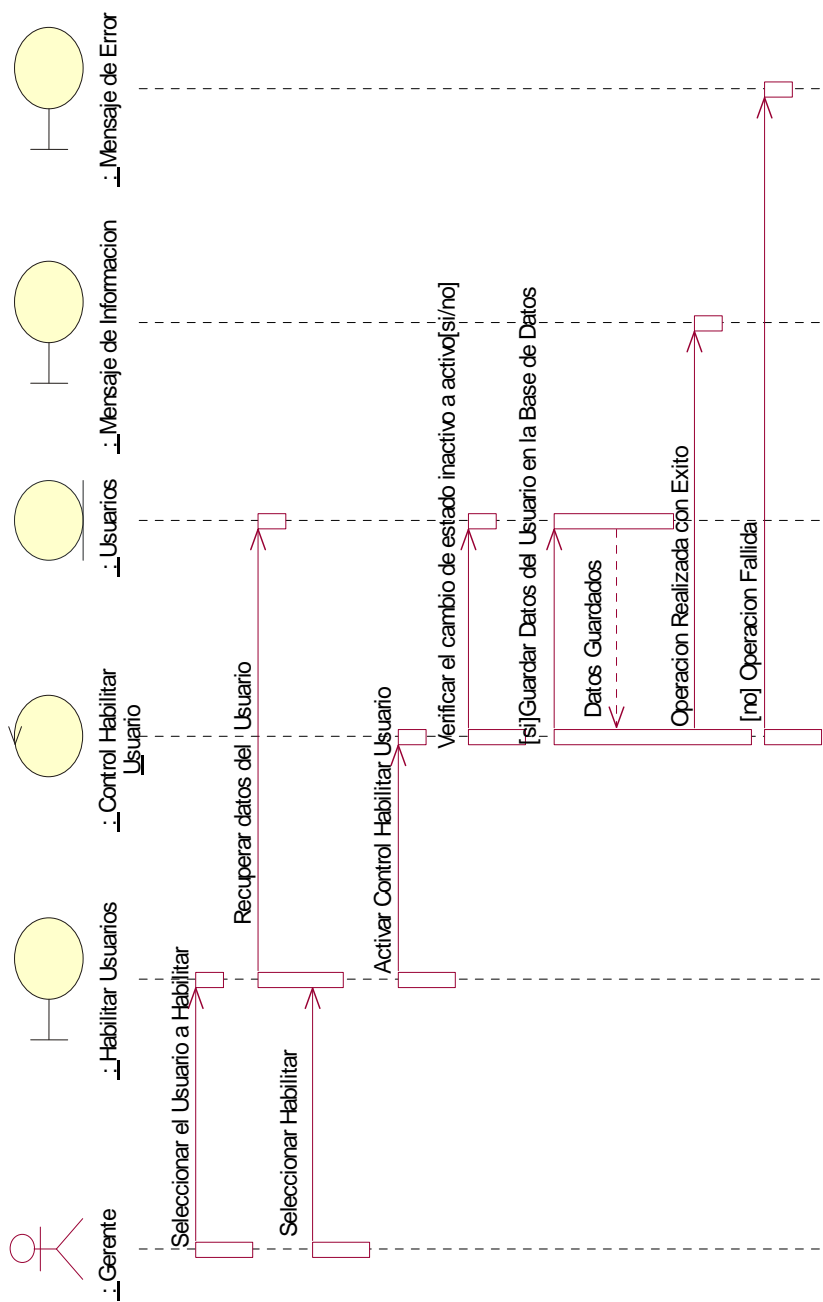
Figura 1305: Diagrama de Secuencia: Modificar Usuario

**Diagrama de Secuencia: Borrar Usuario**



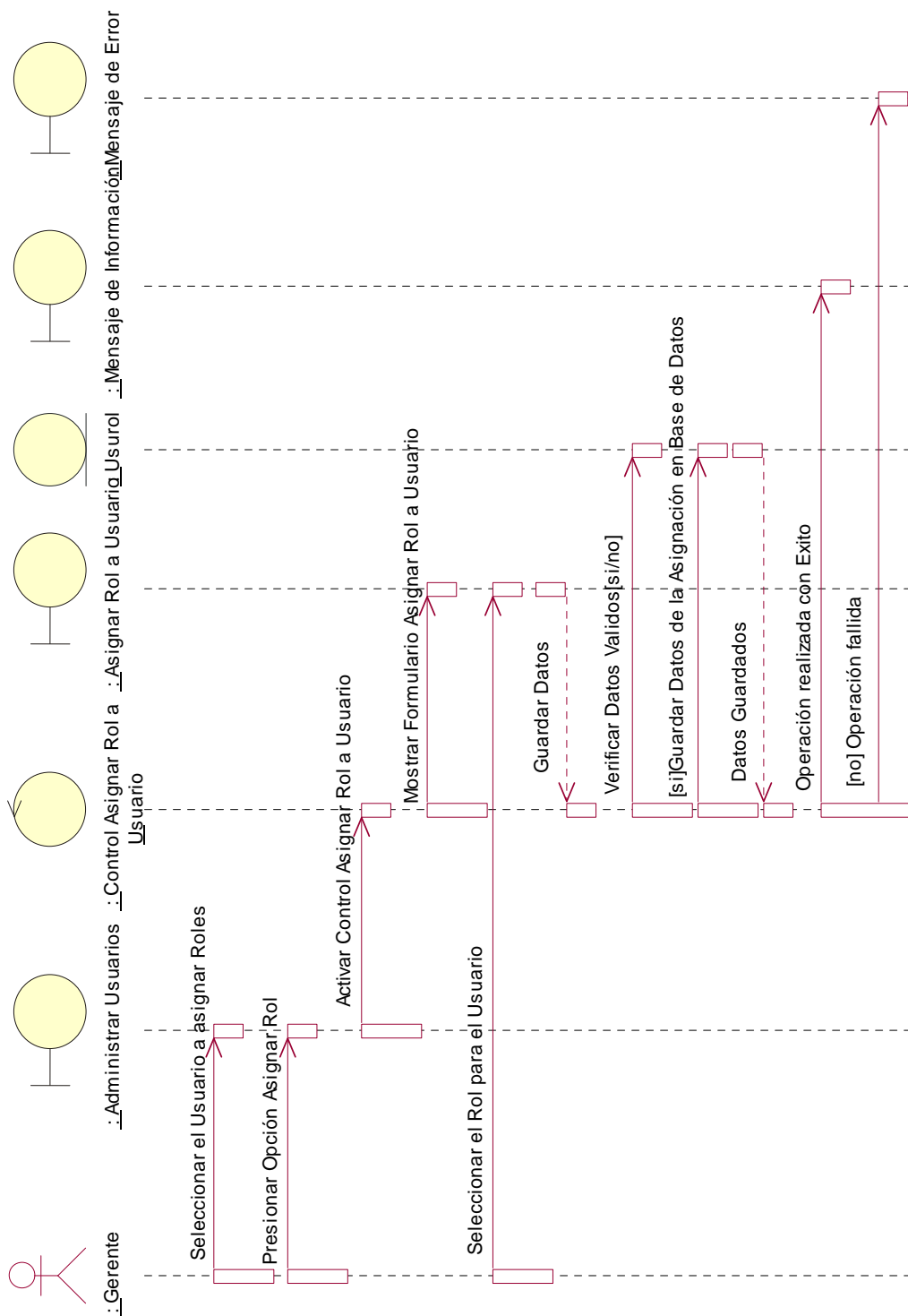
**Figura 1406: Diagrama de Secuencia: Borrar Usuario**

*Diagrama de Secuencia: Habilitar Usuario*



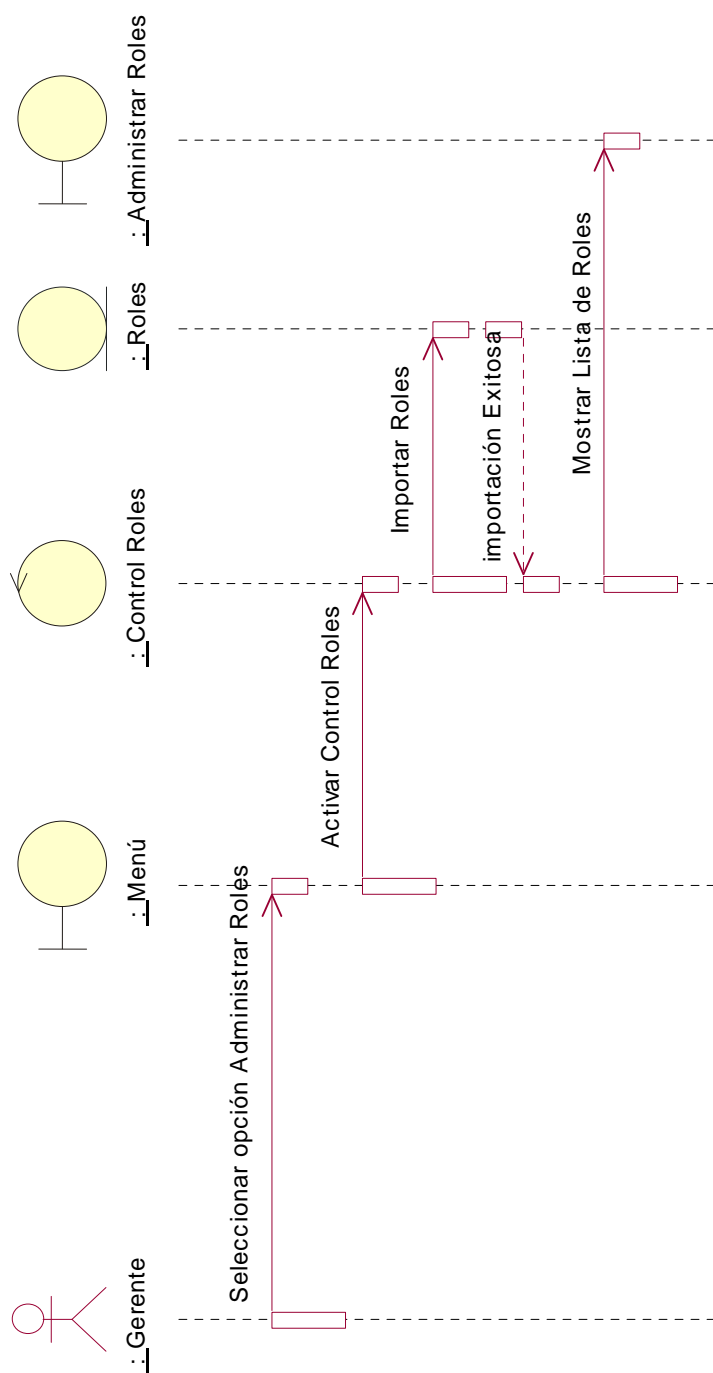
**Figura 1507: Diagrama de Secuencia: Habilitar Usuario**

**Diagrama de Secuencia: Asignar Rol a Usuario**



**Figura 1608: Diagrama Secuencia: Asignar Rol a Usuario**

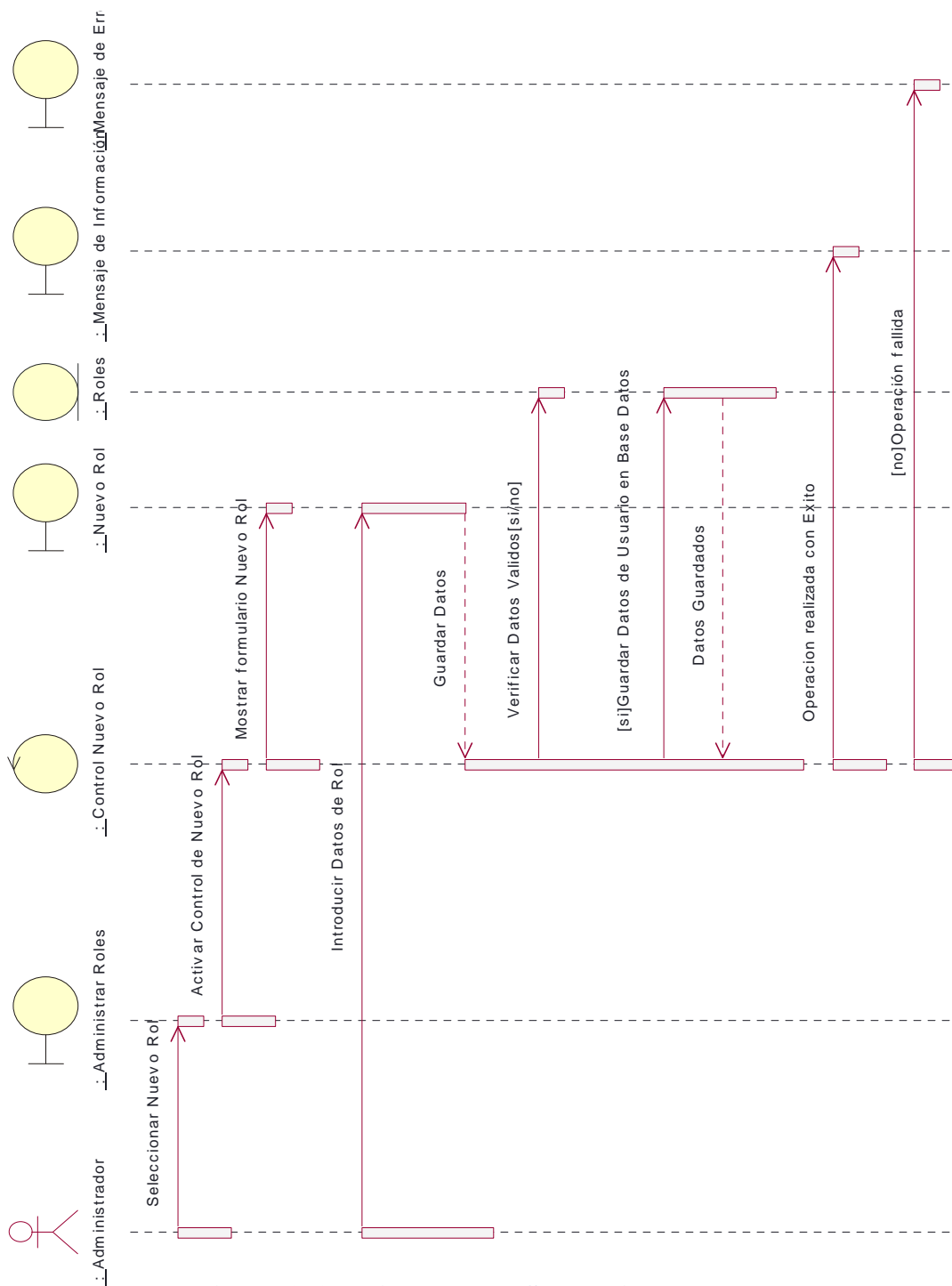
*Diagrama de Secuencia: Administrar Roles*



**Figura 1709: Diagrama de Secuencia: Administrar Roles**

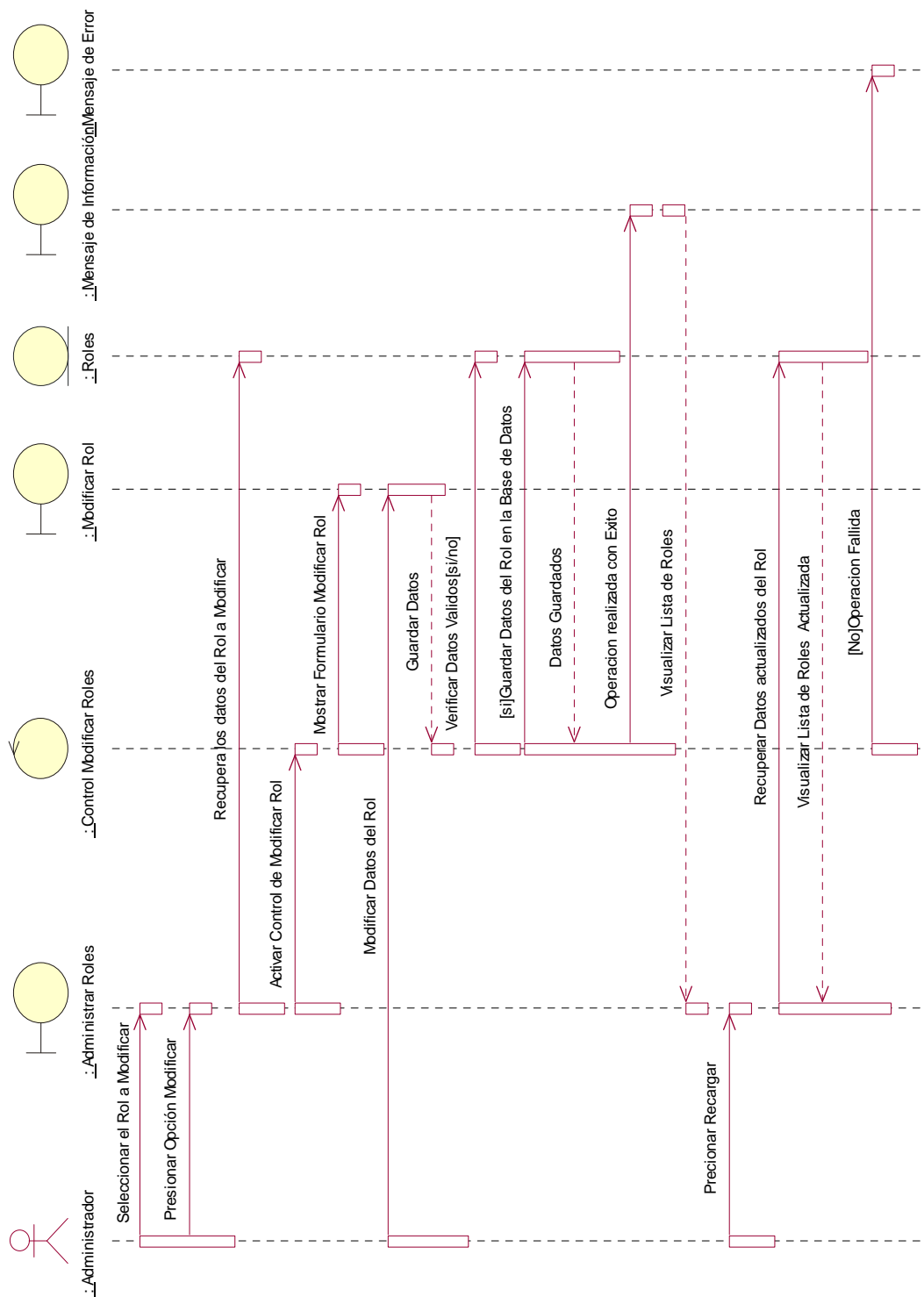


**Diagrama de Secuencia: Nuevo Rol**



**Figura 1810: Diagrama de Secuencia: Nuevo Rol**

**Diagrama de Secuencia: Modificar Rol**



**Figura 1911: Diagrama de Secuencia: Modificar Rol**

### Diagrama de Secuencia: Borrar Rol

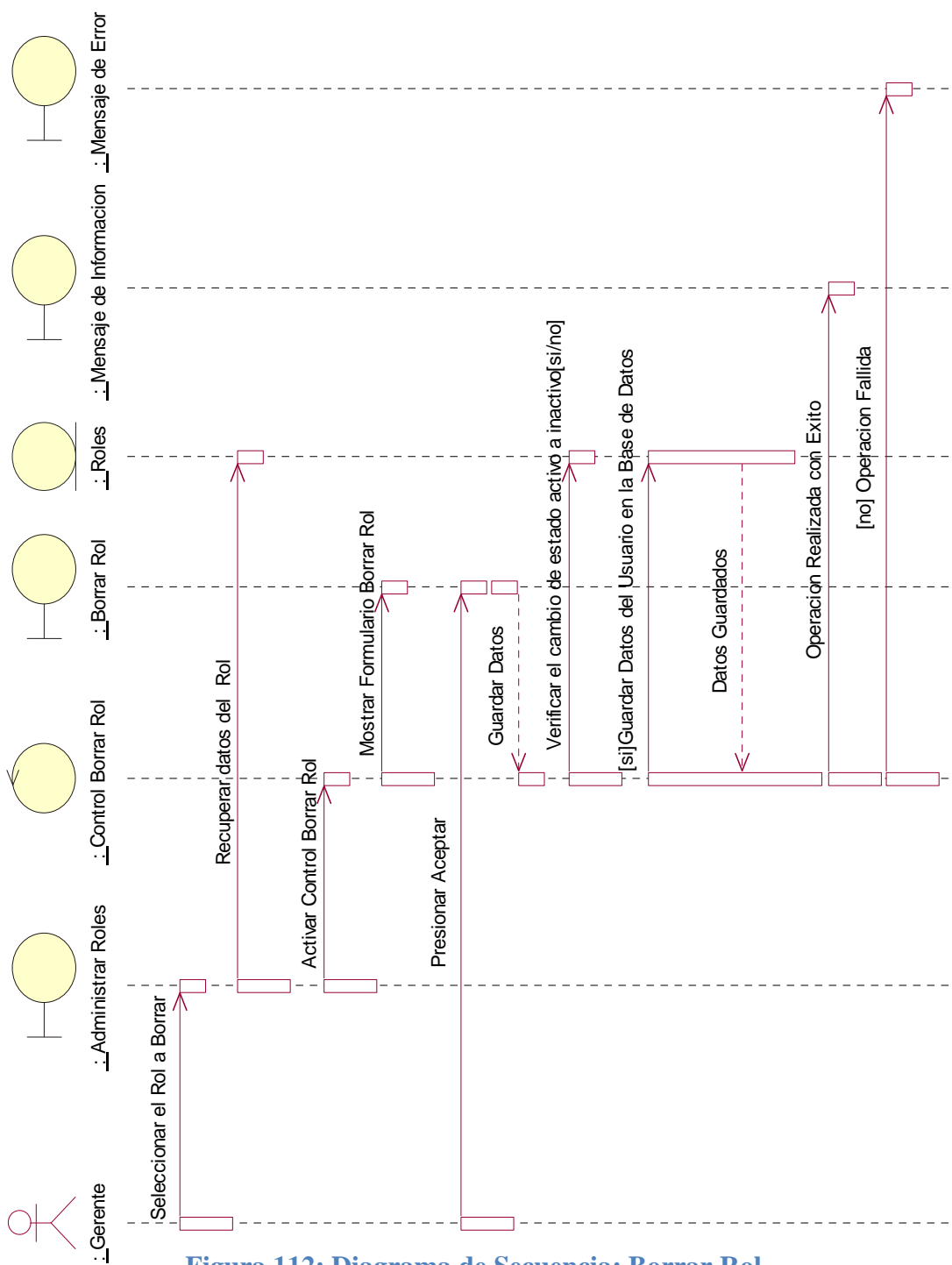
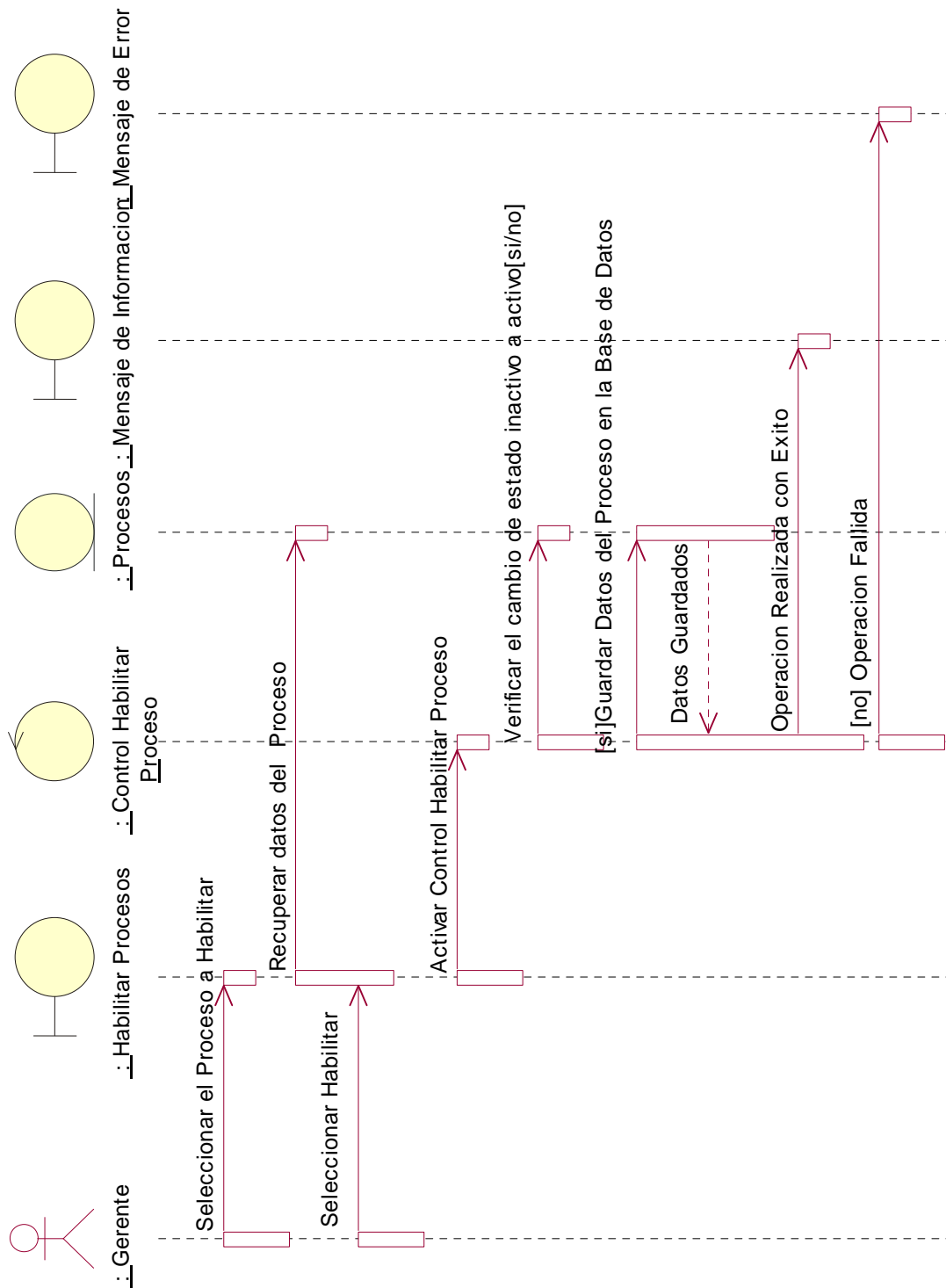


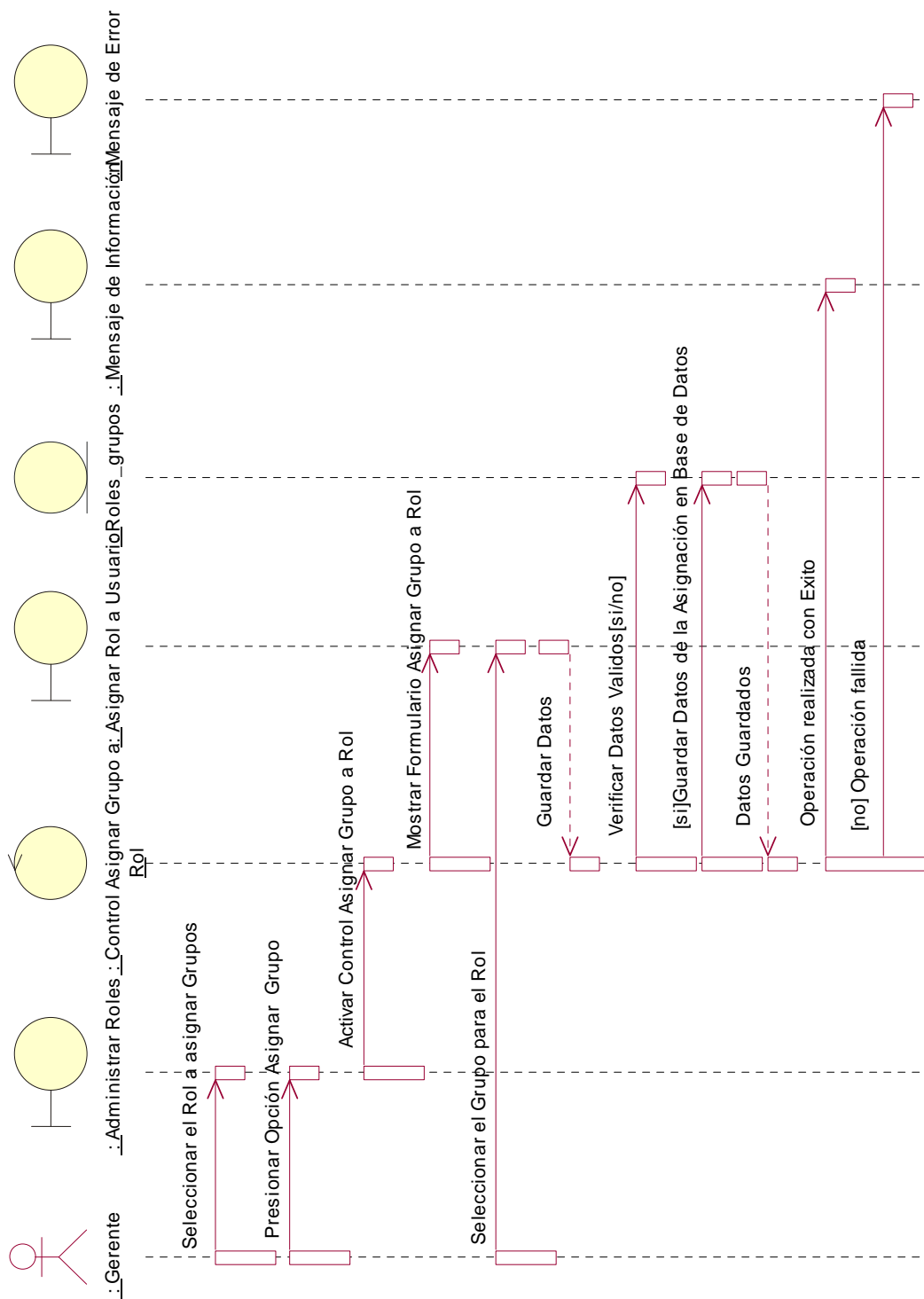
Figura 112: Diagrama de Secuencia: Borrar Rol

**Diagrama de Secuencia: Habilitar Roles**



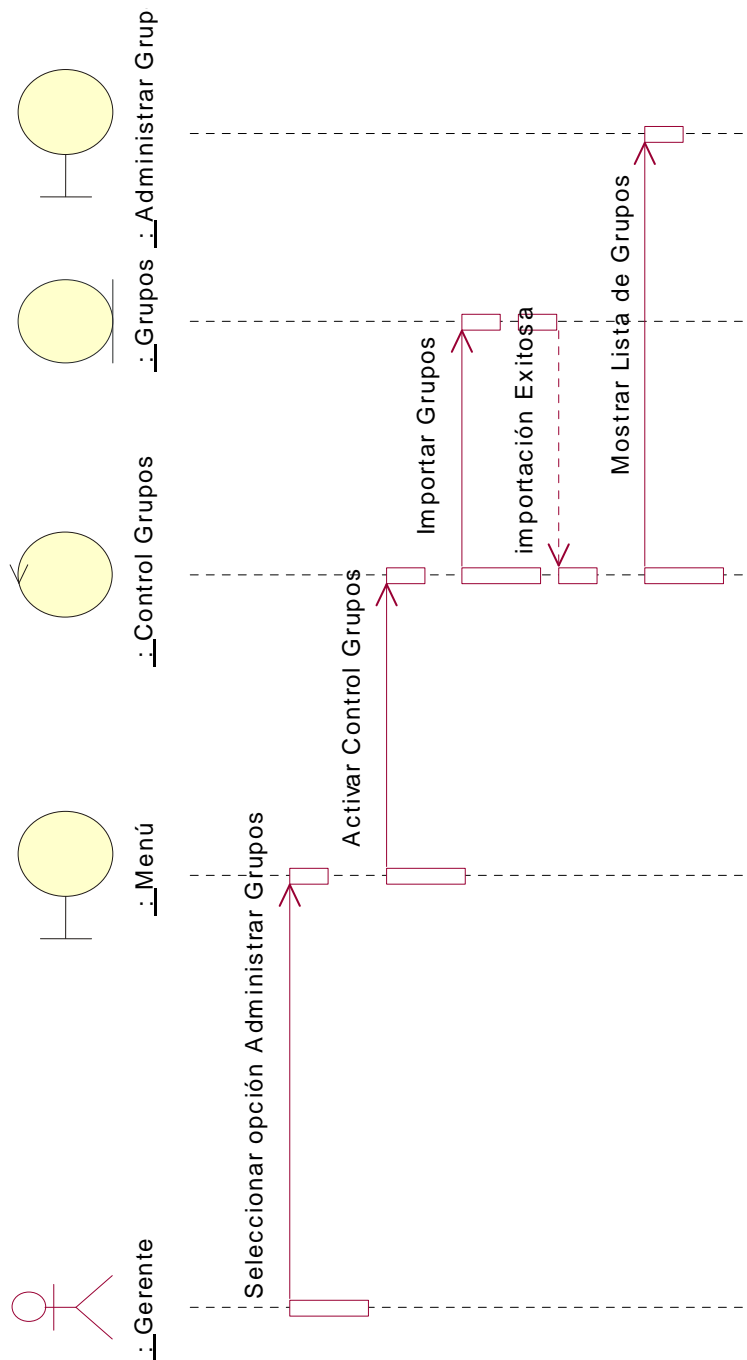
**Figura 113: Diagrama de Secuencia: Habilitar Rol**

**Diagrama de Secuencia: Asignar Grupo a Rol**



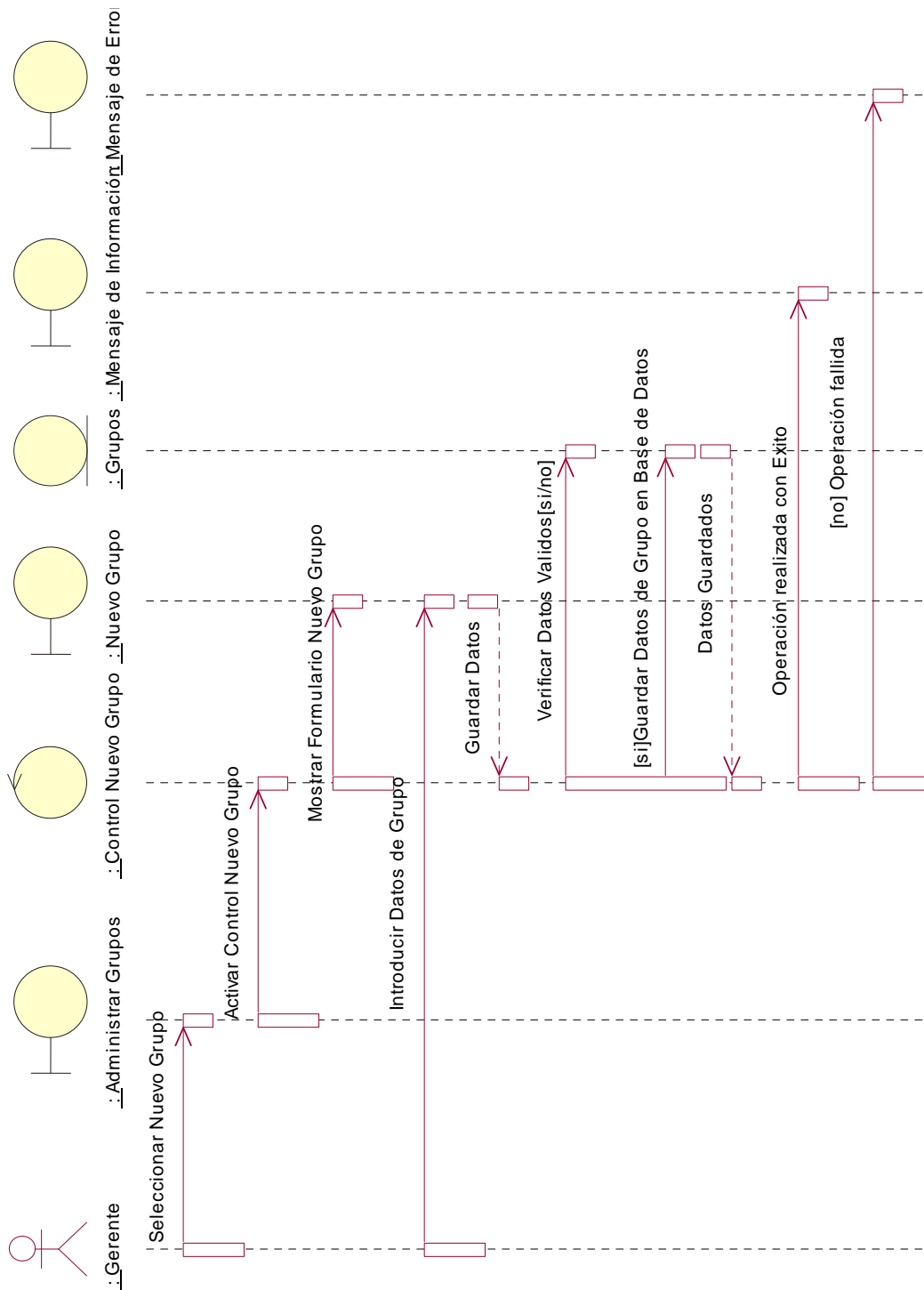
**Figura 114: Diagrama de Secuencia: Asignar Grupo a Rol**

**Diagrama de Secuencia: Administrar Grupos**



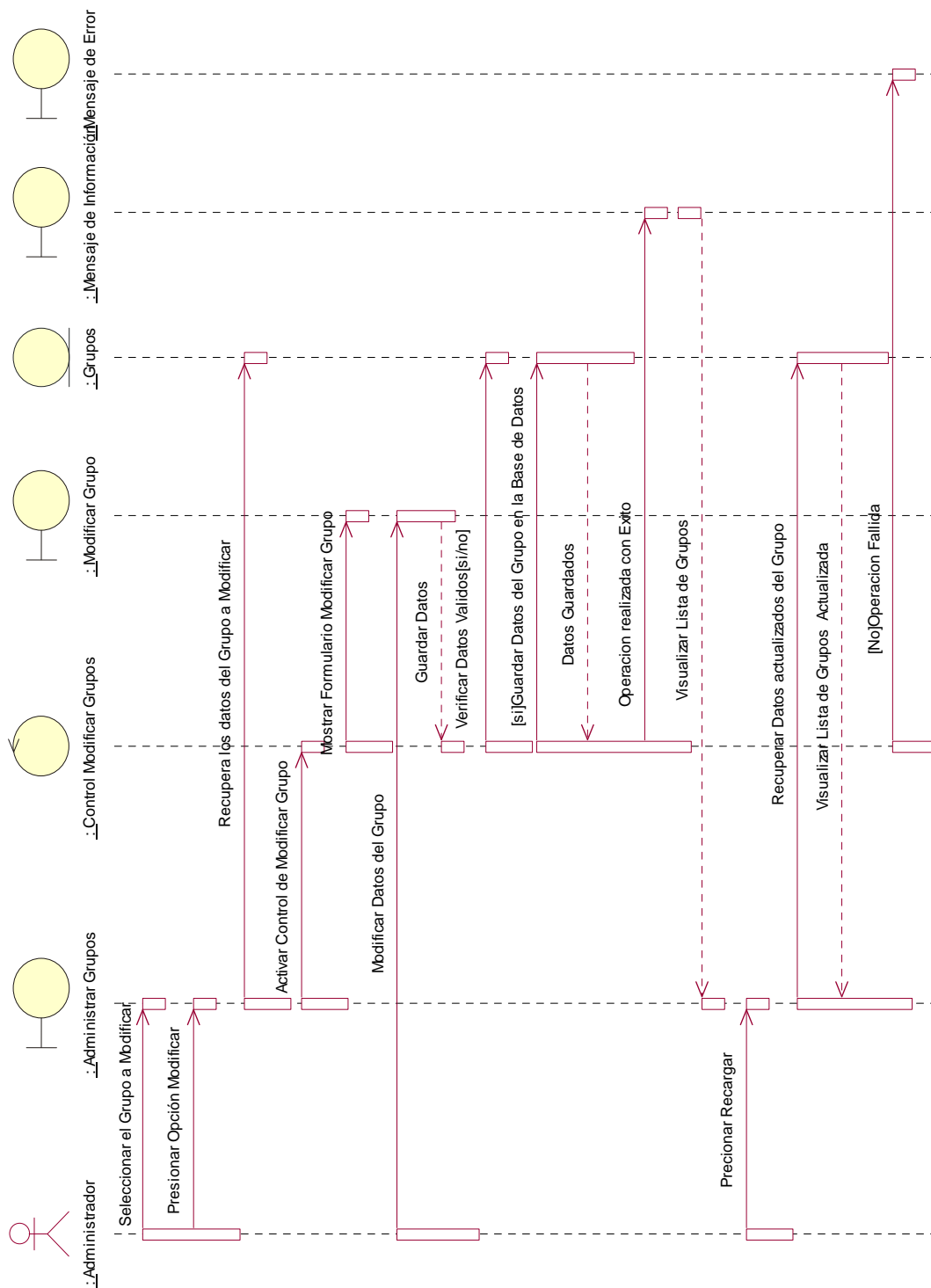
**Figura 115: Diagrama de Secuencia: Administrar Grupos**

**Diagrama de Secuencia: Nuevo Grupo**



**Figura 116: Diagrama de Secuencia: Nuevo Grupo**

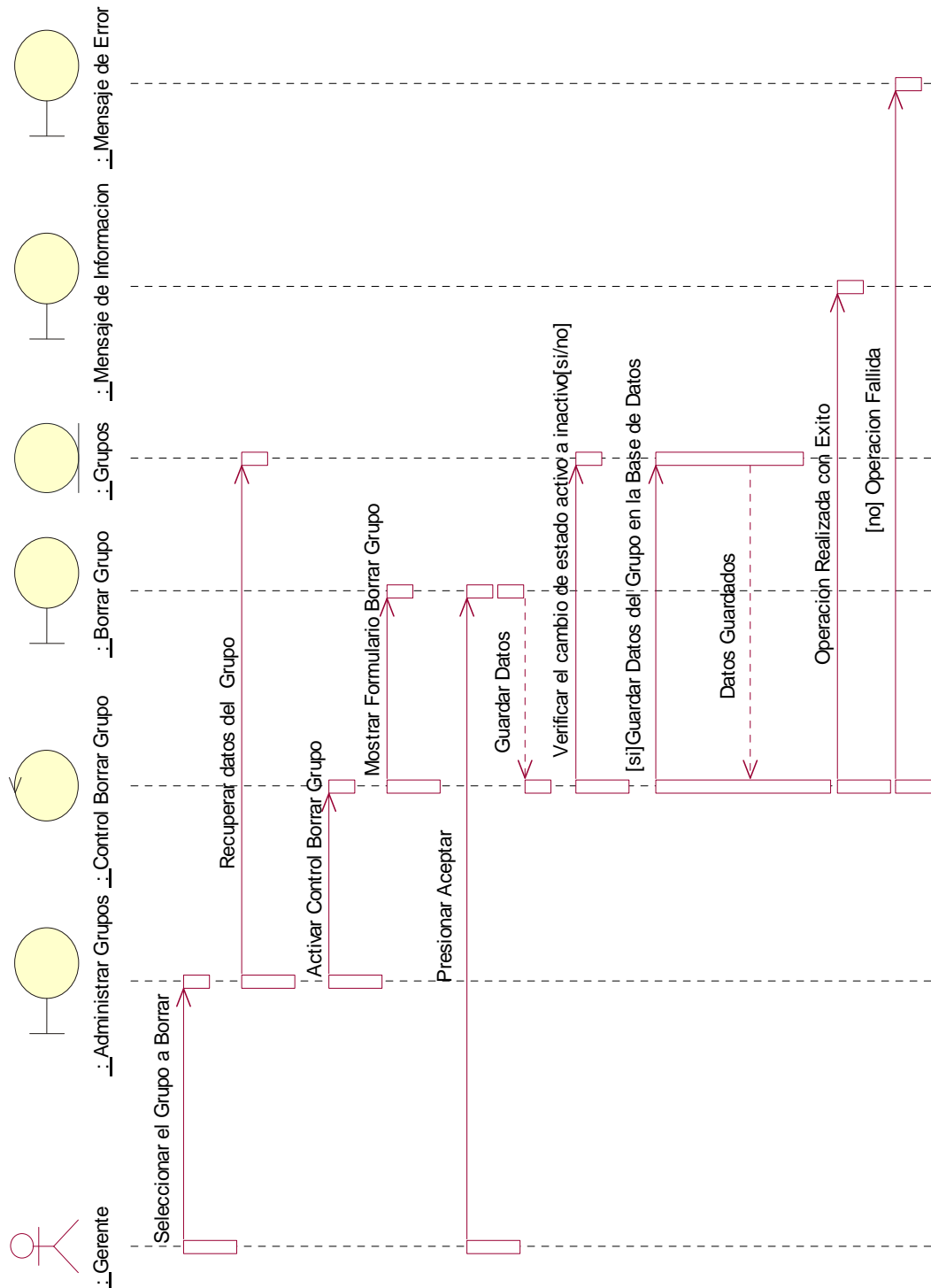
**Diagrama de Secuencia: Modificar Grupo**



**Figura 117: Diagrama de Secuencia: Modificar Grupo**

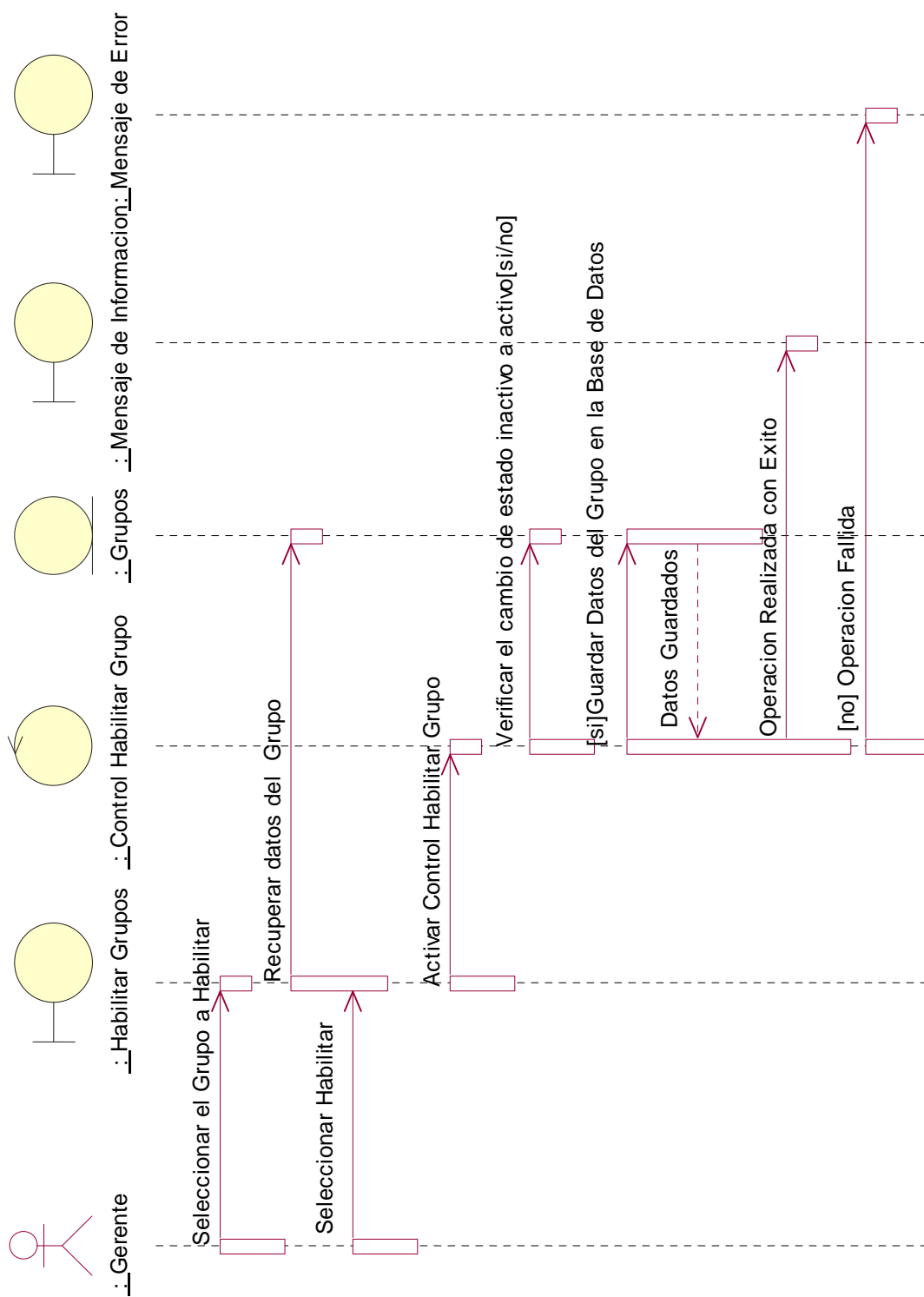


**Diagrama de Secuencia: Borrar Grupo**



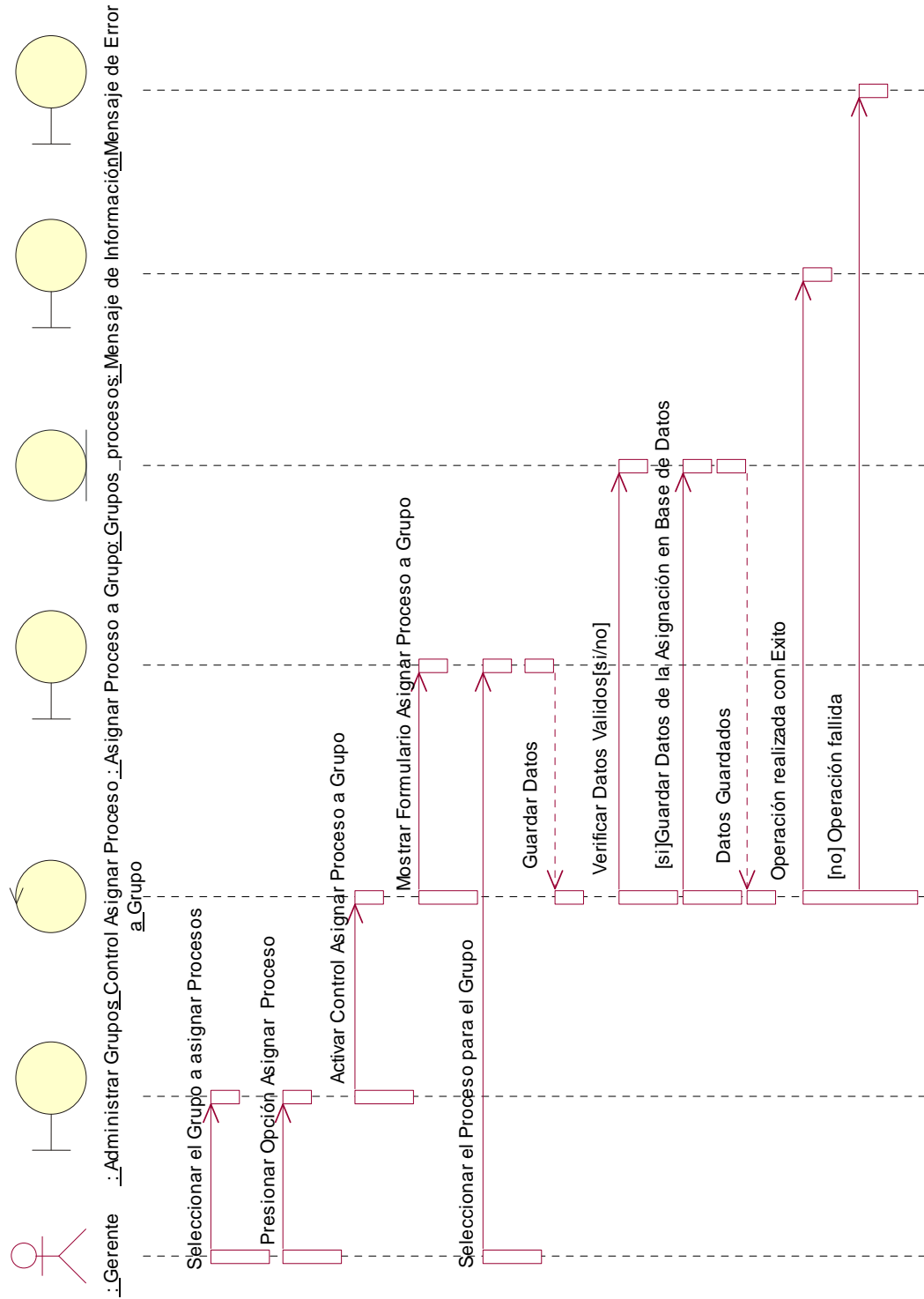
**Figura 118: Diagrama de Secuencia: Borrar Grupo**

**Diagrama de Secuencia: Habilitar Grupo**



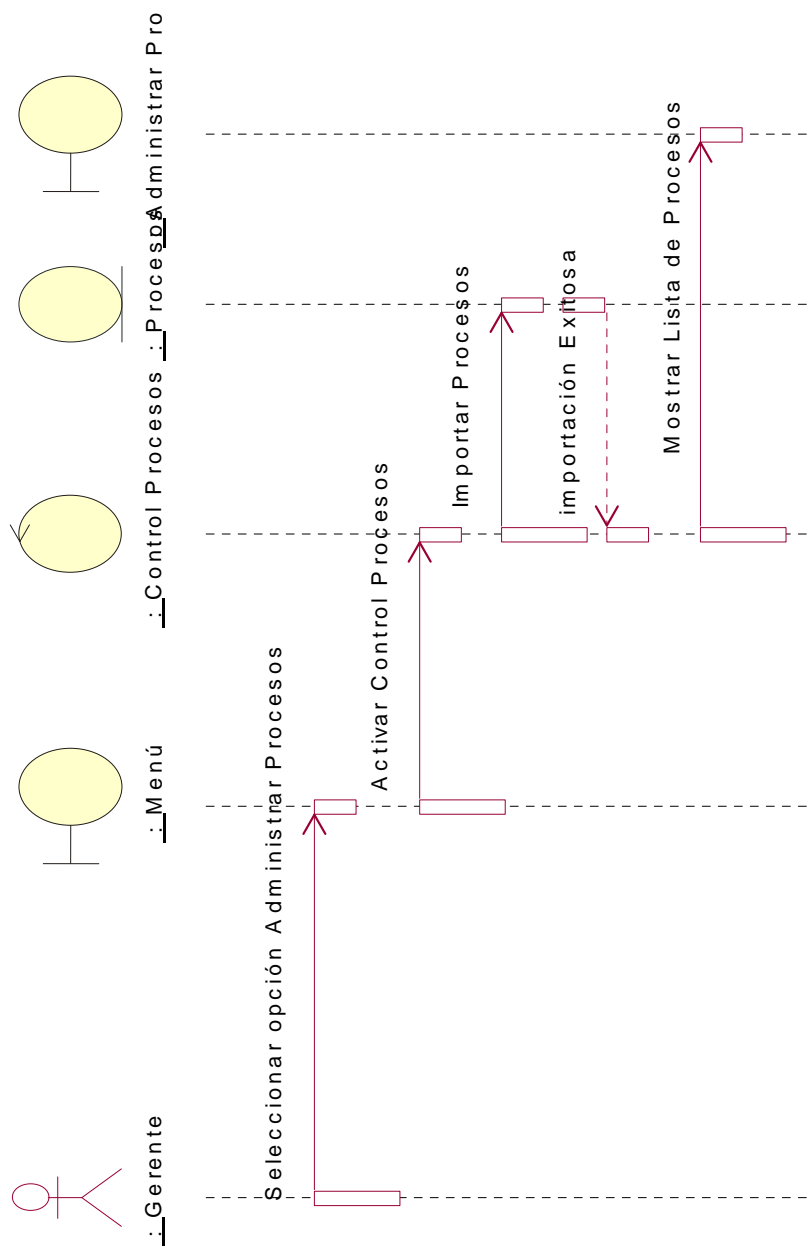
**Figura 119: Diagrama de Secuencia: Habilitar Grupo**

**Diagrama de Secuencia: Asignar Proceso a Grupo**



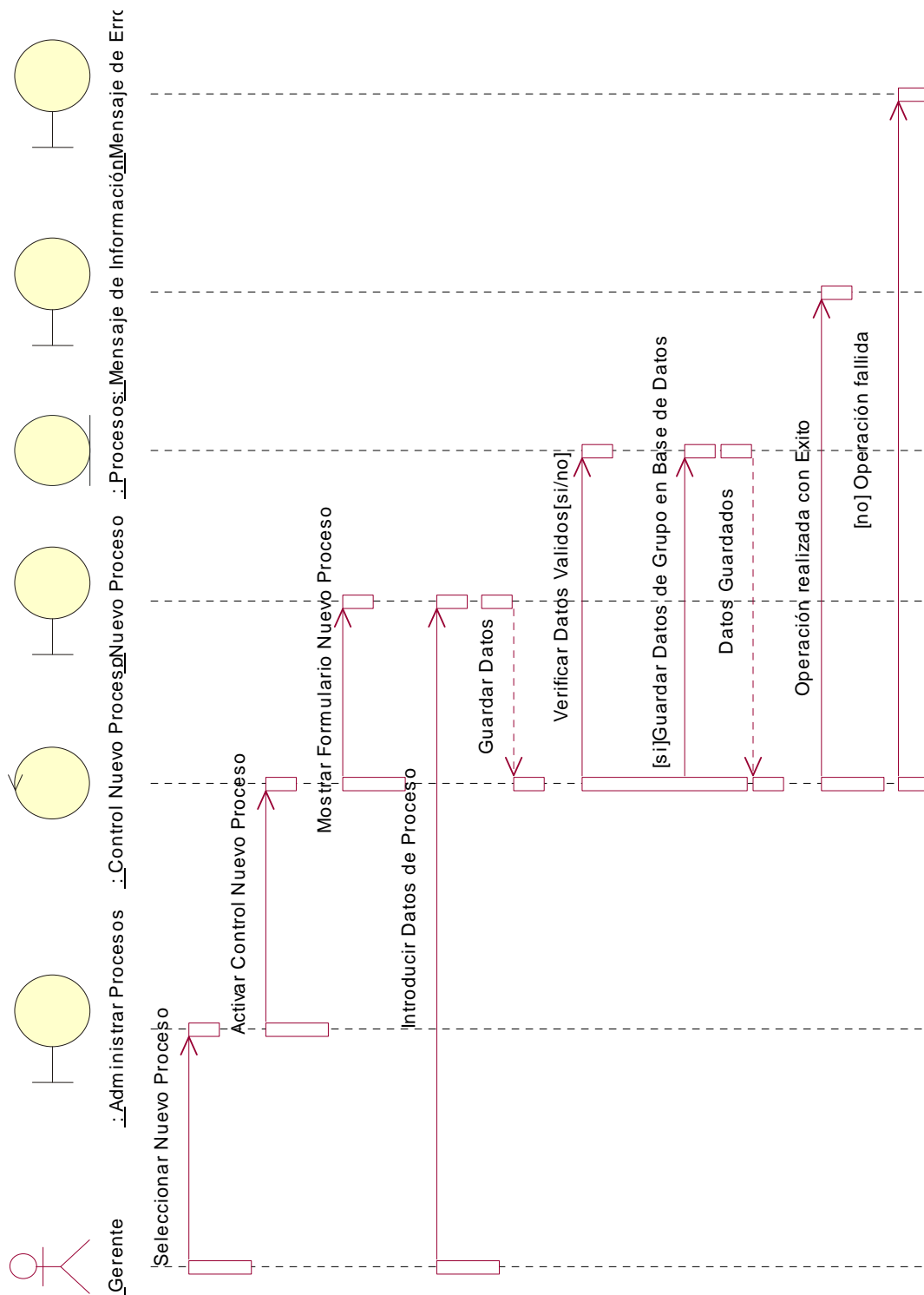
**Figura 120: Diagrama de Secuencia: Asignar Proceso a Grupo**

*Diagrama de Secuencia: Administrar Proceso*



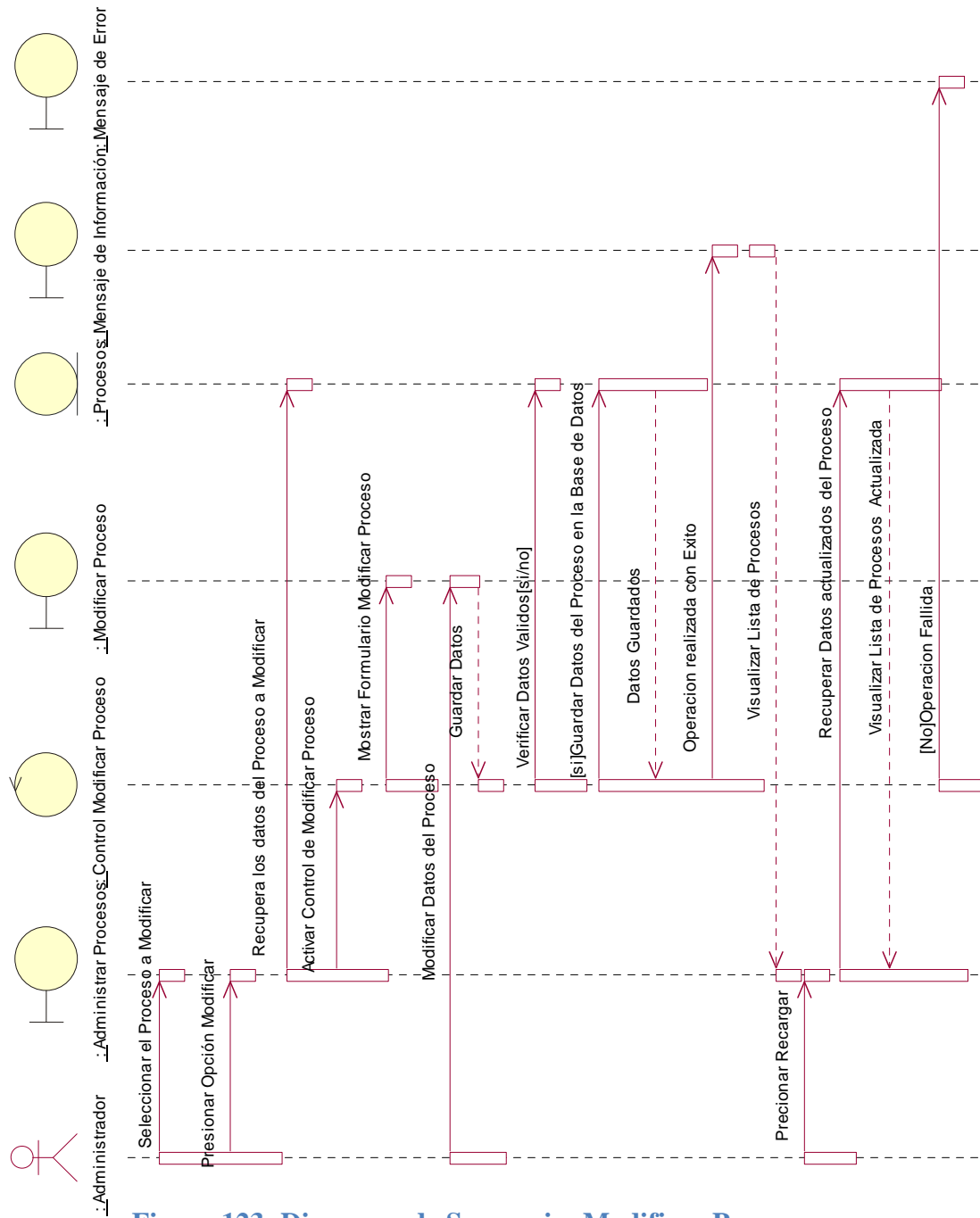
**Figura 121: Diagrama de Secuencia: Administrar Proceso**

**Diagrama de Secuencia: Nuevo Proceso**



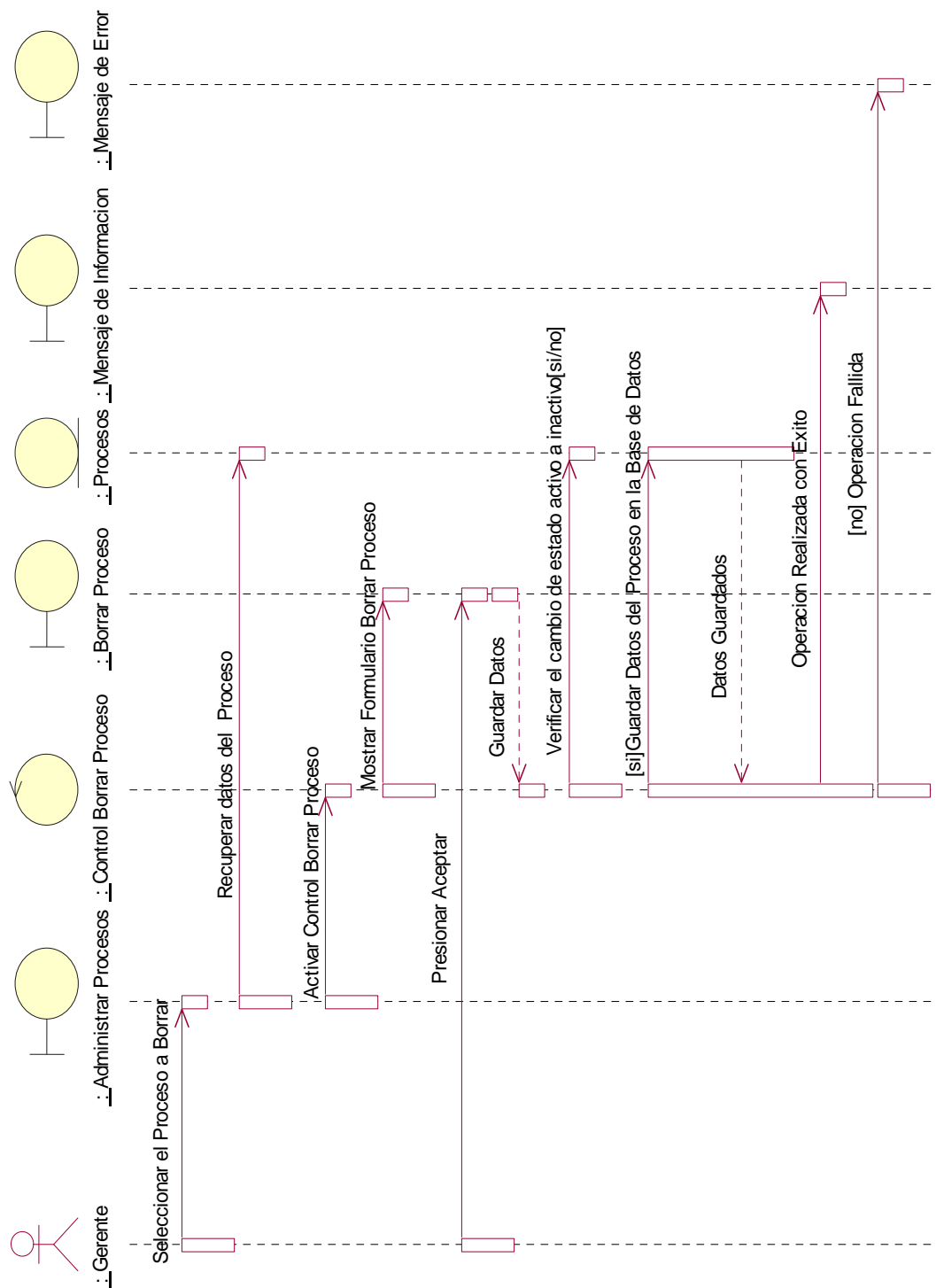
**Figura 122: Diagrama de Secuencia: Nuevo Proceso**

**Diagrama de Secuencia: Modificar Proceso**



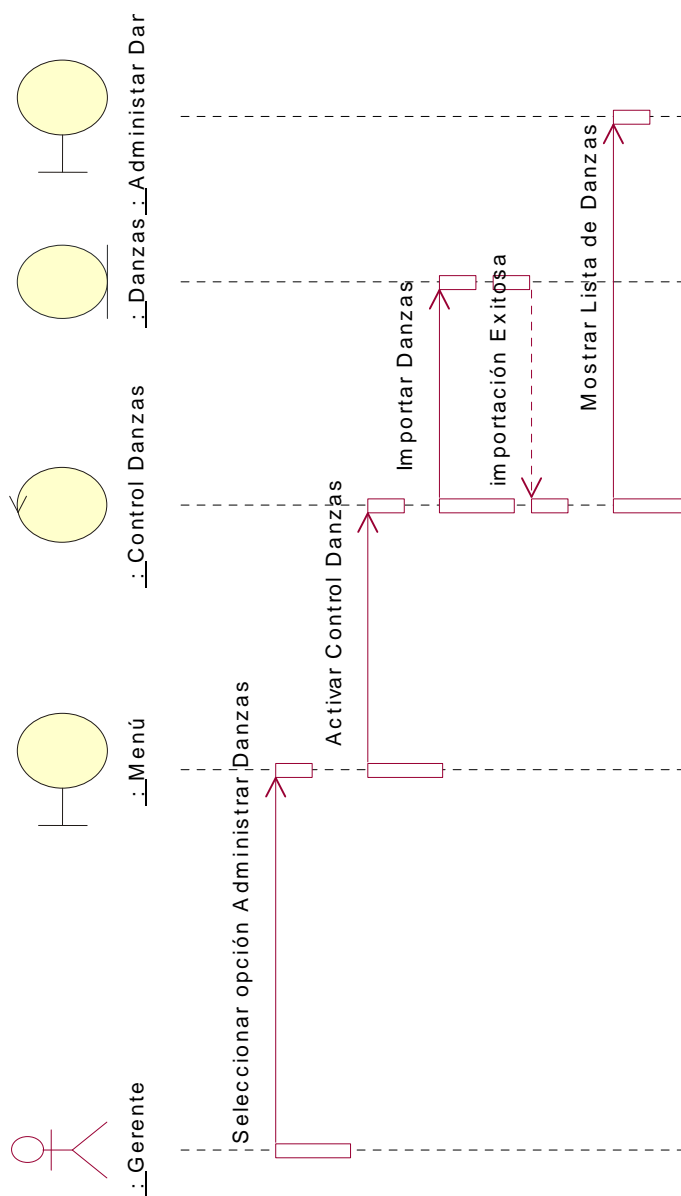
**Figura 123: Diagrama de Secuencia: Modificar Proceso**

**Diagrama de Secuencia: Borrar Proceso**



**Figura 124: Diagrama de Secuencia: Borrar Proceso**

**Diagrama de Secuencia: Administrar Danzas**



**Figura 125: Diagrama de Secuencia: Administrar Danzas**



### Diagrama de Secuencia: Nueva Danza

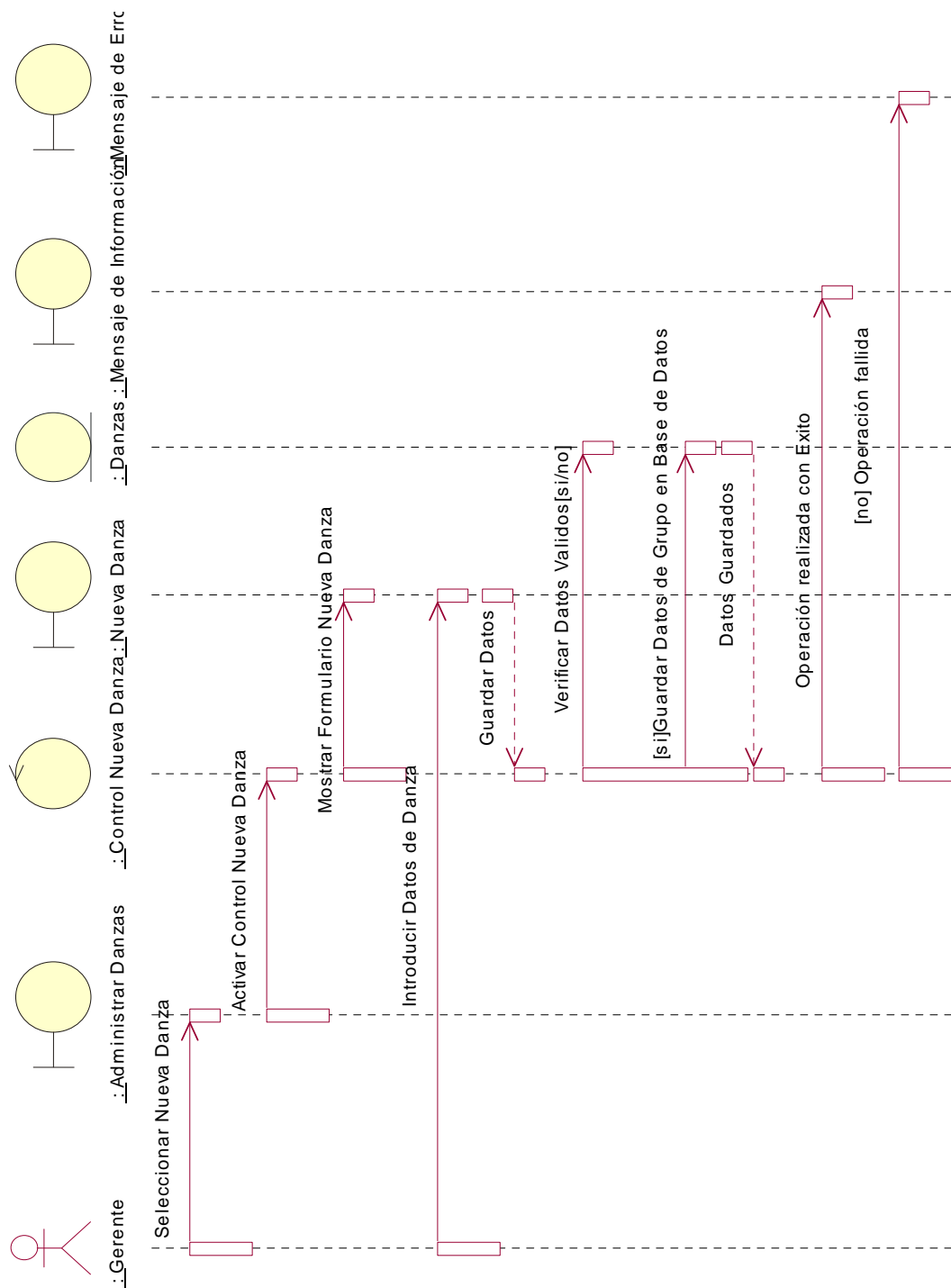


Figura 126: Diagrama de Secuencia: Nueva Danza

### Diagrama de Secuencia: Modificar Danza

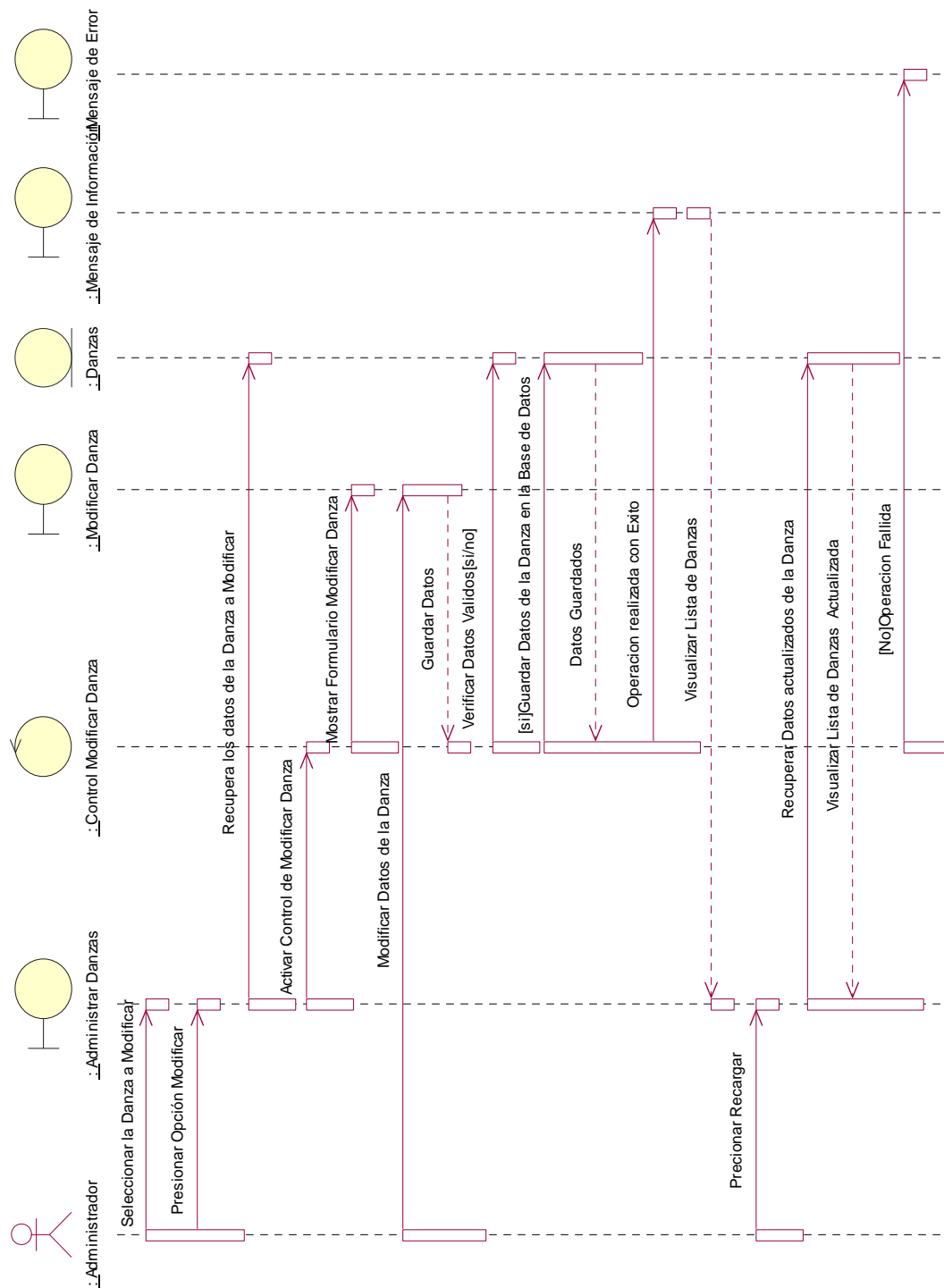
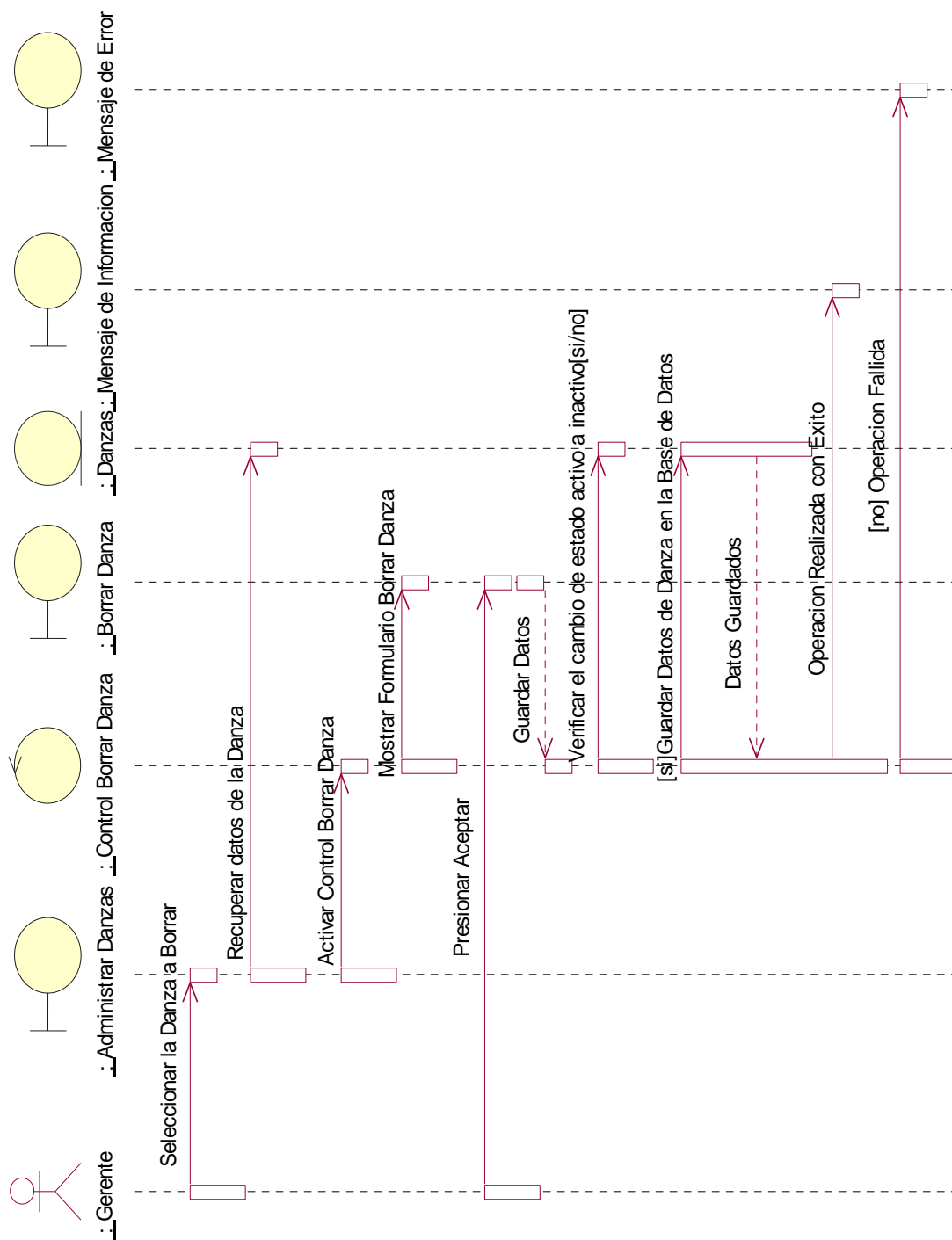


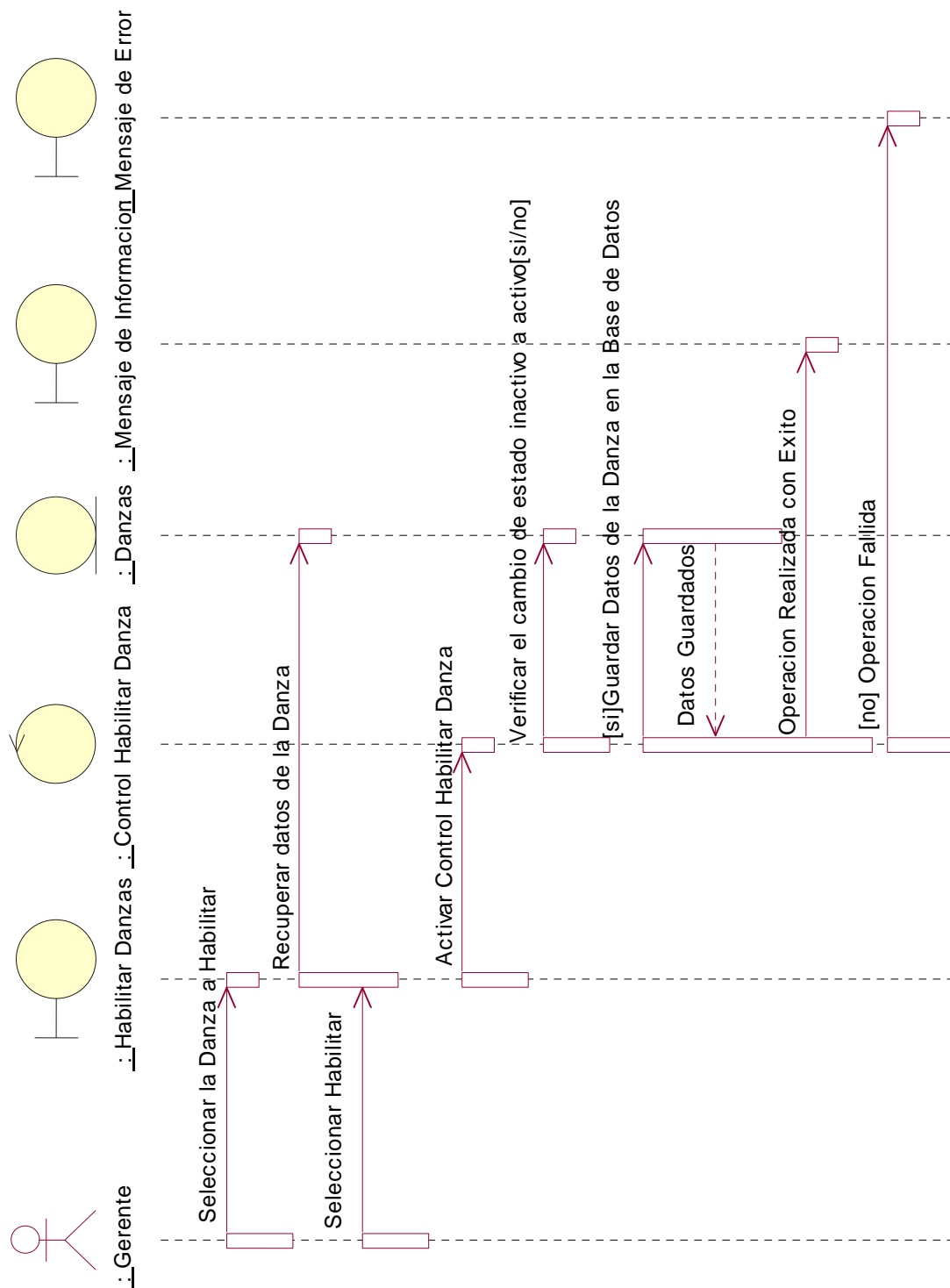
Figura 127: Diagrama de Secuencia: Modificar Danza

**Diagrama de Secuencia: Borrar Danza**



**Figura 128: Diagrama de Secuencia: Borrar Danza**

**Diagrama de Secuencia: Habilitar Danza**



**Figura 129: Diagrama de Secuencia: Habilitar Danza**

### Diagrama de Secuencia: Asignar Prenda a Danza

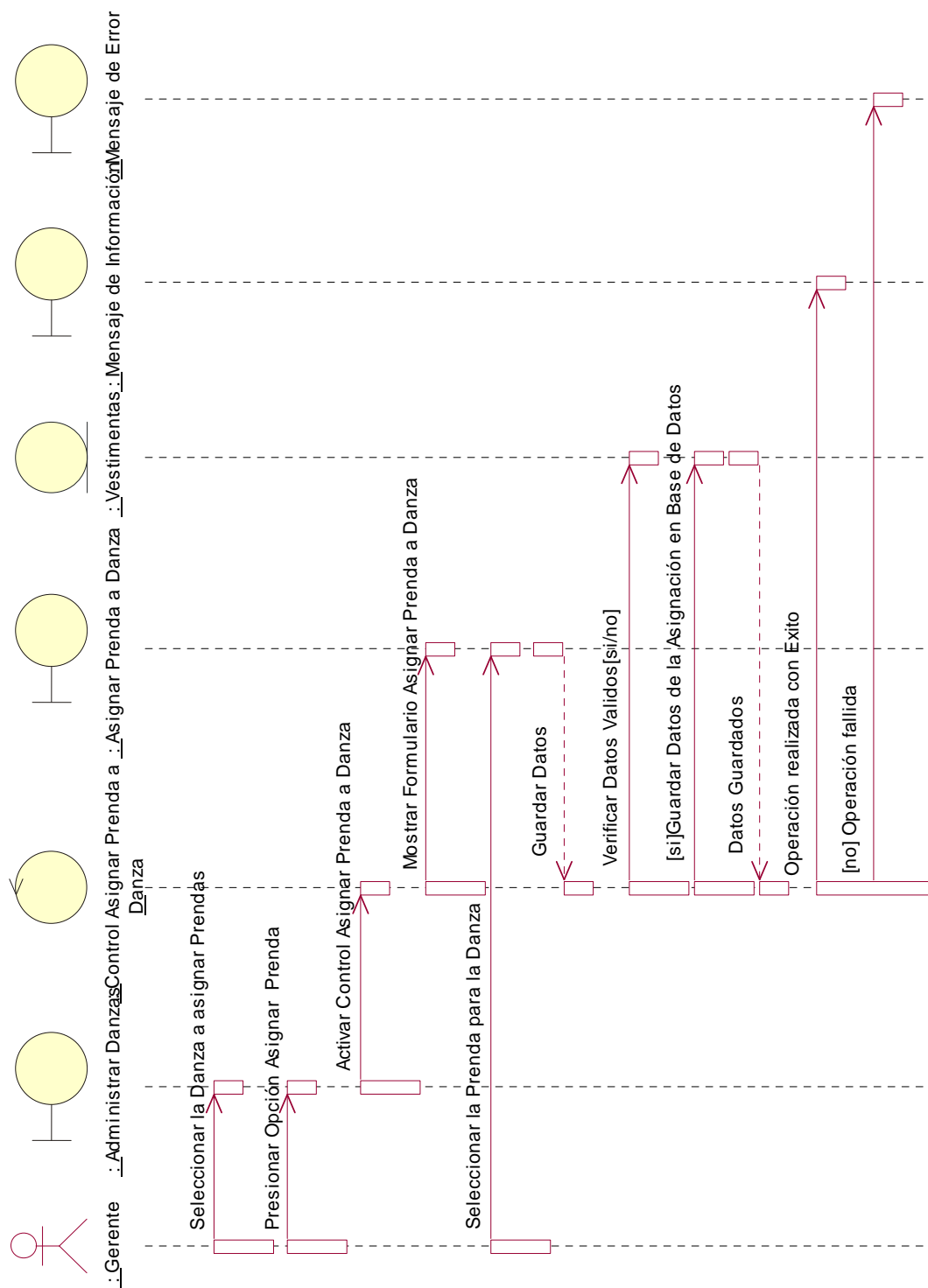
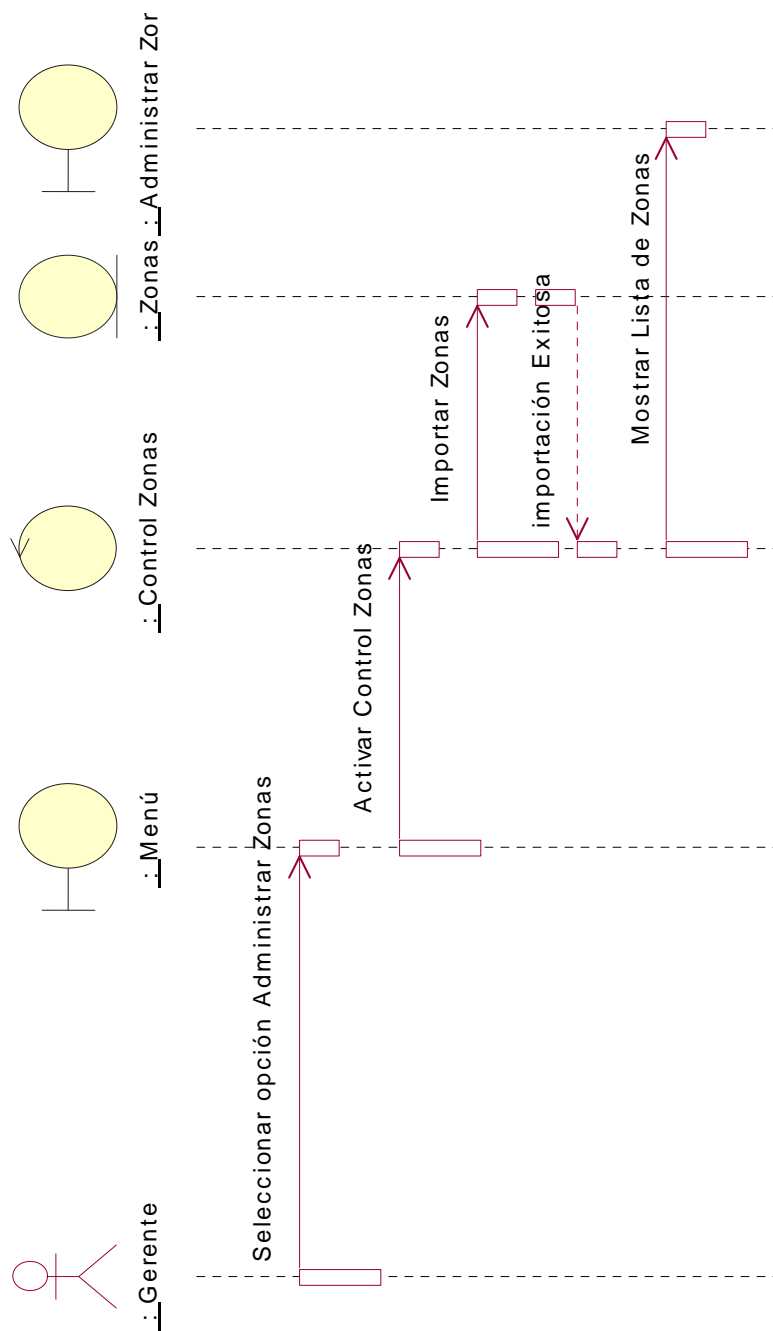


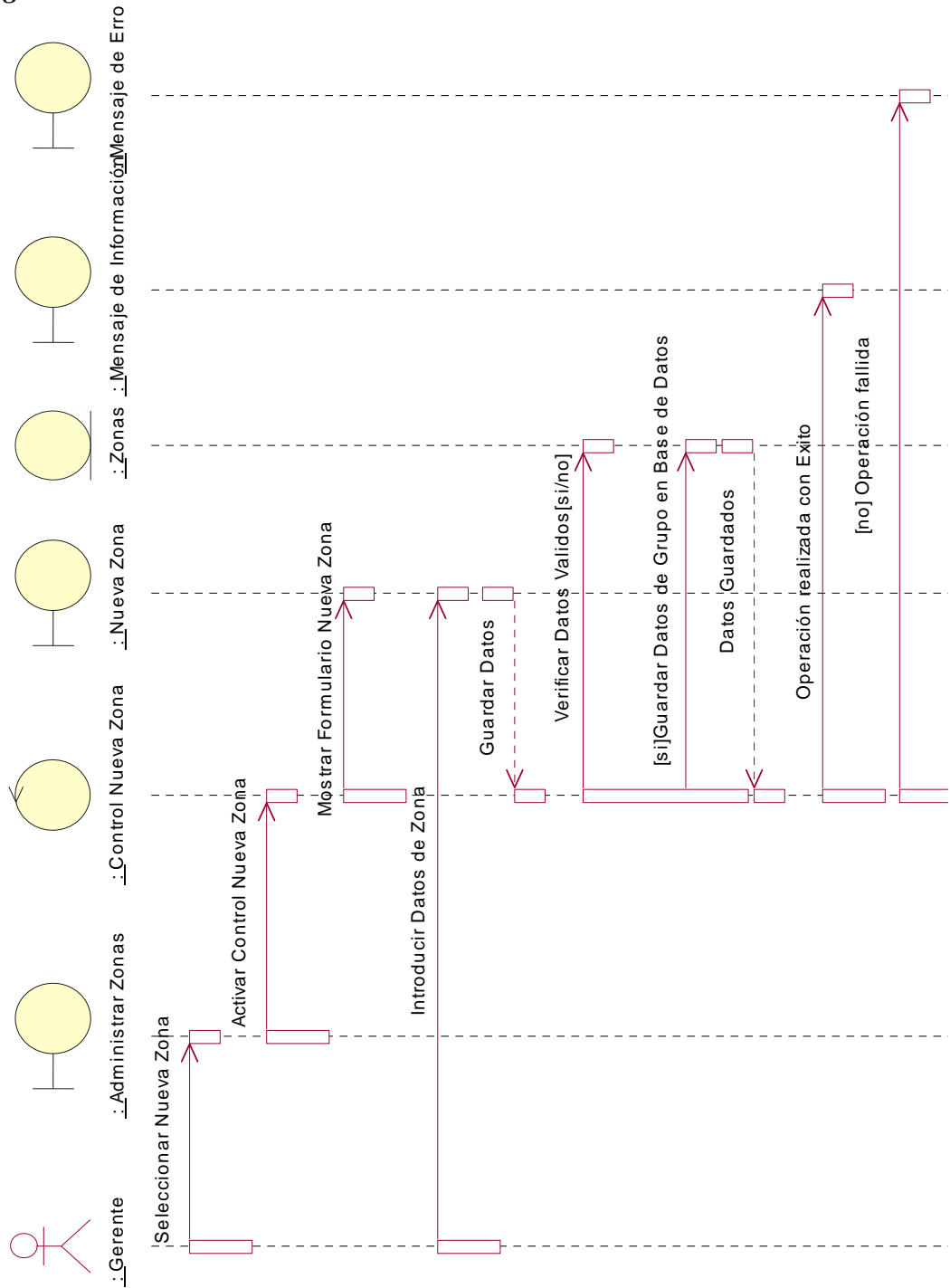
Figura 130: Diagrama de Secuencia: Asignar Prenda a Danza

*Diagrama de Secuencia: Administrar Zonas*



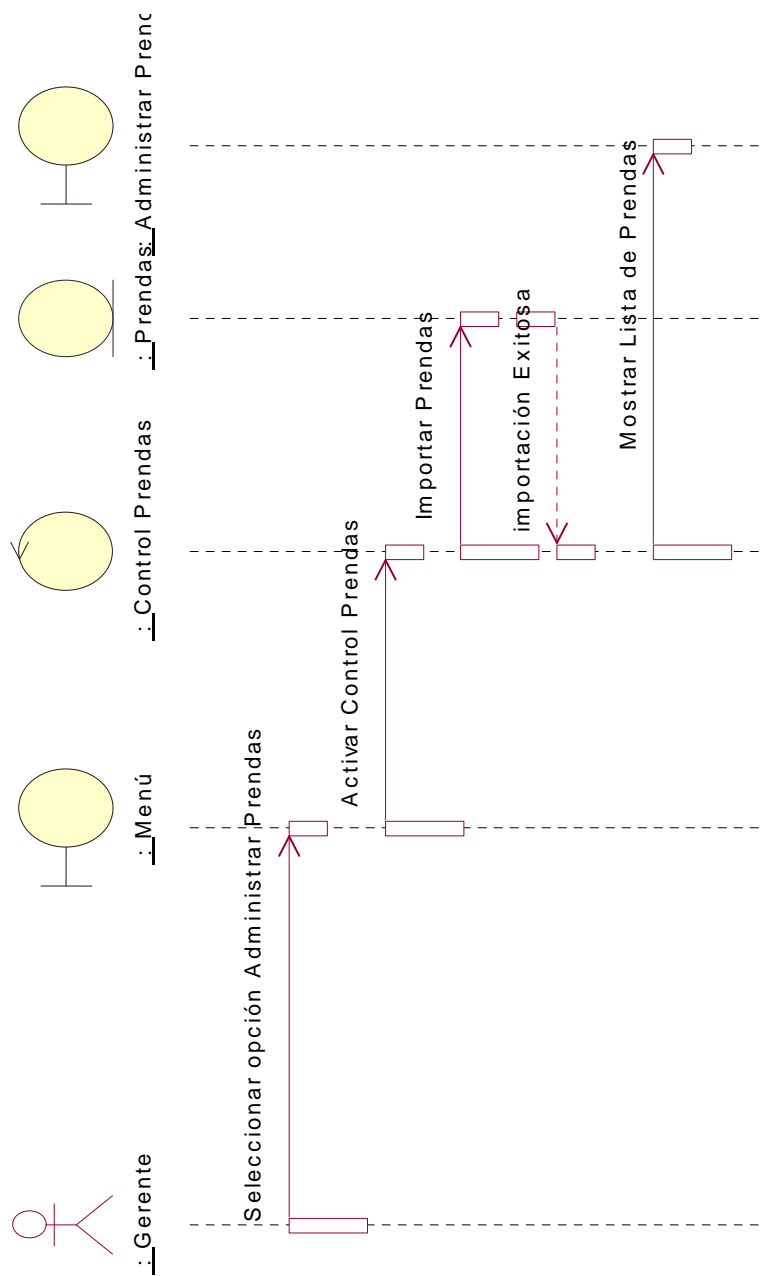
**Figura 131: Diagrama de Secuencia: Administrar Zonas**

**Diagrama de Secuencia: Nueva Zona**



**Figura 132: Diagrama de Secuencia: Nueva Zona**

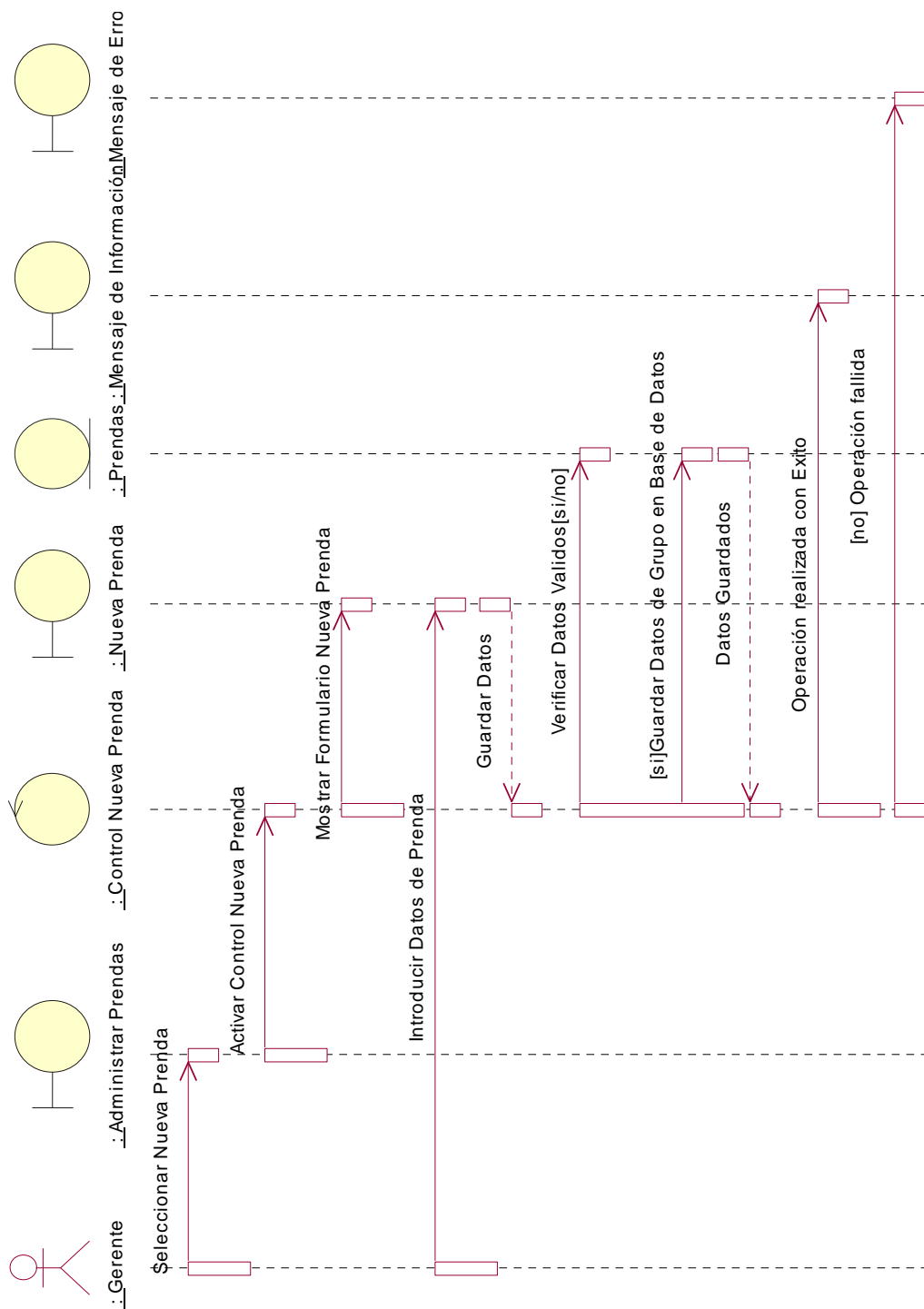
**Diagrama de Secuencia: Administrar Prendas**



**Figura 133: Diagrama de Secuencia: Administrar Prendas**



**Diagrama de Secuencia: Nueva Prenda**



**Figura 134: Diagrama de Secuencia: Nueva Prenda**

### Diagrama de Secuencia: Modificar Prenda

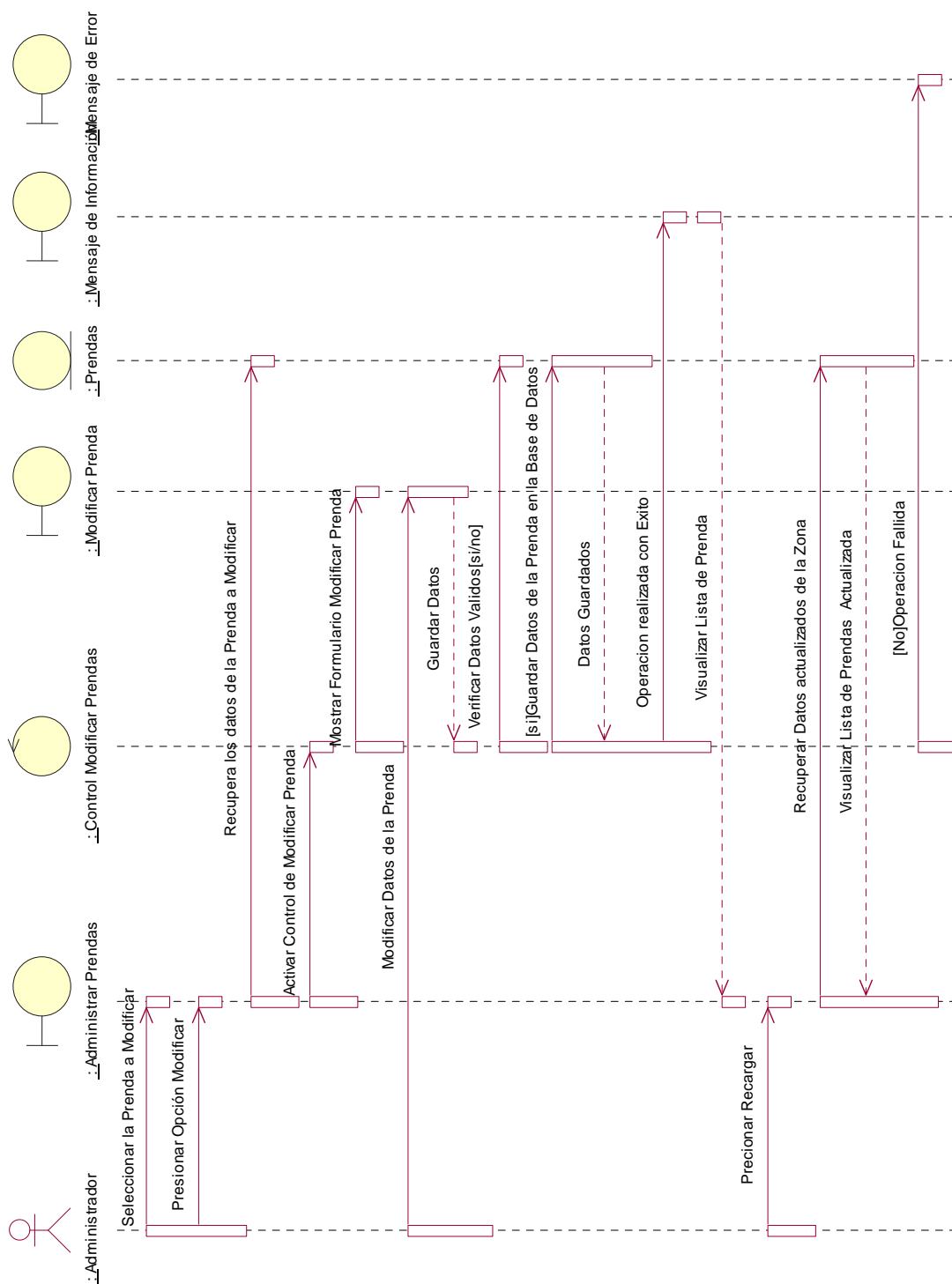
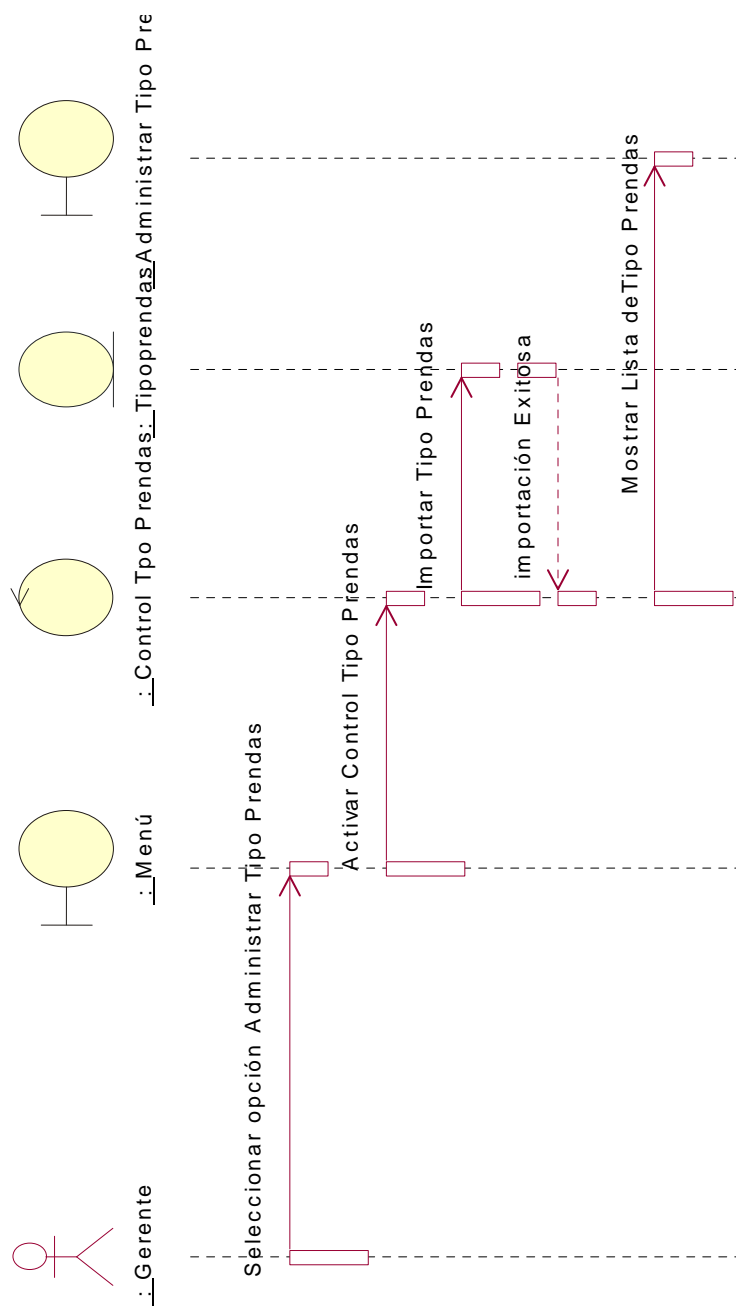


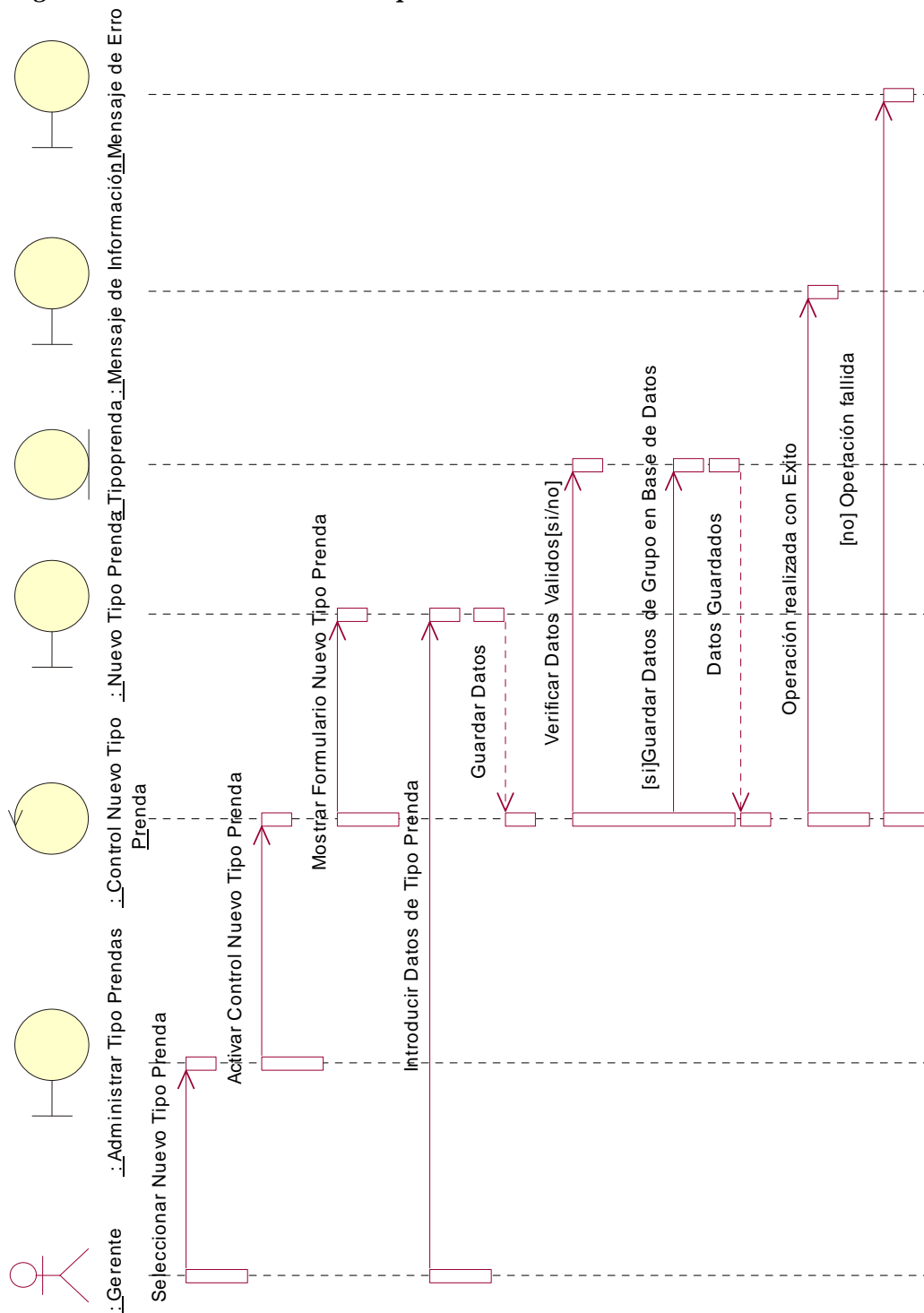
Figura 135: Diagrama de Secuencia: Modificar Prenda

**Diagrama de Secuencia: Administrar Tipo Prenda**



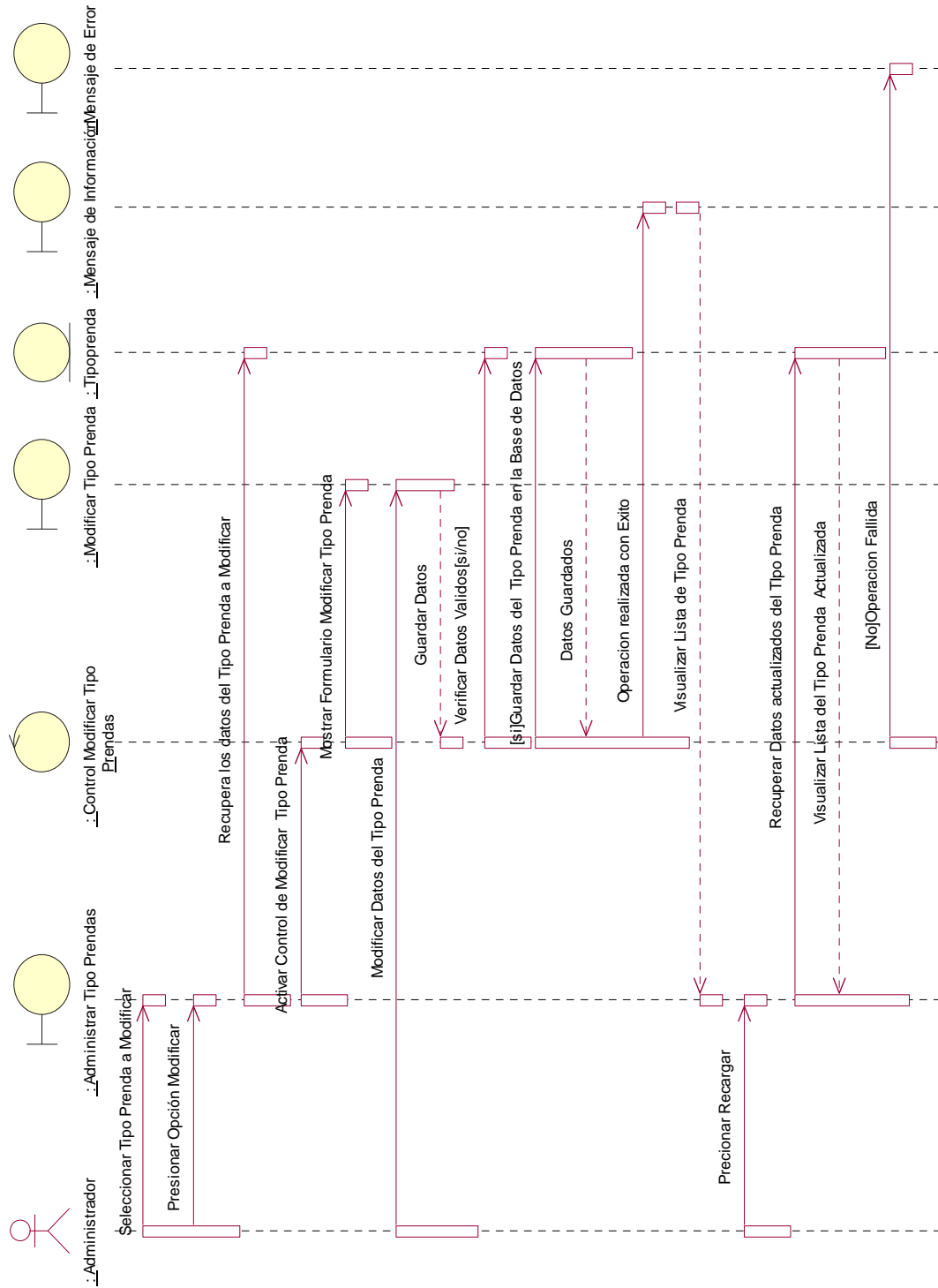
**Figura 136: Diagrama de Secuencia: Administrar Tipo Prenda**

**Diagrama de Secuencia: Nuevo Tipo Prenda**



**Figura 137: Diagrama de Secuencia: Nuevo Tipo Prenda**

**Diagrama de Secuencia: Modificar Tipo Prenda**



**Figura 138: Diagrama de Secuencia: Modificar Tipo Prenda**

### Diagrama de Secuencia: Borrar Tipo Prenda

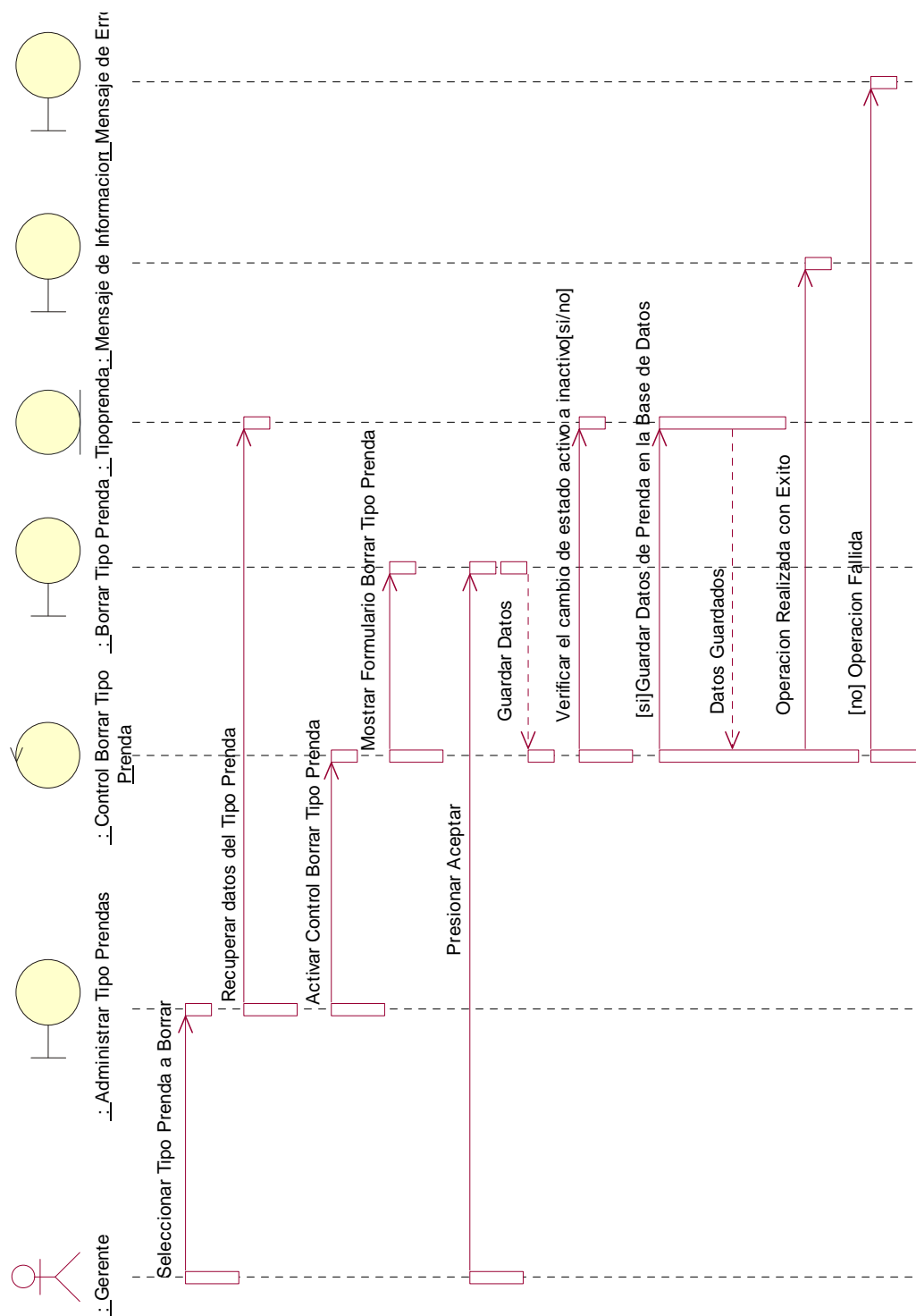
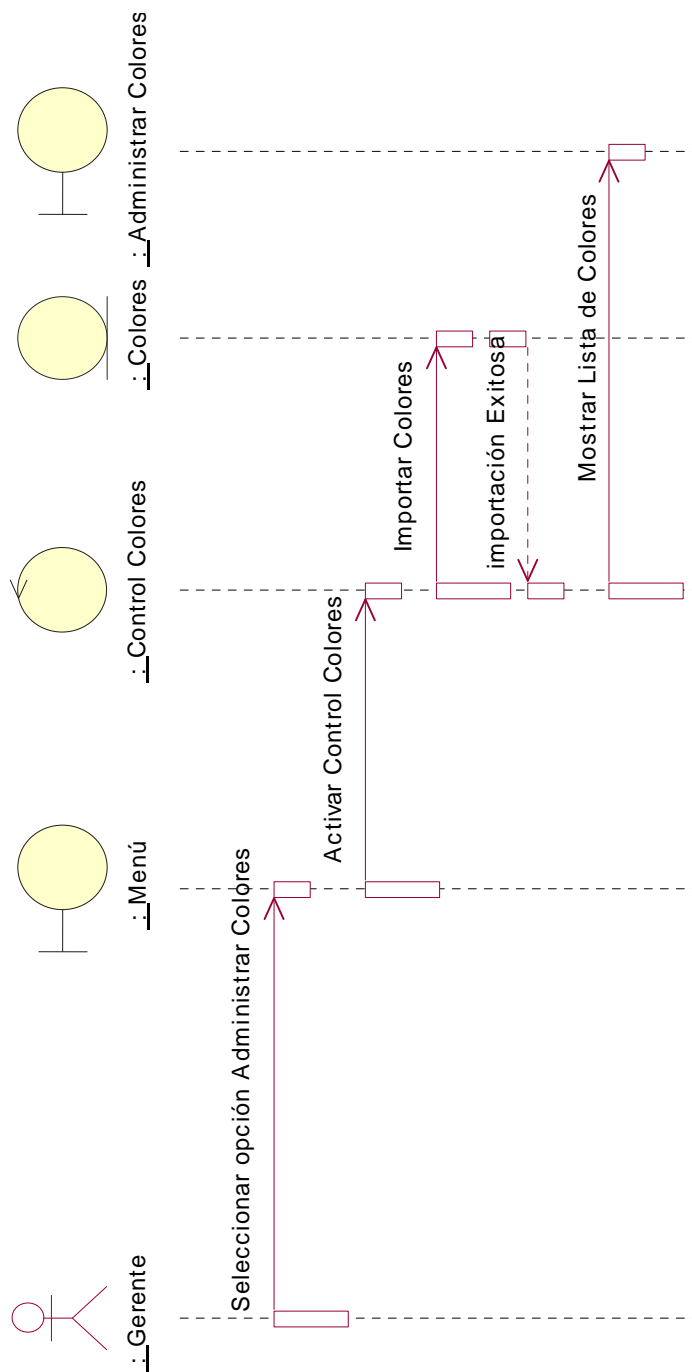


Figura 139: Diagrama de Secuencia: Borrar Tipo Prenda

*Diagrama de Secuencia: Administrar Color*



**Figura 140: Diagrama de Secuencia: Administrar Color**

### Diagrama de Secuencia: Nuevo Color

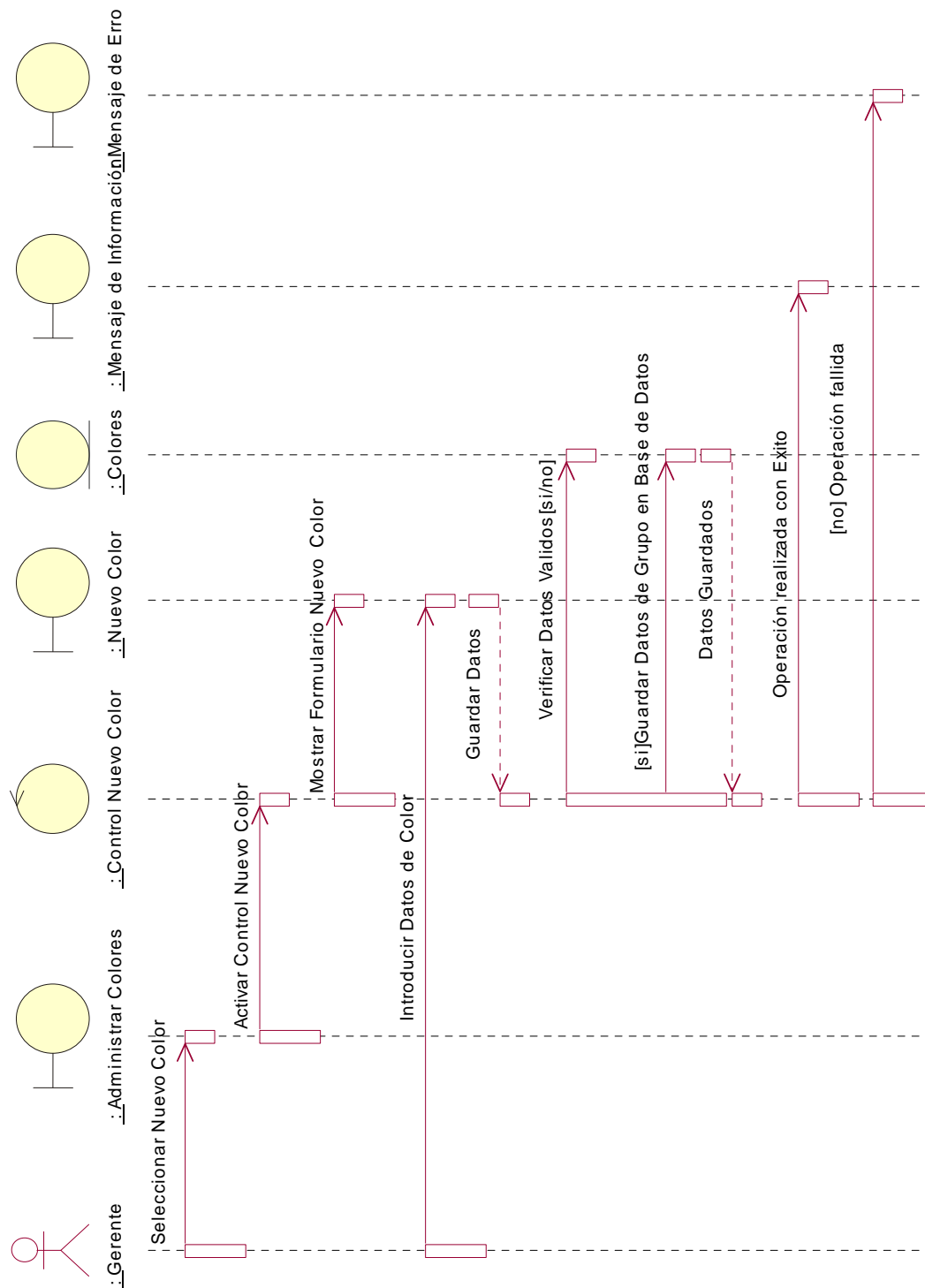


Figura 141: Diagrama de Secuencia: Nuevo Color



### Diagrama de Secuencia: Borrar Color

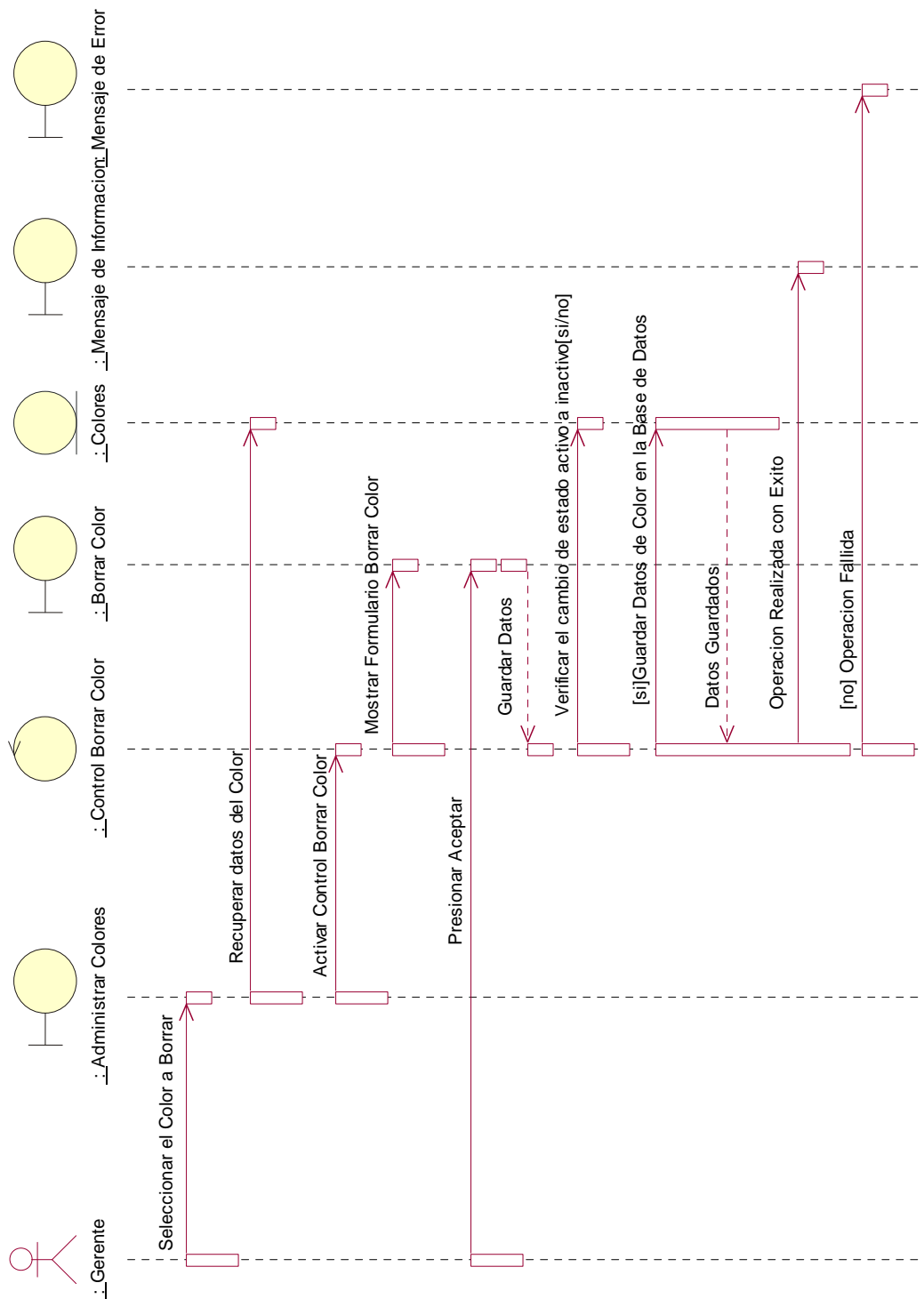
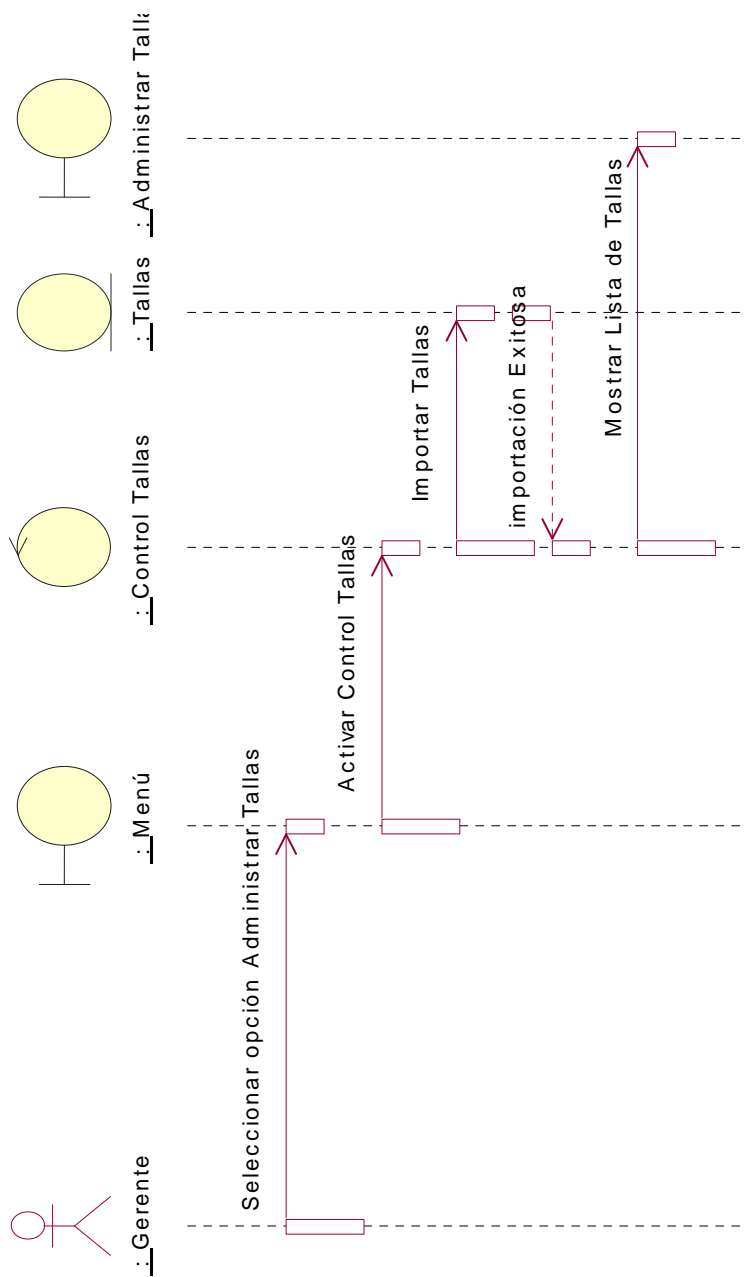


Figura 142: Diagrama de Secuencia: Borrar Color

**Diagrama de Secuencia: Administrar Tallas**



**Figura 143: Diagrama de Secuencia: Administrar Tallas**

### Diagrama de Secuencia: Nueva Talla

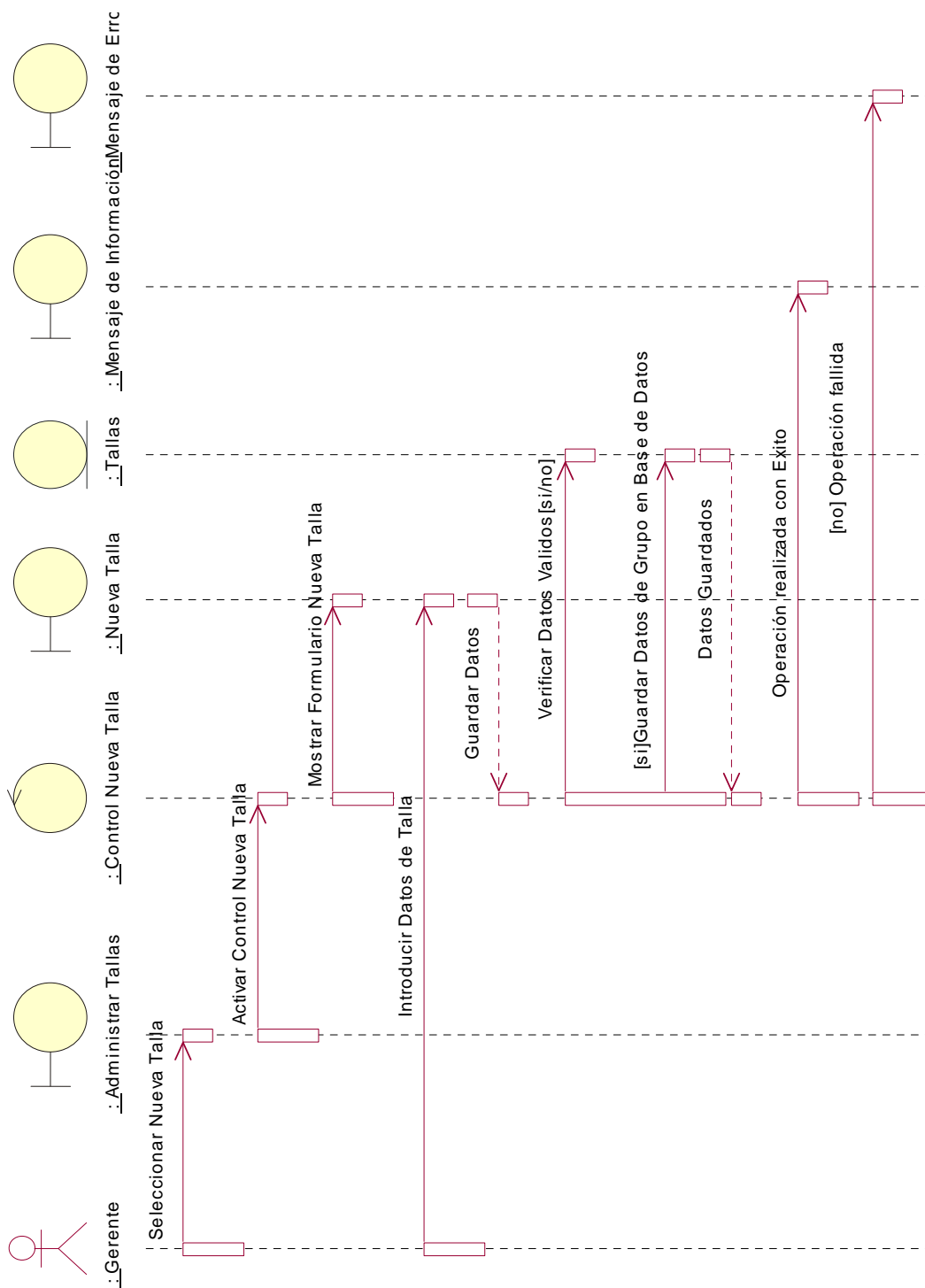


Figura 144: Diagrama de Secuencia: Nueva Talla

### Diagrama de Secuencia: Borrar Talla

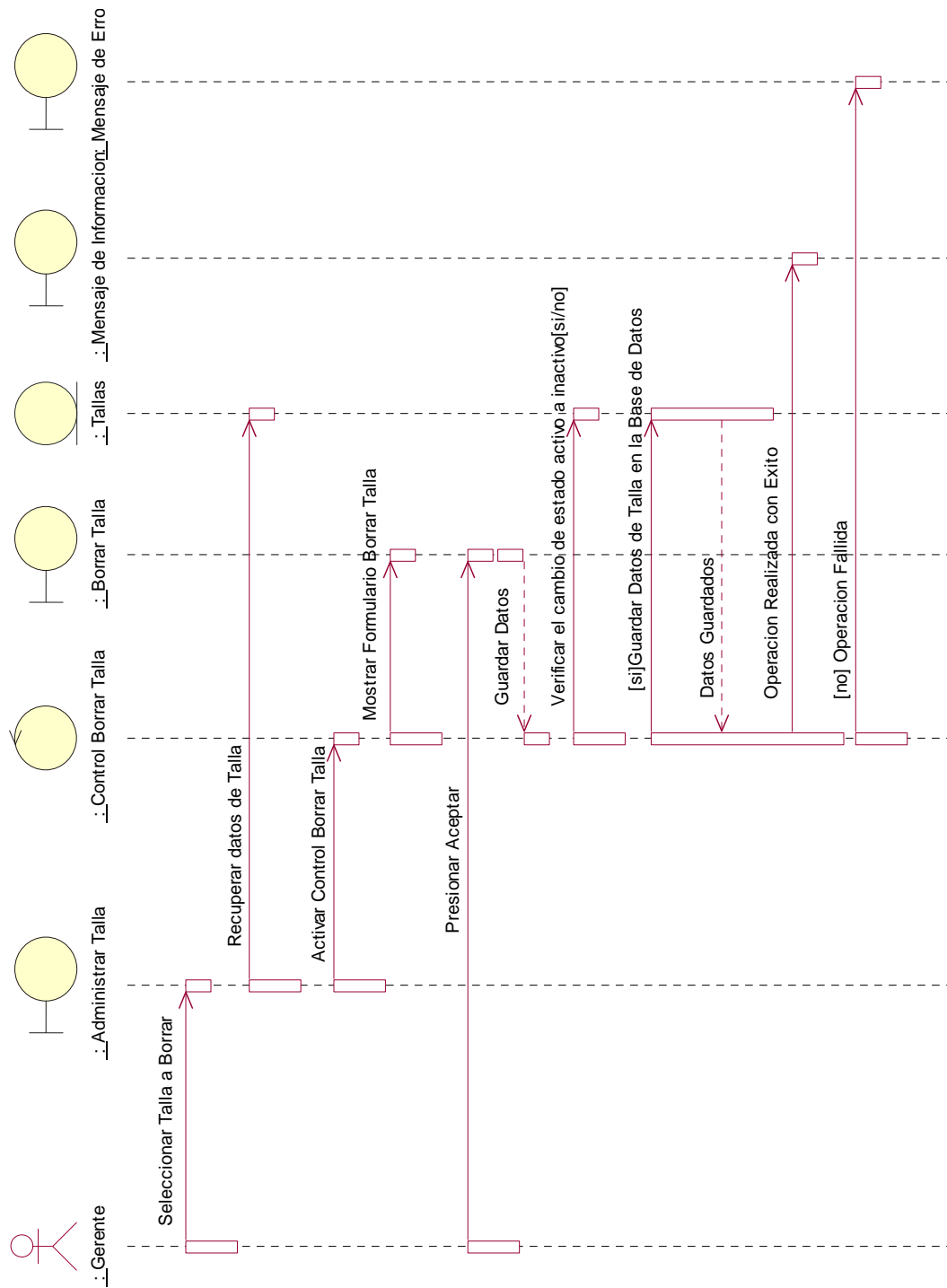
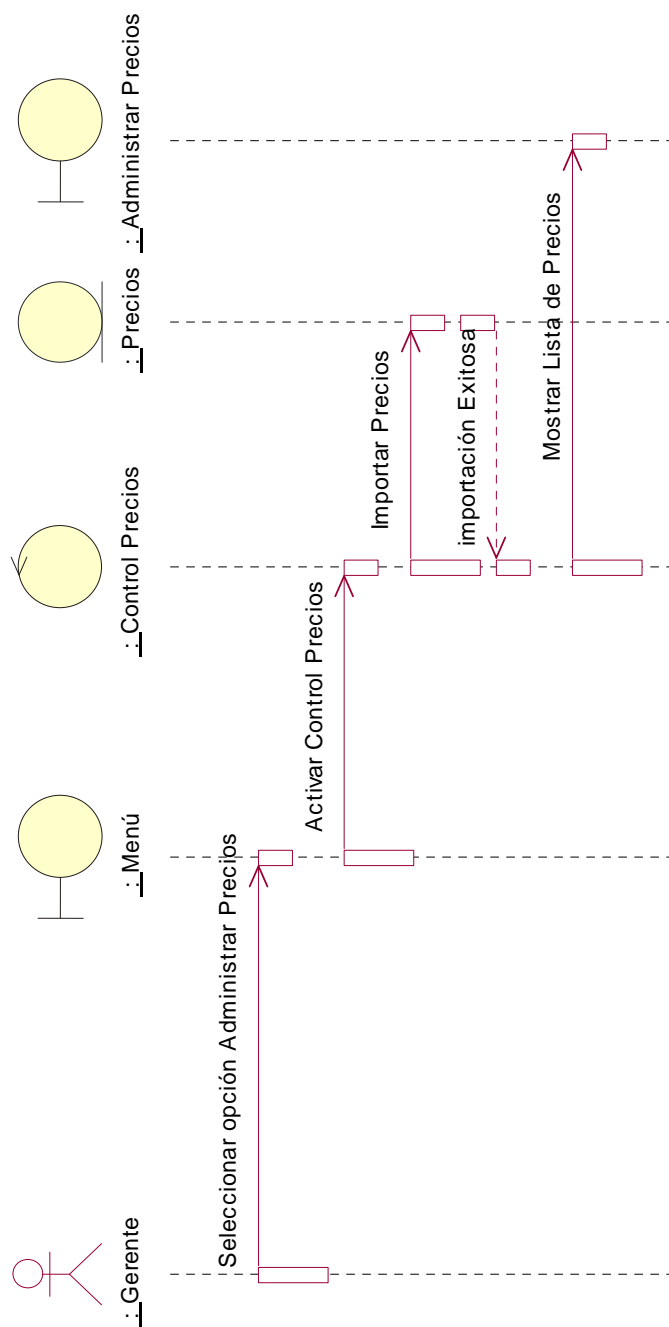


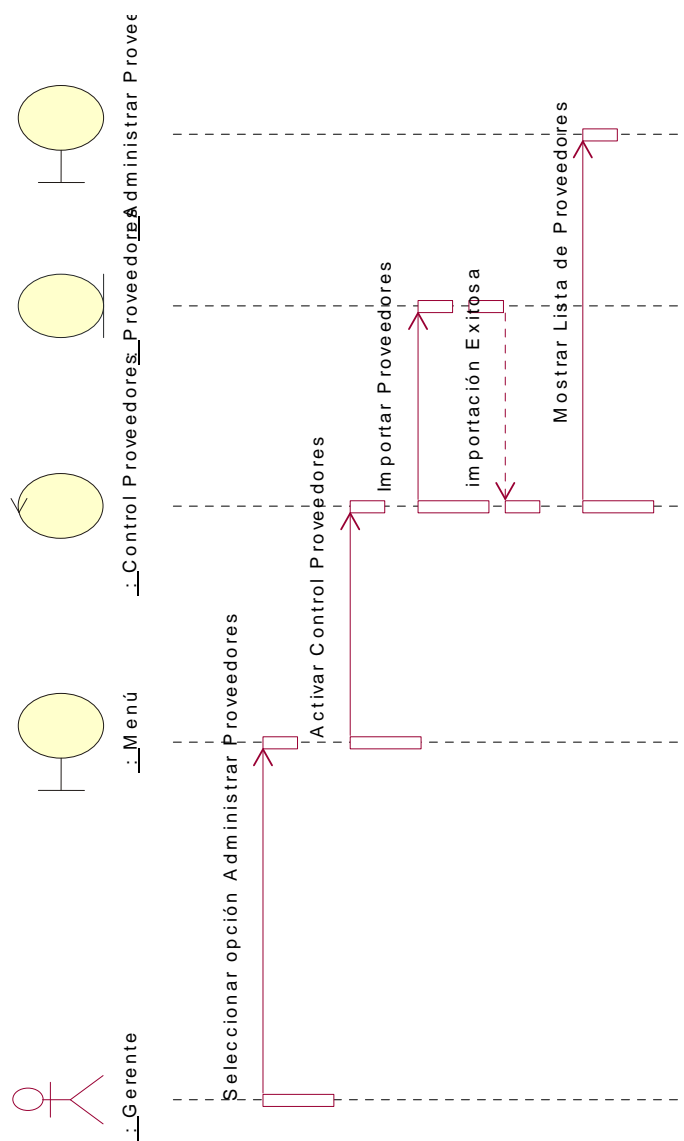
Figura 145: Diagrama de Secuencia: Borrar Talla

*Diagrama de Secuencia: Listar Precios*



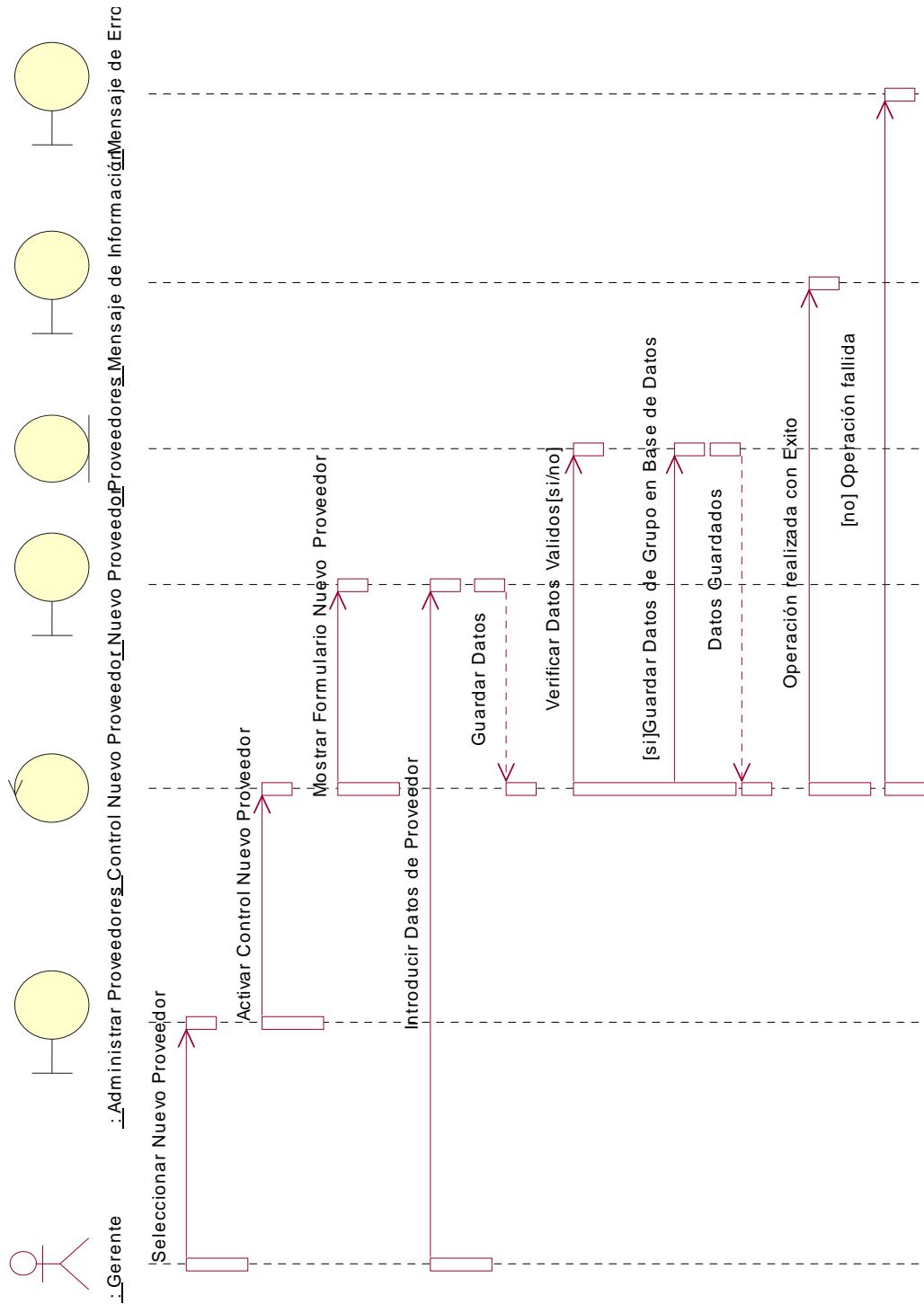
**Figura 146: Diagrama de Secuencia: Administrar Precios**

*Diagrama de Secuencia: Administrar Proveedor*



**Figura 147: Diagrama de Secuencia: Administrar Proveedor**

**Diagrama de Secuencia: Nuevo Proveedor**



**Figura 148: Diagrama de Secuencia: Nuevo Proveedor**

### Diagrama de Secuencia: Modificar Proveedor

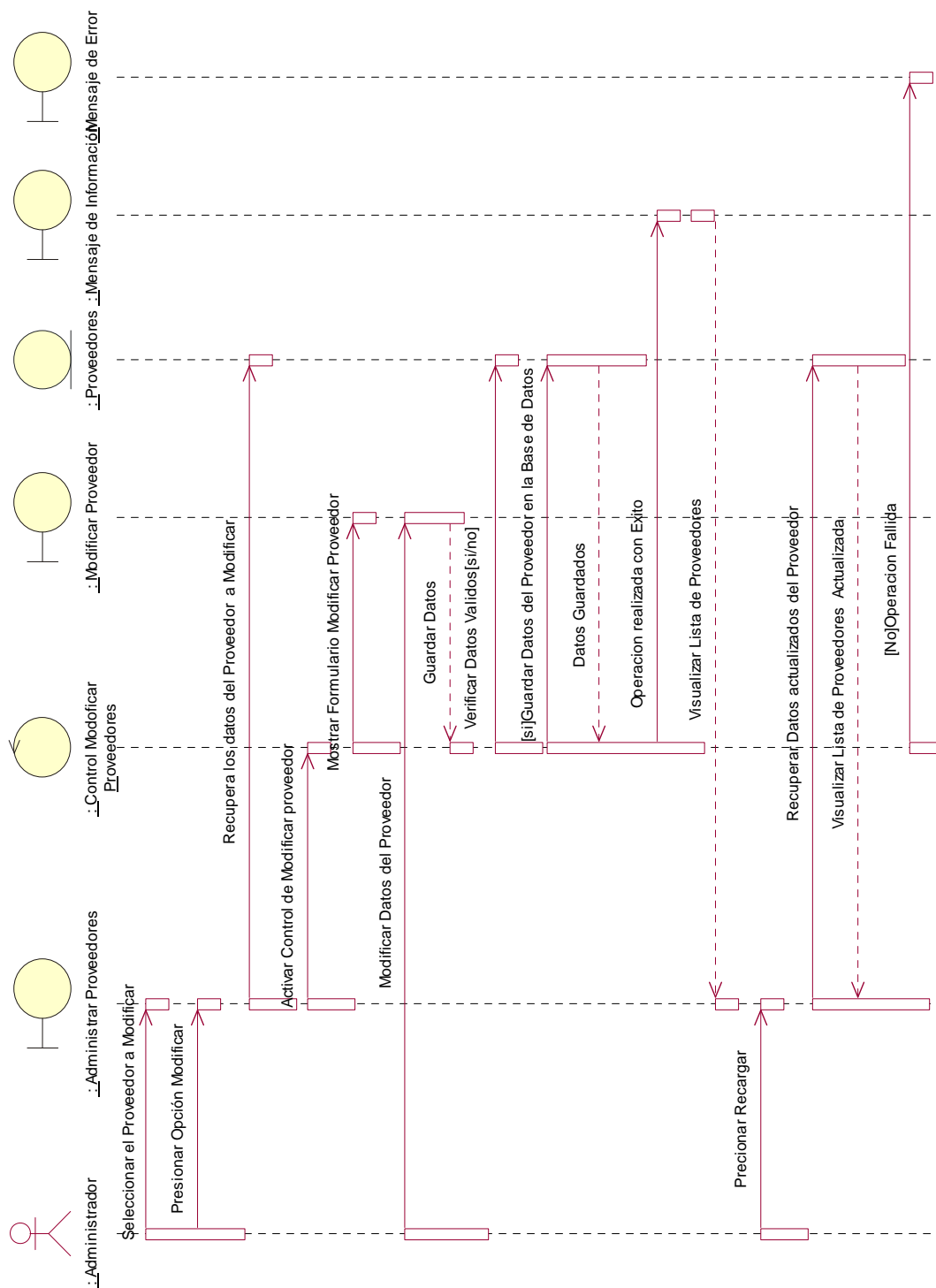


Figura 149: Diagrama de Secuencia: Modificar Proveedor



### Diagrama de Secuencia: Borrar Proveedor

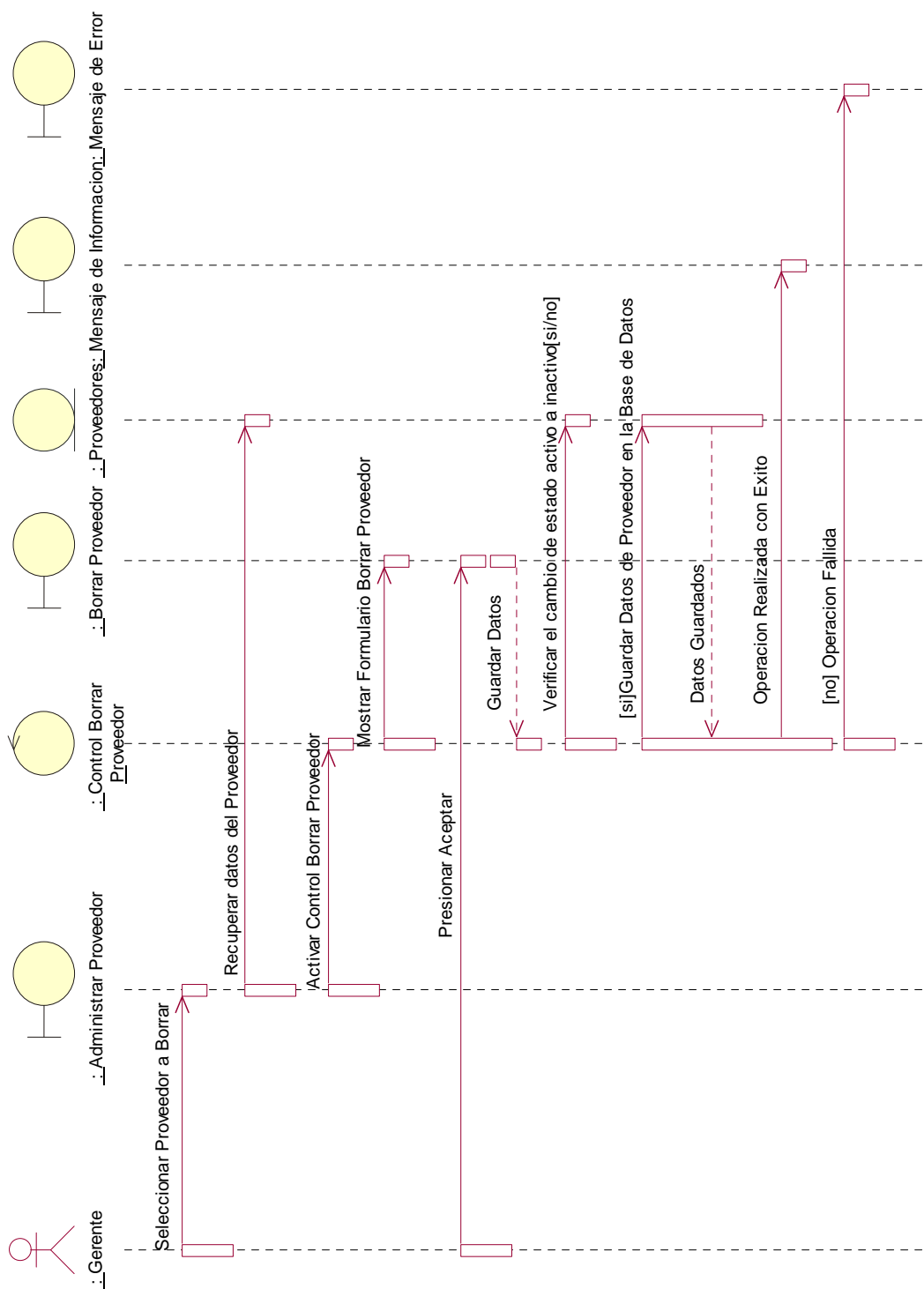
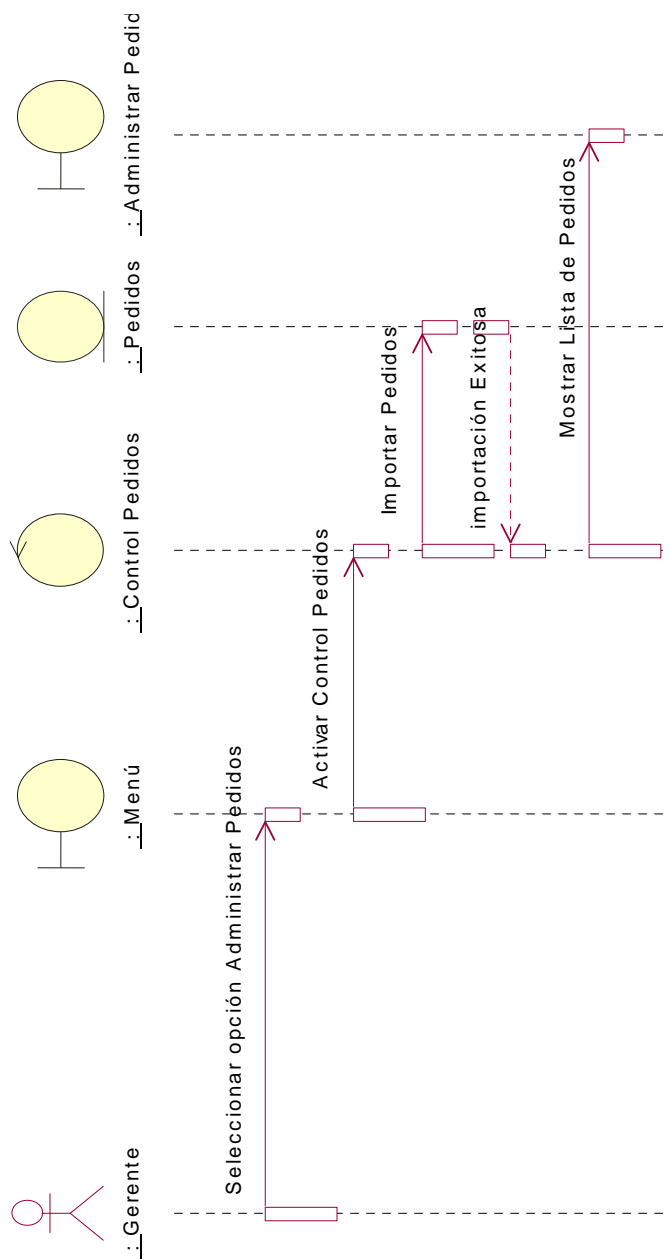


Figura 150: Diagrama de Secuencia: Borrar Proveedor

*Diagrama de Secuencia: Administrar Pedidos*



**Figura 151: Diagrama de Secuencia: Administrar Pedidos**

### Diagrama de Secuencia: Nuevo Pedido

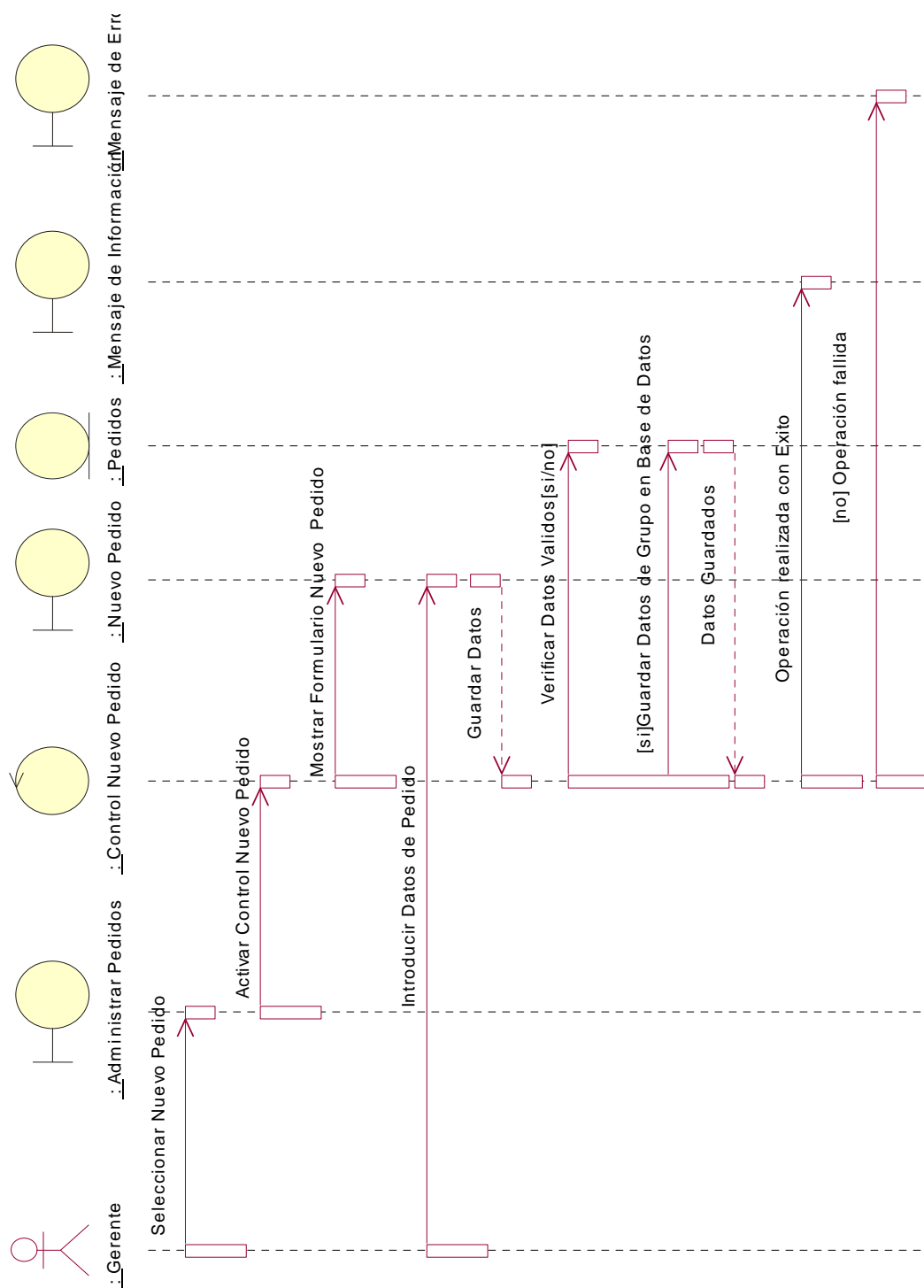
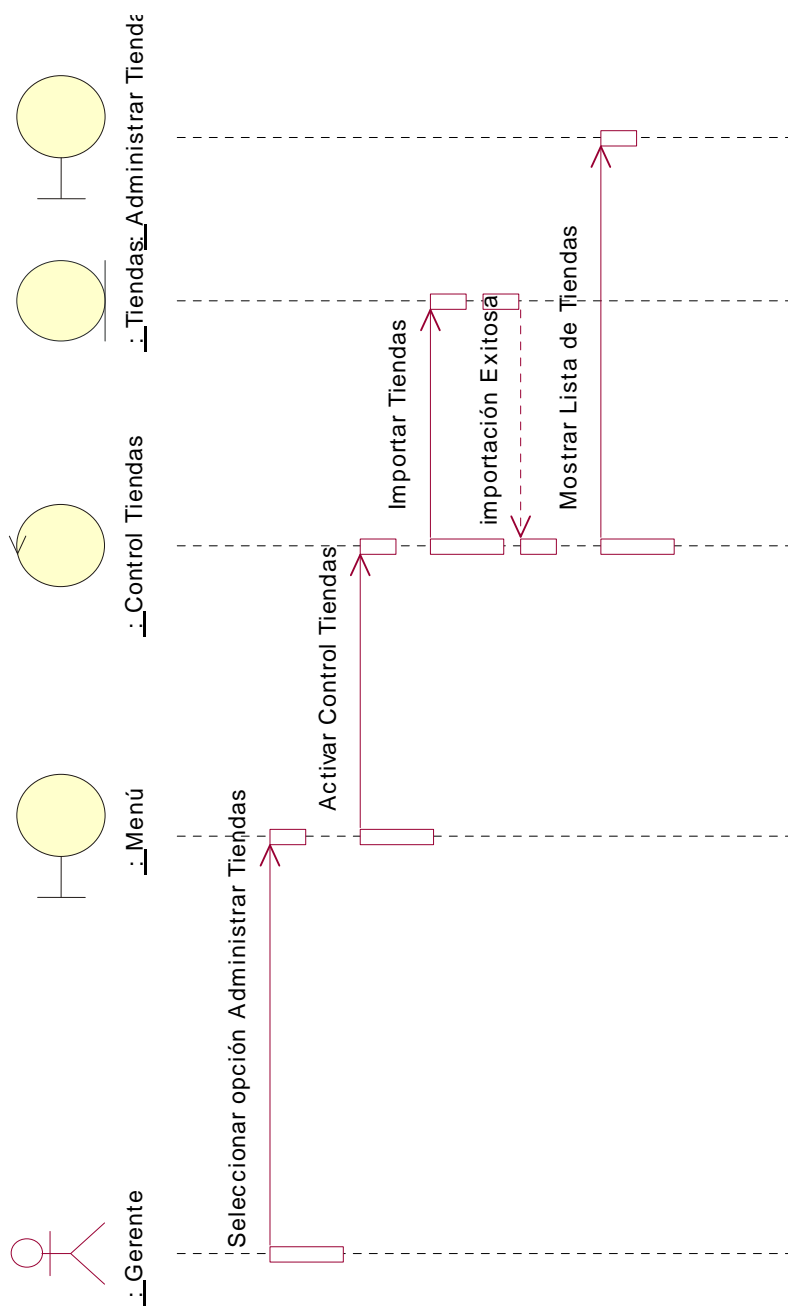


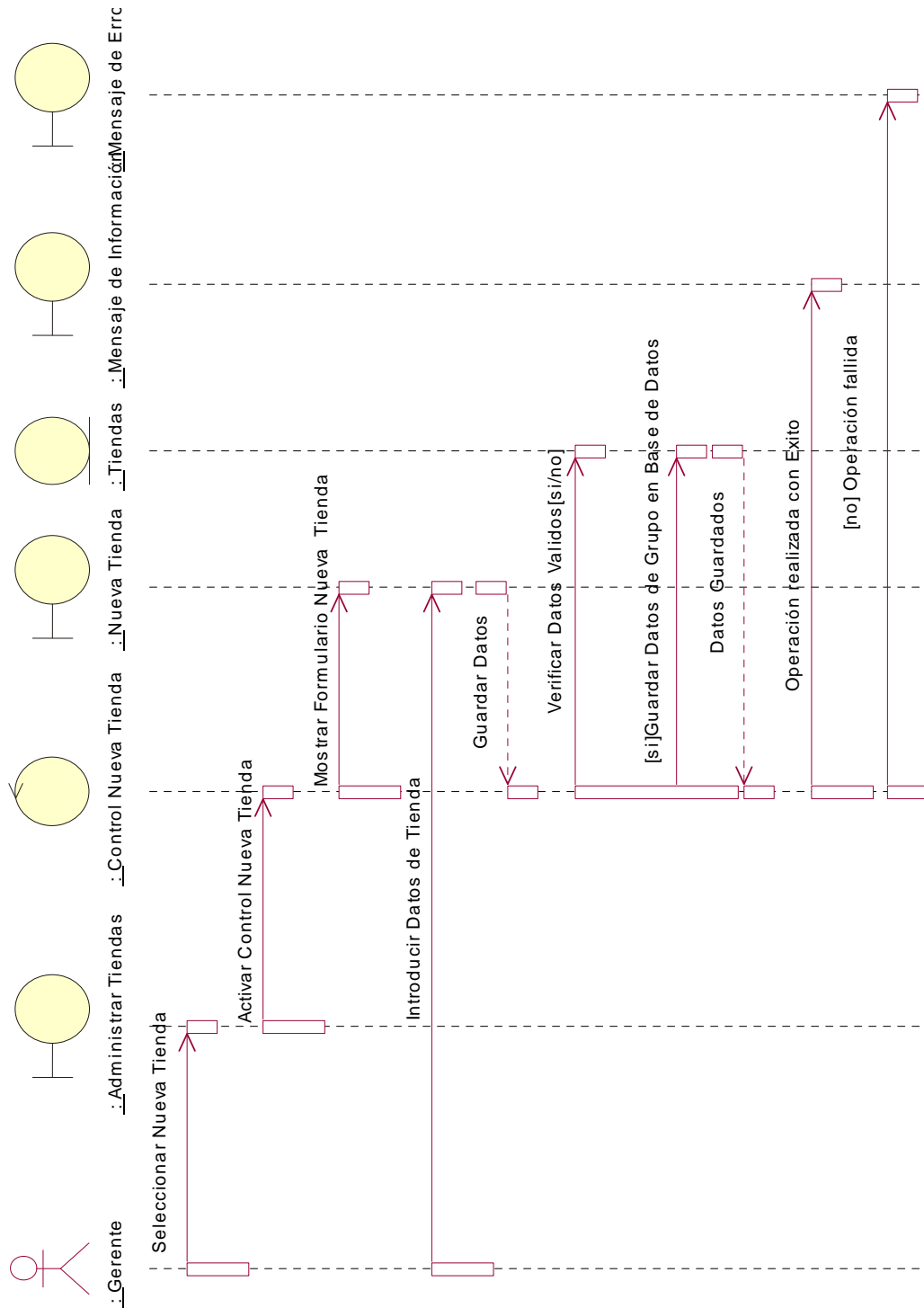
Figura 152: Diagrama de Secuencia: Nuevo Pedido

*Diagrama de Secuencia: Administrar Tiendas*



**Figura 153: Diagrama de Secuencia: Administrar Tiendas**

**Diagrama de Secuencia: Nueva Tienda**



**Figura 154: Diagrama de Secuencia: Nueva Tienda**

### Diagrama de Secuencia: Modificar Tienda

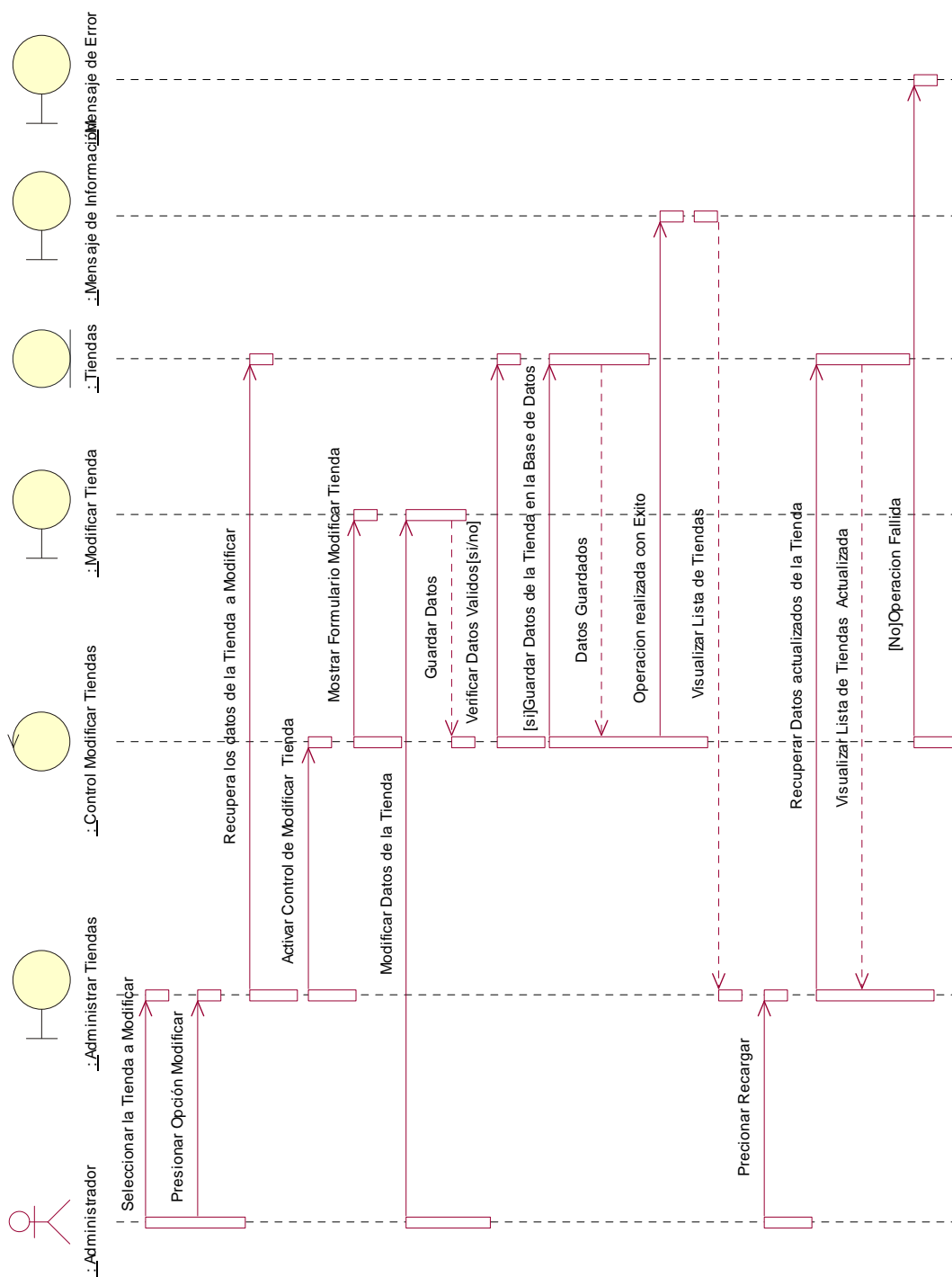


Figura 155: Diagrama de Secuencia: Modificar Tienda

### Diagrama de Secuencia: Borrar Tienda

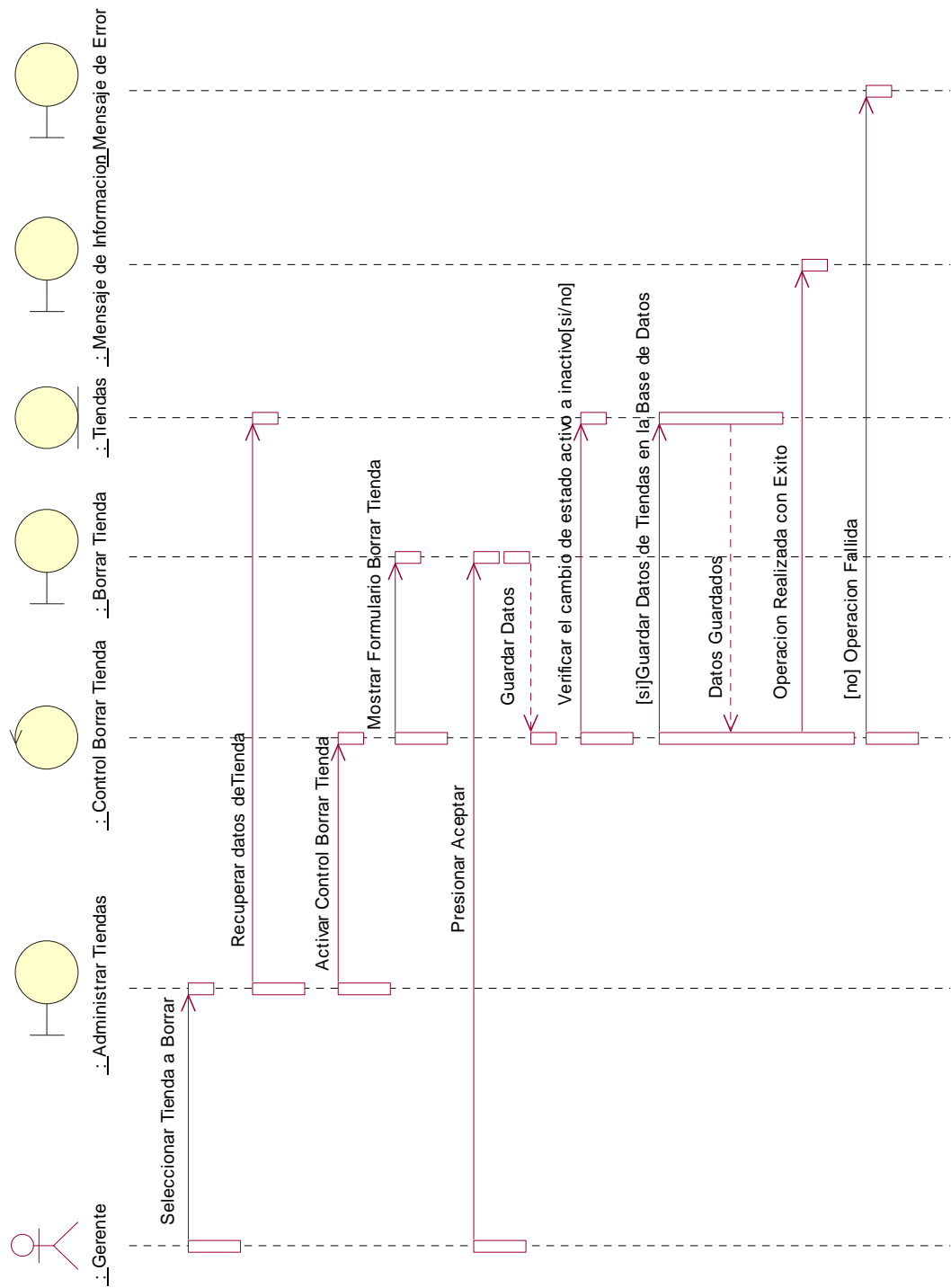
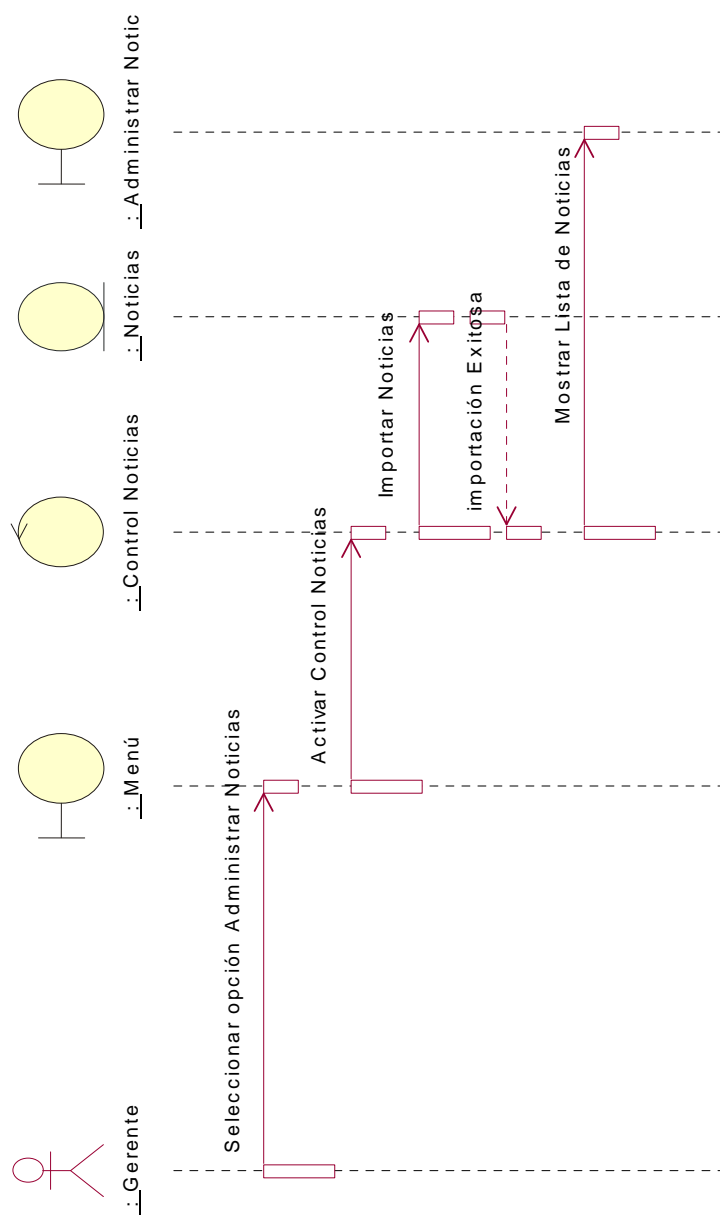


Figura 156: Diagrama de Secuencia: Borrar Tienda

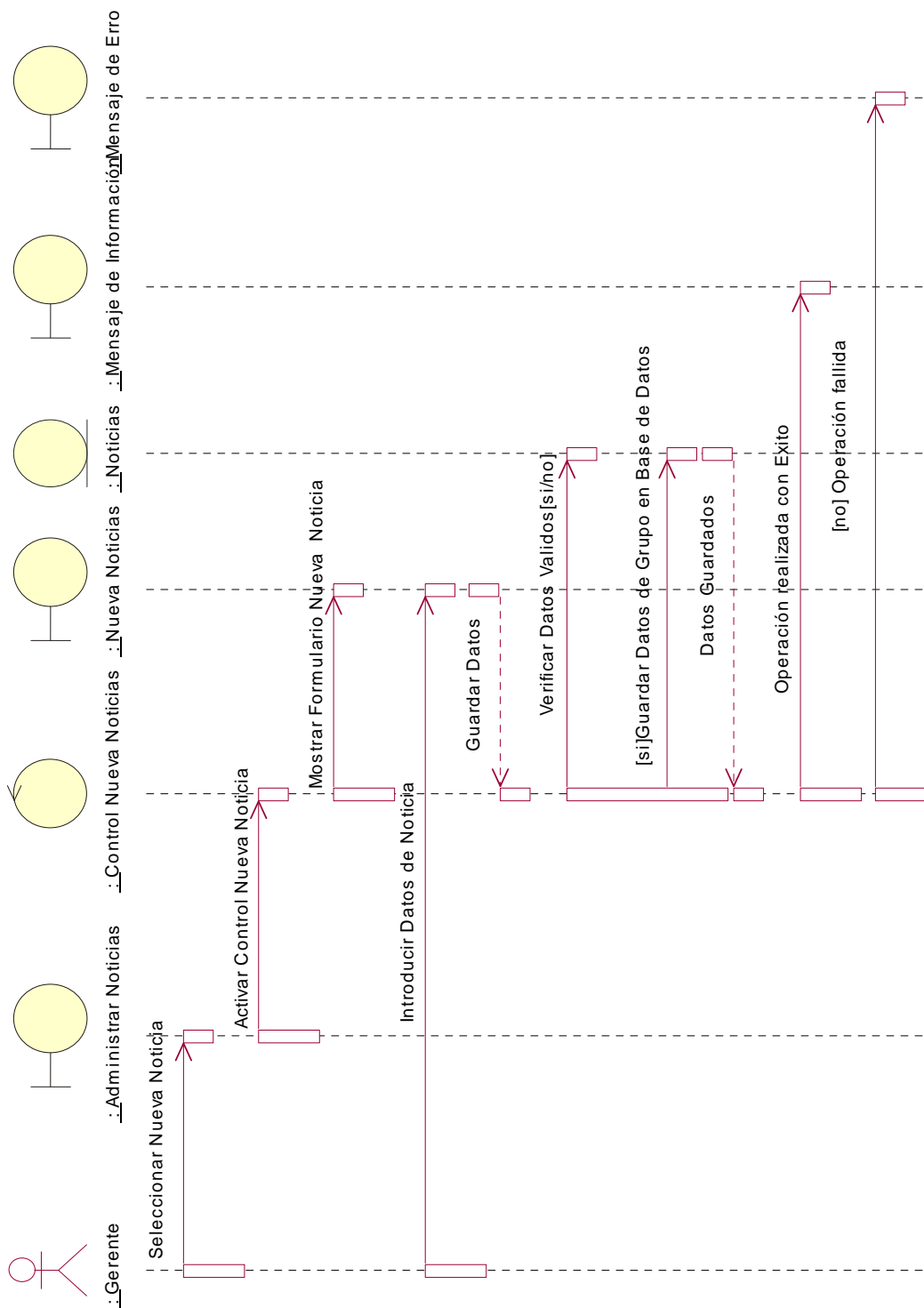
*Diagrama de Secuencia: Administrar Noticias*



**Figura 157: Diagrama de Secuencia: Administrar Noticias**

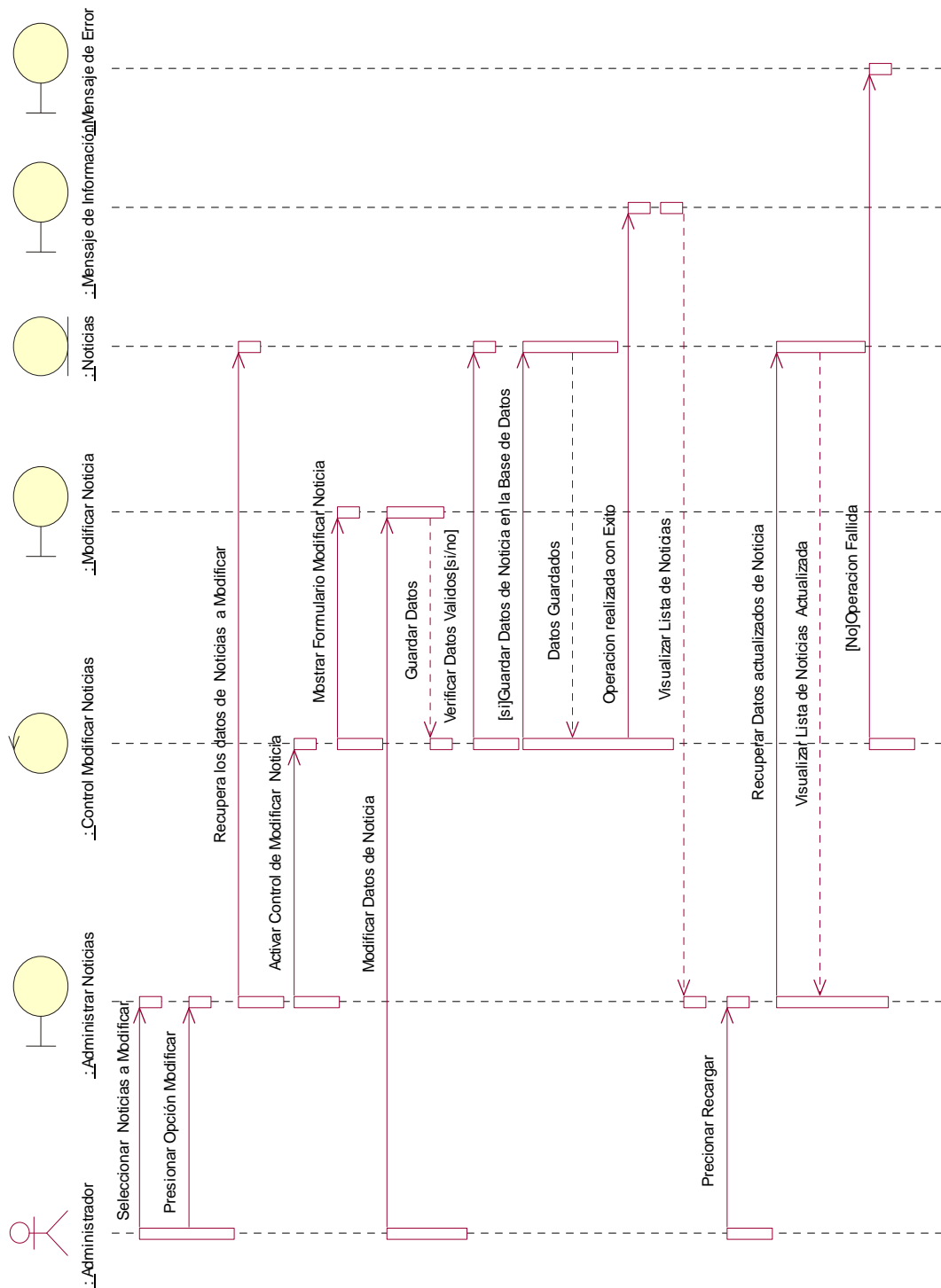


**Diagrama de Secuencia: Nueva Noticia**



**Figura 158: Diagrama de Secuencia: Nueva Noticia**

**Diagrama de Secuencia: Modificar Noticia**



**Figura 159: Diagrama de Secuencia: Modificar Noticia**

### Diagrama de Secuencia: Borrar Noticia

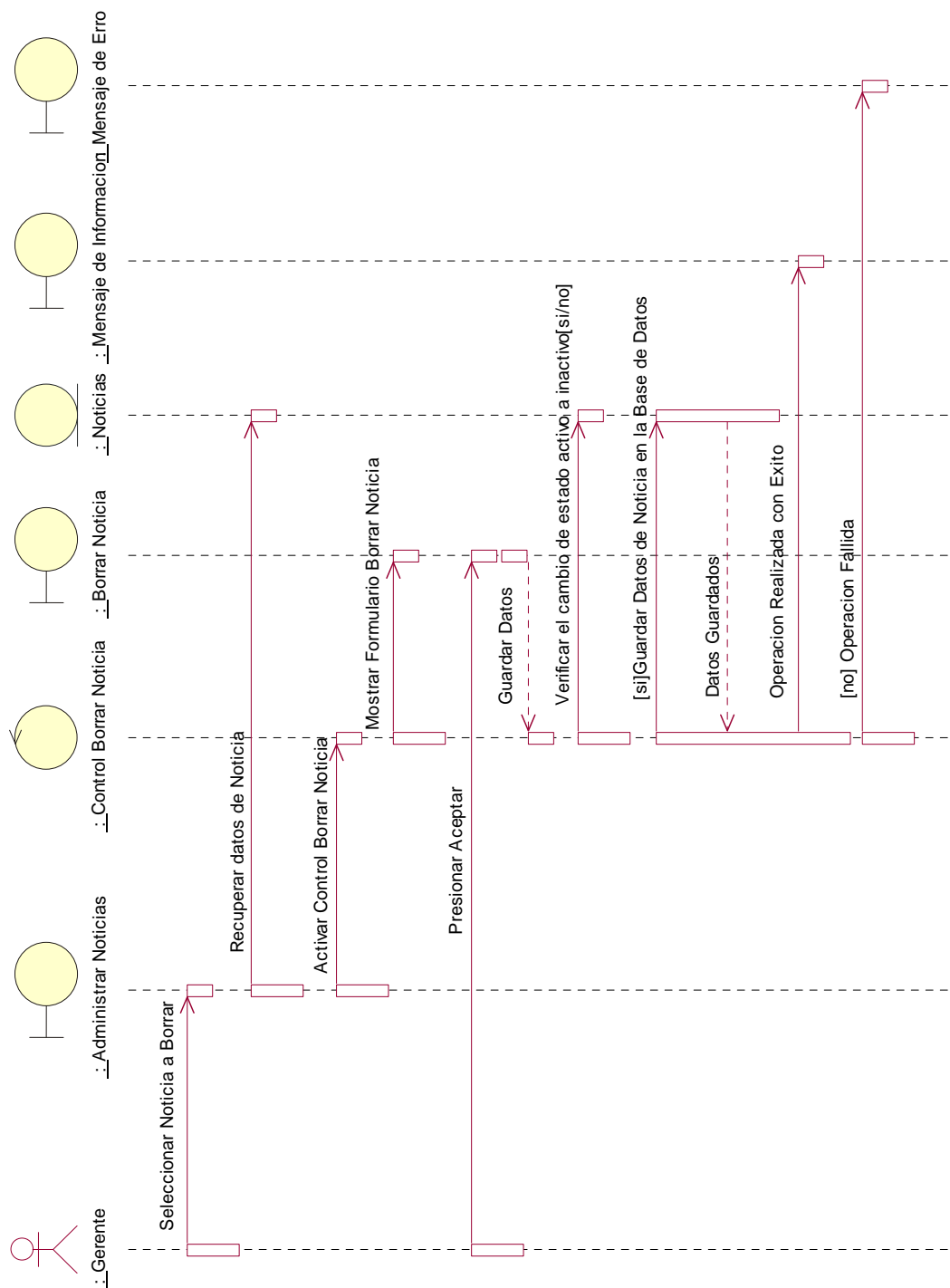
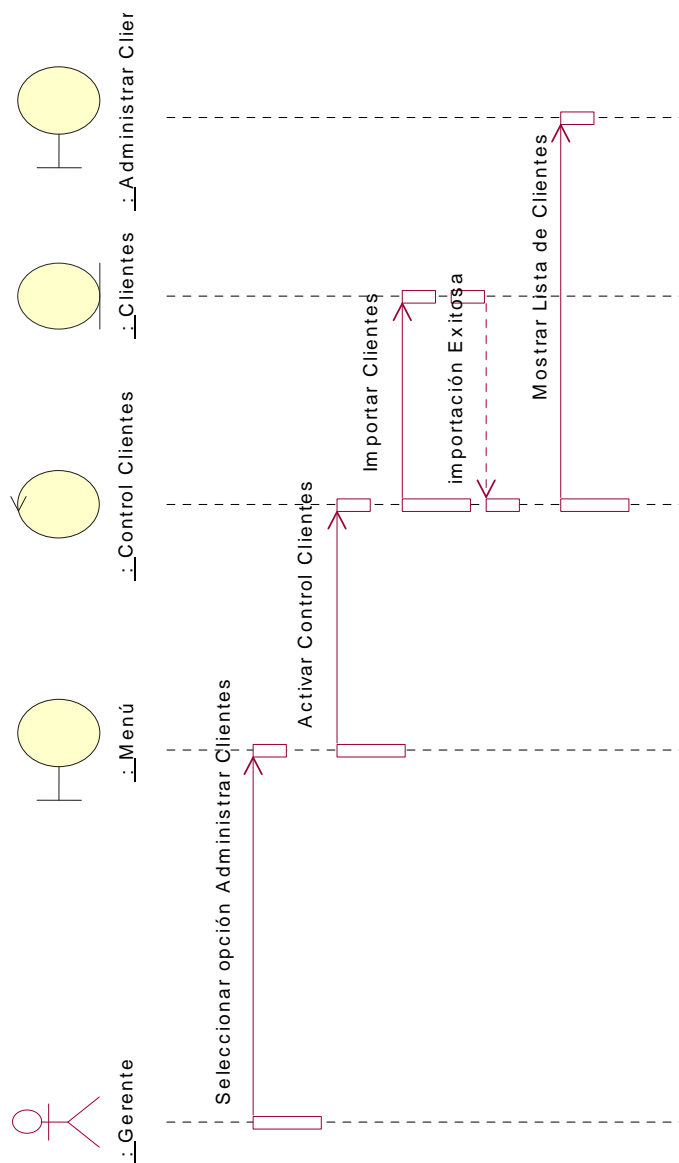


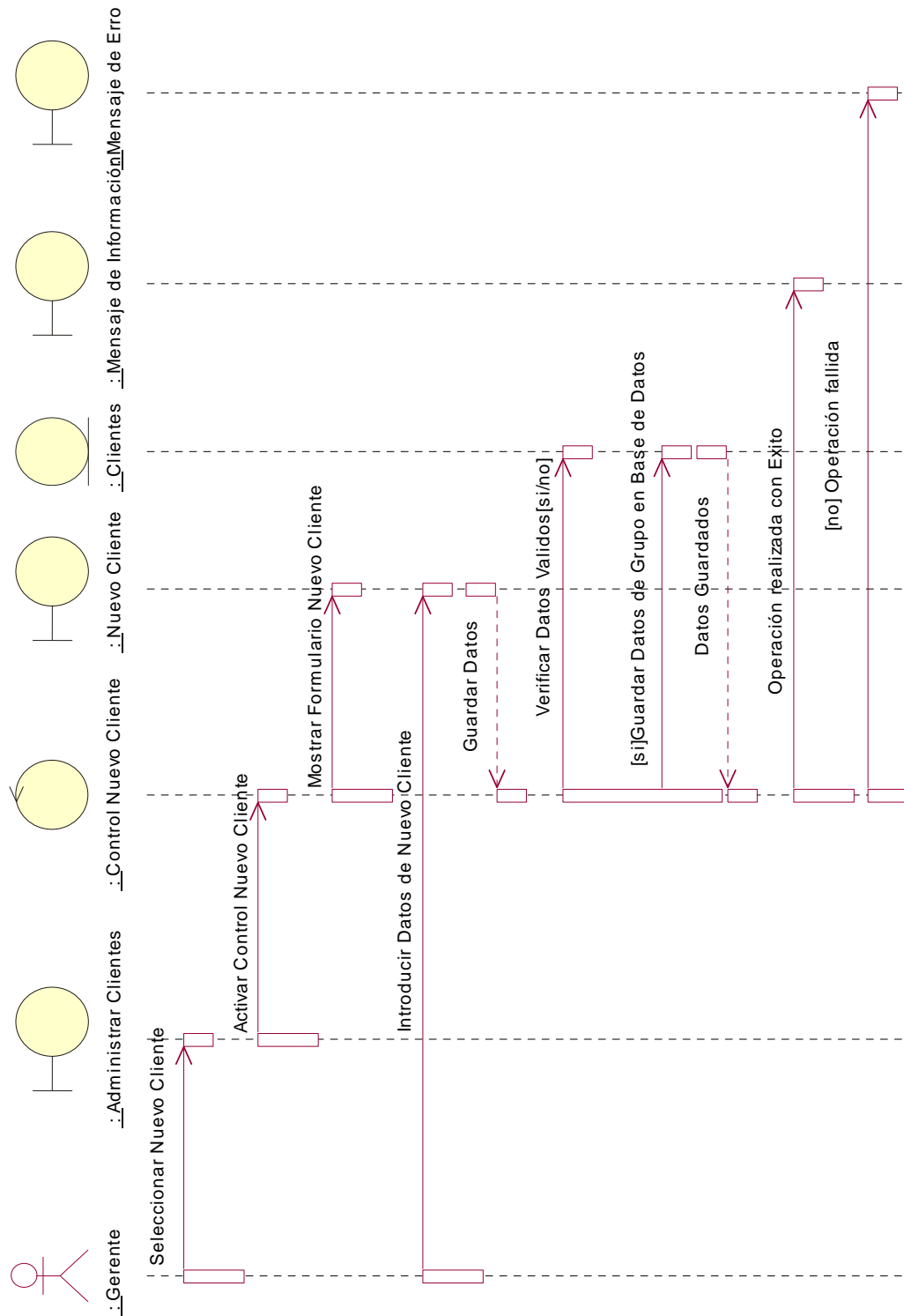
Figura 160: Diagrama de Secuencia: Borrar Noticia

*Diagrama de Secuencia: Administrar Clientes*



**Figura 161: Diagrama de Secuencia: Administrar Clientes**

**Diagrama de Secuencia: Nuevo Cliente**



**Figura 162: Diagrama de Secuencia: Nuevo Cliente**

### Diagrama de Secuencia: Modificar Cliente

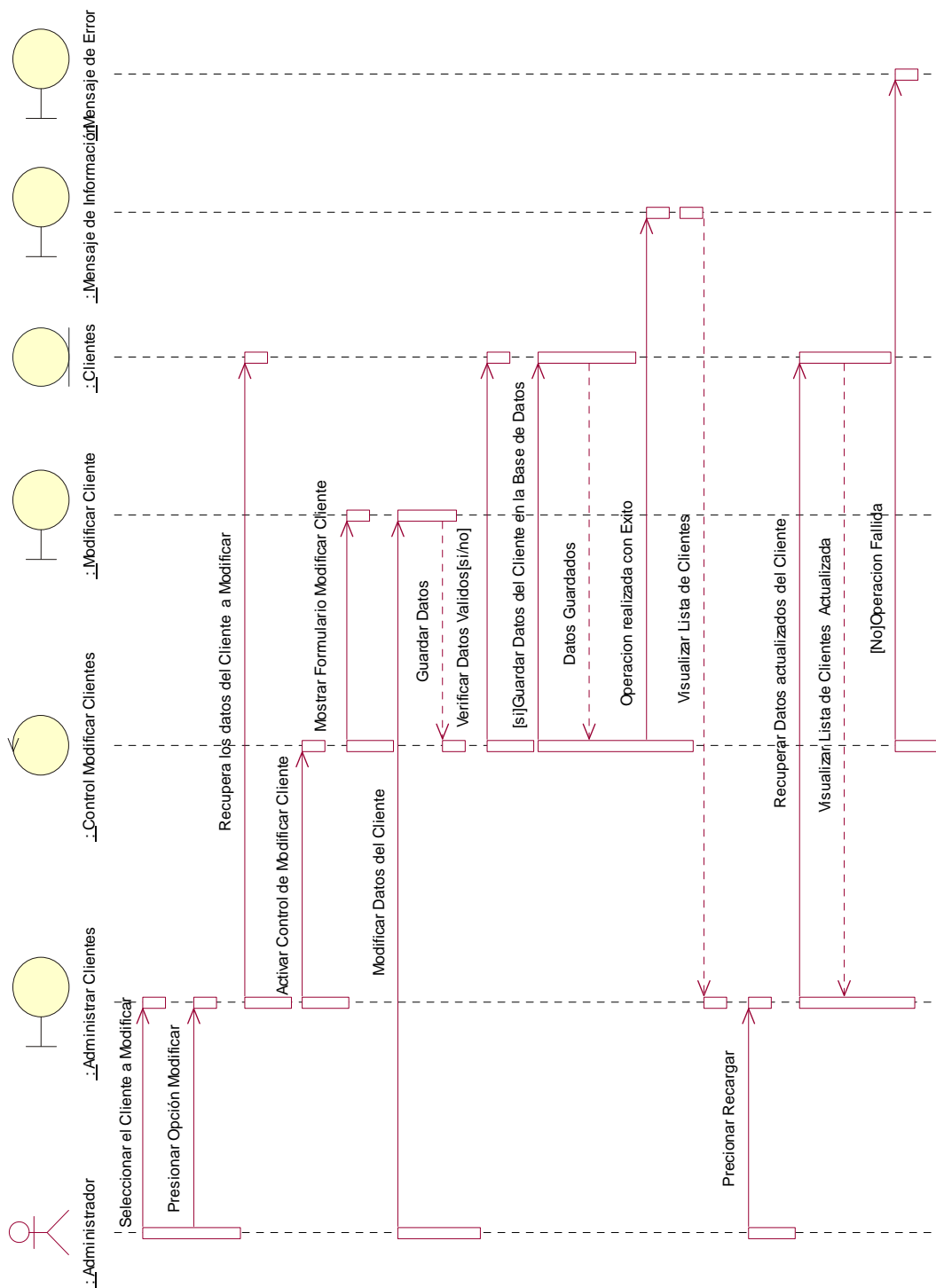
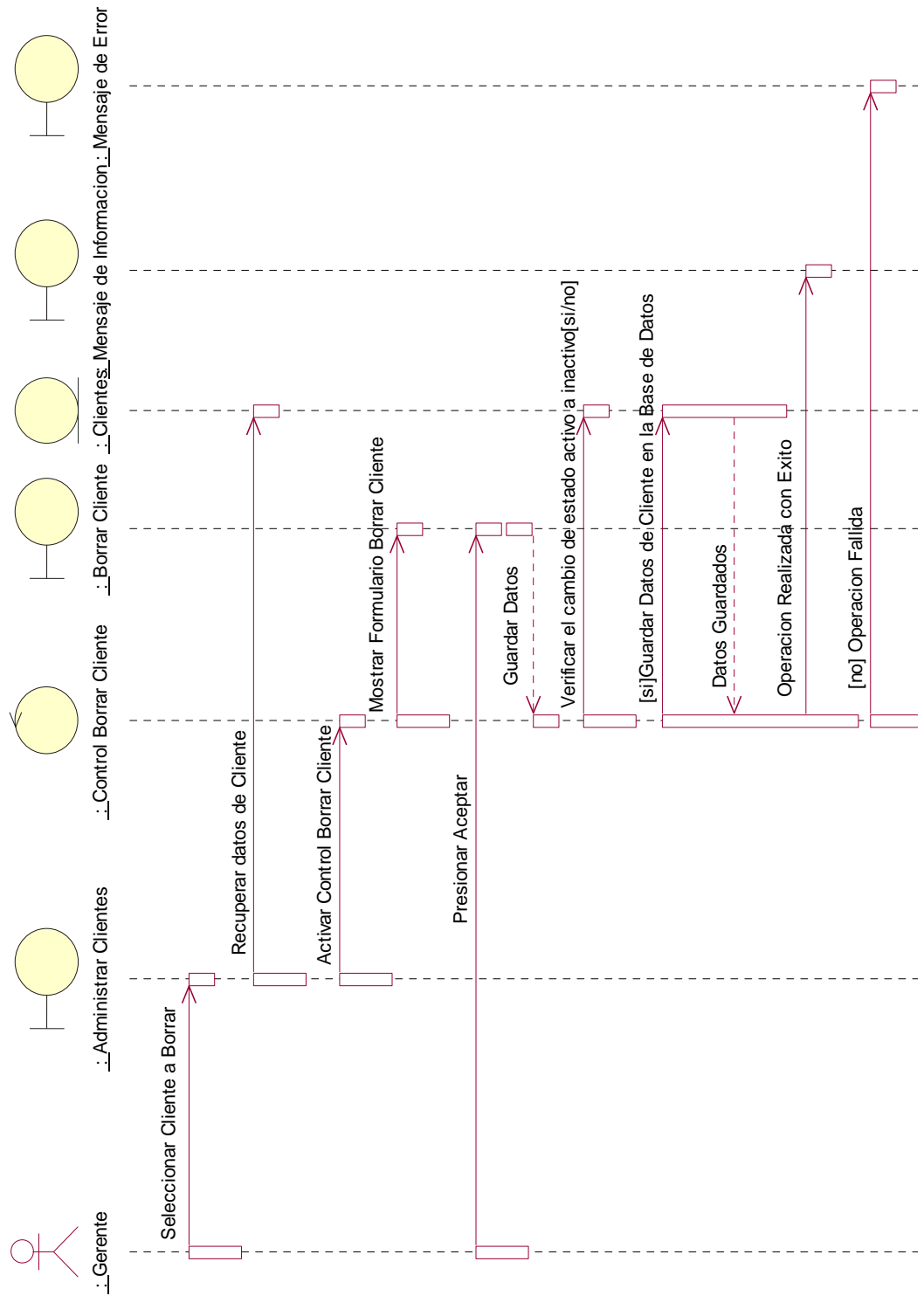


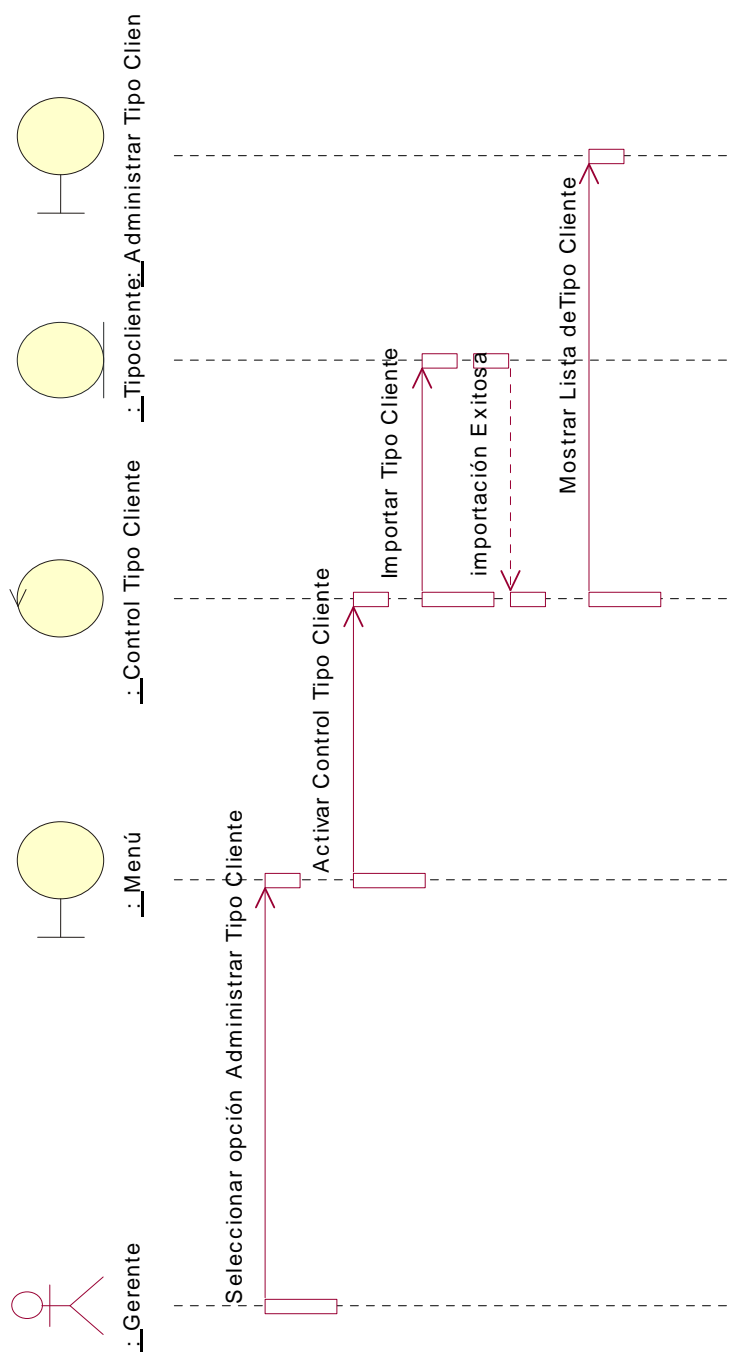
Figura 163: Diagrama de Secuencia: Modificar Cliente

**Diagrama de Secuencia: Borrar Cliente**



**Figura 164: Diagrama de Secuencia: Borrar Cliente**

*Diagrama de Secuencia: Administrar Tipo Clientes*



**Figura 165: Diagrama de Secuencia: Administrar Tipo Clientes**



Diagrama de Secuencia: Nuevo Tipo Cliente

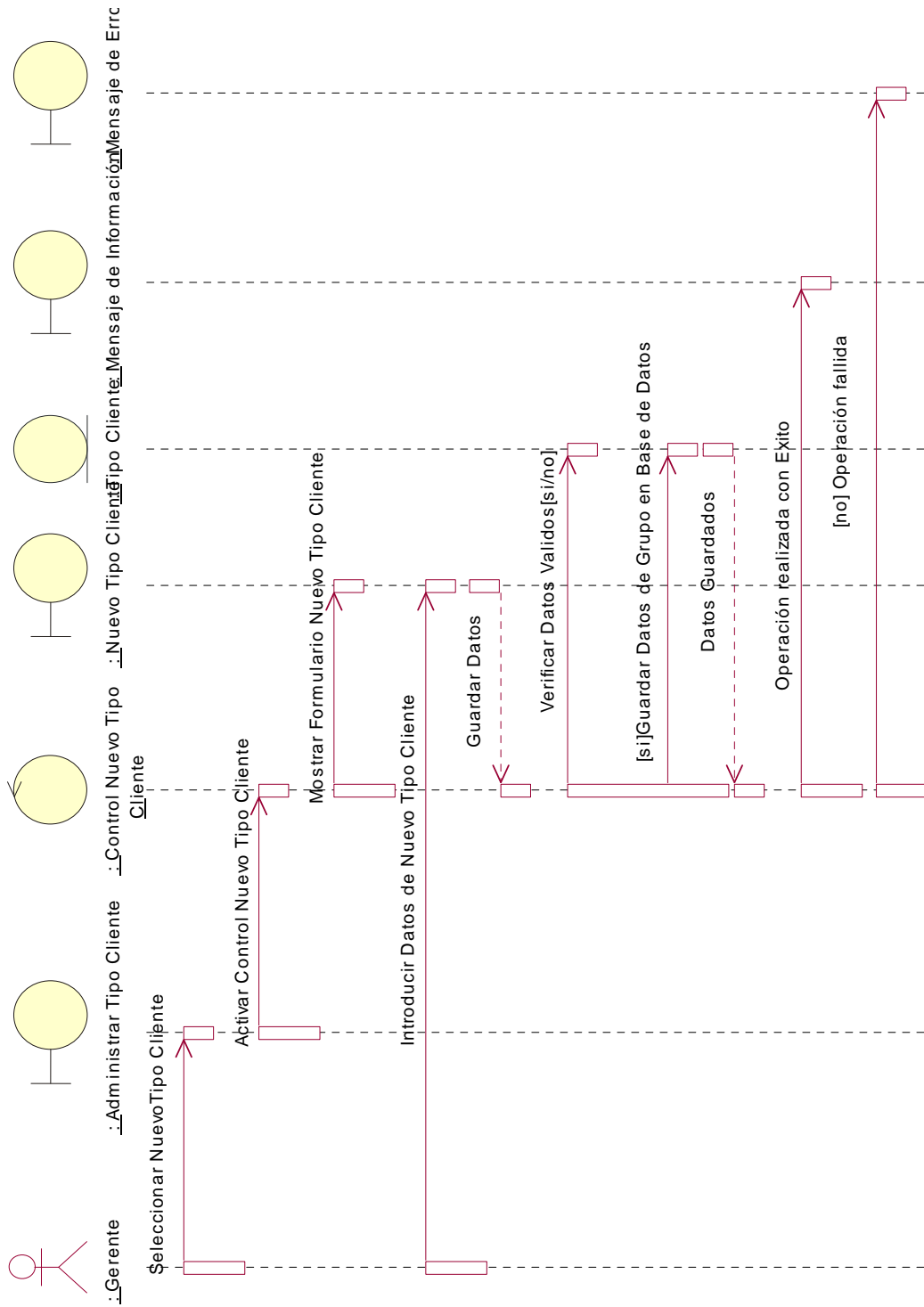
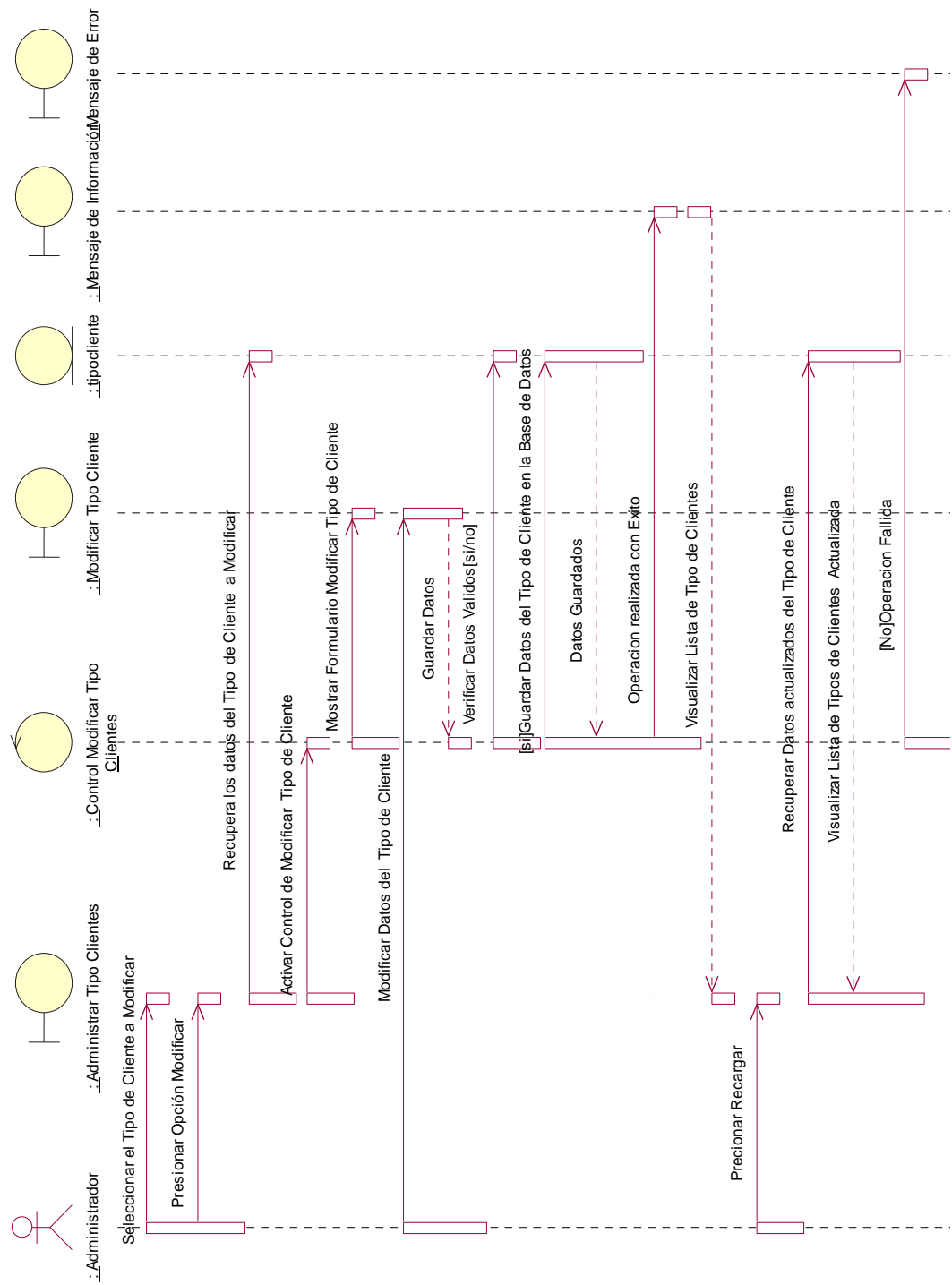


Figura 166: Diagrama de Secuencia: Nuevo Tipo Cliente

**Diagrama de Secuencia: Modificar Tipo Cliente**



**Figura 167: Diagrama de Secuencia: Modificar Tipo Cliente**

### Diagrama de Secuencia: Borrar Tipo Cliente

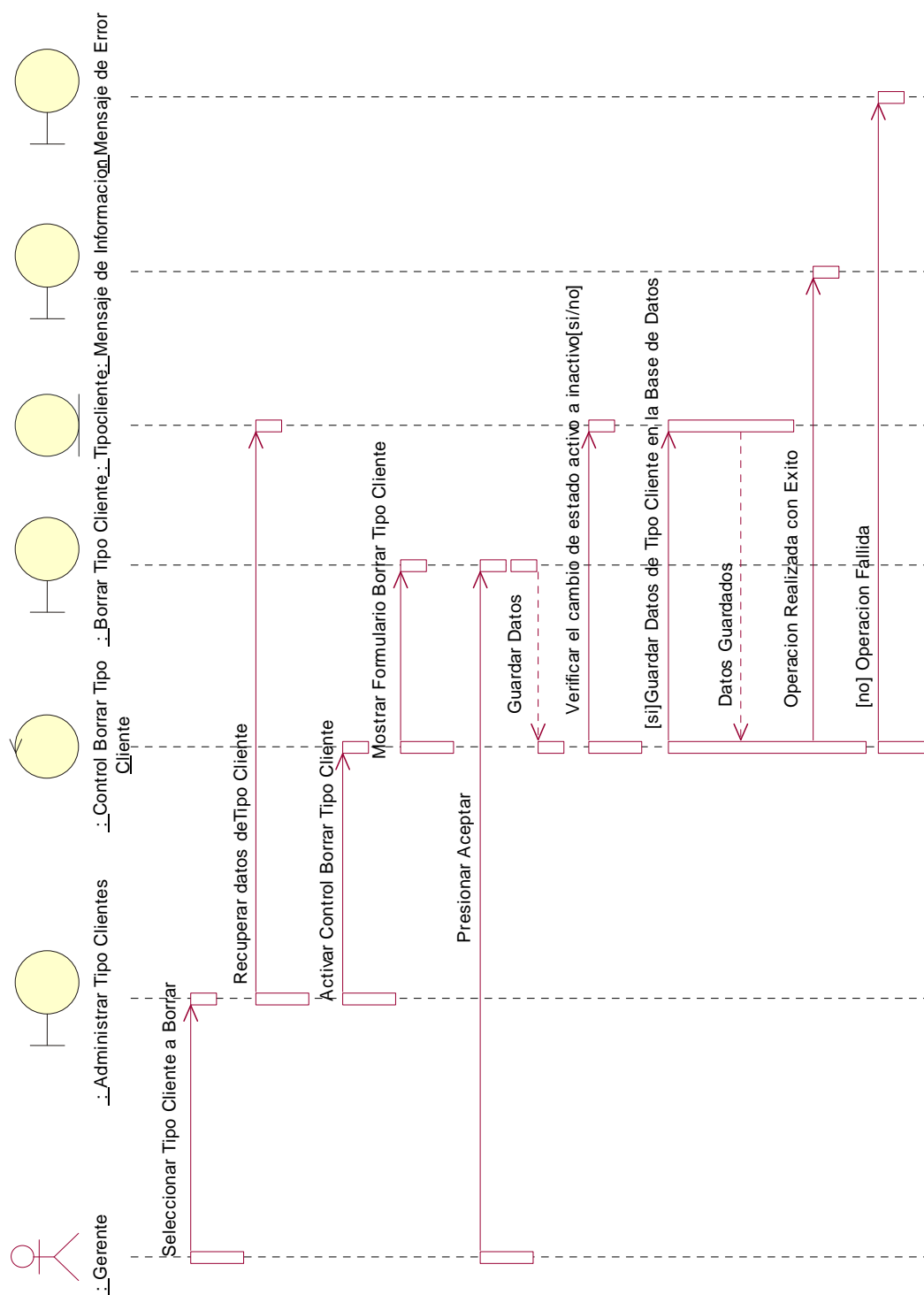
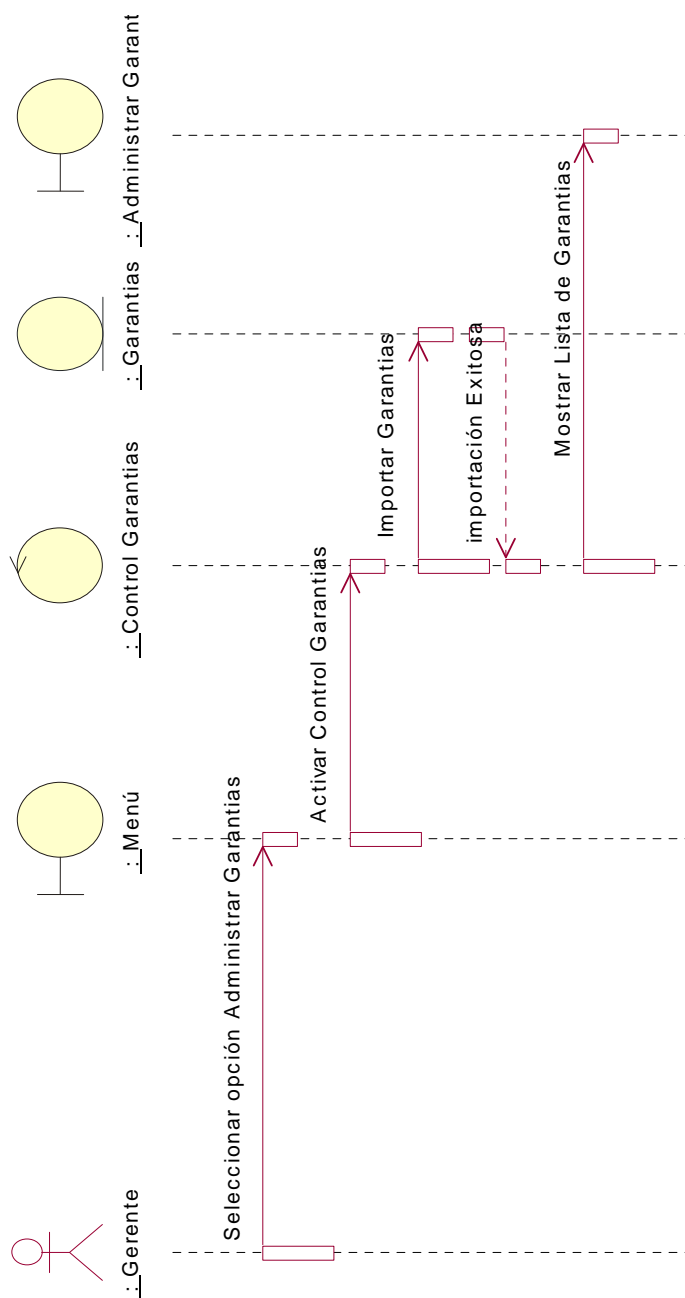


Figura 168: Diagrama de Secuencia: Borrar Tipo Cliente

*Diagrama de Secuencia: Administrar Garantías*



**Figura 169: Diagrama de Secuencia: Administrar Garantías**

### Diagrama de Secuencia: Nueva Garantía

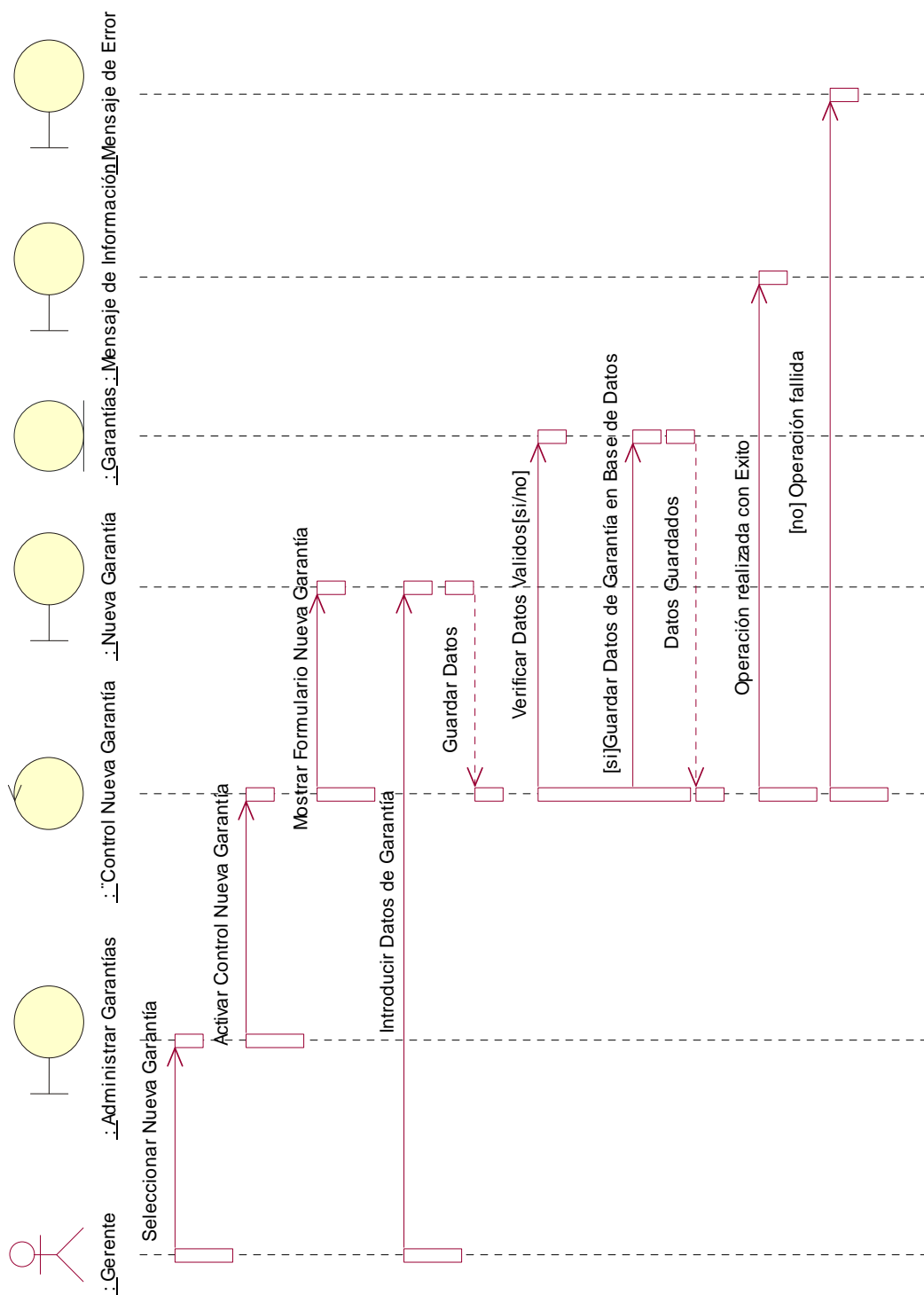


Figura 170: Diagrama de Secuencia: Nueva Garantía

### Diagrama de Secuencia: Modificar Garantía

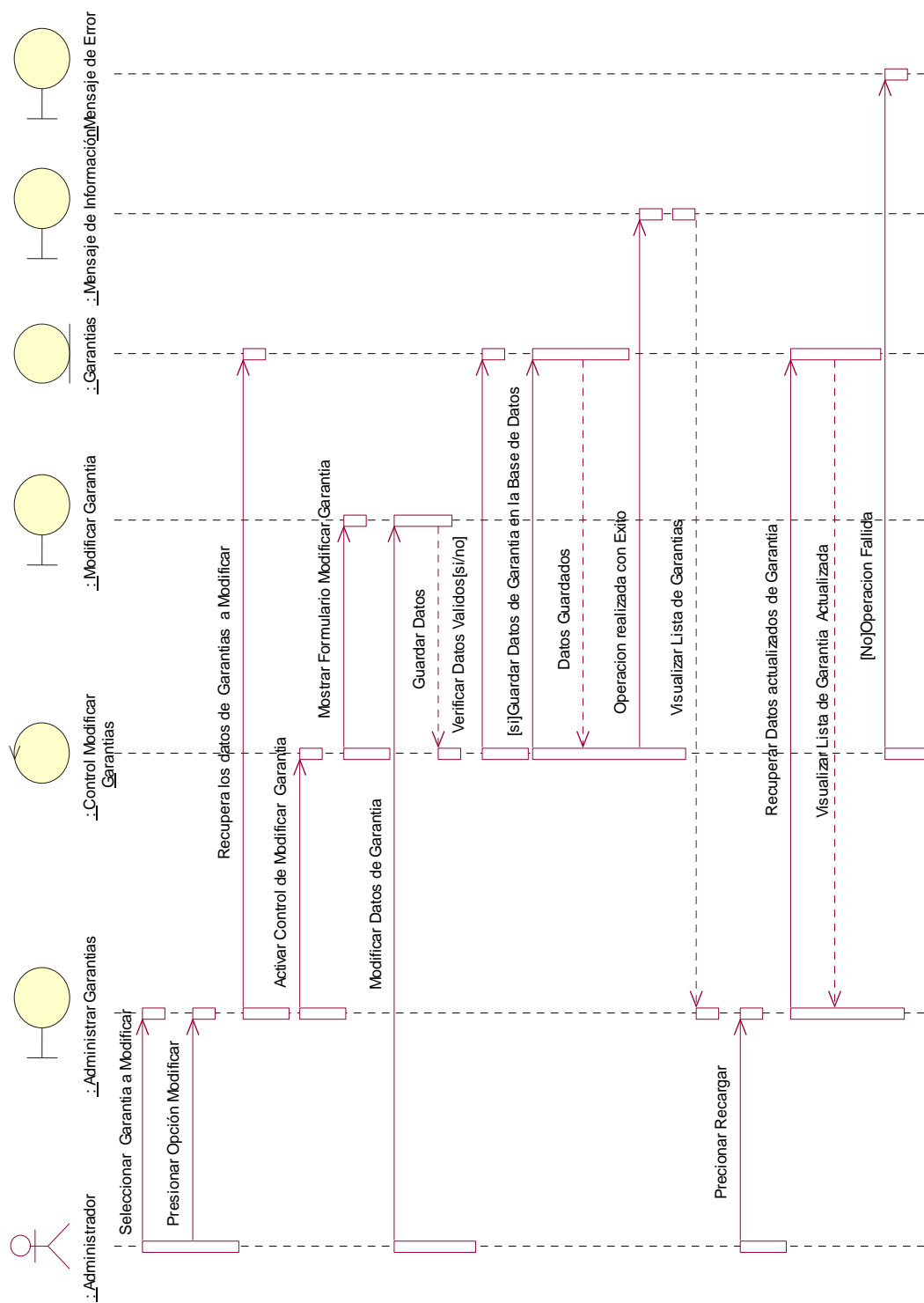


Figura 171: Diagrama de Secuencia: Modificar Garantía

### Diagrama de Secuencia: Borrar Garantía

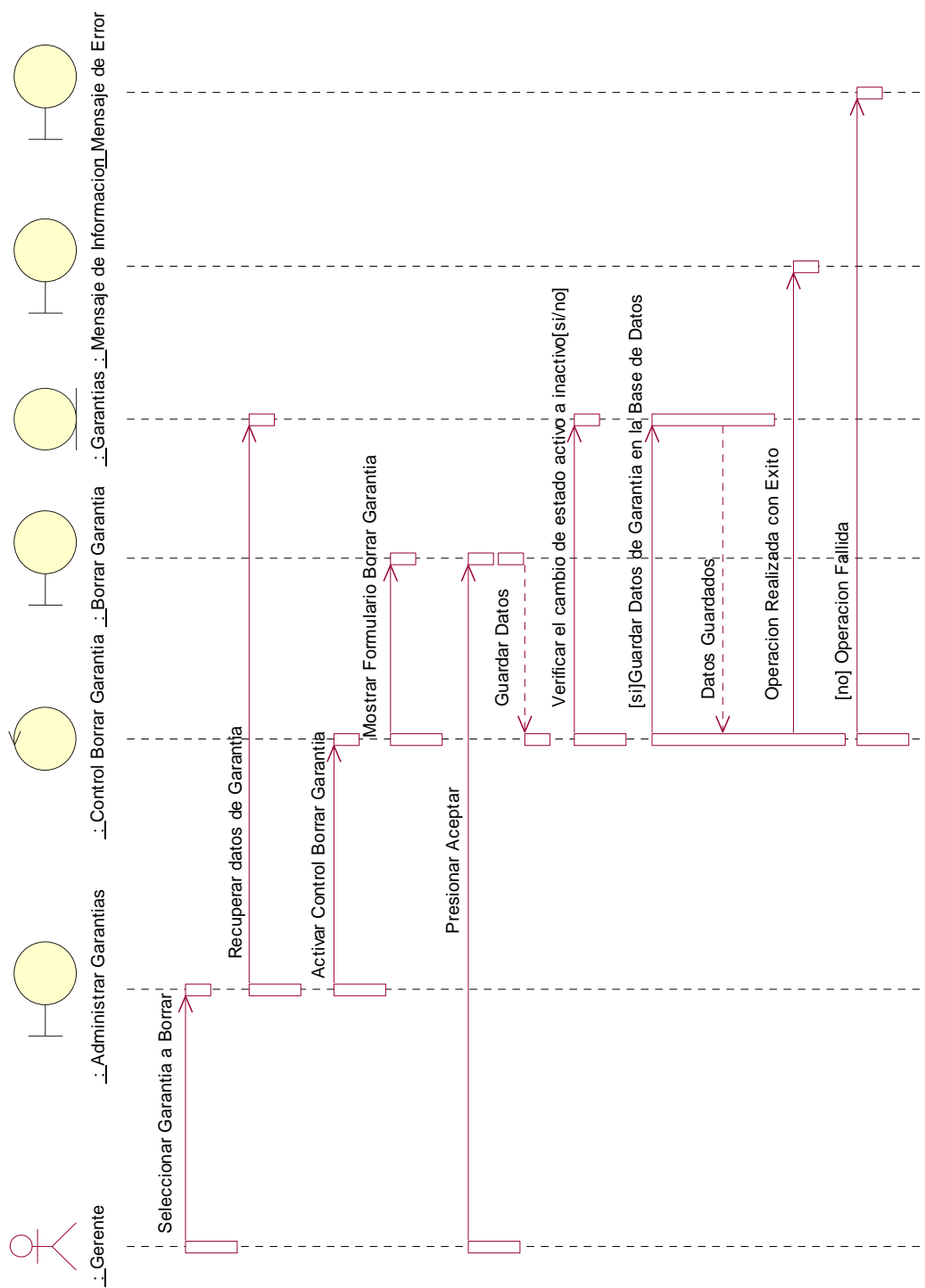
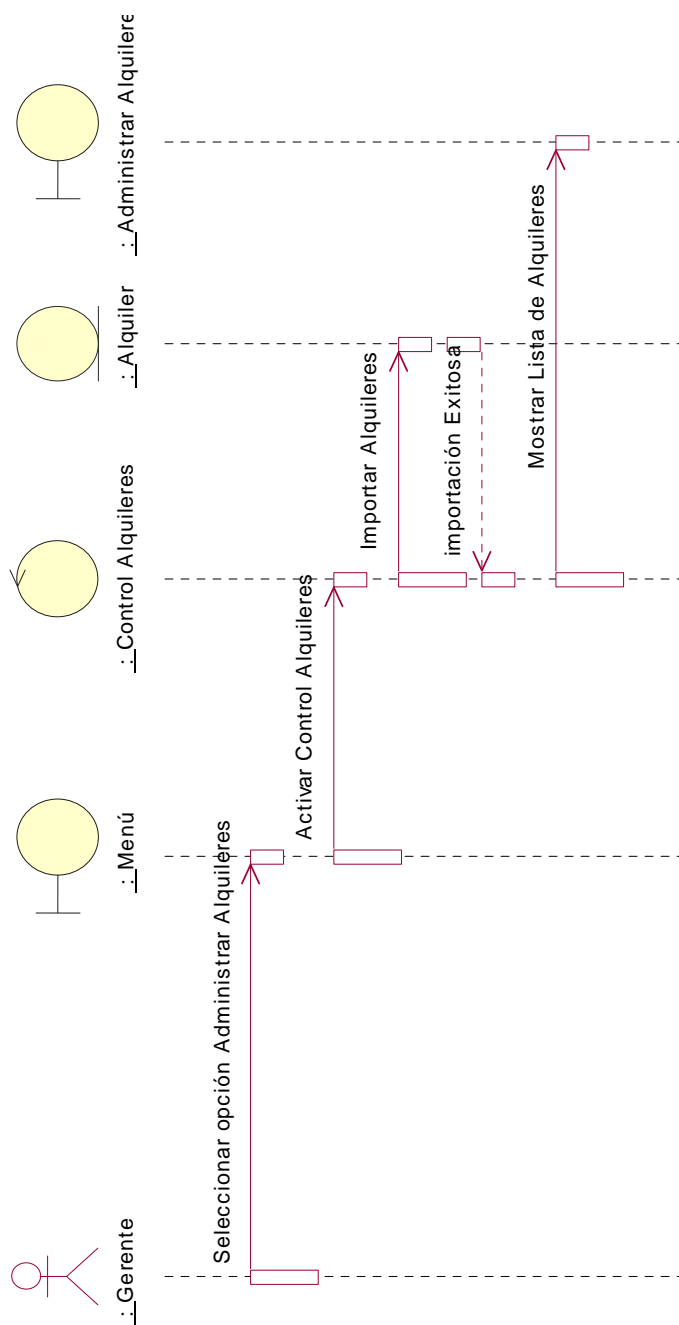


Figura 172: Diagrama de Secuencia: Borrar Garantía

*Diagrama de Secuencia: Administrar Alquileres*



**Figura 173: Diagrama de Secuencia: Administrar Alquileres**



### Diagrama de Secuencia: Nuevo Alquiler

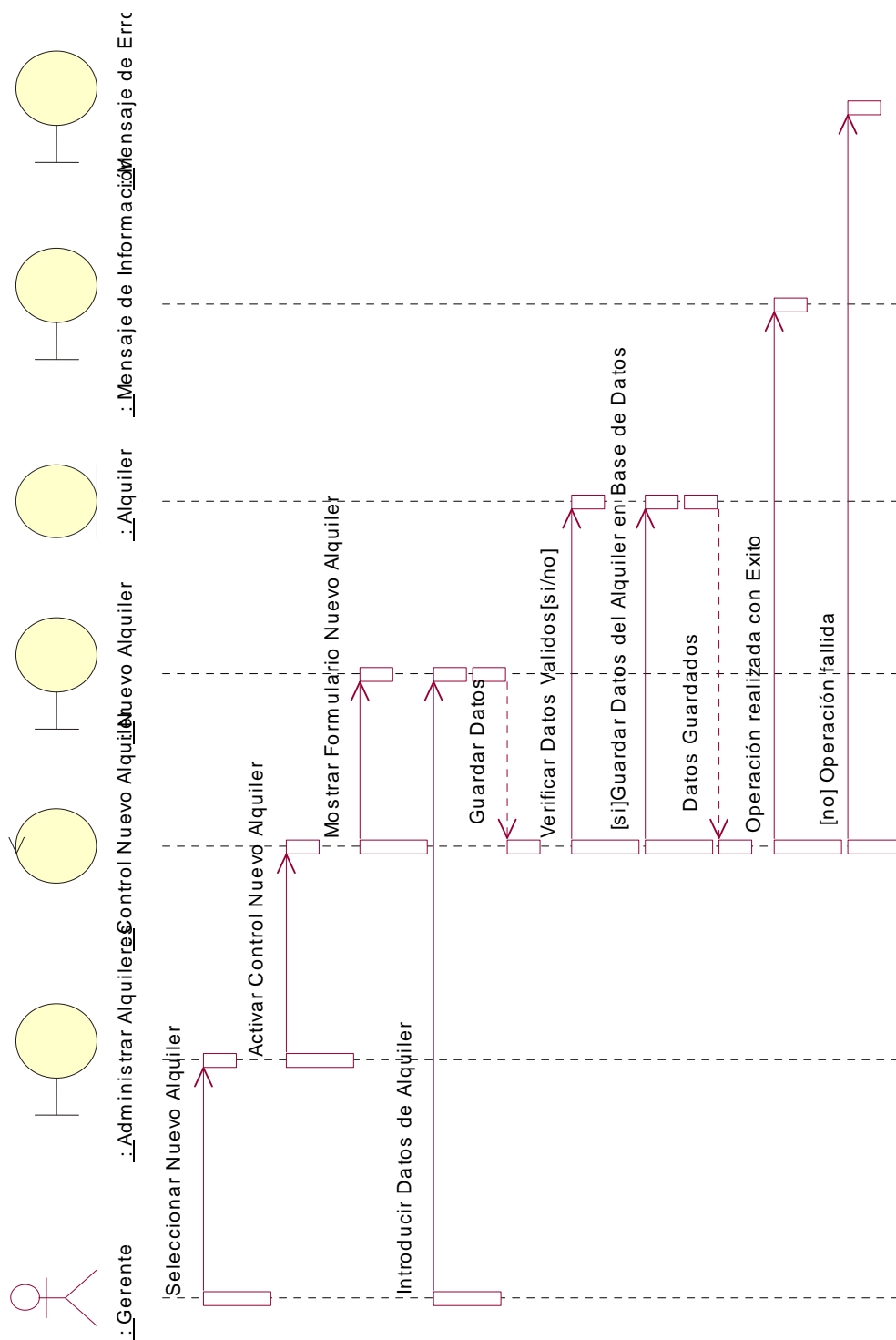
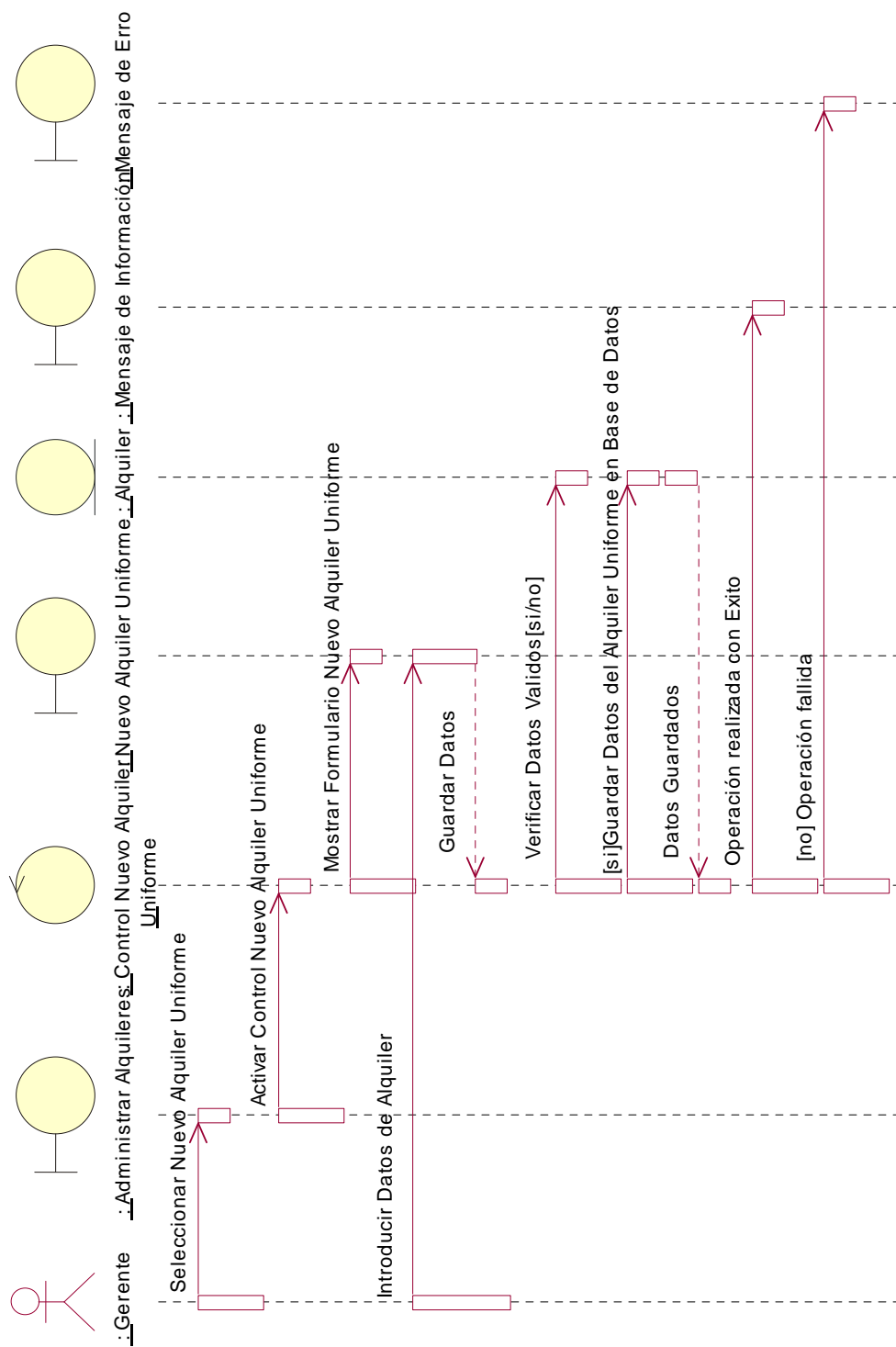


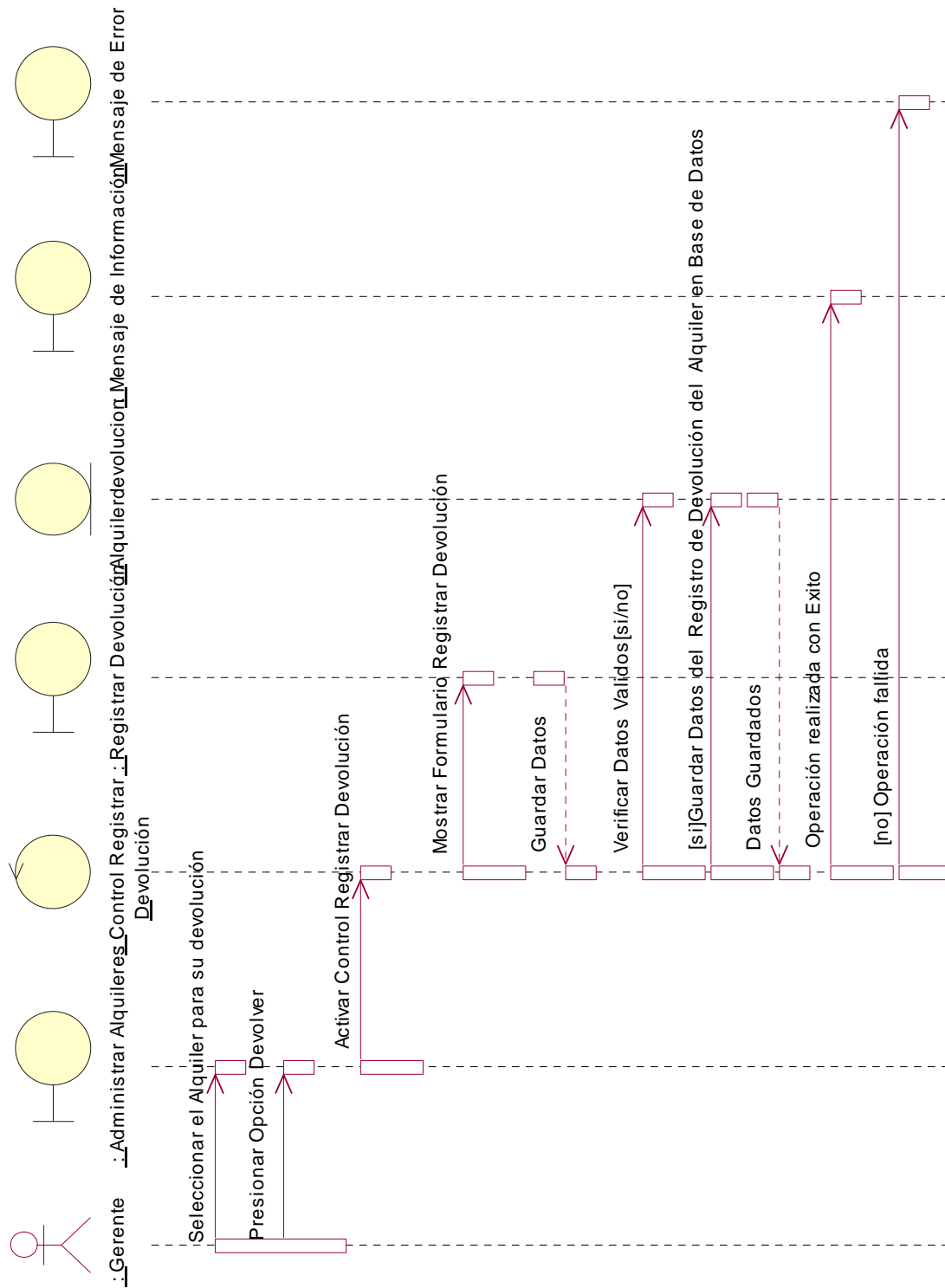
Figura 174: Diagrama de Secuencia: Nuevo Alquiler

*Diagrama de Secuencia: Nuevo Alquiler Uniforme*



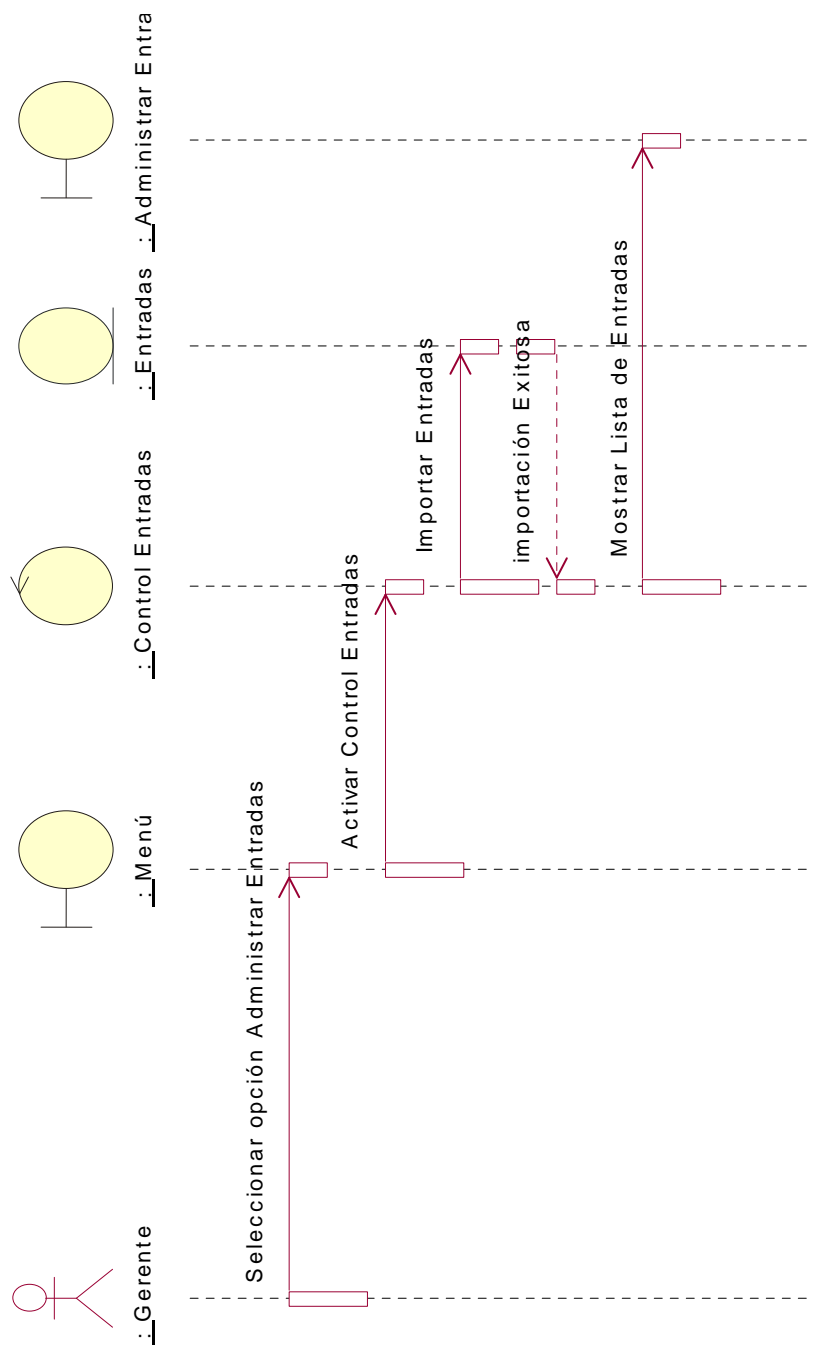
**Figura 175: Diagrama de Secuencia: Nuevo Alquiler Uniforme**

*Diagrama de Secuencia: Registrar Devolución*



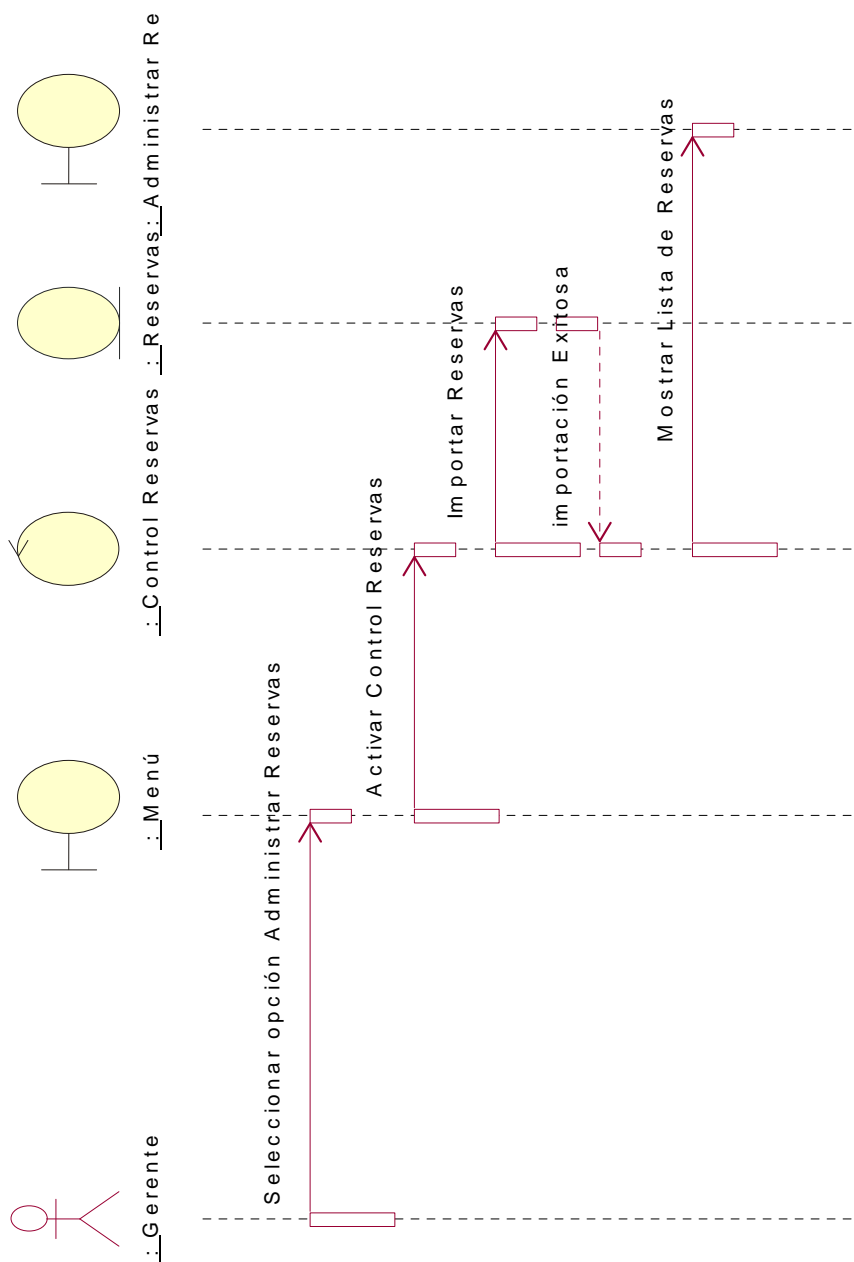
**Figura 176: Diagrama de Secuencia: Registrar Devolución**

*Diagrama de Secuencia: Registrar Entradas de Prenda a Tienda*



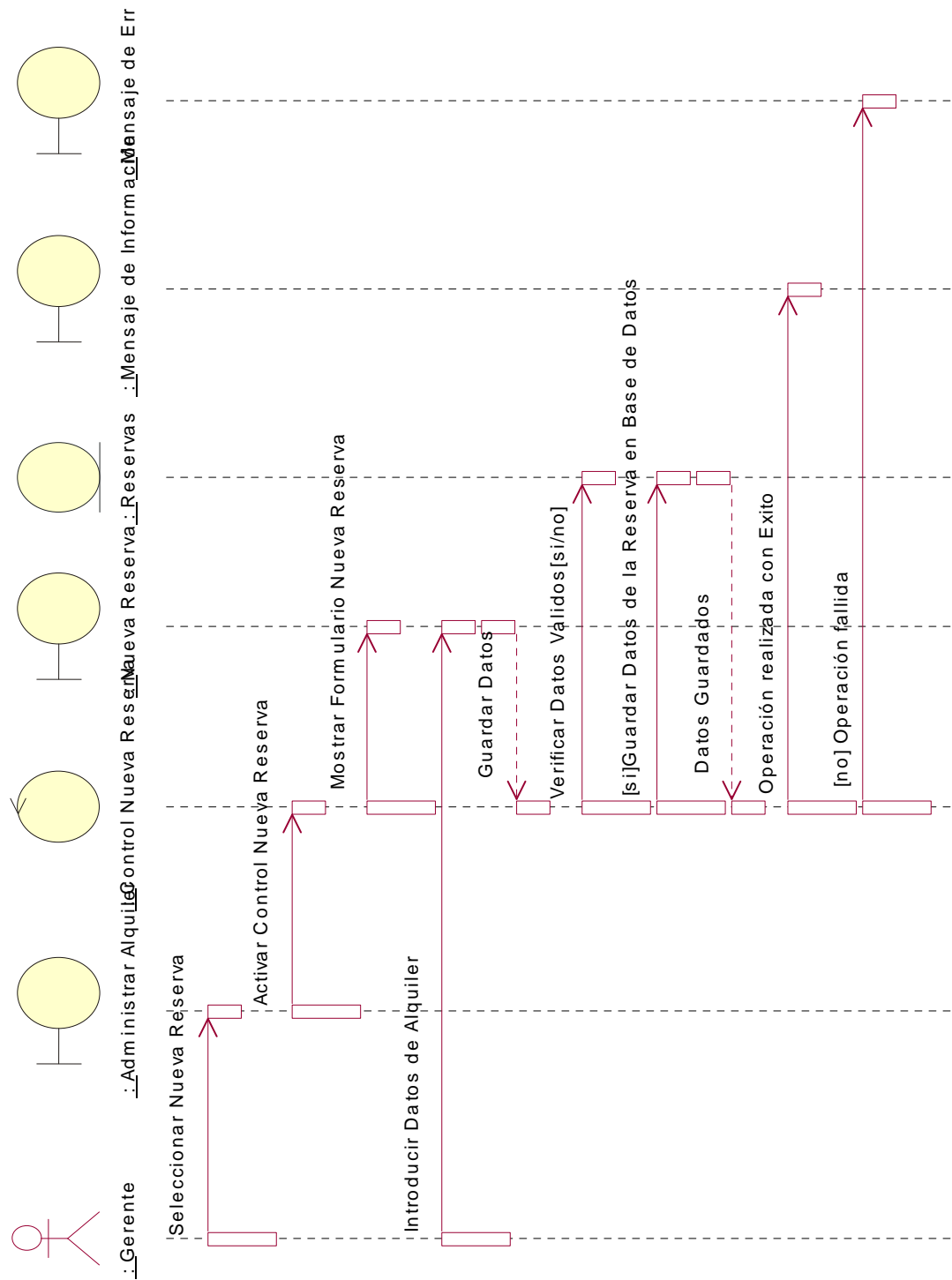
**Figura 177: Diagrama de Secuencia: Registrar Entradas de Prenda a Tienda**

*Diagrama de Secuencia: Administrar Reservas*



**Figura 178: Diagrama de Secuencia: Administrar Reservas**

**Diagrama de Secuencia: Nueva Reserva**



**Figura 179: Diagrama de Secuencia: Nueva Reserva**

## II.1.13 Modelado de Diagrama de Clases

### II.1.13.1 Introducción

El modelado de diagrama de clases es un artefacto de la disciplina análisis diseño en la metodología RUP la cual estamos implementando.

### II.1.13.2 Propósito

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

### II.1.13.3 Alcance

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





## **II.1.14 Modelado de Datos**

### **II.1.14.1 Introducción**

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

### **II.1.14.2 Propósito**

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

### **II.1.14.3 Alcance**

- Describir los campos de cada tabla de la base de datos especificando el tipo, longitud y descripción de cada campo.
- Identificar y definir las relaciones entre las diferentes tablas de la base de datos de nuestro sistema deseado y aprobado por la organización.

### II.1.14.4 Descripción de los campos para la base de Datos

Entity: alquiler	
Primary key constraint name	alquiler_pkey
Comment	
Table options	

#### Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codalq	Yes	INTEGER	Yes	
ci	No	INTEGER	Yes	
codcli	No	INTEGER	No	
codtie	No	INTEGER	Yes	
estado	No	INTEGER	No	
fechaalq	No	DATE	No	
fechadev	No	DATE	No	

#### Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquilerdevolucion_codalq_fkey	Non Identifying	alquiler	alquilerdevolucion	Zero Or More
alquilergarante_codalq_fkey	Identifying	alquiler	alquilergarante	Zero Or More
alquiler_ci_fkey	Non Identifying	usuarios	alquiler	Zero Or More
alquiler_codcli_fkey	Non Identifying	clientes	alquiler	Zero Or More
alquiler_codtie_fkey	Non Identifying	tiendas	alquiler	Zero Or More
detallealquiler_codalq_fkey	Non Identifying	alquiler	detallealquiler	Zero Or More

#### Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codtie)
	Not Null	Column Constraint	NOT NULL (codalq)
	Default	Column Constraint	codalq DEFAULT nextval('alquiler_codalq_seq')
	Not Null	Column Constraint	NOT NULL (ci)
alquiler_ci_fkey	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)
alquiler_codcli_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codcli) REFERENCES clientes(codcli)
alquiler_codtie_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codtie) REFERENCES tiendas(codtie)
alquiler_pkey	Primary Key	Table Constraint	PRIMARY KEY (codalq)

Entity: alquilerdevolucion	
Primary key constraint name	alquilerdevolucion_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
coddevol	Yes	INTEGER	Yes	
codalq	No	INTEGER	Yes	
fecha	No	DATE	Yes	
montoretraso	No	INTEGER	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquilerdevolucion_codalq_fkey	Non Identifying	alquiler	alquilerdevolucion	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (fecha)
	Not Null	Column Constraint	NOT NULL (coddevol)
	Default	Column Constraint	coddevol DEFAULT nextval('alquilerdevolucion_coddevol_seq')
	Not Null	Column Constraint	NOT NULL (codalq)
alquilerdevolucion_codalq_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codalq) REFERENCES alquiler(codalq)
alquilerdevolucion_pkey	Primary Key	Table Constraint	PRIMARY KEY (coddevol)

Entity: alquilergarante	
Primary key constraint name	alquilergarante_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codalq	Yes	INTEGER	Yes	
codgar	Yes	INTEGER	Yes	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquilergarante_codalq_fkey	Identifying	alquiler	alquilergarante	Zero Or More
alquilergarante_codgar_fkey	Identifying	garantias	alquilergarante	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codgar)
	Not Null	Column Constraint	NOT NULL (codalq)
alquilergarante_codalq_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codalq) REFERENCES alquiler(codalq)
alquilergarante_codgar_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codgar) REFERENCES garantias(codgar)
alquilergarante_pkey	Primary Key	Table Constraint	PRIMARY KEY (codalq, codgar)

Entity: clientes	
Primary key constraint name	clientes_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codcli	Yes	INTEGER	Yes	
ci	No	INTEGER	Yes	
codtipocli	No	INTEGER	Yes	
nombreccli	No	CHARACTER VARYING(50)	Yes	Nombre del Cliente
ap	No	CHARACTER VARYING(50)	No	Apellido Paterno del Cliente
am	No	CHARACTER VARYING(50)	No	Apellido Materno del Cliente
estado	No	INTEGER	Yes	Es el estado del Cliente 1 activo y 0 inactivo.
nit	No	INTEGER	Yes	Es el Nit del Cliente
telefono	No	INTEGER	Yes	Teléfono del Cliente
coddir	No	INTEGER	Yes	Código de la Dirección

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquiler_codcli_fkey	Non Identifying	clientes	alquiler	Zero Or More
clientes_codtipocli_fkey	Non Identifying	tipocliente	clientes	Zero Or More
clirol_codcli_fkey	Non Identifying	clientes	clirol	Zero Or More
direccion_coddir_fkey	Non Identifying	direccion	clientes	Zero Or More
reservas_codcli_fkey	Non Identifying	clientes	reservas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (coddir)
	Not Null	Column Constraint	NOT NULL (codcli)
	Not Null	Column Constraint	NOT NULL (ci)

	Not Null	Column Constraint	NOT NULL (codtipocli)
	Not Null	Column Constraint	NOT NULL (nombrecli)
	Not Null	Column Constraint	NOT NULL (estado)
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (nit)
	Not Null	Column Constraint	NOT NULL (telefono)
clientes_codtipocli_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codtipocli) REFERENCES tipocliente(codtipocli)
clientes_pkey	Primary Key	Table Constraint	PRIMARY KEY (codcli)
direccion_coddir_fkey	Foreign Key	Table Constraint	FOREIGN KEY (coddir, nombrecli) REFERENCES direccion(coddir, coddir)

<b>Entity: clirol</b>	
Primary key constraint name	
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codcli	No	INTEGER	Yes	Es el código del cliente
login	No	CHARACTER VARYING(50)	Yes	Login asignado al usuario
clave	No	CHARACTER VARYING(50)	Yes	Clave asignada al usuario
estado	No	INTEGER	Yes	El estado de datos puede ser 1 activo y 0 inactivo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
clirol_codcli_fkey	Non Identifying	clientes	clirol	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (estado)
	Not Null	Column Constraint	NOT NULL (codcli)
	Not Null	Column Constraint	NOT NULL (login)
	Not Null	Column Constraint	NOT NULL (clave)
clirol_codcli_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codcli) REFERENCES clientes(codcli)

Entity: colores	
Primary key constraint name	zcolores_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
color	Yes	CHARACTER VARYING(50)	Yes	Es el nombre del color
estado	No	INTEGER	No	Es el estado del color 1 activo y 0 inactivo.

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
prendas_color_fkey	Non Identifying	colores	prendas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (color)
zcolores_pkey	Primary Key	Table Constraint	PRIMARY KEY (color)

Entity: danzas	
Primary key constraint name	danzas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
coddan	Yes	INTEGER	Yes	Es el código de danza
nombre	No	CHARACTER VARYING(50)	Yes	Es el nombre de la danza
lugar	No	CHARACTER VARYING(50)	No	Es el lugar de danzas
estado	No	INTEGER	No	Es el estado de danza puede ser 1 activo 0 inactivo
zona	No	CHARACTER VARYING	No	Es la zona de que pertenezca la danza

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
danzas_zona_fkey	Non Identifying	zonas	danzas	Zero Or More
vestimentas_coddan_fkey	Non Identifying	danzas	vestimentas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (coddan)

	Default	Column Constraint	coddan DEFAULT nextval('danzas_coddan_seq')
	Not Null	Column Constraint	NOT NULL (nombre)
danzas_pkey	Primary Key	Table Constraint	PRIMARY KEY (coddan)
danzas_zona_fkey	Foreign Key	Table Constraint	FOREIGN KEY (zona) REFERENCES zonas(zona)

<b>Entity: datos</b>	
Primary key constraint name	pk_datos
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
ci	Yes	INTEGER	Yes	Identidad de usuarios
login	No	CHARACTER VARYING(40)	No	Login de usuario
clave	No	CHARACTER VARYING(40)	No	Clave de usuario
estado	No	INTEGER	No	Estado activo 1 y inactivo 0

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
usuarios_datos	Identifying	usuarios	datos	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (ci)
pk_datos	Primary Key	Table Constraint	PRIMARY KEY (ci)
usuarios_datos	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)

<b>Entity: detallealquiler</b>	
Primary key constraint name	detallealquiler_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
coddetallealquiler	Yes	INTEGER	Yes	Código detalle de alquiler
codalq	No	INTEGER	Yes	Código de alquiler
codinv	No	INTEGER	Yes	Código de inventario
precio_unit	No	INTEGER	No	Precio unitario de prendas
estadouso	No	INTEGER	No	Estado de uso

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
detallealquiler_codalq_fkey	Non Identifying	alquiler	detallealquiler	Zero Or More
detallealquiler_codinv_fkey	Non Identifying	inventarioprendas	detallealquiler	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codinv)
	Not Null	Column Constraint	NOT NULL (coddetallealquiler)
	Default	Column Constraint	coddetallealquiler DEFAULT nextval('detallealquiler_coddetallealquiler_seq')
	Not Null	Column Constraint	NOT NULL (codalq)
detallealquiler_codalq_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codalq) REFERENCES alquiler(codalq)
detallealquiler_codinv_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codinv) REFERENCES inventarioprendas(codinv)
detallealquiler_pkey	Primary Key	Table Constraint	PRIMARY KEY (coddetallealquiler)

Entity: detallepedidos	
Primary key constraint name	coddetallepedidos
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codped	No	INTEGER	Yes	Es el código del pedido
codpre	No	INTEGER	Yes	Es el código de prendas
cantidad	No	INTEGER	Yes	Es la cantidad de prendas ingresadas
estado	No	INTEGER	Yes	Es el estado de pedidos 1 activo y 0 inactivo
coddetallepedidos	Yes	INTEGER	Yes	Es el código de detalle pedidos
preciounitario	No	REAL	No	Es el precio unitario de prendas

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
coddetallepedidos	Non Identifying	detallepedidos	inventarioprendas	Zero Or More
detallepedidos_codped_fkey	Non Identifying	pedidos	detallepedidos	Zero Or More
detallepedidos_codpre_fkey	Non Identifying	prendas	detallepedidos	Zero Or More



## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	coddetallepedidos DEFAULT nextval('detallepedidos_coddetallepedidos_seq')
	Not Null	Column Constraint	NOT NULL (codped)
	Not Null	Column Constraint	NOT NULL (codpre)
	Not Null	Column Constraint	NOT NULL (cantidad)
	Not Null	Column Constraint	NOT NULL (estado)
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (coddetallepedidos)
coddetallepedidos	Primary Key	Table Constraint	PRIMARY KEY (coddetallepedidos)
detallepedidos_codped_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codped) REFERENCES pedidos(codped)
detallepedidos_codpre_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codpre) REFERENCES prendas(codpre)

Entity: detallereservas	
Primary key constraint name	detallereservas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codres	Yes	INTEGER	Yes	Código de reservas
codpre	Yes	INTEGER	Yes	Código de prendas
cantidad	No	INTEGER	Yes	Cantidad de prendas
estadosu	No	INTEGER	Yes	Estado de uso
precio_unit	No	INTEGER	No	Precio unitario

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
detallereservas_alq_fkey	Identifying	reservas	detallereservas	Zero Or More
detallereservas_codpre_fkey	Identifying	prendas	detallereservas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (estadosu)
	Not Null	Column Constraint	NOT NULL (codres)
	Not Null	Column Constraint	NOT NULL (codpre)
	Not Null	Column Constraint	NOT NULL (cantidad)

detallereservas_alq_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codres) REFERENCES reservas(codres)
detallereservas_codpre_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codpre) REFERENCES prendas(codpre)
detallereservas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codres, codpre)

<b>Entity: direccion</b>	
Primary key constraint name	direccion_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
coddir	Yes	INTEGER	Yes	Código de dirección
zona	No	CHARACTER VARYING(50)	Yes	Zona o barrio de dirección
calle	No	CHARACTER VARYING(50)	Yes	Calle de dirección
numero	No	INTEGER	Yes	Número de dirección

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
direccion_coddir_fkey	Non Identifying	direccion	clientes	Zero Or More
direccion_proveedores	Non Identifying	direccion	proveedores	Zero Or More
direccion_tiendas	Non Identifying	direccion	tiendas	Zero Or More
direccion_usuarios	Non Identifying	direccion	usuarios	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (numero)
	Not Null	Column Constraint	NOT NULL (coddir)
	Default	Column Constraint	coddir DEFAULT nextval('direccion_coddir_seq')
	Not Null	Column Constraint	NOT NULL (zona)
	Not Null	Column Constraint	NOT NULL (calle)
direccion_pkey	Primary Key	Table Constraint	PRIMARY KEY (coddir)

<b>Entity: entradas</b>	
Primary key constraint name	entradas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codent	Yes	INTEGER	Yes	
codtie	No	INTEGER	Yes	
estado	No	INTEGER	Yes	
fecha	No	DATE	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
entradas_codtie_fkey	Non Identifying	tiendas	entradas	Zero Or More
inventarioentradatienda_codent_fkey	Identifying	entradas	inventarioentradatienda	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (estado)
	Not Null	Column Constraint	NOT NULL (codent)
	Default	Column Constraint	codent DEFAULT nextval('entradas_codent_seq')
	Not Null	Column Constraint	NOT NULL (codtie)
entradas_codtie_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codtie) REFERENCES tiendas(codtie)
entradas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codent)

Entity: garantías	
Primary key constraint name	garantias_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codgar	Yes	INTEGER	Yes	
tipo	No	CHARACTER VARYING(50)	Yes	
valor	No	INTEGER	Yes	
estado	No	INTEGER	No	
codmod	No	INTEGER	Yes	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquilergarante_codgar_fkey	Identifying	garantias	alquilergarante	Zero Or More
monedas_garantias	Non Identifying	monedas	garantias	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column	NOT NULL (codmod)

		Constraint	
	Not Null	Column Constraint	NOT NULL (codgar)
	Default	Column Constraint	codgar DEFAULT nextval('garantias_codgar_seq')
	Not Null	Column Constraint	NOT NULL (tipo)
	Not Null	Column Constraint	NOT NULL (valor)
garantias_pkey	Primary Key	Table Constraint	PRIMARY KEY (codgar)
monedas_garantias	Foreign Key	Table Constraint	FOREIGN KEY (codmod) REFERENCES monedas(codmod)

<b>Entity: grupos</b>	
Primary key constraint name	grupos_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codgru	Yes	INTEGER	Yes	Es el código del grupo
nombregru	No	CHARACTER VARYING(50)	Yes	Es el nombre del grupo
estado	No	INTEGER	Yes	Es el estado del grupo puede ser 1 activo y 2 inactivo
imagen	No	CHARACTER VARYING	Yes	Es el icono que represente el grupo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
grupos_grupos_procesos	Non Identifying	grupos	grupos_procesos	Zero Or More
grupos_rols_grupos	Identifying	grupos	rols_grupos	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	imagen DEFAULT '50'
	Not Null	Column Constraint	NOT NULL (codgru)
	Default	Column Constraint	codgru DEFAULT nextval('grupos_codgru_seq')
	Not Null	Column Constraint	NOT NULL (nombregru)
	Not Null	Column Constraint	NOT NULL (estado)
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (imagen)
grupos_pkey	Primary Key	Table Constraint	PRIMARY KEY (codgru)

Entity: grupos_procesos	
Primary key constraint name	
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codgru	No	INTEGER	Yes	Es el código del grupo
codproc	No	INTEGER	Yes	Es el código de proceso
estado	No	INTEGER	No	Es el estado de asignar procesos a grupos puede ser 1 activo o 2 inactivo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
grupos_grupos_procesos	Non Identifying	grupos	grupos_procesos	Zero Or More
procesos_grupos_procesos	Non Identifying	procesos	grupos_procesos	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codproc)
	Not Null	Column Constraint	NOT NULL (codgru)
grupos_grupos_procesos	Foreign Key	Table Constraint	FOREIGN KEY (codgru) REFERENCES grupos(codgru)
procesos_grupos_procesos	Foreign Key	Table Constraint	FOREIGN KEY (codproc) REFERENCES procesos(codproc)

Entity: imagenes	
Primary key constraint name	imagenes_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codimg	Yes	INTEGER	Yes	Código de imagen
nombre	No	CHARACTER VARYING(50)	Yes	Nombre de la imagen
descripcion	No	CHARACTER VARYING(50)	Yes	Descripción de la imagen
tipo	No	CHARACTER VARYING(50)	No	Tipo de la imagen
url	No	CHARACTER VARYING(50)	No	El url de la imagen
estado	No	INTEGER	No	Estado de la imagen
ci	No	INTEGER	No	Identidad del usuario

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
usuarios_imagenes	Non Identifying	usuarios	imagenes	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (descripcion)
	Not Null	Column Constraint	NOT NULL (coding)
	Default	Column Constraint	coding DEFAULT nextval('imagenes_coding_seq')
	Not Null	Column Constraint	NOT NULL (nombre)
imagenes_pkey	Primary Key	Table Constraint	PRIMARY KEY (coding)
usuarios_imagenes	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)

<b>Entity: inventarioentradienda</b>	
Primary key constraint name	inventarioentradienda_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codinv	Yes	INTEGER	Yes	
codent	Yes	INTEGER	Yes	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
inventarioentradienda_codent_fkey	Identifying	entradas	inventarioentradienda	Zero Or More
inventarioentradienda_codinv_fkey	Identifying	inventarioprendas	inventarioentradienda	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codent)
	Not Null	Column Constraint	NOT NULL (codinv)
inventarioentradienda_codent_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codent) REFERENCES entradas(codent)
inventarioentradienda_codinv_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codinv) REFERENCES inventarioprendas(codinv)
inventarioentradienda_pkey	Primary Key	Table Constraint	PRIMARY KEY (codinv, codent)

<b>Entity: inventarioprendas</b>	
Primary key constraint name	inventarioprendas_pkey
Comment	
Table options	

**Attributes:**

Column name	Primary key	Data type	Not NULL	Comment
codinv	Yes	INTEGER	Yes	Código de inventario
estado	No	INTEGER	No	Estado del inventario
coddetallepedidos	No	INTEGER	No	Código de detalle pedidos
estadouso	No	INTEGER	No	Estado de uso

**Relationships:**

Name	Relationship type	Parent	Child	Cardinality
coddetallepedidos	Non Identifying	detallepedidos	inventarioprendas	Zero Or More
detallealquiler_codinv_fkey	Non Identifying	inventarioprendas	detallealquiler	Zero Or More
inventarioentradatienda_codinv_fkey	Identifying	inventarioprendas	inventarioentradatienda	Zero Or More

**Constraints:**

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 0
	Not Null	Column Constraint	NOT NULL (codinv)
	Default	Column Constraint	codinv DEFAULT nextval('inventarioprendas_codinv_seq')
coddetallepedidos	Foreign Key	Table Constraint	FOREIGN KEY (coddetallepedidos) REFERENCES detallepedidos(coddetallepedidos)
inventarioprendas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codinv)

<b>Entity: monedas</b>	
Primary key constraint name	monedas_pkey
Comment	
Table options	

**Attributes:**

Column name	Primary key	Data type	Not NULL	Comment
codmod	Yes	INTEGER	Yes	
nombre	No	CHARACTER VARYING(50)	Yes	
estado	No	INTEGER	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
monedas_garantias	Non Identifying	monedas	garantias	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (codmod)
	Default	Column Constraint	codmod DEFAULT nextval('monedas_codmod_seq')
	Not Null	Column Constraint	NOT NULL (nombre)
monedas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codmod)

Entity: noticias	
Primary key constraint name	noticias_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codnot	Yes	INTEGER	Yes	
ci	No	INTEGER	Yes	
autor	No	CHARACTER VARYING(50)	Yes	
tipo	No	CHARACTER VARYING(50)	Yes	
contenido	No	CHARACTER VARYING(50)	Yes	
fechaini	No	DATE	No	
fechafin	No	DATE	No	
estado	No	INTEGER	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
noticias_ci_fkey	Non Identifying	usuarios	noticias	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (contenido)
	Not Null	Column Constraint	NOT NULL (codnot)
	Default	Column Constraint	codnot DEFAULT nextval('noticias_codnot_seq')
	Not Null	Column Constraint	NOT NULL (ci)
	Not Null	Column Constraint	NOT NULL (autor)



	Not Null	Column Constraint	NOT NULL (tipo)
noticias_ci_fkey	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)
noticias_pkey	Primary Key	Table Constraint	PRIMARY KEY (codnot)

<b>Entity: pedidos</b>	
Primary key constraint name	pedidos_pkey
Comment	
Table options	

**Attributes:**

Column name	Primary key	Data type	Not NULL	Comment
codped	Yes	INTEGER	Yes	Es el código del pedido
codpro	No	INTEGER	Yes	Es el código de la prenda que hereda de la tabla prendas
ci	No	INTEGER	Yes	Es el ci del usuario
fechaped	No	DATE	No	Representa la fecha en la que realizó el pedido
estado	No	INTEGER	No	Es el estado de Usuarios

**Relationships:**

Name	Relationship type	Parent	Child	Cardinality
detallepedidos_codped_fkey	Non Identifying	pedidos	detallepedidos	Zero Or More
pedidos_ci_fkey	Non Identifying	usuarios	Pedidos	Zero Or More
pedidos_codpro_fkey	Non Identifying	proveedores	Pedidos	Zero Or More

**Constraints:**

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (ci)
	Not Null	Column Constraint	NOT NULL (codped)
	Default	Column Constraint	codped DEFAULT nextval('pedidos_codped_seq')
	Not Null	Column Constraint	NOT NULL (codpro)
pedidos_ci_fkey	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)
pedidos_codpro_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codpro) REFERENCES proveedores(codpro)
pedidos_pkey	Primary Key	Table Constraint	PRIMARY KEY (codped)

<b>Entity: precios</b>	
Primary key constraint name	precios_pkey
Comment	
Table options	

Attributes:Column name	Primary key	Data type	Not NULL	Comment
codprecio	Yes	INTEGER	Yes	Es el código del precio
tipotarifa	No	INTEGER	No	Es el tipo de tarifa
monto	No	INTEGER	Yes	Es el monto del precio
estado	No	INTEGER	No	Es el estado del precio 1 activo y 0 inactivo.
fecha	No	DATE	No	Es la fecha en que registra el precio
codpre	No	INTEGER	No	Es el código de la prenda

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
prendas_codpre_fkey	Non Identifying	prendas	precios	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (codprecio)
	Default	Column Constraint	codprecio DEFAULT nextval('precios_codprecio_seq')
	Not Null	Column Constraint	NOT NULL (monto)
precios_pkey	Primary Key	Table Constraint	PRIMARY KEY (codprecio)
prendas_codpre_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codpre) REFERENCES prendas(codpre)

Entity: prendas	
Primary key constraint name	prendas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codpre	Yes	INTEGER	Yes	Código de prendas
color	No	CHARACTER VARYING(50)	Yes	Clave de colores
talla	No	CHARACTER VARYING(50)	Yes	Clave de tallas
estado	No	INTEGER	Yes	Estado de la tabla
codves	No	INTEGER	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
detallepedidos_codpre_fkey	Non Identifying	prendas	detallepedidos	Zero Or More
detallereservas_codpre_fkey	Identifying	prendas	detallereservas	Zero Or More
prendas_codpre_fkey	Non Identifying	prendas	precios	Zero Or More
prendas_codves_fkey	Non Identifying	vestimentas	prendas	Zero Or More

prendas_color_fkey	Non Identifying	colores	Prendas	Zero Or More
tallas_talla	Non Identifying	tallas	Prendas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (estado)
	Not Null	Column Constraint	NOT NULL (codpre)
	Default	Column Constraint	codpre DEFAULT nextval('prendas_codpre_seq')
	Not Null	Column Constraint	NOT NULL (color)
	Not Null	Column Constraint	NOT NULL (talla)
prendas_codves_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codves) REFERENCES vestimentas(codves)
prendas_color_fkey	Foreign Key	Table Constraint	FOREIGN KEY (color) REFERENCES colores(color)
prendas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codpre)
tallas_talla	Foreign Key	Table Constraint	FOREIGN KEY (talla) REFERENCES tallas(talla)

Entity: procesos	
Primary key constraint name	procesos_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codproc	Yes	INTEGER	Yes	Es el código del proceso
nombreproc	No	CHARACTER VARYING(50)	Yes	Es el nombre del proceso
estado	No	INTEGER	Yes	Es el estado del proceso 1 activo y 0 inactivo
nombreproc_url	No	CHARACTER VARYING(50)	No	Es url del proceso

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
procesos_grupos_procesos	Non Identifying	procesos	grupos_procesos	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (codproc)
	Default	Column Constraint	codproc DEFAULT nextval('procesos_codproc_seq')

	Not Null	Column Constraint	NOT NULL (nombreproc)
	Not Null	Column Constraint	NOT NULL (estado)
procesos_pkey	Primary Key	Table Constraint	PRIMARY KEY (codproc)

<b>Entity: proveedores</b>	
Primary key constraint name	proveedores_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codpro	Yes	INTEGER	Yes	Es el código del proveedor
ci	No	INTEGER	Yes	Es el ci del proveedor
nombre	No	CHARACTER VARYING(50)	Yes	Es el nombre del proveedor
ap	No	CHARACTER VARYING(50)	Yes	Es el apellido paterno del proveedor
am	No	CHARACTER VARYING(50)	Yes	Es el apellido materno del proveedor
direccion	No	CHARACTER VARYING(50)	No	Es el código de dirección q hereda de la tabla dirección
telefono	No	INTEGER	Yes	Teléfono del proveedor
estado	No	INTEGER	Yes	Es el estado de proveedor, puede ser activo 1 o inactivo 0
coddir	No	INTEGER	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
direccion_proveedores	Non Identifying	direccion	proveedores	Zero Or More
pedidos_codpro_fkey	Non Identifying	proveedores	pedidos	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (estado)
	Not Null	Column Constraint	NOT NULL (codpro)
	Default	Column Constraint	codpro DEFAULT nextval('proveedores_codpro_seq')
	Not Null	Column Constraint	NOT NULL (ci)
	Not Null	Column Constraint	NOT NULL (nombre)
	Not Null	Column Constraint	NOT NULL (ap)
	Not Null	Column Constraint	NOT NULL (am)
	Not Null	Column Constraint	NOT NULL (telefono)

direccion_proveedores	Foreign Key	Table Constraint	FOREIGN KEY (coddir) REFERENCES direccion(coddir)
proveedores_pkey	Primary Key	Table Constraint	PRIMARY KEY (codpro)

<b>Entity: reservas</b>	
Primary key constraint name	reservas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codres	Yes	INTEGER	Yes	Es el código de reserva
ci	No	INTEGER	Yes	Es el código del Usuario
codcli	No	INTEGER	Yes	Es el código del Cliente
codtie	No	INTEGER	Yes	Es el código de la Tienda
fechares	No	DATE	No	Es la fecha en que se realiza la reserva
fechadev	No	DATE	No	Es la fecha de devolución
estado	No	INTEGER	No	Es el estado de la categoría puede ser 1 activo o 0 inactivo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
detallereservas_alq_fkey	Identifying	reservas	detallereservas	Zero Or More
reservas_ci_fkey	Non Identifying	usuarios	Reservas	Zero Or More
reservas_codcli_fkey	Non Identifying	clientes	Reservas	Zero Or More
reservas_codtie_fkey	Non Identifying	tiendas	Reservas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codtie)
	Not Null	Column Constraint	NOT NULL (codres)
	Default	Column Constraint	codres DEFAULT nextval('reservas_codres_seq')
	Not Null	Column Constraint	NOT NULL (ci)
	Not Null	Column Constraint	NOT NULL (codcli)
reservas_ci_fkey	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)
reservas_codcli_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codcli) REFERENCES clientes(codcli)
reservas_codtie_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codtie) REFERENCES tiendas(codtie)
reservas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codres)

<b>Entity: roles</b>
----------------------

Primary key constraint name	roles_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codrol	Yes	INTEGER	Yes	Es el código del Rol
nombrerol	No	CHARACTER VARYING(50)	Yes	Es el nombre del rol
estado	No	INTEGER	No	Es el estado del rol puede ser 1 actico y 0 inactivo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
roles_rols_grupos	Identifying	roles	roles_grupos	Zero Or More
usurol_codrol_fkey	Identifying	roles	usurol	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (nombrerol)
	Not Null	Column Constraint	NOT NULL (codrol)
	Default	Column Constraint	codrol DEFAULT nextval('roles_codrol_seq')
roles_pkey	Primary Key	Table Constraint	PRIMARY KEY (codrol)

**Entity: roles\_grupos**

Primary key constraint name	pk_roles_grupos
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codrol	Yes	INTEGER	Yes	Es el código de la tabla rol
codgru	Yes	INTEGER	Yes	Es el código de la tabla grupos
estado	No	INTEGER	No	El el estado de roles_grupos

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
grupos_roles_grupos	Identifying	grupos	roles_grupos	Zero Or More
roles_roles_grupos	Identifying	roles	roles_grupos	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (codgru)
	Not Null	Column Constraint	NOT NULL (codrol)

grupos_rols_grupos	Foreign Key	Table Constraint	FOREIGN KEY (codgru) REFERENCES grupos(codgru)
pk_rols_grupos	Primary Key	Table Constraint	PRIMARY KEY (codrol, codgru)
rols_rols_grupos	Foreign Key	Table Constraint	FOREIGN KEY (codrol) REFERENCES rols(codrol)

<b>Entity: sesiones</b>	
Primary key constraint name	sesionhbn_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
id	Yes	INTEGER	Yes	
ci	No	INTEGER	No	
fecha	No	DATE	No	
hora	No	TIMESTAMP	No	
accion	No	CHARACTER VARYING	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
usuario_ci_fkey	Non Identifying	usuarios	Sesiones	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	id DEFAULT nextval('sesionhbn_id_seq')
	Not Null	Column Constraint	NOT NULL (id)
sesionhbn_pkey	Primary Key	Table Constraint	PRIMARY KEY (id)
usuario_ci_fkey	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)

<b>Entity: tallas</b>	
Primary key constraint name	tallas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
talla	Yes	CHARACTER VARYING(50)	Yes	Es el nombre de la talla
categoria	No	CHARACTER VARYING(50)	Yes	Es la categoría de la talla
estado	No	INTEGER	No	Es el estado de la talla 1

				activo y 0 inactivo
--	--	--	--	---------------------

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
tallas_talla	Non Identifying	tallas	prendas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (talla)
	Not Null	Column Constraint	NOT NULL (categoria)
tallas_pkey	Primary Key	Table Constraint	PRIMARY KEY (talla)

Entity: tiendas	
Primary key constraint name	tiendas_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codtie	Yes	INTEGER	Yes	Es el código de tienda
telefono	No	INTEGER	No	Teléfono de tienda
estado	No	INTEGER	No	Es estado de tiendas 1 activo o 0 inactivo
coddir	No	INTEGER	No	Es el código de dirección que hereda de la tabla dirección
nombre	No	CHARACTER VARYING	No	Es el nombre de la Tienda
ci	No	INTEGER	No	Es el ci q hereda de la tabla Usuarios

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquiler_codtie_fkey	Non Identifying	tiendas	alquiler	Zero Or More
direccion_tiemdas	Non Identifying	direccion	tiendas	Zero Or More
entradas_codtie_fkey	Non Identifying	tiendas	entradas	Zero Or More
reservas_codtie_fkey	Non Identifying	tiendas	reservas	Zero Or More
usuarios_ci	Non Identifying	usuarios	tiendas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (codtie)
	Default	Column Constraint	codtie DEFAULT nextval('tiendas_codtie_seq')
direccion_tiemdas	Foreign Key	Table Constraint	FOREIGN KEY (coddir) REFERENCES direccion(coddir)
tiendas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codtie)
usuarios_ci	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES



			usuarios(ci)
--	--	--	--------------

Entity: tipocliente	
Primary key constraint name	tipocliente_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codtipocli	Yes	INTEGER	Yes	
tipo	No	CHARACTER VARYING(50)	Yes	
estado	No	INTEGER	Yes	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
clientes_codtipocli_fkey	Non Identifying	tipocliente	Clients	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Not Null	Column Constraint	NOT NULL (estado)
	Not Null	Column Constraint	NOT NULL (codtipocli)
	Default	Column Constraint	codtipocli DEFAULT nextval('tipocliente_codtipocli_seq')
	Not Null	Column Constraint	NOT NULL (tipo)
tipocliente_pkey	Primary Key	Table Constraint	PRIMARY KEY (codtipocli)

Entity: tipoprenda	
Primary key constraint name	tipoprenda_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codtipopren	Yes	INTEGER	Yes	Es el código del tipo prenda
nombre	No	CHARACTER VARYING(50)	Yes	Es el nombre de la prenda
estado	No	INTEGER	Yes	Es el estado de tipo prenda 1 activo y 0 inactivo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
vestimentas_codtipopren_fkey	Non Identifying	tipoprenda	vestimentas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (codtipopren)
	Default	Column Constraint	codtipopren DEFAULT nextval('tipoprenda_codtipopren_seq')
	Not Null	Column Constraint	NOT NULL (nombre)
	Not Null	Column Constraint	NOT NULL (estado)
tipoprenda_pkey	Primary Key	Table Constraint	PRIMARY KEY (codtipopren)

Entity: usuarios	
Primary key constraint name	usuarios_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
ci	Yes	INTEGER	Yes	Es el identificador o código del usuario del sistema
nombre	No	CHARACTER VARYING(50)	Yes	Es el nombre del Usuario
ap	No	CHARACTER VARYING(50)	No	Es el apellido paterno del usuario
am	No	CHARACTER VARYING(50)	No	Es el apellido materno del usuario
estado	No	INTEGER	Yes	Es el estado del usuario este puede ser 1 activo o inactivo 0
email	No	CHARACTER VARYING(120)	No	Es el correo electrónico del usuario
foto	No	CHARACTER VARYING(120)	No	Es la foto del Usuario
coddir	No	INTEGER	No	Es el código de dirección q hereda de la tabla dirección
telefono	No	CHARACTER VARYING	No	Teléfono del usuario

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
alquiler_ci_fkey	Non Identifying	usuarios	alquiler	Zero Or More
direccion_usuarios	Non Identifying	direccion	usuarios	Zero Or More
noticias_ci_fkey	Non Identifying	usuarios	noticias	Zero Or More
pedidos_ci_fkey	Non Identifying	usuarios	pedidos	Zero Or More
reservas_ci_fkey	Non Identifying	usuarios	reservas	Zero Or More
usuarios_ci	Non Identifying	usuarios	tiendas	Zero Or More
usuarios_datos	Identifying	usuarios	datos	Zero Or More
usuarios_imagenes	Non Identifying	usuarios	imagenes	Zero Or More

usuario_ci_fkey	Non Identifying	usuarios	sesiones	Zero Or More
usurol_ci_fkey	Identifying	usuarios	usurol	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (ci)
	Not Null	Column Constraint	NOT NULL (nombre)
	Not Null	Column Constraint	NOT NULL (estado)
direccion_usuarios	Foreign Key	Table Constraint	FOREIGN KEY (coddir) REFERENCES direccion(coddir)
usuarios_pkey	Primary Key	Table Constraint	PRIMARY KEY (ci)

Entity: usurol	
Primary key constraint name	usurol_pkey
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codrol	Yes	INTEGER	Yes	Código del Rol
ci	Yes	INTEGER	Yes	Identidad
estado	No	INTEGER	No	

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
usurol_ci_fkey	Identifying	usuarios	usurol	Zero Or More
usurol_codrol_fkey	Identifying	roles	usurol	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (codrol)
	Not Null	Column Constraint	NOT NULL (ci)
usurol_ci_fkey	Foreign Key	Table Constraint	FOREIGN KEY (ci) REFERENCES usuarios(ci)
usurol_codrol_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codrol) REFERENCES roles(codrol)
usurol_pkey	Primary Key	Table Constraint	PRIMARY KEY (codrol, ci)

Entity: vestimentas	
Primary key constraint	vestimentas_pkey

name	
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
codtipopren	No	INTEGER	Yes	Código de tipo prendas
coddan	No	INTEGER	Yes	Código de danza
genero	No	CHARACTER VARYING(50)	Yes	Es el género
estado	No	INTEGER	Yes	Es el estado de vestimentas 1 activo y 0 inactivo
codves	Yes	INTEGER	Yes	Código de vestimentas

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
prendas_codves_fkey	Non Identifying	vestimentas	prendas	Zero Or More
vestimentas_coddan_fkey	Non Identifying	danzas	vestimentas	Zero Or More
vestimentas_codtipopren_fkey	Non Identifying	tipoprenda	vestimentas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	codves DEFAULT nextval('vestimentas_codves_seq')
	Not Null	Column Constraint	NOT NULL (codtipopren)
	Not Null	Column Constraint	NOT NULL (coddan)
	Not Null	Column Constraint	NOT NULL (genero)
	Not Null	Column Constraint	NOT NULL (estado)
	Not Null	Column Constraint	NOT NULL (codves)
vestimentas_coddan_fkey	Foreign Key	Table Constraint	FOREIGN KEY (coddan) REFERENCES danzas(coddan)
vestimentas_codtipopren_fkey	Foreign Key	Table Constraint	FOREIGN KEY (codtipopren) REFERENCES tipoprenda(codtipopren)
vestimentas_pkey	Primary Key	Table Constraint	PRIMARY KEY (codves)

Entity: zonas	
Primary key constraint	zonas_pkey
name	
Comment	
Table options	

## Attributes:

Column name	Primary key	Data type	Not NULL	Comment
zona	Yes	CHARACTER	Yes	Es el nombre de la zona

		VARYING(50)		
estado	No	INTEGER	No	Es el estado de zonas 1 activo y 0 inactivo

## Relationships:

Name	Relationship type	Parent	Child	Cardinality
danzas_zona_fkey	Non Identifying	zonas	danzas	Zero Or More

## Constraints:

Name	Type	Level	Constraint
	Default	Column Constraint	estado DEFAULT 1
	Not Null	Column Constraint	NOT NULL (zona)
zonas_pkey	Primary Key	Table Constraint	PRIMARY KEY (zona)

## Código de la Base de Datos

```

/* -----
----- */
/* Tables
*/
/* -----
----- */

/* -----
----- */
/* Add table "detallereservas"
*/
/* -----
----- */

CREATE TABLE public.detallereservas (
    codres INTEGER NOT NULL,
    codpre INTEGER NOT NULL,
    cantidad INTEGER NOT NULL,
    estadousu INTEGER NOT NULL,
    precio_unit INTEGER,
    CONSTRAINT detallereservas_pkey PRIMARY KEY (codres, codpre)
);

/* -----
----- */
/* Add table "procesos"
*/
/* -----
----- */

CREATE TABLE public.procesos (
    codproc INTEGER DEFAULT nextval('procesos_codproc_seq') NOT
NULL,
    nombreproc CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1 NOT NULL,
    nombreproc_url CHARACTER VARYING(50),
    CONSTRAINT procesos_pkey PRIMARY KEY (codproc)
);

/* -----
----- */
/* Add table "roles_grupos"
*/
/* -----
----- */

CREATE TABLE public.roles_grupos (
    codrol INTEGER NOT NULL,
    codgru INTEGER NOT NULL,
    estado INTEGER,
    CONSTRAINT pk_roles_grupos PRIMARY KEY (codrol, codgru)
);

/* -----
----- */
/* Add table "garantias"
*/

```

```

/* -----
----- */

CREATE TABLE public.garantias (
    codgar INTEGER DEFAULT nextval('garantias_codgar_seq') NOT
NULL,
    tipo CHARACTER VARYING(50) NOT NULL,
    valor INTEGER NOT NULL,
    estado INTEGER,
    codmod INTEGER NOT NULL,
    CONSTRAINT garantias_pkey PRIMARY KEY (codgar)
);

/* -----
----- */
/* Add table "precios"
*/
/* -----
----- */

CREATE TABLE public.precios (
    codprecio INTEGER DEFAULT nextval('precios_codprecio_seq')
NOT NULL,
    tipotarifa INTEGER,
    monto INTEGER NOT NULL,
    estado INTEGER DEFAULT 1,
    fecha DATE,
    codpre INTEGER,
    CONSTRAINT precios_pkey PRIMARY KEY (codprecio)
);

/* -----
----- */
/* Add table "monedas"
*/
/* -----
----- */

CREATE TABLE public.monedas (
    codmod INTEGER DEFAULT nextval('monedas_codmod_seq') NOT
NULL,
    nombre CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1,
    CONSTRAINT monedas_pkey PRIMARY KEY (codmod)
);

/* -----
----- */
/* Add table "grupos"
*/
/* -----
----- */

CREATE TABLE public.grupos (
    codgru INTEGER DEFAULT nextval('grupos_codgru_seq') NOT NULL,
    nombregru CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1 NOT NULL,
    imagen CHARACTER VARYING DEFAULT '50' NOT NULL,
    CONSTRAINT grupos_pkey PRIMARY KEY (codgru)
);

```

```

);

/* ----- */
/* Add table "inventarioentradatienda"
*/
/* ----- */
----- */

CREATE TABLE public.inventarioentradatienda (
    codinv INTEGER NOT NULL,
    codent INTEGER NOT NULL,
    CONSTRAINT inventarioentradatienda_pkey PRIMARY KEY (codinv,
codent)
);

/* ----- */
----- */
/* Add table "tallas"
*/
/* ----- */
----- */

CREATE TABLE public.tallas (
    talla CHARACTER VARYING(50) NOT NULL,
    categoria CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1,
    CONSTRAINT tallas_pkey PRIMARY KEY (talla)
);

/* ----- */
----- */
/* Add table "colores"
*/
/* ----- */
----- */

CREATE TABLE public.colores (
    color CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1,
    CONSTRAINT zcolores_pkey PRIMARY KEY (color)
);

/* ----- */
----- */
/* Add table "alquilergarante"
*/
/* ----- */
----- */

CREATE TABLE public.alquilergarante (
    codalq INTEGER NOT NULL,
    codgar INTEGER NOT NULL,
    valor REAL,
    CONSTRAINT alquilergarante_pkey PRIMARY KEY (codalq, codgar)
);

/* ----- */
----- */

```



```

/* Add table "zonas"
*/
/* -----
----- */

CREATE TABLE public.zonas (
    zona CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1,
    CONSTRAINT zonas_pkey PRIMARY KEY (zona)
);

/* -----
----- */
/* Add table "pedidos"
*/
/* -----
----- */

CREATE TABLE public.pedidos (
    codped INTEGER DEFAULT nextval('pedidos_codped_seq') NOT
NULL,
    codpro INTEGER NOT NULL,
    ci INTEGER NOT NULL,
    fechaped DATE,
    estado INTEGER,
    CONSTRAINT pedidos_pkey PRIMARY KEY (codped)
);

/* -----
----- */
/* Add table "alquiler"
*/
/* -----
----- */

CREATE TABLE public.alquiler (
    codalq INTEGER DEFAULT nextval('alquiler_codalq_seq') NOT
NULL,
    ci INTEGER NOT NULL,
    codcli INTEGER,
    codtie INTEGER NOT NULL,
    estado INTEGER,
    fechaalq DATE,
    fechadev DATE,
    CONSTRAINT alquiler_pkey PRIMARY KEY (codalq)
);

/* -----
----- */
/* Add table "roles"
*/
/* -----
----- */

CREATE TABLE public.roles (
    codrol INTEGER DEFAULT nextval('roles_codrol_seq') NOT NULL,
    nombrerol CHARACTER VARYING(50) NOT NULL,
    estado INTEGER,
    CONSTRAINT roles_pkey PRIMARY KEY (codrol)
);

```

```

);

COMMENT ON COLUMN public.roles.codrol IS 'c';

/* ----- */
/* Add table "imagenes"
*/
/* ----- */
----- */

CREATE TABLE public.imagenes (
    coding INTEGER DEFAULT nextval('imagenes_coding_seq') NOT
NULL,
    nombre CHARACTER VARYING(50) NOT NULL,
    descripcion CHARACTER VARYING(250) NOT NULL,
    tipo CHARACTER VARYING(50),
    url CHARACTER VARYING(150),
    estado INTEGER,
    CONSTRAINT imagenes_pkey PRIMARY KEY (coding)
);

/* ----- */
/* Add table "alquilerdevolucion"
*/
/* ----- */
----- */

CREATE TABLE public.alquilerdevolucion (
coddevol INTEGER DEFAULT
nextval('alquilerdevolucion_coddevol_seq') NOT NULL,
    codalq INTEGER NOT NULL,
    fecha DATE NOT NULL,
    montoretraso INTEGER,
    CONSTRAINT alquilerdevolucion_pkey PRIMARY KEY (coddevol)
);

/* ----- */
/* Add table "tipocliente"
*/
/* ----- */
----- */

CREATE TABLE public.tipocliente (
    codtipocli INTEGER DEFAULT
nextval('tipocliente_codtipocli_seq') NOT NULL,
    tipo CHARACTER VARYING(50) NOT NULL,
    estado INTEGER NOT NULL,
    CONSTRAINT tipocliente_pkey PRIMARY KEY (codtipocli)
);

/* ----- */
/* Add table "datos"
*/
/* ----- */
----- */

```

```

CREATE TABLE public.datos (
    ci INTEGER NOT NULL,
    login CHARACTER VARYING(40),
    clave CHARACTER VARYING(40),
    estado INTEGER,
    CONSTRAINT pk_datos PRIMARY KEY (ci)
);

/* -----
----- */
/* Add table "noticias"
*/
/* -----
----- */

CREATE TABLE public.noticias (
    codnot INTEGER DEFAULT nextval('noticias_codnot_seq') NOT
NULL,
    ci INTEGER NOT NULL,
    tipo CHARACTER VARYING(50) NOT NULL,
    contenido CHARACTER VARYING(400) NOT NULL,
    fechaini DATE,
    estado INTEGER,
    CONSTRAINT noticias_pkey PRIMARY KEY (codnot)
);

/* -----
----- */
/* Add table "detallealquiler"
*/
/* -----
----- */

CREATE TABLE public.detallealquiler (
    coddetallealquiler INTEGER DEFAULT
nextval('detallealquiler_coddetallealquiler_seq') NOT NULL,
    codalq INTEGER NOT NULL,
    codinv INTEGER NOT NULL,
    precio_unit INTEGER,
    estadouso INTEGER,
    CONSTRAINT detallealquiler_pkey PRIMARY KEY
(coddetallealquiler)
);

/* -----
----- */
/* Add table "grupos_procesos"
*/
/* -----
----- */

CREATE TABLE public.grupos_procesos (
    codgru INTEGER NOT NULL,
    codproc INTEGER NOT NULL,
    estado INTEGER
);

```

```

/* -----
----- */
/* Add table "vestimentas"
*/
/* -----
----- */

CREATE TABLE public.vestimentas (
    codtipopren INTEGER NOT NULL,
    coddan INTEGER NOT NULL,
    genero CHARACTER VARYING(50) NOT NULL,
    estado INTEGER NOT NULL,
    codves INTEGER DEFAULT nextval('vestimentas_codves_seq') NOT
NULL,
    CONSTRAINT vestimentas_pkey PRIMARY KEY (codves)
);

/* -----
----- */
/* Add table "usuarios"
*/
/* -----
----- */

CREATE TABLE public.usuarios (
    ci INTEGER NOT NULL,
    nombre CHARACTER VARYING(50) NOT NULL,
    ap CHARACTER VARYING(50),
    am CHARACTER VARYING(50),
    estado INTEGER DEFAULT 1 NOT NULL,
    email CHARACTER VARYING(120),
    foto CHARACTER VARYING(120),
    coddir INTEGER,
    telefono CHARACTER VARYING,
    CONSTRAINT usuarios_pkey PRIMARY KEY (ci)
);

/* -----
----- */
/* Add table "danzas"
*/
/* -----
----- */

CREATE TABLE public.danzas (
    coddan INTEGER DEFAULT nextval('danzas_coddan_seq') NOT NULL,
    nombre CHARACTER VARYING(50) NOT NULL,
    lugar CHARACTER VARYING(50),
    estado INTEGER DEFAULT 1,
    zona CHARACTER VARYING,
    CONSTRAINT danzas_pkey PRIMARY KEY (coddan)
);

/* -----
----- */
/* Add table "prendas"
*/
/* -----
----- */

```

```

CREATE TABLE public.prendas (
    codpre INTEGER DEFAULT nextval('prendas_codpre_seq') NOT
NULL,
    color CHARACTER VARYING(50) NOT NULL,
    talla CHARACTER VARYING(50) NOT NULL,
    estado INTEGER NOT NULL,
    codves INTEGER,
    CONSTRAINT prendas_pkey PRIMARY KEY (codpre)
);

/* -----
----- */
/* Add table "proveedores"
*/
/* -----
----- */

CREATE TABLE public.proveedores (
    codpro INTEGER DEFAULT nextval('proveedores_codpro_seq') NOT
NULL,
    ci INTEGER NOT NULL,
    nombre CHARACTER VARYING(50) NOT NULL,
    ap CHARACTER VARYING(50) NOT NULL,
    am CHARACTER VARYING(50) NOT NULL,
    telefono INTEGER NOT NULL,
    estado INTEGER NOT NULL,
    coddire INTEGER,
    CONSTRAINT proveedores_pkey PRIMARY KEY (codpro)
);

/* -----
----- */
/* Add table "entradas"
*/
/* -----
----- */

CREATE TABLE public.entradas (
    codent INTEGER DEFAULT nextval('entradas_codent_seq') NOT
NULL,
    codtie INTEGER NOT NULL,
    estado INTEGER NOT NULL,
    fecha DATE,
    CONSTRAINT entradas_pkey PRIMARY KEY (codent)
);

/* -----
----- */
/* Add table "detallepedidos"
*/
/* -----
----- */

CREATE TABLE public.detallepedidos (
    codped INTEGER NOT NULL,
    codpre INTEGER NOT NULL,
    cantidad INTEGER NOT NULL,
    estado INTEGER DEFAULT 1 NOT NULL,

```

```

        coddetallepeditos INTEGER DEFAULT
nextval('detallepeditos_coddetallepeditos_seq') NOT NULL,
preciounitario REAL,
        CONSTRAINT coddetallepeditos PRIMARY KEY (coddetallepeditos)
);

/* ----- */
/* Add table "clirol"
*/
/* ----- */
----- */

CREATE TABLE public.clirol (
        codcli INTEGER NOT NULL,
        login CHARACTER VARYING(50) NOT NULL,
        clave CHARACTER VARYING(50) NOT NULL,
        estado INTEGER NOT NULL
);

/* ----- */
/* Add table "inventarioprendas"
*/
/* ----- */
----- */

CREATE TABLE public.inventarioprendas (
        codinv INTEGER DEFAULT nextval('inventarioprendas_codinv_seq')
NOT NULL,
estado INTEGER DEFAULT 0,
        coddetallepeditos INTEGER,
        estadouso INTEGER,
        CONSTRAINT inventarioprendas_pkey PRIMARY KEY (codinv)
);

/* ----- */
/* Add table "direccion"
*/
/* ----- */
----- */

CREATE TABLE public.direccion (
        coddir INTEGER DEFAULT nextval('direccion_coddir_seq') NOT
NULL,
zona CHARACTER VARYING(50) NOT NULL,
        calle CHARACTER VARYING(50) NOT NULL,
        numero INTEGER NOT NULL,
        CONSTRAINT direccion_pkey PRIMARY KEY (coddir)
);

/* ----- */
/* Add table "clientes"
*/
/* ----- */
----- */

```

```

CREATE TABLE public.clientes (
    codcli INTEGER NOT NULL,
    ci INTEGER NOT NULL,
    codtipocli INTEGER NOT NULL,
    nombrecli CHARACTER VARYING(50) NOT NULL,
    ap CHARACTER VARYING(50),
    am CHARACTER VARYING(50),
    estado INTEGER DEFAULT 1 NOT NULL,
    nit INTEGER NOT NULL,
    telefono INTEGER NOT NULL,
    coddir INTEGER NOT NULL,
    CONSTRAINT clientes_pkey PRIMARY KEY (codcli)
);

/* -----
----- */
/* Add table "reservas"
*/
/* -----
----- */

CREATE TABLE public.reservas (
    codres INTEGER DEFAULT nextval('reservas_codres_seq') NOT
NULL,
    ci INTEGER NOT NULL,
    codcli INTEGER NOT NULL,
    codtie INTEGER NOT NULL,
    fechares DATE,
    fechadev DATE,
    estado INTEGER,
    CONSTRAINT reservas_pkey PRIMARY KEY (codres)
);

/* -----
----- */
/* Add table "usurol"
*/
/* -----
----- */

CREATE TABLE public.usurol (
    codrol INTEGER NOT NULL,
    ci INTEGER NOT NULL,
    estado INTEGER DEFAULT 1,
    CONSTRAINT usurol_pkey PRIMARY KEY (codrol, ci)
);

/* -----
----- */
/* Add table "tipoprenda"
*/
/* -----
----- */

CREATE TABLE public.tipoprenda (
    codtipopren INTEGER DEFAULT
nextval('tipoprenda_codtipopren_seq') NOT NULL,
    nombre CHARACTER VARYING(50) NOT NULL,
    estado INTEGER DEFAULT 1 NOT NULL,

```

```

        CONSTRAINT tipoprenda_pkey PRIMARY KEY (codtipopren)
    );

/* ----- */
/* Add table "tiendas"
*/
/* ----- */
/* ----- */

CREATE TABLE public.tiendas (
    codtie INTEGER DEFAULT nextval('tiendas_codtie_seq') NOT
NULL,
    telefono INTEGER,
    estado INTEGER DEFAULT 1,
    coddir INTEGER,
    nombre CHARACTER VARYING,
    ci INTEGER,
    CONSTRAINT tiendas_pkey PRIMARY KEY (codtie)
);

/* ----- */
/* ----- */
/* Foreign key constraints
*/
/* ----- */
/* ----- */

ALTER TABLE public.detallereservas ADD CONSTRAINT
detallereservas_alq_fkey
    FOREIGN KEY (codres) REFERENCES public.reservas (codres);

ALTER TABLE public.detallereservas ADD CONSTRAINT
detallereservas_codpre_fkey
    FOREIGN KEY (codpre) REFERENCES public.prendas (codpre);

ALTER TABLE public.roles_grupos ADD CONSTRAINT grupos_rols_grupos
    FOREIGN KEY (codgru) REFERENCES public.grupos (codgru);

ALTER TABLE public.roles_grupos ADD CONSTRAINT roles_roles_grupos
    FOREIGN KEY (codrol) REFERENCES public.roles (codrol);

ALTER TABLE public.garantias ADD CONSTRAINT monedas_garantias
    FOREIGN KEY (codmod) REFERENCES public.monedas (codmod);

ALTER TABLE public.precios ADD CONSTRAINT prendas_codpre_fkey
    FOREIGN KEY (codpre) REFERENCES public.prendas (codpre);

ALTER TABLE public.inventarioentradatienda ADD CONSTRAINT
inventarioentradatienda_codent_fkey
    FOREIGN KEY (codent) REFERENCES public.entradas (codent);

ALTER TABLE public.inventarioentradatienda ADD CONSTRAINT
inventarioentradatienda_codinv_fkey
    FOREIGN KEY (codinv) REFERENCES public.inventarioprendas
(codinv);

ALTER TABLE public.alquilergarante ADD CONSTRAINT
alquilergarante_codalq_fkey

```



```
FOREIGN KEY (codalq) REFERENCES public.alquiler (codalq);

ALTER TABLE public.alquilergarante ADD CONSTRAINT
alquilergarante_codgar_fkey
FOREIGN KEY (codgar) REFERENCES public.garantias (codgar);

ALTER TABLE public.pedidos ADD CONSTRAINT pedidos_ci_fkey
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);

ALTER TABLE public.pedidos ADD CONSTRAINT pedidos_codpro_fkey
FOREIGN KEY (codpro) REFERENCES public.proveedores (codpro);

ALTER TABLE public.alquiler ADD CONSTRAINT alquiler_ci_fkey
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);

ALTER TABLE public.alquiler ADD CONSTRAINT alquiler_codcli_fkey
FOREIGN KEY (codcli) REFERENCES public.clientes (codcli);

ALTER TABLE public.alquiler ADD CONSTRAINT alquiler_codtie_fkey
FOREIGN KEY (codtie) REFERENCES public.tiendas (codtie);

ALTER TABLE public.alquilerdevolucion ADD CONSTRAINT
alquilerdevolucion_codalq_fkey
FOREIGN KEY (codalq) REFERENCES public.alquiler (codalq);

ALTER TABLE public.datos ADD CONSTRAINT usuarios_datos
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);

ALTER TABLE public.noticias ADD CONSTRAINT noticias_ci_fkey
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);

ALTER TABLE public.detallealquiler ADD CONSTRAINT
detallealquiler_codalq_fkey
FOREIGN KEY (codalq) REFERENCES public.alquiler (codalq);

ALTER TABLE public.detallealquiler ADD CONSTRAINT
detallealquiler_codinv_fkey
FOREIGN KEY (codinv) REFERENCES public.inventarioprendas
(codinv);

ALTER TABLE public.vestimentas ADD CONSTRAINT
vestimentas_coddan_fkey
FOREIGN KEY (coddan) REFERENCES public.danzas (coddan);

ALTER TABLE public.vestimentas ADD CONSTRAINT
vestimentas_codtipopren_fkey
FOREIGN KEY (codtipopren) REFERENCES public.tipoprenda
(codtipopren);

ALTER TABLE public.usuarios ADD CONSTRAINT direccion_usuarios
FOREIGN KEY (coddir) REFERENCES public.direccion (coddir);

ALTER TABLE public.danzas ADD CONSTRAINT danzas_zona_fkey
FOREIGN KEY (zona) REFERENCES public.zonas (zona);

ALTER TABLE public.prendas ADD CONSTRAINT prendas_codves_fkey
FOREIGN KEY (codves) REFERENCES public.vestimentas (codves);

ALTER TABLE public.prendas ADD CONSTRAINT prendas_color_fkey
```

```
FOREIGN KEY (color) REFERENCES public.colores (color);

ALTER TABLE public.prendas ADD CONSTRAINT tallas_talla
FOREIGN KEY (talla) REFERENCES public.tallas (talla);

ALTER TABLE public.proveedores ADD CONSTRAINT
direccion_proveedores
FOREIGN KEY (coddir) REFERENCES public.direccion (coddir);

ALTER TABLE public.entradas ADD CONSTRAINT entradas_codtie_fkey
FOREIGN KEY (codtie) REFERENCES public.tiendas (codtie);

ALTER TABLE public.detallepedidos ADD CONSTRAINT
detallepedidos_codped_fkey
FOREIGN KEY (codped) REFERENCES public.pedidos (codped);

ALTER TABLE public.detallepedidos ADD CONSTRAINT
detallepedidos_codpre_fkey
FOREIGN KEY (codpre) REFERENCES public.prendas (codpre);

ALTER TABLE public.clirol ADD CONSTRAINT clirol_codcli_fkey
FOREIGN KEY (codcli) REFERENCES public.clientes (codcli);

ALTER TABLE public.inventarioprendas ADD CONSTRAINT
coddetallepedidos
FOREIGN KEY (coddetallepedidos) REFERENCES
public.detallepedidos (coddetallepedidos);

ALTER TABLE public.clientes ADD CONSTRAINT
clientes_codtipocli_fkey
FOREIGN KEY (codtipocli) REFERENCES public.tipocliente
(codtipocli);

ALTER TABLE public.clientes ADD CONSTRAINT direccion_coddir_fkey
FOREIGN KEY (coddir, nombrecli) REFERENCES public.direccion
(coddir,coddir);

ALTER TABLE public.reservas ADD CONSTRAINT reservas_ci_fkey
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);

ALTER TABLE public.reservas ADD CONSTRAINT reservas_codcli_fkey
FOREIGN KEY (codcli) REFERENCES public.clientes (codcli);

ALTER TABLE public.reservas ADD CONSTRAINT reservas_codtie_fkey
FOREIGN KEY (codtie) REFERENCES public.tiendas (codtie);

ALTER TABLE public.usurol ADD CONSTRAINT usurol_ci_fkey
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);

ALTER TABLE public.usurol ADD CONSTRAINT usurol_codrol_fkey
FOREIGN KEY (codrol) REFERENCES public.roles (codrol);

ALTER TABLE public.tiendas ADD CONSTRAINT direccion_tiendas
FOREIGN KEY (coddir) REFERENCES public.direccion (coddir);

ALTER TABLE public.tiendas ADD CONSTRAINT usuarios_ci
FOREIGN KEY (ci) REFERENCES public.usuarios (ci);
```

## **II.1.15 Modelado de Implementación**

### **II.1.15.1 Introducción**

Este modelo es una colección de componentes y de subsistemas que los contienen.

Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente y de tipo necesarios para la implantación y despliegue del sistema (este modelo es sólo una versión preliminar al final de la fase de elaboración, posteriormente tiene bastante refinamiento).

## **II.1.16 Prototipos de Interfaz de Usuario**

### **II.1.16.1 Introducción**

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

### **II.1.16.2 Propósito**

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

### **II.1.16.3 Alcance**

- Mostrar los prototipos de pantallas, sujeto a modificaciones a lo largo del desarrollo del sistema.

## II.1.16.4 Prototipo de Pantallas

*Descripción: Pantalla Ingresar al Sistema*



The screenshot shows a web browser window displaying the login page for 'Ballet Sin Fronteras'. The header features the company logo and name, 'Alquiler de Vestuarios en General', along with navigation links: 'Propietario', 'Prendas', 'Disponibles', 'Avisos', 'Tienda', and 'Estrenos'. The main content area is titled 'Iniciar sesion' and contains a login form with fields for 'Nombre de usuario:' and 'Contraseña:', and an 'Iniciar sesion' button. Below the form, there are four small images showing a woman in various colorful, floral-patterned dresses and hats, likely representing the types of costumes available for rental.


**Ballet "Sin Fronteras"**  
Alquiler de Vestuarios en General

Propietario Prendas Disponibles Avisos Tienda Estrenos

# Iniciar sesion

Nombre de usuario:

Contraseña:



### ***Pantalla: Gestionar Menú***

Esta pantalla contiene el menú gerente o administrador, el cual contiene los enlaces a los distintos procesos del sistema.

### ***Usuario del Sistema: Gerente***

The screenshot shows the web interface for the Gerente user. At the top, there is a header with the company logo and name "Ballet Sin Fronteras" and "Alquiler de Vestuarios en General". A welcome message reads: "Bienvenido Usuario: Rosely Veldaz Aramayo C#: 5818447 Fecha: Fecha: Lunes 10-10-3030 Salir". Below the header is a navigation menu with buttons for "Vestimentas", "Pedidos", "Administracion del sistema", "Reportes", "Tiendas", "Reservas", and "Alquileres". The main content area features a "Servicios Artísticos" button, the company name "Ballet Sin Fronteras", and the tagline "Alquiler de Vestuarios de Danzas en Graf". It also lists the "Gerente Propietario: Prof. Mario Colpari Leyton" and includes a logo of two dancers in red and blue.

### ***Usuario del Sistema: Administrador***

The screenshot shows the web interface for the Administrador user. The header is identical to the Gerente user's view, with the same logo, name, and welcome message. The navigation menu is different, showing buttons for "Vestimentas", "Tiendas", "Reservas", "Alquileres", and "Publicaciones". The main content area is also identical to the Gerente user's view, featuring the "Servicios Artísticos" button, company name, tagline, and proprietor information.

## ADMINISTRACION DEL SISTEMA

### *Pantalla: Administrar Usuarios*

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5818447 Fecha: Fecha:Lunes 10-10-2030 Salir

**Ballet con Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administracion del sistema Reportes Tiendas Reservas Alquileres

Administrar Usuarios

id	Ci	Nombre	Email	Telefono	Direccion
1	453278	Jorge Gamica Marella	jorge_e@hotmail.com	51237994	Zona:campeino Calle:comercio Numero:333
2	763421	Miguel Jaramillo Lopez	miguel_j@hotmail.com	73439785	Zona:san jorge Calle:venezuela Numero:777
3	132456	Gustavo Torres Mujca	gus_t@hotmail.com	6615259	Zona:san roque Calle:casaco Numero:2234
4	768904	Julio Morales Vallivan	julio_23@hotmail.com	73546289	Zona:le oma Calle:cochabamba Numero:555555
5	236578	Isen Torres Lamas	isen_t@hotmail.com	75123457	Zona:san roqueeee Calle:casaco Numero:2234
6	5818447	Rosely Valdez Aramayo	deby_dey4@hotmail.com	72954553	Zona:LaLoma Calle:Damas de Aguirre Numero:66
7	5678906	Juan Jose Vargas Murillo	juan_jose@hotmail.com	76543299	Zona:El molino Calle:Ramon Rojas Numero:55
8	2346789	Luisa Paredes Lima	luisa_l@hotmail.com	73456789	Zona:Las Panosas Calle:15 de Abril Numero:444
9	1030518	Nataly Romero Jaramillo	nataly_r_romero@hotmail.com	8847945	Zona:Las Panosas Calle:15 de Abril Numero:444
10	1030513	Edujo Valdez Aramayo	edu_diu@hotmail.com	79252998	Zona:Londiles Calle:Bustamante Numero 777

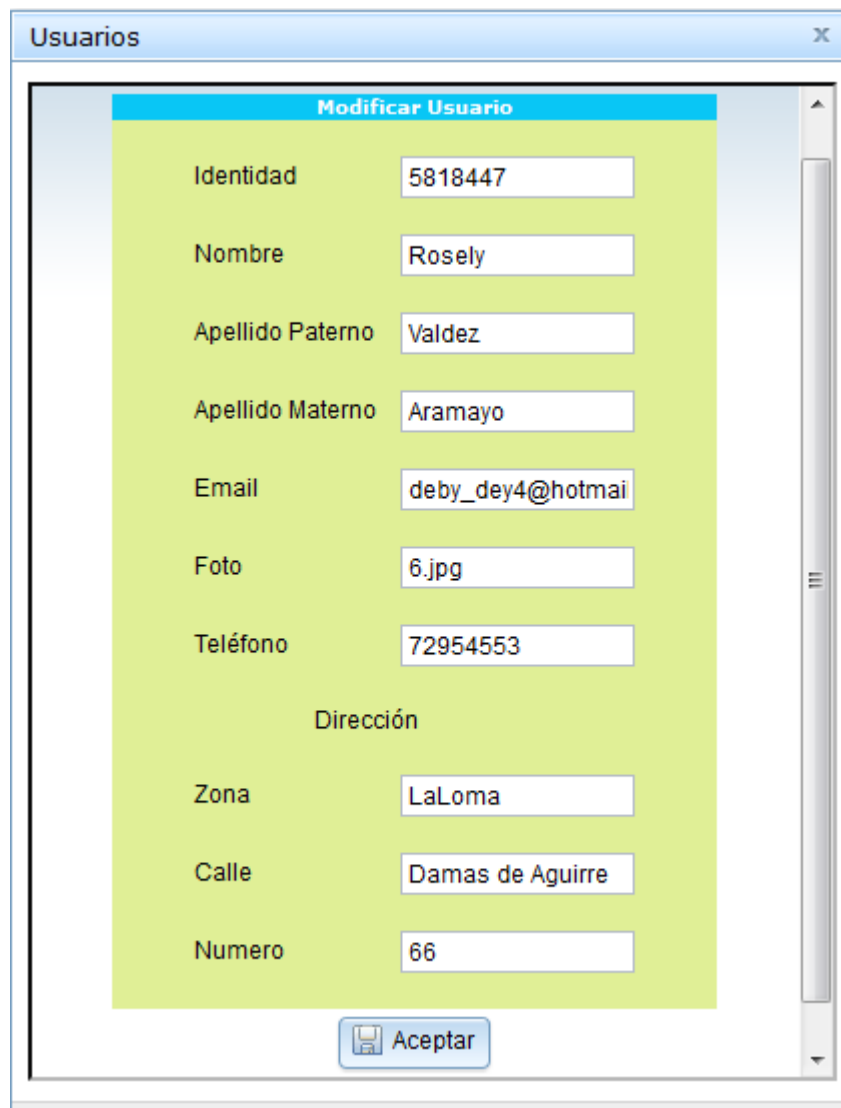
1 - 10 de 11 elementos

*Pantalla: Nuevo Usuario*

The image shows a software window titled 'Usuarios' with a sub-header 'Nuevo Usuario'. The form contains the following fields and controls:

- Identidad:
- Nombre:
- Apellido Paterno:
- Apellido Materno:
- Email:
- Foto:  [Examinar...](#)
- Teléfono:
- Dirección:
  - Zona:
  - Calle:
  - Numero:

At the bottom of the form, there is a button labeled 'Aceptar'.

*Pantalla: Modificar Usuarios*

The image shows a software window titled 'Usuarios' with a close button (X) in the top right corner. Inside the window is a form titled 'Modificar Usuario' with a light green background. The form contains several input fields for user information:

Identidad	5818447
Nombre	Rosely
Apellido Paterno	Valdez
Apellido Materno	Aramayo
Email	deby_dey4@hotmail
Foto	6.jpg
Teléfono	72954553
Dirección	
Zona	LaLoma
Calle	Damas de Aguirre
Numero	66

At the bottom center of the form is a button with a floppy disk icon and the text 'Aceptar'.



*Pantalla: Borrar Usuarios*

Usuarios

**Borrar Usuario**

Identidad	5818447
Nombre	Rosely
Apellido Paterno	Valdez
Apellido Materno	Aramayo
Email	15
Foto	6.jpg
Teléfono	72954553
Dirección	
Zona	LaLoma
Calle	Damas de Aguirre
Numero	66

Aceptar

*Pantalla: Asignar Rol a Usuario*

The image shows a software window titled "Usuarios" with a close button (X) in the top right corner. Inside the window is a dialog box titled "Asignar Rol a Usuario" with a light green background. The dialog box contains two input fields: "Usuario" with the value "5818447" and "Rol" with a dropdown menu showing "Seleccione". Below the input fields are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

Asignar Rol a Usuario	
Usuario	5818447
Rol	Seleccione
Aceptar Cancelar	

*Pantalla: Roles Asignados*

Usuarios

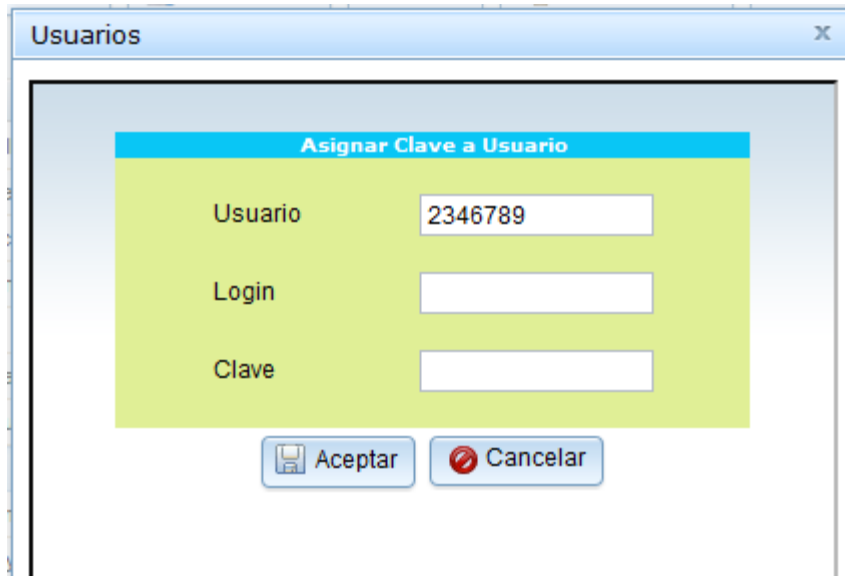
Roles Asignados

Borrar Recargar

Id	Rol
1	Vestimentas
2	Pedidos
3	Administracion del sistema
4	Reportes
5	Tiendas
6	Reservas
7	Alquileres

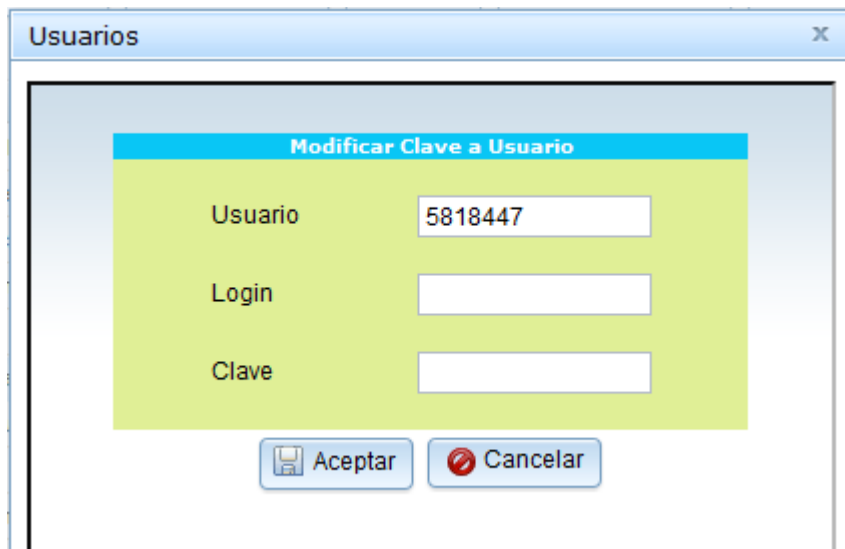
1 - 7 de 7 elementos

*Pantalla: Modificar Clave Usuario*



The screenshot shows a window titled "Usuarios" with a close button (X) in the top right corner. Inside the window is a dialog box titled "Asignar Clave a Usuario" with a light green background. The dialog contains three input fields: "Usuario" with the value "2346789", "Login" (empty), and "Clave" (empty). Below the fields are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

*Pantalla: Modificar Clave Usuario*





The screenshot shows a window titled "Usuarios" with a close button (X) in the top right corner. Inside the window is a dialog box titled "Modificar Clave a Usuario" with a light green background. The dialog contains three input fields: "Usuario" with the value "5818447", "Login" (empty), and "Clave" (empty). Below the fields are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

**Pantalla: Usuarios con Tiendas Habilitadas**

Usuarios

Lista de Tiendas

 **Buscar**  **Todos**

Id	Nombre	Tienda
1	Rosely Valdez Aramayo	tienda1
2	Eladio Valdez Aramayo	tienda3

1 - 2 de 2 elementos

**Pantalla: Habilitar Usuarios**

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5918447 Fecha: Fecha:lunes 10-10-2030 [Salir](#)

**Bella Sin Fronteras**  
Alquiler de Vestuarios en General

[Vestimentas](#) [Pedidos](#) [Administracion del sistema](#) [Reportes](#) [Tiendas](#) [Reservas](#) [Alquileres](#)

Usuarios No Habilitados

 **Buscar**  **Todos**  **Habilitar**  **Recargar**

Id	Ci	Nombre	Email	Telefono	Direccion
1	5566	Cinta Valdez Caceres	cinta_v@hotmail.com	455532	Zona campesino Calle:comercio Numero:122
2	324575	Marcos Flores Gamica	mar_cos@hotmail.com	72966554	Zona campesino Calle:comercio Numero:666
3	455	Gonzalo Vedia Jaramillo	gonso_o_56@hotmail.com	564566	Zona san roque Calle:damaso Numero 2234
4	987	Nely Romero Casco	nel_r@hotmail.com	65556	Zona Central Calle:Santa Cruz esq. Domingo Paz Numero 886
5	4677	Pablo galarza tejerna	pablo@hotmail.com	665455	Zona Mercado Campesino (rotonda fe y Alegria) Calle:Comercial 1o de Abril Caseta N 45(plania baja) Numero 45
6	456666	Veronica Gutierrez Flores	veronica_p@hotmail.com	666666	Zona Senac Calle Julio Arce Numero 777
7	5665	Nely Coronado Chaury	nel_co@hotmail.com	654545	Zona Cammpesino Calle:Comercio Numero:43554
8	4567854	Marcela Gremajc Vidaurte	marce_p@hotmail.com	7295080	Zona Aeropuerto Calle:helix aisa Numero:555

1 - 8 de 8 elementos

**Pantalla: Administrar Roles**

Diervención Usuario: Rosely Valdez Aramayo Ci: 5010447 Fecha: Fecha:lunes 10 10 2020 Salir

**Bella Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentos Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administración de Roles

Buscar Todos + Nuevo Modificar Borrar Asignar Grupo Grupos Recargar

Id	Codigo	Rol
1	2	Vestimentos
2	6	Pedidos
3	1	Administración del sistema
4	5	Reportes
5	10	Tiendas
6	4	Reservas
7	3	Alquileres
8	40	Publicaciones

1 - 8 de 8 elementos

**Pantalla: Nuevo Rol**

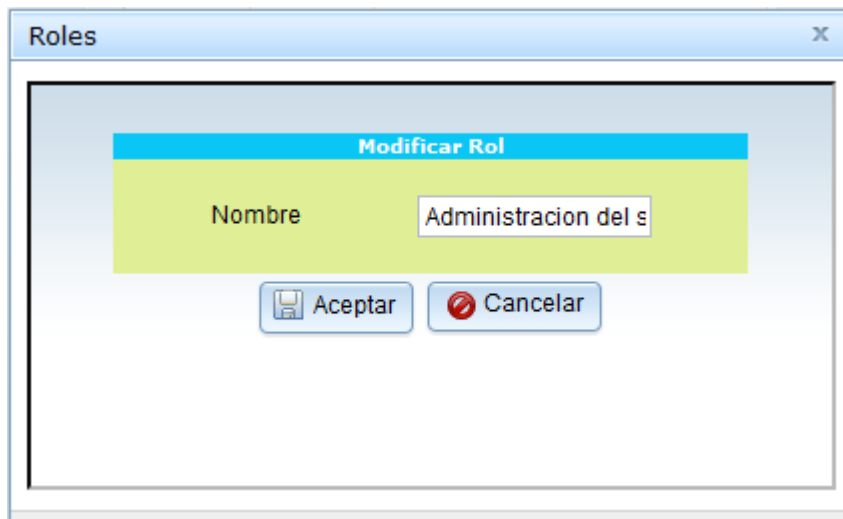
Roles

**Nuevo Rol**

Nombre

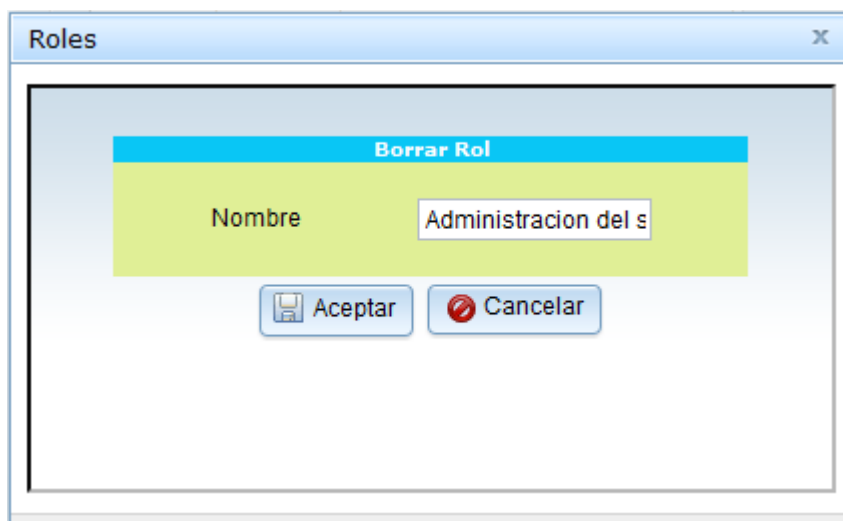
Aceptar Cancelar

*Pantalla: Modificar Rol*



The image shows a software window titled "Roles" with a close button (X) in the top right corner. Inside the window, there is a dialog box titled "Modificar Rol" with a light blue header. The dialog box has a light green background and contains a label "Nombre" followed by a text input field containing the text "Administracion del s". Below the input field, there are two buttons: "Aceptar" with a floppy disk icon and "Cancelar" with a red circle and slash icon.

*Pantalla: Borrar Rol*



The image shows a software window titled "Roles" with a close button (X) in the top right corner. Inside the window, there is a dialog box titled "Borrar Rol" with a light blue header. The dialog box has a light green background and contains a label "Nombre" followed by a text input field containing the text "Administracion del s". Below the input field, there are two buttons: "Aceptar" with a floppy disk icon and "Cancelar" with a red circle and slash icon.

*Pantalla: Asignar Grupo a Rol*

The screenshot shows a window titled 'Roles' with a sub-dialog box titled 'Asignar Grupos'. The dialog contains two input fields: 'Rol' with the value '1' and 'Grupo' with a dropdown menu showing 'Seleccione'. Below the fields are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

*Pantalla: Grupos Asignados*

The screenshot shows a window titled 'Roles' with a table titled 'Grupos Asignados al Rol'. Above the table are two buttons: 'Borrar' (Delete) and 'Recargar' (Refresh). The table has two columns: 'Id' and 'Grupo'. The data rows are as follows:

Id	Grupo
1	Gestionar Grupos
2	Gestionar Usuarios
3	Gestionar Roles
4	Gestionar Procesos
5	Gestinar Publicaciones



### Pantalla: Asignar Habilitar Roles

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha:lunes 10 10 3030 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Habilitar Roles

Id	Codigo	Nombre
1	21	rosely
2	22	fuli
3	23	ulhi
4	15	superusuariroot
5	12	Tienda
6	11	garantias
7	13	vestimentas
8	14	prendas

1 - 10 de 20 elementos

### Pantalla: Administrar Grupos

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha:lunes 10-10-3030 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General


Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administrar Grupos

Id	Codigo	Nombre	imagen
1	1	Inicio	inicio.png
2	5	Gestionar Garantias	garantias.png
3	7	Gestionar Grupos	admgrupos.png
4	9	Gestionar Lanzas	admlanzas.png
5	11	Gestionar Reservas	reservas.png
6	12	musica	musica.png
7	13	Calendario	calendario.png
8	14	Impresora	impresora.png
9	16	Gestionar Proveedores	admproveedores.png
10	15	Gestionar Pedidos	pedidos.png

1 - 10 de 22 elementos

*Pantalla: Nuevo Grupo*



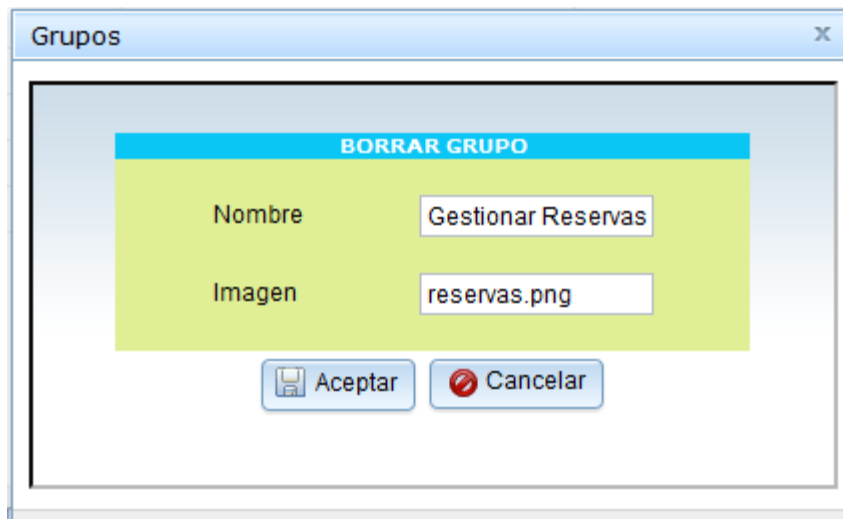
The screenshot shows a window titled "Grupos" with a close button (X) in the top right corner. Inside the window is a dialog box titled "NUEVO GRUPO" with a light green background. The dialog contains two input fields: "Nombre" and "Imagen". Below the "Imagen" field is a button labeled "Examinar...". At the bottom of the dialog are two buttons: "Aceptar" (with a floppy disk icon) and "Cancelar" (with a red circle and slash icon). The dialog box has a scroll bar on the right side.

*Pantalla: Modificar Grupo*



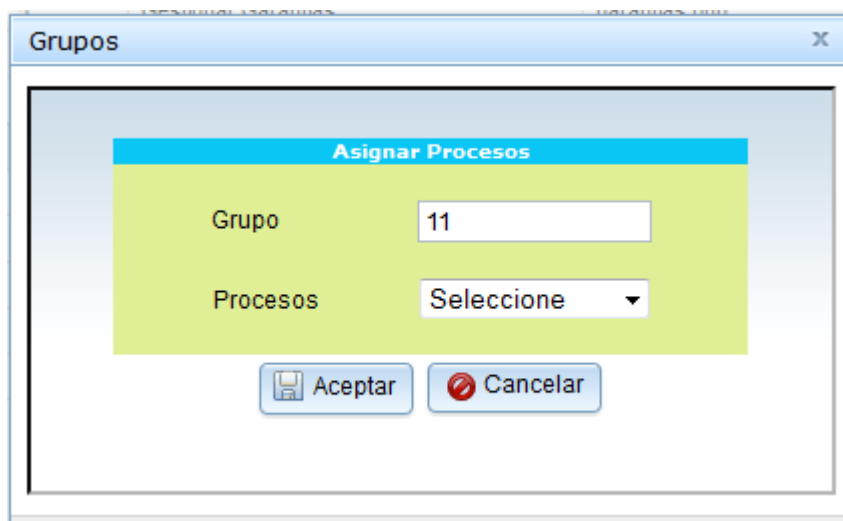
The screenshot shows a window titled "Grupos" with a close button (X) in the top right corner. Inside the window is a dialog box titled "MODIFICAR GRUPO" with a light green background. The dialog contains two input fields: "Nombre" and "Imagen". The "Nombre" field contains the text "Gestionar Reservas". Below the "Imagen" field is a button labeled "Examinar...". At the bottom of the dialog are two buttons: "Aceptar" (with a floppy disk icon) and "Cancelar" (with a red circle and slash icon). The dialog box has a scroll bar on the right side.

*Pantalla: Borrar Grupo*



The screenshot shows a window titled "Grupos" with a close button (X) in the top right corner. Inside the window is a dialog box titled "BORRAR GRUPO" with a light green background. The dialog contains two text input fields: "Nombre" with the value "Gestionar Reservas" and "Imagen" with the value "reservas.png". Below the fields are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

*Pantalla: Asignar Procesos a Grupo*



The screenshot shows a window titled "Grupos" with a close button (X) in the top right corner. Inside the window is a dialog box titled "Asignar Procesos" with a light green background. The dialog contains two input fields: "Grupo" with the value "11" and "Procesos" with a dropdown menu showing "Seleccione". Below the fields are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

**Pantalla: Procesos a Asignados**

Procesos Asignados al Grupo

Borrar Recargar

Id	Proceso
1	Administrar Tipo Prendas
2	Administrar Prendas
3	Administrar Precios
4	Administrar Tallas
5	Administrar Colores

**Pantalla: Habilitar Grupos**

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5813447 Fecha: Fecha:lunas 10-10-2030 Salir

Ballet Sin Fronteras Alquiler de Vestuarios en General

Vestimentas Pedidos Administracion del sistema Reportes Tiendas Reservas Alquileres

Habilitar Grupos

Buscar Todos Habilitar Recargar

id	Codigo	Nombre	Imagen
1	3	Ocasionar Almacen	cdma/traocer.png
2	25	aac	lal-add.png

1 - 2 de 2 elementos

**Pantalla: Administrar Procesos**

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha:lunes 10-10-2020 [Salir](#)

**Sala Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administracion del sistema Reportes Tiendas Reservas Alquileres

Administrar Procesos

Buscar + Nuevo Modificar Borrar Restaurar

Id	Codigo	Nombre	Nombre Url
1	14	Asignar Prenda a Danza	asignarprendaadanza.html
2	39	Reporte Usuarios	reporteusuarios.html
3	37	Asignar Proceso a Grupo	asignargrupoproceso.html
4	36	Asignar Grupo a Rol	asignargruporol.html
5	2	Asignar Rol a Usuario	asignarrolausuario.html
6	5	Asignar Rol a Proceso	asignarrolaproceso.html
7	45	Administracion de Alquiler Por Traje	admin_alquiler_uniforme
8	32	Habilitar Danzas	danzasnohabilitadas.html
9	29	Habilitar Grupos	gruposnohabilitados.html
10	4	Habilitar Roles	rolesnohabilitados.html

1 - 10 de 50 elementos

**Pantalla: Nuevo Procesos**

Procesos

**NUEVO PROCESO**

Nombre

Nombre\_Url

Aceptar Cancelar

*Pantalla: Modificar Procesos*

The screenshot shows a window titled "Procesos" with a close button (X) in the top right corner. Inside the window, there is a light green rectangular area with a blue header that reads "MODIFICAR PROCESO". Below the header, there are two text input fields: "Nombre" with the value "Reporte Usuarios" and "Nombre\_Url" with the value "reporteusuarios.htm". At the bottom of the green area, there are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

*Pantalla: Borrar Procesos*

The screenshot shows a window titled "Procesos" with a close button (X) in the top right corner. Inside the window, there is a light green rectangular area with a blue header that reads "BORRAR PROCESO". Below the header, there are two text input fields: "Nombre" with the value "Reporte Usuarios" and "Nombre\_Url" with the value "reporteusuarios.htm". At the bottom of the green area, there are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

### Pantalla: Habilitar Procesos

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5813447 Fecha: Fecha: lunas 10-10-2030 Salir

**Ballet con Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Habilitar Procesos

Buscar Todos Habilitar Recargar

Id	Codigo	Nombre	Nombre Url
1	41	dUsuarios	reporteusuario.html
2	35	Usuariosdd	reporteusuario.html
3	40	Administracion de Danzas	admdanzas.html
4	44	fghdfggdh	fhyhgvhjg
5	25	Listar Usuarios	listarusuarios.html
6	26	Listar Roles	listarroles.htm
7	27	Listar Procesos	listarprocesos.html
8	28	Listar Garantias	listargarantias.html
9	30	Listar Grupos	listargrupos.html

1 - 10 de 12 elementos

### Pantalla: Administrar Danzas

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818417 Fecha: Fecha: lunes 10-10-2030 Salir

**Ballet con Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administración del sistema Reportes Tiendas Reservas Alquileres

Administrar Danzas

Buscar Todos Nuevo Modificar Eliminar Asignar Prenda Prendas Recargar

Id	Codigo	Nombre	Lugar	Zona
1	2	a'troeva	la paz	allipano
2	4	arabe	arabe	internacional
3	16	argentin	chacarera	internacional
4	30	afico	tarja	regional
5	23	brincac	oriental	oriente
6	30	cacharpaya	tarja	regional
7	28	calchonoa	poroal	allipano
8	4	capu al	la paz	oriente
9	20	carnevalc	cochabamba	vcllo

1 - 10 de 46 elementos

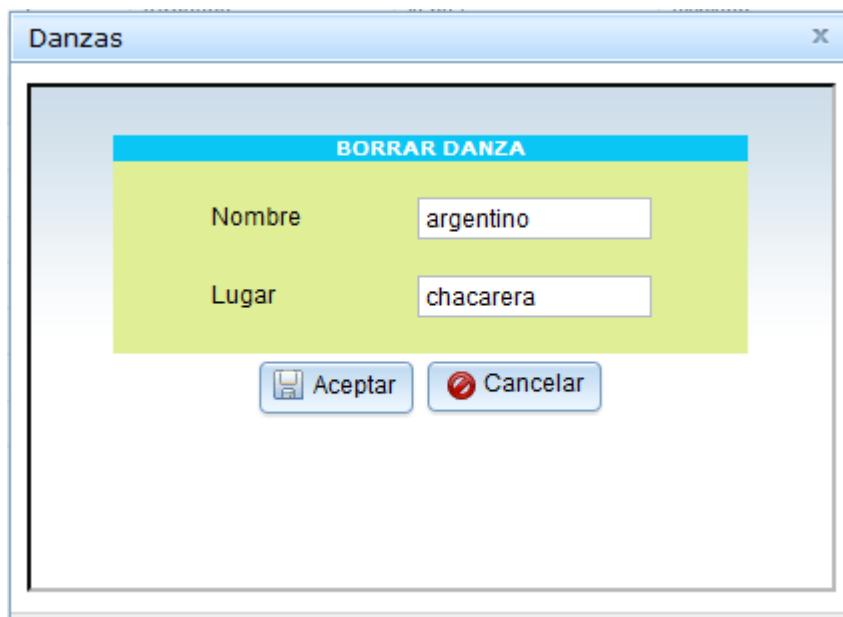
*Pantalla: Nueva Danza*

The screenshot shows a window titled "Danzas" with a close button (x) in the top right corner. Inside the window is a form titled "Nuevo Danza" with a light green background. The form contains three input fields: "Nombre" (text input), "Lugar" (text input), and "Zona" (dropdown menu with "Seleccione" selected). Below the form are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

*Pantalla: Modificar Danza*

The screenshot shows a window titled "Danzas" with a close button (x) in the top right corner. Inside the window is a form titled "MODIFICAR DANZA" with a light green background. The form contains three input fields: "Nombre" (text input with "argentino" entered), "Lugar" (text input with "chacarera" entered), and "Zona" (dropdown menu with "Seleccione" selected). Below the form are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).



***Pantalla: Borrar Danzas***


**Danzas**

**BORRAR DANZA**

Nombre:

Lugar:

***Pantalla: Habilitar Danzas***


Bienvenido Usuario: Rosely Valdez Arzamayo G: 5818447 Fecha: Fecha:lunes 10-10-2030 Salir

**Bala Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Habilitar Danzas

Id	Código	Nombre	Lugar
1	61	aas	gvv
2	24	chcoquonc	tarija
3	62	9999	hhhh
4	24	negritos	endno

1 - 4 de 4 elementos

### Pantalla: Administrar Zonas

Dienvenido Usuario: Rosely Valdez Aramayo CI: 5816447 Fecha: Fecha:lunes 10 10 3030 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administración de Zonas

Id	Zona	Estado
1	oriente	1
2	internacional	1
3	regional	1
4	altplano	1
5	valle	1
6	ppp	1

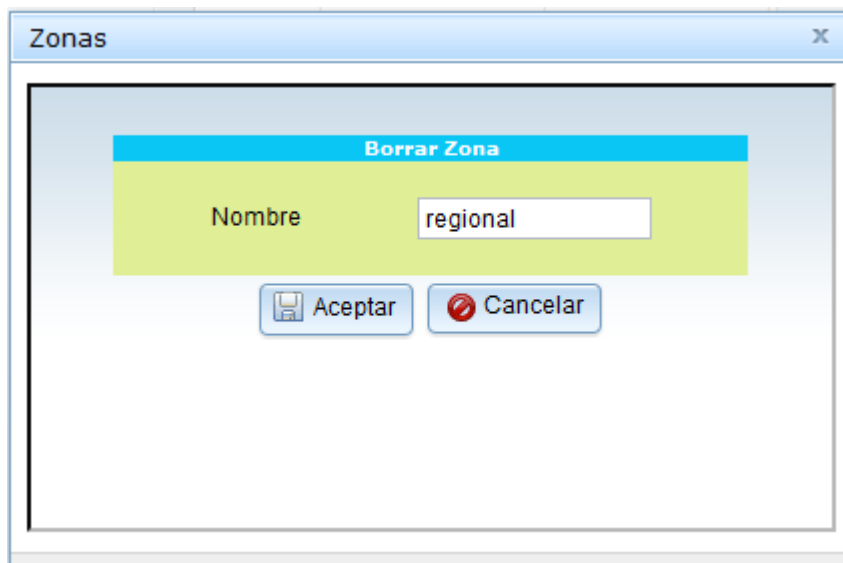
1 - 6 de 6 elementos

### Pantalla: Nueva Zona

Zonas

**Nueva Zona**

Zona

**Pantalla: Borrar Zona**


Nombre regional

Aceptar Cancelar

**Pantalla: Administrar Prendas**

Dienvenido Usuario: Rosely Valdez Aramayo C: 5618447 Fecha: Fecha:lunes 10 10 3030 [Salir](#)

**Balla con Fronteras**  
Alquiler de Vestuarios en General

Vestimentos Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administrar Prendas

Buscar Todo Nuevo Modificar Recargar

Id	Prenda	Color	Talla	Genero	Manza
1	falda	verde lienzo	M	femenino	sfrosays
2	falda	foxia lienzo	M	femenino	sfrosays
3	blusa	morado s'rena	S	femenino	sfrosays
4	blusa	morado	M	femenino	sfrosays
5	blusa	verde lienzo	M	femenino	sfrosays
6	blusa	foxia lienzo	M	femenino	sfrosays
7	camisa	morado s'rena	S	masculino	cfrosayc
8	camisa	morado	M	masculino	cfrosayc
9	camisa	verde lienzo	M	masculino	cfrosayc
10	mangas	verde blanco	M	masculino	cfrosaye

1 - 10 de 00 elementos

*Pantalla: Nueva Prenda*

Prendas

**Nueva Prenda**

Danza

Tipo Prenda

Color

Talla

Precio



Estreno	<input type="text" value="25"/>
Normal	<input type="text" value="20"/>
Antiguo	<input type="text" value="15"/>

*Pantalla: Modificar Prenda*

Prendas

Nueva Prenda

Danza afrosaya



Tipo Prenda blusa

Color morado

Talla M

Precio 204

Estreno	25
Normal	15
Antiguo	20

 Aceptar  Cancelar

**Pantalla: Administrar Tipo Prendas**

The screenshot shows the 'Administrar Tipo Prendas' interface. At the top, there is a header with the logo 'Bella Sin Fronteras' and the text 'Alquiler de Vestuarios en General'. Below the header is a navigation menu with options: Vestimentas, Pedidos, Administracion del sistema, Reportes, Tiendas, Reservas, and Alquileres. The main content area is titled 'Administrar Tipo Prendas' and includes a search bar, a 'Buscar' button, and a toolbar with buttons for 'Todos', 'Nuevo', 'Modificar', 'Borrar', and 'Recargar'. A table displays a list of clothing types with columns for 'Codigo' and 'Nombre'.

	Codigo	Nombre
1	78	ajilayn
2	3	axcas
3	58	huusa
4	86	bombachon
5	47	bolas
6	46	buzo
7	7	camisa
8	75	cape
9	79	casco
10	84	chaleco

At the bottom of the table, there is a pagination control showing '1 - 10 de 63 elementos' and navigation arrows.

**Pantalla: Nuevo Tipo Prenda**

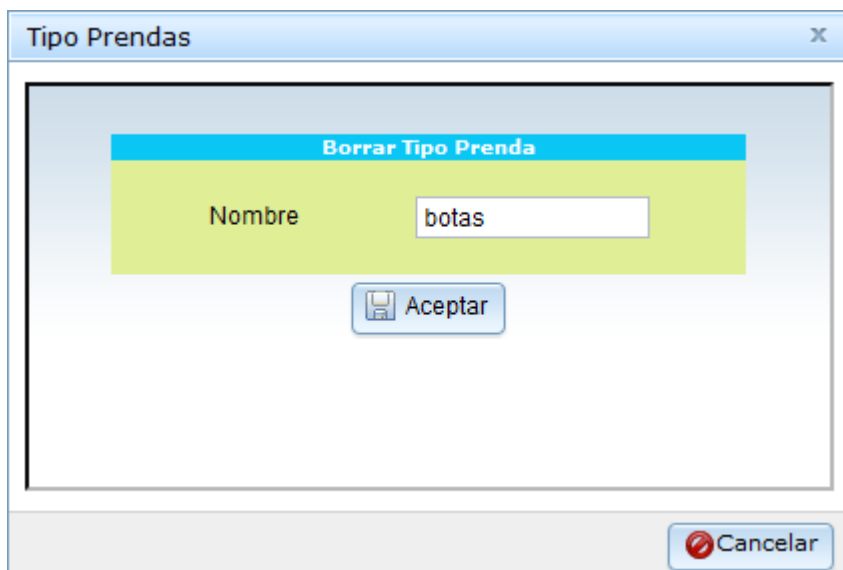
The screenshot shows a dialog box titled 'Tipo Prendas' with a sub-header 'Nuevo Tipo Prenda'. Inside the dialog, there is a text input field labeled 'Nombre'. Below the input field is an 'Aceptar' button with a floppy disk icon. At the bottom right of the dialog is a 'Cancelar' button with a red circle and slash icon.

*Pantalla: Modificar Tipo Prenda*



The screenshot shows a window titled "Tipo Prendas" with a close button (X) in the top right corner. Inside the window, there is a dialog box titled "Modificar Tipo Prenda" with a light blue header. The dialog box has a light green background and contains a label "Nombre" followed by a text input field containing the word "botas". Below the input field is a button labeled "Aceptar" with a floppy disk icon. At the bottom right of the main window, there is a button labeled "Cancelar" with a red prohibition sign icon.

*Pantalla: Borrar Tipo Prenda*



The screenshot shows a window titled "Tipo Prendas" with a close button (X) in the top right corner. Inside the window, there is a dialog box titled "Borrar Tipo Prenda" with a light blue header. The dialog box has a light green background and contains a label "Nombre" followed by a text input field containing the word "botas". Below the input field is a button labeled "Aceptar" with a floppy disk icon. At the bottom right of the main window, there is a button labeled "Cancelar" with a red prohibition sign icon.

### Pantalla: Administrar Colores

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5818417 Fecha: Fecha:lunes 10-10-2020 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administración de Colores

Id	Color	Estado
1	amarillo	1
2	amarillo nro	1
3	amarillo patito	1
4	amarillo rojo	1
5	azul	1
6	azul blanco	1
7	blanco	1
8	cafe	1
9	calipso	1
10	calipso jaspeado	1

1 - 10 de 47 elementos

### Pantalla: Nuevo Tipo Prenda

Colores

**Nuevo Color**

Color

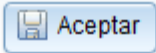


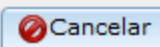
**Pantalla: Borrar Tipo Prenda**

Tipo Prendas

**Borrar Tipo Prenda**

Nombre

 Aceptar

 Cancelar

**Pantalla: Administrar Tallas**

Bienvenido Usuario: Rosely Valdez Aramayo C: 5618447 Fecha: Fecha:lunes 10-10-3030 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administración de Tallas

 Buscar  Todos  Nuevo  Borrar  Recargar

Id	Talla	Categoría	Estado
1	S	juvenil	juvenil
2	XS	infantil	infantil
3	M	juvenil	juvenil
4	L	juvenil	juvenil
5	XL	juvenil	juvenil
6	36	juvenil	juvenil
7	35	juvenil	juvenil
8	37	juvenil	juvenil
9	36	juvenil	juvenil
10	39	juvenil	juvenil

1 - 10 de 11 elementos

### Pantalla: Administrar Precios

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha: Lunes 10-10-2030 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentos Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Listar Precios

Id	Código	Tip. Lanta	Monto	Fecha	Prenda	Color	Talla
1	250	1	25	2012-03-23	blusa	azul	S
2	251	2	20	2012-03-23	blusa	azul	S
3	252	3	15	2012-03-23	blusa	azul	S
4	253	1	25	2012-03-23	blusa	azul	XS
5	254	2	20	2012-03-23	blusa	azul	XS
6	255	3	15	2012-03-23	blusa	azul	XS
7	256	1	25	2012-03-23	blusa	fuxie	XS
8	257	2	20	2012-03-20	blusa	fuxie	XS
9	258	3	15	2012-03-20	blusa	fuxie	XS
10	259	1	25	2012-03-23	blusa	fuxie	S

1 - 10 de 103 elementos

### Pantalla: Administrar Pedidos

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha: Lunes 10-10-2030 [Salir](#)

**Balla Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentos Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administrar Pedidos

Id	Código	Proveedor	Usuario	Fecha Pedido
1	83	Marta Valdez Valca	5818447	2012-03-23
2	82	Idia Valdez Aramayo	5818447	2012-03-23
3	85	Marcos mansillaaa Romero	5818447	2012-03-23
4	80	Carla Michel Rios	5810447	2012-04-10
5	84	Carla Michel Rios	5810447	2012-05-20

1 - 5 de 5 elementos

### Pantalla: Nuevo Pedido

Pedidos

**NUEVO PEDIDO**

++ | --

Genero	Danza	Tipo	Color	Talla	Cantidad	Precio Unitario
masculino	arabe	chaleco	cafe	M	10	12

Proveedor:

Fecha Realización:

Estado Uso:

### Pantalla: Administrar Proveedores

Bienvenido Usuario: Rogoly Valdez Aramayo C: 5818417 Fecha: Fecha: lunes 10-10-3030 [Salir](#)

**Baila Sin Fronteras**  
Alquiler de Vestuarios en General

[Vestimentas](#)
[Pedidos](#)
[Administración del sistema](#)
[Reportes](#)
[Tiendas](#)
[Reservas](#)
[Alquileres](#)

Administrar Proveedores

Id	Usuario	Proveedor	Telefono	Dirección
1	543465	Juan Vallejos Torres	662345	Zone: campesino Calle: comercio Numero:122
2	55555	Carla Michel Ros	444	Zone: campesino Calle: comercio Numero:333
3	555	Idia Valdez Aramayo	72949066	Zone: san roque Calle: danzasa Numero:2234
4	345567	Marta Valdes Valde	72965248	Zone: Central Calle: Santa Cruz esq. Domingo Paz Numero:695
5	6712345	Luisa Martelanas Vidaurre	8826745	Zone: tmallitas Calle: Rustamate Numero: 777

1 - 5 de 5 elementos

*Pantalla: Nuevo Proveedor*

The image shows a software window titled "Proveedores" with a close button (X) in the top right corner. Inside the window is a form titled "Nuevo Proveedor" with a light green background. The form contains the following fields:

- Identidad:
- Nombre:
- Apellido Paterno:
- Apellido Materno:
- Telefono:
- Dirección (Section Header):
  - Zona:
  - Calle:
  - Numero:

At the bottom of the form are two buttons: "Aceptar" (with a floppy disk icon) and "Cancelar" (with a red circle and slash icon).

*Pantalla: Modificar Proveedor*

Proveedores x

**Modificar Proveedor**

Identidad	<input type="text" value="555"/>
Nombre	<input type="text" value="lidia"/>
Apellido Paterno	<input type="text" value="valdez"/>
Apellido Materno	<input type="text" value="Aramayo"/>
Telefono	<input type="text" value="72946086"/>
Dirección	
Zona	<input type="text" value="san roque"/>
Calle	<input type="text" value="damaso"/>
Numero	<input type="text" value="2234"/>

*Pantalla: Borrar Proveedor*

Proveedores x

**Borrar Proveedor**

Identidad	<input type="text" value="555"/>
Nombre	<input type="text" value="lidia"/>
Apellido Paterno	<input type="text" value="valdez"/>
Apellido Materno	<input type="text" value="Aramayo"/>
Telefono	<input type="text" value="72946086"/>
Dirección	
Zona	<input type="text" value="san roque"/>
Calle	<input type="text" value="damaso"/>
Numero	<input type="text" value="2234"/>



### Pantalla: Modificar Noticia

Bienvenido Usuario: Rosely Valdez Aramayo C: 5818447 Fecha: Fecha: Lunes 10-10-2010 Salir

Vestimentas Pedidos Noticias Administración del sistema Reportes Tiendas Reservas Alquileres

Lista de Noticias

Buscar Todos Nuevo Modificar Borrar Recargar

Id	Código	Ci	Autor	Fecha Inicio	Fecha Fin
1	3	1030513	Juan	2012-03-23	2012-03-23
2	4	5818447	Jorge	2012-03-23	2012-03-23
3	10	5818447	leonor	2012-03-23	2012-03-23
4	9	5818447	99999	2012-03-23	2012-03-23
5	8	5818447	leonard	2012-03-23	2012-03-23
6	1	5818447	hujc	2012-03-23	2012-03-23
7	12	5818447	rospah	2012-03-23	2012-03-23

1 / de 7 elementos

**MODIFICAR NOTICIA**

Usuario:

Autor:

Tipo:

Contenido:

Formato: Ninguno

avisos

### Pantalla: Borrar Noticia

Bienvenido Usuario: Rosely Valdez Aramayo C: 5818447 Fecha: Fecha: Lunes 10-10-2010 Salir

Vestimentas Pedidos Noticias Administración del sistema Reportes Tiendas Reservas Alquileres

Recargar

Id	Código	Ci	Autor	Fecha Inicio	Fecha Fin
1	3	1030513	Juan	2012-03-23	2012-03-23
2	11	5818447	Jorge	2012-03-23	2012-03-23
3	10	5818447	leonor	2012-03-23	2012-03-23
4	9	5818447	99999	2012-03-23	2012-03-23
5	8	5818447	leonard	2012-03-23	2012-03-23
6	1	5818447	hujc	2012-03-23	2012-03-23
7	12	5818447	rospah	2012-03-23	2012-03-23

1 - 7 de 7 elementos

**MODIFICAR NOTICIA**

Usuario:

Autor:

Tipo:

Contenido:

Formato: Ninguno



### Pantalla: Administrar Tiendas

Bienvenido Usuario: Rosely Valdez Aramayo C: 5618447 Fecha: Fecha:lunes 10-10-2030 [Salir](#)

**Sala Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Lista de Tiendas

ID	Codigo	Nombre	Telefono	Dirección	Usuario
1	4	tienda1	76193345	Zona Central Calle: Santa Cruz esq. Domingo Paz Numero:896	Nombre: Rosely Ap: Valdez Am: Aramayo
2	5	tienda3	74540405	Zona Mercado Campesino (rotonda Fe y Alegria) Calle Comercial "5 de Abril" Caseta N 45 (planta baja) Numero:45	Nombre: Flavin Ap: Valdez Am: Aramayo

### Pantalla: Nueva Tiendas

Tiendas

**Nueva Tienda**

Nombre

Telefono

Dirección

Zona

Calle

Numero

**Pantalla: Administrar Garantías**

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5816447 Fecha: Fecha: lunes 10-10-2020 [Salir](#)

**Sala Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Administra: Garantías

Id	Código	tipo	Valor	tipo de Moneda
1	1	documentos	1000	boliviano
2	2	dinero	400	dolar
3	4	dinerok	5445	boliviano

1 - 3 de 3 elementos

**Pantalla: Nueva Garantía**

Garantías

**NUEVA GARANTIA**

Tipo

Valor

Moneda

*Pantalla: Modificar Garantía*



The screenshot shows a window titled "Garantias" with a close button (X) in the top right corner. Inside the window is a dialog box titled "MODIFICAR GARANTIA" with a light green background. The dialog contains three input fields: "Tipo" with the value "dinero", "Valor" with the value "400", and "Tipo de Moneda" with a dropdown menu showing "dolar". At the bottom of the dialog are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

*Pantalla: Borrar Garantía*



The screenshot shows a window titled "Garantias" with a close button (X) in the top right corner. Inside the window is a dialog box titled "BORRAR GARANTIA" with a light green background. The dialog contains three input fields: "Tipo" with the value "dinero", "Valor" with the value "400", and "Tipo Moneda" with the value "2". At the bottom of the dialog are two buttons: "Aceptar" (Accept) with a floppy disk icon and "Cancelar" (Cancel) with a red circle and slash icon.

*Pantalla: Administrar Clientes*

Bienvenido Usuario: Rosely Valdez Aramayo C: 5618447 Fecha: Fecha: lunes 10-10-2020 [Salir](#)

**Sala Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos **Administración del sistema** Reportes Tiendas Reservas Alquileres

Administrar Clientes

[Buscar](#) [Todos](#) [+ Nuevo](#) [Modificar](#) [Borrar](#) [Asignar Clave](#) [Recargar](#)

id	Identidad	tipo	Cliente	Nit	leetcno	Direccion
1	4444	comun	valdez juan valdez	4555	5555	Zona:san roque Calle:damaso Numero:2234
2	83/887	comun	dominguez Luis2 tec	78/87	88/787	Zona:romanas Calle:bus:amate Numero:111
3	84/888	comun	dominguez Luis juarez	78/88	88/788	Zona:romanas Calle:bus:amate Numero:111

1 - 3 de 3 elementos

*Pantalla: Nuevo Cliente*

The image shows a software window titled "Clientes" with a close button (X) in the top right corner. Inside the window is a form titled "NUEVO CLIENTE" with a light green background. The form contains the following fields:

- Codigo de Cliente:
- Identidad:
- Tipocliente:  (dropdown menu)
- Nombre:
- Apellido Paterno:
- Apellido Materno:
- Nit:
- Telefono:
- Dirección: (Section header)
- Zona:
- Calle:
- Numero:

A vertical scrollbar is visible on the right side of the form area.

*Pantalla: Modificar Cliente*

The image shows a software window titled 'Clientes' with a close button (X) in the top right corner. Inside the window is a form titled 'MODIFICAR CLIENTE' with a light green background. The form contains the following fields:

Codigo de Cliente	6677
Identidad	887666
Tipo Cliente	comun
Nombre	luis
Apellido Paterno	domingues
Apellido Materno	juares
Nit	78766
Telefono	987766
Dirección	
Zona	tomatitas
Calle	Bustamate
Numero	777

*Pantalla: Borrar Cliente*

The image shows a software window titled "Cientes" with a close button (X) in the top right corner. Inside the window, there is a form titled "MODIFICAR CLIENTE" with a light green background. The form contains the following fields:

Codigo de Cliente	6677
Identidad	887666
Tipo Cliente	Seleccione ▼
Nombre	luis
Apellido Paterno	domingues
Apellido Materno	juares
Nit	78766
Telefono	987766
Dirección	
Zona	tomatitas
Calle	Bustamate
Numero	777

*Pantalla: Administrar Tipo Clientes*

Bienvenido Usuario: Rosely Valdez Aramayo CI: 551844 / Fecha: Fecha: lunes 10-10-2020 Salir

**Ballet "Sin Fronteras"**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administración del sistema Reportes Tiendas Reservas Alquileres

Administrar Tipo Clientes

Descargar Todos Nuevo Modificar Borrar Recargar

id	Código	Tipo
1	1	común
2	3	favulita

1 - 2 de 2 elementos

*Pantalla: Nuevo Tipo Cliente*

Tipo Cliente

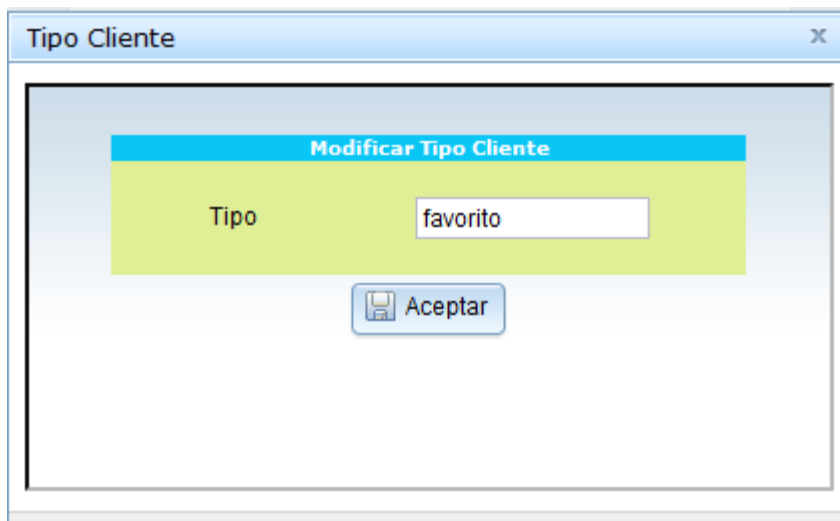
**Nuevo Tipo Cliente**

Tipo

Aceptar

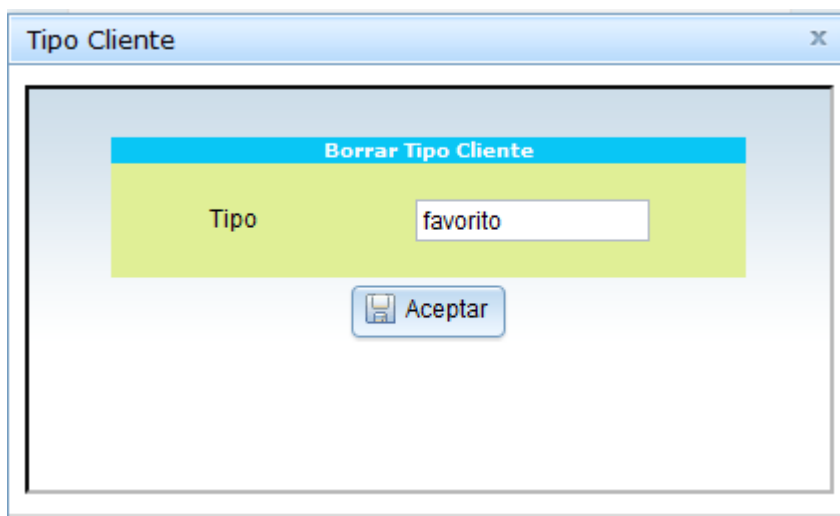


*Pantalla: Modificar Tipo Cliente*



The screenshot shows a window titled "Tipo Cliente" with a close button (X) in the top right corner. Inside the window, there is a dialog box titled "Modificar Tipo Cliente" with a light green background. The dialog box contains a label "Tipo" followed by a text input field containing the word "favorito". Below the input field is a button labeled "Aceptar" with a floppy disk icon.

*Pantalla: Borrar Tipo Cliente*



The screenshot shows a window titled "Tipo Cliente" with a close button (X) in the top right corner. Inside the window, there is a dialog box titled "Borrar Tipo Cliente" with a light green background. The dialog box contains a label "Tipo" followed by a text input field containing the word "favorito". Below the input field is a button labeled "Aceptar" with a floppy disk icon.

### *Pantalla: Administrar Alquiler*

The screenshot shows a web browser window displaying the 'Administración de Alquileres' page. The browser's address bar shows the URL 'http://www.baladonfronteras.com'. The page header includes a welcome message: 'Bienvenido Usuario: Rosely Valdez Aramayo CI: 5618447 Fecha: Fecha: lunes 10-10-2030 Salir'. The main navigation menu contains the following items: 'Vestimentas', 'Pedidos', 'Noticias', 'Administración del sistema', 'Reportes', 'Tiendas', 'Reservas', and 'Alquileres'. The 'Alquileres' section is active, showing a sub-header 'Administración de Alquileres'. Below this, there is a search bar and several action buttons: 'Buscar', 'Todos', '+ Nuevo', 'Modificar', 'Borrar', 'Devolucion', 'Grupos', and 'Recargar'. A table with the following columns is visible: 'Id', 'Codigo', 'Nombre', 'Nombre', 'Nombre', and 'Nombre'. The table is currently empty, and the status at the bottom left indicates '0 elementos'.

**Pantalla: Registrar Entrada de Prendas a Tienda**

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha:lunes 10-10-3030 [Salir](#)

**Ballet Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administracion del sistema Reportes Tiendas Reservas Alquileres

ASIGNACION PRENDAS A TIENDA

Tienda tienda1 35356 4

Usuario Rosely Valdez Aramayo  
Fecha

++ 3 --

zona Estado Uso  
Seleccione Seleccione

Genero	Danza	Tipo	Color	Talla	Cant Disp
femeninc	Seleccione	Seleccione	Seleccione	Seleccione	
femeninc	Seleccione	Seleccione	Seleccione	Seleccione	
femeninc	Seleccione	Seleccione	Seleccione	Seleccione	

ACEPTAR CANCELAR

**Pantalla: Nuevo Alquiler**

Bienven do Usuario: Rosely Valdez Aramayo C: 5818447 Fecha: Fecha:lunes 10-10-3030 [Salir](#)

**Ballet Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administracion del sistema Reportes Tiendas Reservas Alquileres

NUEVO ALQJILER VESTIMENTA

Tienca tienda1 35355 4

Usuario Rosely Valdez Aramayo

Cliente 6677

Garantias

mueble /yyyyy

inmueble bicicleta /yyyyy

dnerc /yyyyy

camara /yyyyy

4

Fecha alquiler\_uniforme

Fecha Devolucion

++ 0 --

zona Seleccione

Genero	Danza	Estado Uso	Nro Uniformes
--------	-------	------------	---------------

Total costo alquiler\_uniforme = \$total

ACEPTAR CANCELAR

III

**Pantalla: Nuevo Alquiler Uniforme**

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5818447 Fecha: Fecha:lunes 10-10-3030 [Salir](#)

**Bala Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administracion del sistema Reportes Tiendas Reservas Alquileres

NUEVO ALQUILER VESTIMENTA

Tienda tenda1 35355 4

Usuario Rosely Valdez Aramayo  
 Cliente 6677

Garantias

mueb e yyyy  
 inmueble bicicleta yyyy  
 dinerc yyyy  
 camara yyyy  
 4

Fecha alquiler\_uniforme

Fecha Devolucion

++ 0 --

zona Seleccione

Genero	Danza	Estado Uso	Nro Uniformes
--------	-------	------------	---------------

Total costo alquiler\_uniforme = \$total

### *Pantalla: Administrar Reservas*

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5818\*17 Fecha: Fecha: lunes 10-10-2030 Salir

**Bata Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administración del sistema Reportes Tiendas Reservas Alquileres

Administración de Alquileres

Buscar Todas Nuevas Modificar Borrar Devolución Grupos Imprimir

id	Código	Nombre	Nombre	Nombre	Nombre
0 elementos					

### *Pantalla: Nueva Reserva*

\_detalle/menus.html

**Bata Sin Fronteras**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Noticias Administración del sistema Reportes Tiendas Reservas Alquileres

\*\*\*NUEVA RESERVA DE PRENDAS\*\*\*

Tienda tienda1 33356 4

Usuario Rosely Valdez Aramayo

Cliente GG77

Gatallas

mueble

inmueble bicideta

dinero

camara

Fecha

Fecha Devolucion

zona Selección

Genero	Edad	Tipo	Color	Talla	Estado Us	Cant. Disp	Cant. Res	Precio Unitario
Total costo reserva = \$0.00								

### Pantalla: Administrar Reservas

### Pantalla: Nueva Reserva

## Reporte Prendas

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha: Lunes 10-10-3030 Salir

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

HTML Ver en Pdf Guardar en Excel

### Reporte Prendas

Usuario: Rosely Valdez Aramayo Tienda: tienda1 Fecha: Mon Apr 23 06:25:18 MDT

Codigo	Prenda	Color	Talla	Genero	Estado Uso	Precio	Cantidad
235	blusa	verde lienzo	M	femenino	Estrecho	25	50
235	blusa	verde lienzo	M	femenino	Normal	20	50
235	blusa	verde lienzo	M	femenino	Antiguo3	15	50
237	camisa	morado	M	masculino	Estrecho	25	50
237	camisa	morado	M	masculino	Normal	20	50
237	camisa	morado	M	masculino	Antiguo3	15	50
248	falda	morado siena	S	femenino	Estrecho	25	50

## Reporte Pedidos

Bienvenido Usuario: Rosely Valdez Aramayo Ci: 5818447 Fecha: Fecha: Lunes 10-10-3030 Salir

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

HTML Ver en Pdf Guardar en Excel

### REPORTE INGRESO DE PEDIDOS

Usuario: Rosely Valdez Aramayo Fecha: Monday 23 April 2014

CODIGO	PROVEEDOR	USUARIO	FECHA
91	Georgeta Mendez P. Os	Rosely Valdez Aramayo	2012-04-23

Danza	Prenda	Genero	Precio	Cantidad	Costo
diablada	blusa blanco M	femenino	20	50	1000
diablada	botas blanco rojo 22	femenino	20	50	1000
diablada	botas blanco rojo 26	femenino	20	50	1000
diablada	botas blanco rojo 28	femenino	20	50	1000
diablada	botas blanco rojo 30	femenino	20	50	1000
diablada	botas blanco rojo 32	femenino	20	50	1000
diablada	botas naranja M	femenino	20	50	1000
diablada	falda naranja plateado M	femenino	20	50	1000
diablada	capa naranja M	masculino	20	50	1000



## Reporte Proveedores

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5818447 Fecha: Fecha: Lunes 10-10-3030 Salir

**Ballet "Sin Fronteras"**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Html Ver en Pdf Guardar en Excel

### PROVEEDORES

Código	Proveedor	C.I.	Teléfono	Dirección
12	Lorenzo Mamani Velasquez	2942447	6820452	San Roque Umaso de Aguirre 1048
13	Yasmani Mamani Aramayo	1030515	72807821	San Roque Umaso de Aguirre 11049
14	Maria Gomez Casera	5818441	72054558	Campesino Comercio 765
15	Erika Chambi Siquit	8818447	78254179	Ran. Jorge Averina America 7344
16	Rafaela Mendez Rios	2948780	76198435	Quito Centenario avenida chaco
17	Tania Martinez Gorzon	0008302	72879098	La Luna Cudatumbia 7004
18	Pazilda Chavez Arizalde	5818440	72800433	Paroti Carapan 6606
19	Rodrigo Aramayo Mamani	70037970	79264180	San Roque Frey Manuel Minga 3000

## Reporte Prenda por Danza

Bienvenido Usuario: Rosely Valdez Aramayo CI: 5818447 Fecha: Fecha: Lunes 10-10-3030 Salir

**Ballet "Sin Fronteras"**  
Alquiler de Vestuarios en General

Vestimentas Pedidos Administración del sistema Reportes Tiendas Reservas Alquileres

Html Ver en Pdf Guardar en Excel

### Reporte Prendas por Danzas

Usuario: Rosely Valdez Fecha: Mier Apr 23 08:31:

orienta	casual	femenino	masculino	total	disponible	alquilado
		buzo negro M	20	1	0	0
		buzo negro M	20	2	0	0
		huro rojo M	18	3	0	0
		huro azul XS	26	1	0	0
		buzo azul XS	20	2	0	0
		buzo azul XS	15	3	0	0
		chacueta negro M	25	1	0	0
		chacueta negro M	20	2	0	0
		chacueta negro M	15	3	0	0

## **II.1.17 Modelo de Implementación**

### **II.1.17.1 Introducción**

La implementación se empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de códigos fuente, scripts, tablas de la base de datos y otros similares.

### **II.1.17.2 Propósito**

El propósito de la implementación de la arquitectura es imaginar el modelo de implementación y su arquitectura mediante identificación de componentes significativos arquitectónicamente.

### **II.1.17.3 Alcance**

Mostrar cómo distintos subsistemas de software conforman la estructura general del sistema, que se crea en una base de datos centralizada.

## **II.1.18 Modelo de Despliegue**

### **II.1.18.1 Diagrama de Desplazamiento**

### **II.1.18.2 Introducción**

El modelo de despliegue es el que representa o muestra la parte física de la arquitectura del sistema que se está modelando.

### **II.1.18.3 Propósito**

Modelar la arquitectura del sistema.

### I.1.1.1 Diagrama de Despliegue

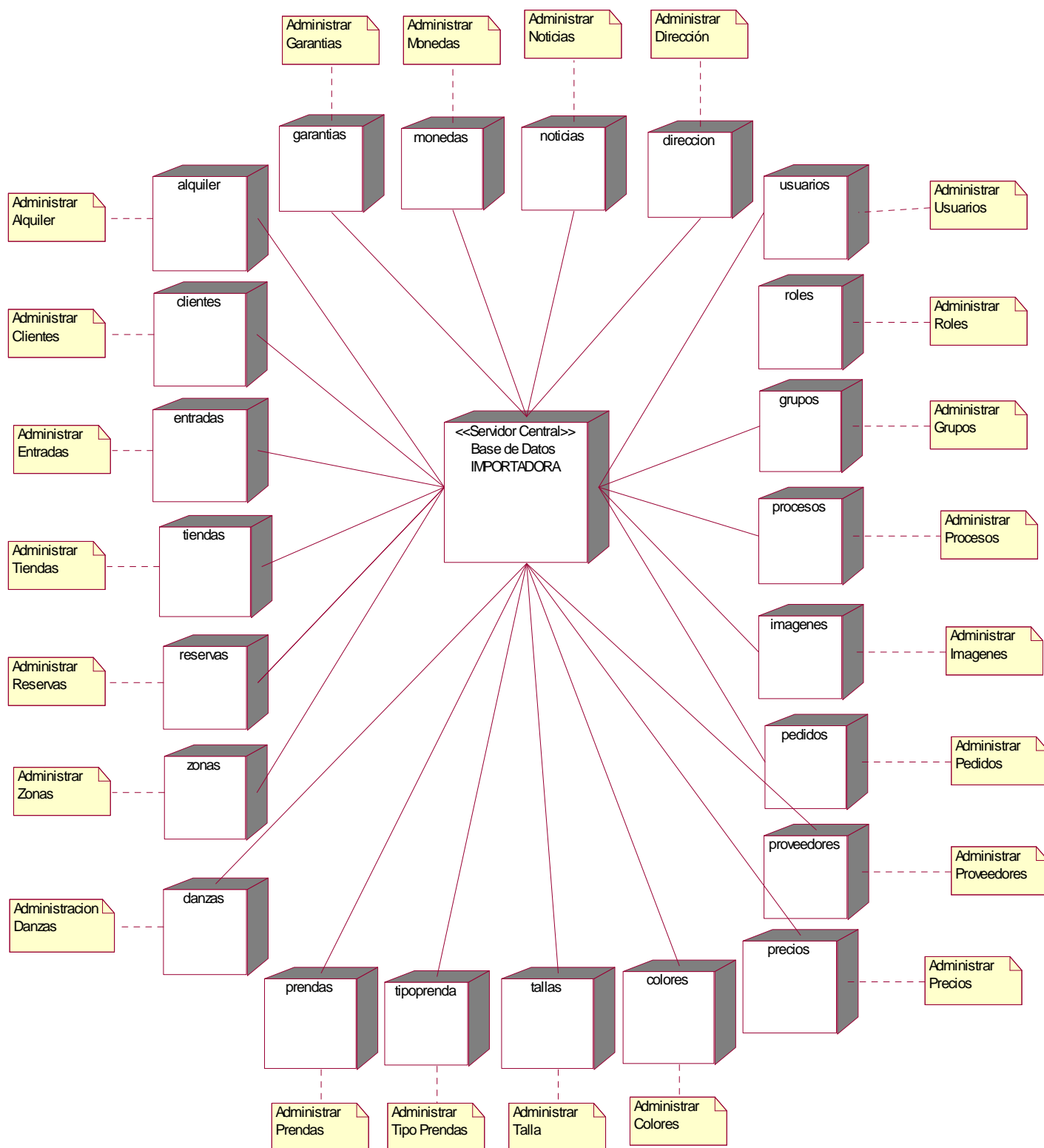


Figura 181: Diagrama de Despliegue

## **II.1.19 Casos de Prueba**

### **II.1.19.1 Introducción**

Las pruebas de software son un artefacto de la disciplina en la metodología RUP la cual se está implementando.

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de las pruebas y los resultados esperados, estos casos de prueba llevarán asociado un procedimiento prueba con las instrucciones para realizar la misma y dependiendo del tipo que se utilice, dicho procedimiento podrá ser automatizado mediante un script de prueba.

### **II.1.19.2 Propósito**

- Verificar el correcto funcionamiento del sistema.
- Identificar posibles mejoras.

### **II.1.19.3 Alcance**

Describe la funcionalidad de cada componente individualmente una vez codificado.

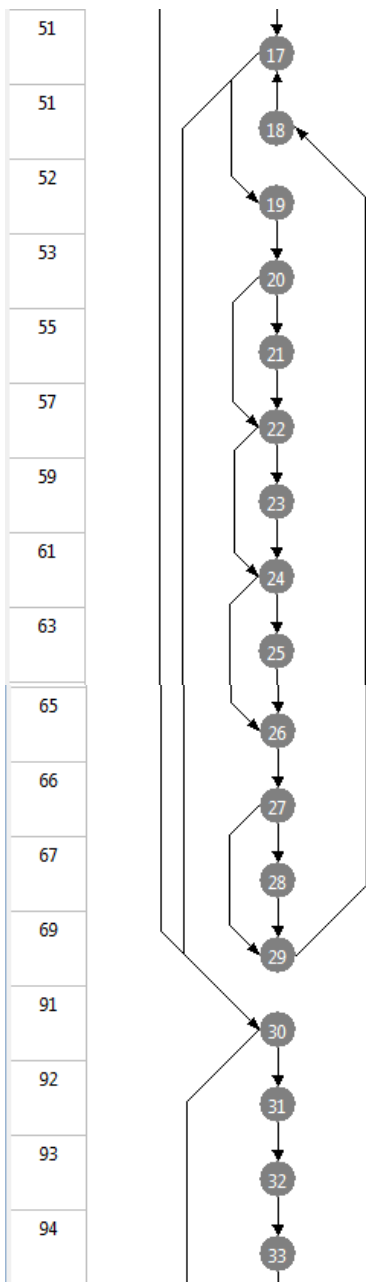
### **II.1.19.4 Pruebas de Caja Blanca**

En programación, se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas. Entre las técnicas usadas se encuentran; la cobertura de caminos pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos.

En los sistemas orientados a objetos, las pruebas de caja blanca pueden aplicarse a los métodos de la clase, pero según varias opiniones, ese esfuerzo debería dedicarse a otro tipo de pruebas más especializadas (un argumento podría ser que los métodos de una clase suelen ser menos complejos que los de una función de programación estructurada). Dentro de las Pruebas de Caja Blanca encontramos las llamadas coberturas (sentencia, decisión, condición y múltiple además de los mencionados caminos ciclomáticos).

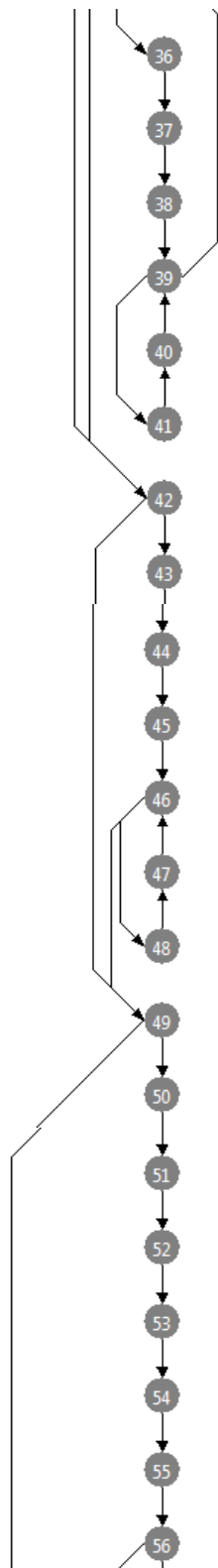
## Pruebas de Caja Blanca Prueba de Caja Blanca: Reportes

28	<pre> graph TD     0((0)) --&gt; 1((1))     1 --&gt; 2((2))     2 --&gt; 3((3))     3 --&gt; 4((4))     4 --&gt; 5((5))     5 --&gt; 6((6))     6 --&gt; 7((7))     7 --&gt; 8((8))     8 --&gt; 9((9))     9 --&gt; 10((10))     10 --&gt; 11((11))     11 --&gt; 12((12))     12 --&gt; 13((13))     13 --&gt; 14((14))     14 --&gt; 15((15))     15 --&gt; 16((16)) </pre>	1	<code>u(lisgru), u(j), u(hasta), u(desde), u(listadanzas), u(linedidoprendas), u(i), u(fecha)</code>	<code>public Template handleRequest(HttpServletRequest request</code>
28		2		<code>public Template handleRequest(HttpServletRequest request</code>
31		3	<code>d(template)</code>	<code>Template template = null;</code>
32		4	<code>d(sesion)</code>	<code>try {HttpSession sesion=request.getSession(true);</code>
33		5	<code>d(ci), r(sesion)</code>	<code>String ci=(String) sesion.getAttribute("ci");</code>
34		6, 8	<code>r(ci)</code>	<code>if(ci==null){</code>
35		7		<code>System.out.println(" sesion invalida");</code>
36		8	<code>d(template)</code>	<code>template = Velocity.getTemplate("templates/Pr esentacionWeb/index.vml");</code>
39		9	<code>d(listien)</code>	<code>ArrayList listien=new manager.managertiendas().listarAct ivo("1");</code>
40		10	<code>r(listien)</code>	<code>context.put("listatiendas",listien);</code>
43		11	<code>d(outFile)</code>	<code>File outFile = new File(request.getParameter("reporte" )+toString()+".vml");</code>
45		12	<code>d(writer), r(outFile)</code>	<code>BufferedWriter writer = new BufferedWriter(new FileWriter(outFile));</code>
47		13	<code>r(writer)</code>	<code>writer.write("&lt;?xml version=\"1.0\" encoding=\"UTF-8\"?&gt;&lt;Sinfrontera es");</code>
48		14, 30		<code>if(request.getParameter("reporte").t oString().equals("ReportePrendas")) {</code>
49		15	<code>d(lisgru)</code>	<code>String [][] lisgru=new manager.managerreportes().listarpr endado();</code>
50		16	<code>r(lisgru)</code>	<code>System.out.println("Tamaño de la lista j"+lisgru.length);</code>
51		17	<code>d(i)</code>	<code>for (int i=0;i&lt;lisgru.length ;i++) {</code>



19, 30	r(lisgru), r(i)	for (int i=0;i<lisgru.length ;i++) {
17	d(i)	for (int i=0;i<lisgru.length ;i++) {
20	d(uso)	String uso="";
21, 22	r(lisgru), r(i)	if(lisgru[i][5].equals("1"))
22	d(uso)	uso="Estreno";
23, 24	r(lisgru), r(i)	if(lisgru[i][5].equals("2"))
24	d(uso)	uso="Normal";
25, 26	r(lisgru), r(i)	if(lisgru[i][5].equals("3"))
26	d(uso)	uso="Antiguo";
27	d(cantidad)	String cantidad="0";
28, 29	r(lisgru), r(i)	if(lisgru[i][7]!=null)
29	r(lisgru), d(cantidad), r(i)	cantidad=lisgru[i][7];
18	r(lisgru), r(lisgru), r(lisgru), r(lisgru), r(lisgru), r(lisgru), r(writer), r(cantidad), r(uso), r(i), r(i), r(i), r(i)	writer.write("<prendas> " +
31, 42		if(request.getParameter("reporte").t oString().equals("Pedidos")){
32	d(lispedidos)	String [][] lispedidos=new manager.managerreportes().listarpe didos();
33	r(lispedidos)	System.out.println("Tamaño de la lista p"+ lispedidos.length);
34	d(i)	for (int i=0;i<lispedidos.length ;i++) {

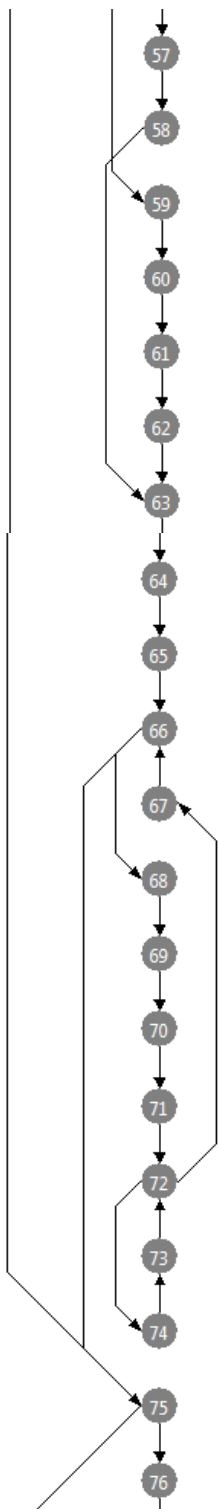
95
102
103
103
103
104
119
120
121
122
122
122
123
135
136
137
138
139
140
141
142



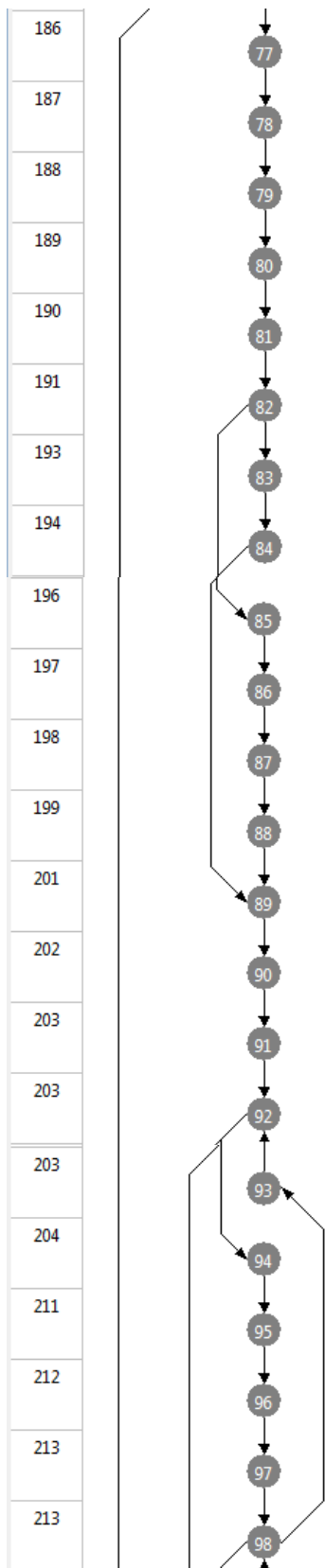
37	<code>r(i), r(i), r(i), r(i), r(i), r(i), r(i), r(i), r(writer), r(lispedidos), r(lispedidos), r(lispedidos), r(lispedidos))</code>	<code>writer.write("&lt; pedidos&gt;" +</code>
38	<code>d(lispedidosprendas), r(i), r(lispedidos)</code>	<code>String [][] lispedidosprendas=new manager.managerreportes().listarpedidosprendas(lispedidos);</code>
39	<code>d(j)</code>	<code>for (int j=0;j&lt;lispedidosprendas.length;i++)</code>
41, 35	<code>r(lispedidosprendas), r(j)</code>	<code>for (int j=0;j&lt;lispedidosprendas.length;i++)</code>
39	<code>d(j)</code>	<code>for (int j=0;j&lt;lispedidosprendas.length;i++)</code>
40	<code>r(lispedidosprendas), r(lispedidosprendas), r(lispedidosprendas)</code>	<code>writer.write("&lt; prendas&gt;" +</code>
43, 49		<code>if(request.getParameter("reporte").toString().equals(" Proveedores")){</code>
44		<code>String [][] lispedidos=new manager.managerreportes().listarproveedores();</code>
45		<code>System.out.println("Tamaño de la lista p"+ lispedidos.length);</code>
46		<code>for (int i=0;i&lt;lispedidos.length;i++)</code>
48, 49		<code>for (int i=0;i&lt;lispedidos.length;i++)</code>
46		<code>for (int i=0;i&lt;lispedidos.length;i++)</code>
47	<code>r(writer)</code>	<code>writer.write("&lt; proveedores&gt;" +</code>
50, 75		<code>if(request.getParameter("reporte").toString().equals("Reportealquiler")){</code>
51	<code>d(date)</code>	<code>Date date=new Date();</code>
52	<code>d(simpleDateFormat)</code>	<code>SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");</code>
53	<code>r(simpleDateFormat)</code>	<code>simpleDateFormat.setLenient(false);</code>
54	<code>d(fecha), r(simpleDateFormat), r(date)</code>	<code>String fecha = simpleDateFormat.format(date);</code>
55	<code>d(lista)</code>	<code>String [][] lista=null;</code>
56	<code>d(tipo)</code>	<code>String tipo=null;</code>
57, 59		<code>if(request.getParameter("tipo")!=null)</code>



144
145
147
148
149
150
152
153
154
154
154
155
163
164
165
165
165
166
184
185

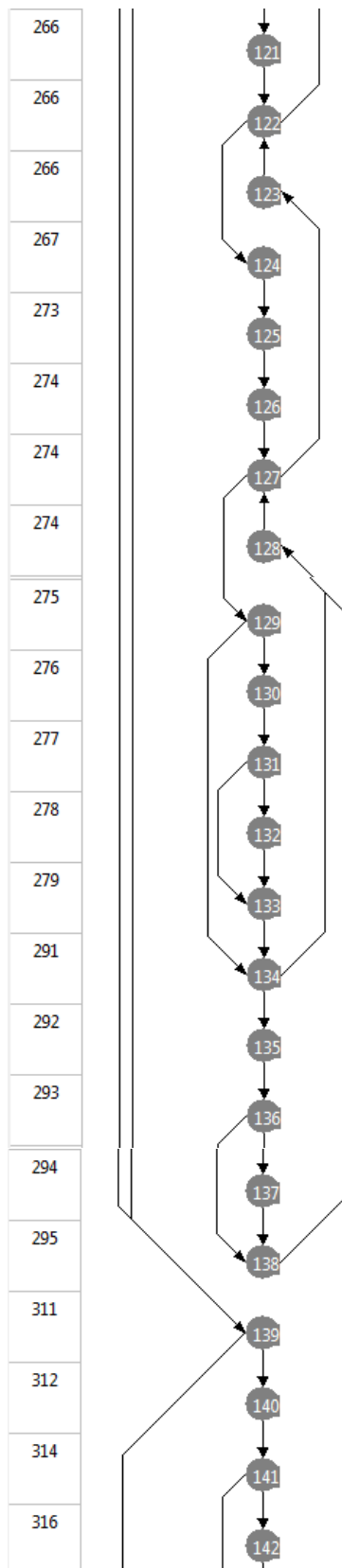


58	r(fecha), r(fecha), d(lista)	lista=new manager.managerreportes().listaralquileres(fecha fecha);
63	d(tipo)	tipo="Reporte diario de alquileres";
60	d(desde)	String desde= request.getParameter("desde");
61	d(hasta)	String hasta= request.getParameter("hasta");
62	r(hasta), r(desde), d(lista)	lista=new manager.managerreportes().listaralquileres(desde hasta);
63	d(tipo)	tipo="Reporte de alquileres";
64	r(tipo)	request.getSession().setAttribute("tipo", tipo);
65	r(lista)	System.out.println("Tamano de la lista p"+ lista.length);
66	d(i)	for (int i=0;i<lista.length;i++) {
68, 75	r(lista), r(i)	for (int i=0;i<lista.length;i++) {
66	d(i)	for (int i=0;i<lista.length;i++) {
69	r(lista), r(lista), r(lista), r(lista), r(lista), r(lista), r(lista), r(writer), r(i), r(i), r(i), r(i), r(i), r(i))	writer.write("<alquiler>" +
70	d(listasub)	String [][] listasub=new manager.managerreportes().listarsubalquileres();
71	r(listasub)	System.out.println("tamano de sublistasub"+ listasub.length);
72	d(j)	for (int j=0;j<listasub.length;j++) {
74, 67	r(j), r(listasub)	for (int j=0;j<listasub.length;j++) {
72	d(j)	for (int j=0;j<listasub.length;j++) {
73	r(j), r(j), r(j), r(j), r(j), r(j), r(j), r(j), r(j), r(j), r(writer), r(listasub), r(listasub), r(listasub), r(listasub), r(listasub))	writer.write("<subalquiler>" +
76, 101		if(request.getParameter("reporte").toString().equals("Reportereserva")){
77		Date date=new Date();



78		SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");
79		simpleDateFormat.setLenient(false);
80		String fecha = simpleDateFormat.format(date);
81		String [][] lista=null;
82		String tipo=null;
83, 85		if(request.getParameter("tipo")!=null)
84		lista=new manager.managerreportes().listarreportes(fecha, fecha);
89		tipo="Reporte diario de reserva";
86		String desde= request.getParameter("desde");
87		String hasta= request.getParameter("hasta");
88		lista=new manager.managerreportes().listarreportes(desde, hasta);
89		tipo="Reporte de reserva";
90		request.getSession().setAttribute("tipo", tipo);
91		System.out.println("Tamano de la lista p"+ lista.length);
92		for (int i=0;i<lista.length;i++) {
94, 101		for (int i=0;i<lista.length;i++) {
92		for (int i=0;i<lista.length;i++) {
95	r(writer)	writer.write("< reserva> " +
96		String [][] listasub=new manager.managerreportes().listarsubreportes();
97		System.out.println("tamano de sublista"+ listasub.length);
98		for (int j=0;j<listasub.length;j++) {
100, 93		for (int j=0;j<listasub.length;j++) {

213		
214		
229		
230		
231		
232		
232		
232		
233		
241		
242		
243		
243		
243		
244		
259		
260		
261		
261		
261		
262		
265		
98		for (int j=0;j<listasub.length ;j++) {
99	r(writer)	writer.write("<subreserva> " +
102, 114		if(request.getParameter("reporte").t oString().equals("NoDevueltos")){
103	d(lista)	String [][] lista=new manager.managerreportes().listarde vdares();
104	r(lista)	System.out.println("Tamano de la lista p"+lista.length);
105		for (int i=0;i<lista.length ;i++) {
107, 114	r(lista)	for (int i=0;i<lista.length ;i++) {
105		for (int i=0;i<lista.length ;i++) {
108	r(writer), r(lista), r(lista), r(lista), r(lista), r(lista), r(lista), r(lista), r(lista), r(lista)	writer.write("<alquiler> " +
109		String [][] listasub=new manager.managerreportes().listarsu balquileres();
110		System.out.println("tamano de sublista"+listasub.length);
111		for (int j=0;j<listasub.length ;j++) {
113, 106		for (int j=0;j<listasub.length ;j++) {
111		for (int j=0;j<listasub.length ;j++) {
112	r(writer)	writer.write("<subalquiler> " +
115, 139		if(request.getParameter("reporte").t oString().equals("StockPrendas")){
116	d(listazona)	String [][] listazona=new manager.managerreportes().listasto ckzonas();
117	d(i)	for (int i=0;i<listazona.length ;i++) {
119, 139	r(listazona), r(i)	for (int i=0;i<listazona.length ;i++) {
117	d(i)	for (int i=0;i<listazona.length ;i++) {
120	r(listazona), r(writer), r(i)	writer.write("<StockPrendas> "+
121	d(listadanzas), r(listazona), r(i)	String [][] listadanzas=new manager.managerreportes().listasto ckdanzas(listazona[i][0].toString());



122	d(j)	for (int j=0;j<listadanzas.length;j++) {
124, 118	r(j), r(listadanzas)	for (int j=0;j<listadanzas.length;j++) {
122	d(j)	for (int j=0;j<listadanzas.length;j++) {
125	r(j), r(j), r(listadanzas), r(listadanzas), r(listazona), r(writer), r(i)	writer.write("< danzas> " +
126	r(j), r(listadanzas), d(listaprendas)	String [][] listaprendas=new manager.managerreportes().listastorendas(listadanzasfil[0].toString()
127	d(k)	for (int k=0;k<listaprendas.length;k++) {
129, 123	r(listaprendas), r(k)	for (int k=0;k<listaprendas.length;k++) {
127	d(k)	for (int k=0;k<listaprendas.length;k++) {
130, 134	r(listaprendas), r(k)	if(listaprendas[k][5].toString().equals("masculino")){
131	d(cantidad)	String cantidad="0";
132, 133	r(listaprendas), r(k)	if(listaprendas[k][8]!=null)
133	d(cantidad), r(listaprendas), r(k)	cantidad=listaprendas[k][8].toString();
134	r(cantidad), r(writer), r(listaprendas), r(listaprendas), r(listaprendas), r(listaprendas)	writer.write("< prendasmasculino> " +
135, 128	r(listaprendas), r(k)	if(listaprendas[k][5].toString().equals("femenino")){
136		String cantidad="0";
137, 138	r(listaprendas), r(k)	if(listaprendas[k][8]!=null)
138	r(listaprendas), r(k)	cantidad=listaprendas[k][8].toString();
128	r(writer), r(listaprendas), r(listaprendas), r(listaprendas), r(listaprendas)	writer.write("< prendasfemenino> "+
140, 163		if(request.getParameter("reporte")!=null){
141	d(listie)	String [][] listie=new manager.managerreportes().listartienda();
142, 143		if(request.getParameter("tienda")!=null)
143	d(listie)	listie=new manager.managerreportes().listartienda(Integer.parseInt(request.getParameter("tienda")));



368	↓ 166	167		context.put("reporte",request.getParameter("reporte").toString());
370	↓ 167	168	d(template)	template = Velocity.getTemplate("templates/reporter/Reporte.vm");
373	↓ 168	169		System.err.println("Exception caught en reportes: " + e.getMessage());
375	↓ 169	170	r(template)	return template;
376	↓ 170		u(lisgru), u(j), u(hasta), u(desde), u(listadanzas), u(linedidoprendar), u(i), u(fecha)	}

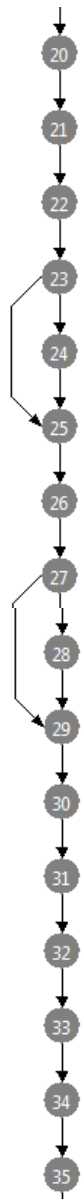
### Prueba de Caja Blanca:Index

17	0	1	u(template), u(sesion)	public Template handleRequest(HttpServletRequest request)
17	↓ 1	2		public Template handleRequest(HttpServletRequest request)
20	↓ 2	3	d(sesion)	HttpSession sesion=request.getSession(true);
21	↓ 3	4	r(sesion)	sesion.invalidate();
23	↓ 4	5	d(template)	Template template = null;
28	↓ 5	6	d(template)	template = Velocity.getTemplate("templates/PrerentacionWeb/index.vm");
34	↓ 6	7		System.err.println("Exception caught: " + e.getMessage());
36	↓ 7	8	r(template)	return template;
34	↓ 6	7		System.err.println("Exception caught: " + e.getMessage());
36	↓ 7	8	r(template)	return template;
37	↓ 8		u(template), u(sesion)	}

### Prueba de Caja Blanca: Prendas Disponibles

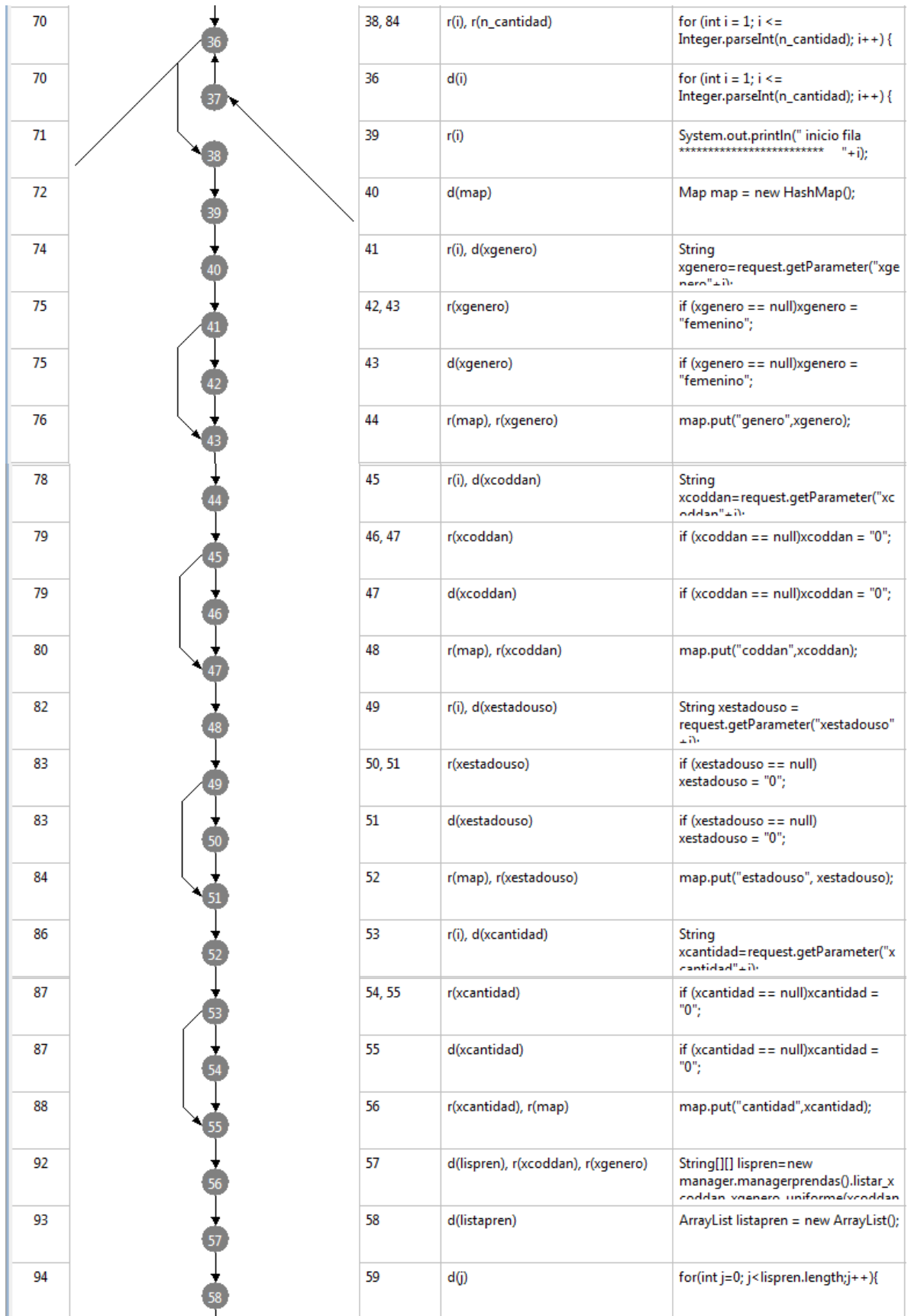
21	<pre> graph TD     0((0)) --&gt; 1((1))     1 --&gt; 2((2))     2 --&gt; 3((3))     3 --&gt; 4((4))     4 --&gt; 5((5))     5 --&gt; 6((6))     6 --&gt; 7((7))     7 --&gt; 8((8))     8 --&gt; 9((9))     8 --&gt; 10((10))     8 --&gt; 11((11))     9 --&gt; 10     10 --&gt; 11     11 --&gt; 12((12))     12 --&gt; 13((13))     13 --&gt; 14((14))     14 --&gt; 15((15))     15 --&gt; 16((16))     16 --&gt; 17((17))     17 --&gt; 18((18))     17 --&gt; 19((19))     18 --&gt; 19 </pre>	1	u(i), u(template), u(lisgar), u(tie), u(xzona), u(liszona), u(ci), u(usu), u(sesion), u(lista), u(lidat)	public Template handleRequest(HttpServletRequest request)
21		2		public Template handleRequest(HttpServletRequest request)
24		3	d(template)	Template template = null;
28		4	d(sesion)	HttpSession sesion=request.getSession(true);
29		5	d(ci), r(sesion)	String ci=(String) sesion.getAttribute("ci");
30		6	d(usu)	domain.Usuarios usu=null;
31		7	d(tie)	domain.Tiendas tie=null;
32		8	d(ci)	ci="5818447";
33		9, 11	r(ci)	if(ci==null){
34		10		System.out.println(" sesion invalida");
35		13	d(template)	template = Velocity.getTemplate("templates/Pr esentacionWeb/index.vm");
39		12	r(ci), d(usu)	usu=new manager.managerusuarios().getusu ario(ci);
40		13	d(tie), r(ci)	tie=new manager.managertiendas().gettiend a(ci);
43		14	r(usu), r(usu), r(usu)	context.put("xnombrecompleto", usu.getNombre()+ " "+usu.getApellido()+" "+usu.getApellido());
44		15	r(tie), r(tie)	context.put("xnombrecompletotien da", tie.getNombre()+ " "+tie.getTelefono()+"");
45		16	r(tie)	context.put("xcodtie",tie.getCodtie( ));
47		17	d(xcodgar)	String xcodgar = request.getParameter("xcodgar");
48		18, 19	r(xcodgar)	if (xcodgar == null)xcodgar = "0";
48		19	d(xcodgar)	if (xcodgar == null)xcodgar = "0";
49		20	r(xcodgar)	context.put("xcodgar", xcodgar);

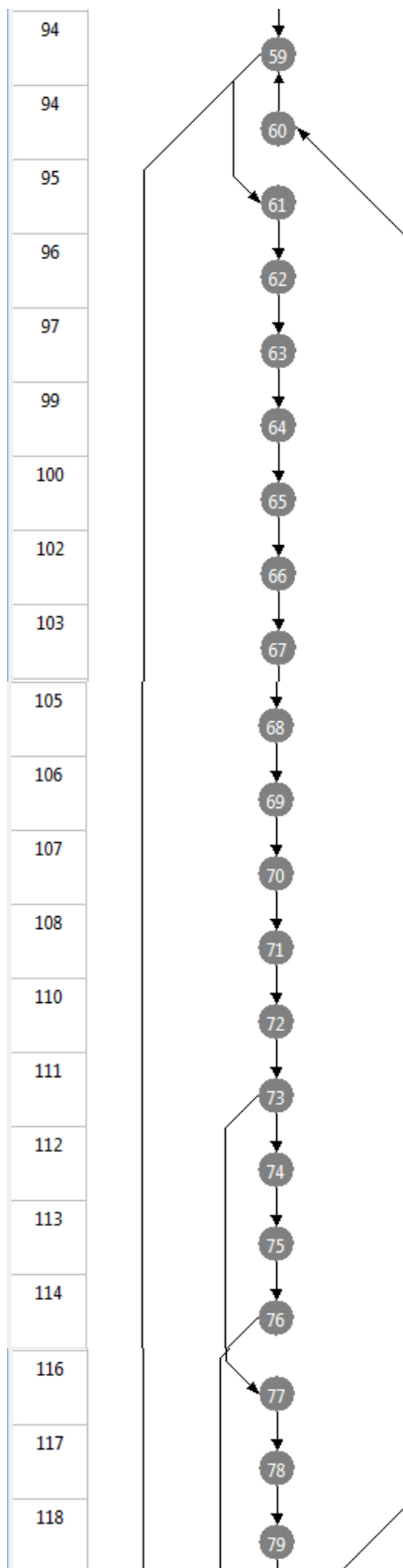
52
53
55
56
56
57
59
60
60
61
63
64
66
67
69
70



21	d(lisgar)	ArrayList lisgar = new manager.manageralquileres().listagarantias("1");
22	r(lisgar)	context.put("listagarantias", lisgar);
23	d(xzona)	String xzona = request.getParameter("xzona");
24, 25	r(xzona)	if (xzona == null)xzona = "ninguna";
25	d(xzona)	if (xzona == null)xzona = "ninguna";
26	r(xzona)	context.put("xzona", xzona);
27	d(n_cantidad)	String n_cantidad = request.getParameter("n_cantidad");
28, 29	r(n_cantidad)	if(n_cantidad==null) n_cantidad="0";
29	d(n_cantidad)	if(n_cantidad==null) n_cantidad="0";
30	r(n_cantidad)	context.put("n_cantidad",n_cantidad);
31	d(liszona)	ArrayList liszona=new manager.managerzonas().listaxesta("1");
32	r(liszona)	context.put("listazonas",liszona);
33	r(xzona), d(lisdan)	ArrayList lisdan = new manager.managerdanzas().listaxzona("1", xzona);
34	r(lisdan)	context.put("listadanzas", lisdan);
35	d(lista)	ArrayList lista = new ArrayList();
36	d(i)	for (int i = 1; i <= Integer.parseInt(n_cantidad); i++) {







61, 81	r(j), r(lispren)	for(int j=0; j<lispren.length;j++){
59	d(j)	for(int j=0; j<lispren.length;j++){
62	d(mappre)	Map mappre = new HashMap();
63	r(j), d(xcodtipopren), r(lispren)	String xcodtipopren = lispren[j][2];
64	r(mappre), r(xcodtipopren)	mappre.put("codtipopren",xcodtipopren);
65	r(j), d(xcodpre), r(lispren)	String xcodpre= lispren[j][0];
66	r(mappre), r(xcodpre)	mappre.put("xcodpre",xcodpre);
67	r(j), d(xcolor), r(lispren)	String xcolor = lispren[j][3];
68	r(xcolor), r(mappre)	mappre.put("color",xcolor);
69	r(j), d(xtalla), r(lispren)	String xtalla= lispren[j][4];
70	r(mappre), r(xtalla)	mappre.put("talla",xtalla);
71	r(tie), r(xcodpre), d(lista_inventario), r(xestadouso)	ArrayList lista_inventario=new manager.managerprendas().listar_inventario_novaEntradaTienda/tia.net
72	r(mappre), r(lista_inventario)	mappre.put("lista_inventario",lista_inventario);
73	r(tie), d(lis_stock), r(xcodpre), r(xestadouso)	String[][] lis_stock=new manager.managerprendas().listar_inventario_xcodprevestadouso/tia.net
74, 77	r(lis_stock)	if(lis_stock.length==0){
75	r(mappre)	mappre.put("precio_unit", "");
76	r(mappre)	mappre.put("cantidad_ped", "");
80	r(mappre)	mappre.put("cantidad_dis", "");
78	r(lis_stock), r(mappre)	mappre.put("precio_unit",lis_stock[0][1]);
79	r(lis_stock), r(mappre)	mappre.put("cantidad_ped",lis_stock[0][2]);
80	r(lis_stock), r(mappre)	mappre.put("cantidad_dis",lis_stock[0][3]);

121		60	<code>r(mappre), r(listapren)</code>	<code>listapren.add(mappre);</code>
123		82	<code>r(map), r(listapren)</code>	<code>map.put("listapren", listapren);</code>
124		83	<code>r(lista), r(map)</code>	<code>lista.add(map);</code>
125		37	<code>r(i)</code>	<code>System.out.println(" fila ***** "+ i);</code>
127		85	<code>r(lista)</code>	<code>context.put("lista", lista);</code>
130		86	<code>d(template)</code>	<code>template = Velocity.getTemplate("templates/PrerentacionWeb/rendardimnonbler</code>
132		87		<code>System.err.println("Exception caught: " + e.getMessage());</code>
134		88	<code>r(template)</code>	<code>return template;</code>
135			<code>u(i), u(template), u(lisgar), u(tie), u(xzona), u(liszona), u(ci), u(usu), u(sesion), u(lista), u(lidan)</code>	<code>}</code>

### Prueba de Caja Blanca: Galería

19		1	<code>u(template), u(sesion), u(lisimg)</code>	<code>public Template handleRequest(HttpServletRequest request</code>
19		2		<code>public Template handleRequest(HttpServletRequest request</code>
22		3	<code>d(sesion)</code>	<code>HttpSession sesion=request.getSession(true);</code>
23		4	<code>r(sesion)</code>	<code>sesion.invalidate();</code>
25		5	<code>d(template)</code>	<code>Template template = null;</code>
29		6	<code>d(lisimg)</code>	<code>ArrayList lisimg=new manager.managerimagenes().listax estado("1");</code>
30		7	<code>r(lisimg)</code>	<code>context.put("listaimagenes", lisimg);</code>
31		8	<code>d(template)</code>	<code>template = Velocity.getTemplate("templates/PrerentacionWeb/galeria.vm");</code>
37		9		<code>System.err.println("Exception caught: " + e.getMessage());</code>
39		10	<code>r(template)</code>	<code>return template;</code>
40			<code>u(template), u(sesion), u(lisimg)</code>	<code>}</code>

## Prueba de Caja Blanca: Noticias

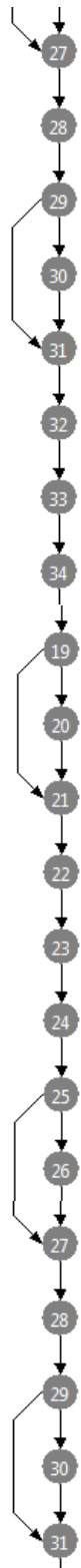
19	<pre> graph TD     0((0)) --&gt; 1((1))     1 --&gt; 2((2))     2 --&gt; 3((3))     3 --&gt; 4((4))     4 --&gt; 5((5))     5 --&gt; 6((6))     6 --&gt; 7((7))     7 --&gt; 8((8))     8 --&gt; 9((9))     9 --&gt; 10((10))     10 --&gt; 11((11))     11 --&gt; 12((12))     12 --&gt; 13((13))     13 --&gt; 14((14))     14 --&gt; 15((15))     15 --&gt; 16((16))     16 --&gt; 17((17))     8 --&gt; 10     10 --&gt; 14           </pre>	1	u(tie), u(ci), u(usu), u(sesion), u(lisnot), u(template)	public Template handleRequest(HttpServletRequest request
19		2		public Template handleRequest(HttpServletRequest request
22		3	d(template)	Template template = null;
26		4	d(sesion)	HttpSession sesion=request.getSession(true);
27		5	d(ci), r(sesion)	String ci=(String) sesion.getAttribute("ci");
28		6	d(usu)	domain.Usuarios usu=null;
29		7	d(tie)	domain.Tiendas tie=null;
30		8	d(ci)	ci="5818447";
31		9, 11	r(ci)	if(ci==null){
32		10		System.out.println(" sesion invalida");
33		14	d(template)	template = Velocity.getTemplate("templates/Pr esentacionWeb/index.vm");
35		12	d(lisnot)	ArrayList lisnot=new manager.managemnoticias().listarxes tado("1");
36		13	r(lisnot)	context.put("listanoticias",lisnot);
37		14	d(template)	template = Velocity.getTemplate("templates/Pr esentacionWeb/noticias.vm");
40		15	d(template)	template = Velocity.getTemplate("templates/Pr esentacionWeb/noticias.vm");
42		16		System.err.println("Exception caught: " + e.getMessage());
44		17	r(template)	return template;
45		u(tie), u(ci), u(usu), u(sesion), u(lisnot), u(template)	}	

## Prueba de Caja Blanca: nuevoalquiler

Método: `handleRequest`

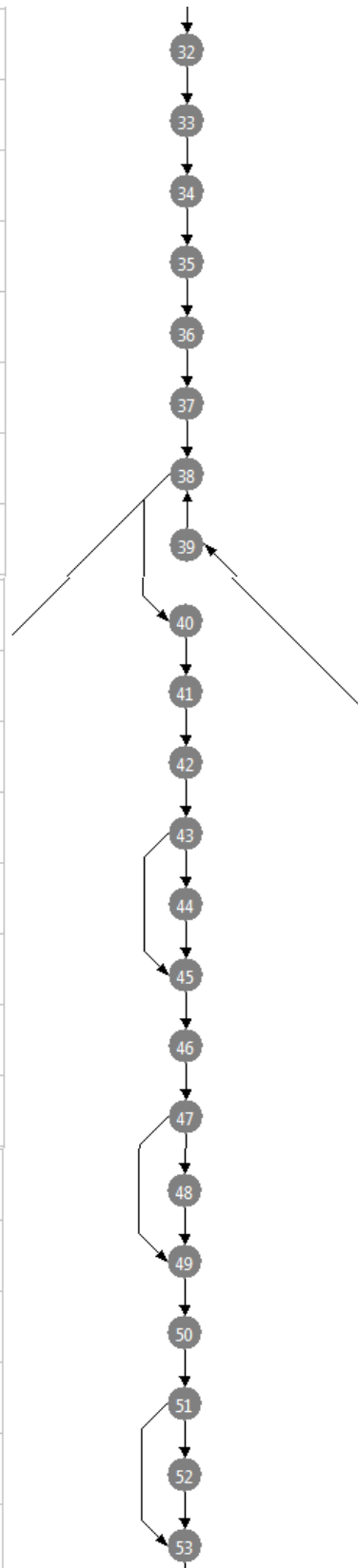
24	<pre> graph TD     0((0)) --&gt; 1((1))     1 --&gt; 2((2))     2 --&gt; 3((3))     3 --&gt; 4((4))     4 --&gt; 5((5))     5 --&gt; 6((6))     6 --&gt; 7((7))     7 --&gt; 8((8))     8 --&gt; 9((9))     8 --&gt; 11((11))     9 --&gt; 10((10))     10 --&gt; 11     11 --&gt; 12((12))     12 --&gt; 13((13))     13 --&gt; 14((14))     14 --&gt; 15((15))     15 --&gt; 16((16))     16 --&gt; 17((17))     17 --&gt; 18((18)) </pre>	1	<code>u(template), u(lis_stock), u(xcolor), u(xpreciounitario), u(liscolor), u(xestadouso), u(xcodtinopren)</code>	<code>public Template handleRequest(HttpServletRequest request)</code>
24		2		<code>public Template handleRequest(HttpServletRequest request)</code>
26		3	<code>d(template)</code>	<code>Template template = null;</code>
30		4	<code>d(sesion)</code>	<code>HttpSession sesion=request.getSession(true);</code>
31		5	<code>d(ci), r(sesion)</code>	<code>String ci=(String) sesion.getAttribute("ci");</code>
32		6	<code>d(usu)</code>	<code>domain.Usuarios usu=null;</code>
33		7	<code>d(tie)</code>	<code>domain.Tiendas tie=null;</code>
34		8	<code>d(ci)</code>	<code>ci="5818447";</code>
35		9, 11	<code>r(ci)</code>	<code>if(ci==null){</code>
36		10		<code>System.out.println(" sesion invalida");</code>
37		13	<code>d(template)</code>	<code>template = Velocity.getTemplate("templates/Pr esentacionWeb/index.vm");</code>
41		12	<code>r(ci), d(usu)</code>	<code>usu=new manager.managerusuarios().getusu ario(ci);</code>
42		13	<code>d(tie), r(ci)</code>	<code>tie=new manager.managertiendas().gettiend ad(ci);</code>
45		14	<code>r(usu), r(usu), r(usu)</code>	<code>context.put("xnombrecompleto", usu.getNombre()+ " "+usu.getApellido());</code>
46		15	<code>r(tie)</code>	<code>context.put("xnombrecompletotien da", tie.getNombre()+ " ");</code>
47		16	<code>r(tie)</code>	<code>context.put("xcodtie",tie.getCodtie( ));</code>
48		17	<code>d(liscli)</code>	<code>ArrayList liscli=new manager.managerclientes().listaxes tacli("1");</code>
49		18	<code>r(liscli)</code>	<code>context.put("listaclientes",liscli);</code>
51		19	<code>d(xcodgar)</code>	<code>String xcodgar = request.getParameter("xcodgar");</code>

61
63
64
64
65
67
68
70
52
52
53
56
57
59
60
60
61
63
64
64
65



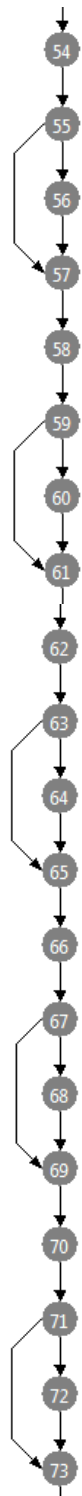
28	r(xzona)	context.put("xzona", xzona);
29	d(n_cantidad)	String n_cantidad = request.getParameter("n_cantidad");
30, 31	r(n_cantidad)	if(n_cantidad==null) n_cantidad="0";
31	d(n_cantidad)	if(n_cantidad==null) n_cantidad="0";
32	r(n_cantidad)	context.put("n_cantidad",n_cantidad);
33	d(liszona)	ArrayList lizzona=new manager.managerzonas().listaxestada("1");
34	r(liszona)	context.put("listazonas",lizzona);
35	d(lisdan), r(xzona)	ArrayList lisdan = new manager.managerdanzas().listaxzona("1", xzona);
20, 21	r(xcodgar)	if (xcodgar == null)xcodgar = "0";
21	d(xcodgar)	if (xcodgar == null)xcodgar = "0";
22	r(xcodgar)	context.put("xcodgar", xcodgar);
23	d(lisgar)	ArrayList lisgar = new manager.manageralquileres().listagarantias("1");
24	r(lisgar)	context.put("listagarantias", lisgar);
25	d(xzona)	String xzona = request.getParameter("xzona");
26, 27	r(xzona)	if (xzona == null)xzona = "ninguna";
27	d(xzona)	if (xzona == null)xzona = "ninguna";
28	r(xzona)	context.put("xzona", xzona);
29	d(n_cantidad)	String n_cantidad = request.getParameter("n_cantidad");
30, 31	r(n_cantidad)	if(n_cantidad==null) n_cantidad="0";
31	d(n_cantidad)	if(n_cantidad==null) n_cantidad="0";
32	r(n_cantidad)	context.put("n_cantidad",n_cantidad);

67
68
70
71
73
74
74
74
75
76
78
79
79
80
82
83
83
84
86
87
87
88



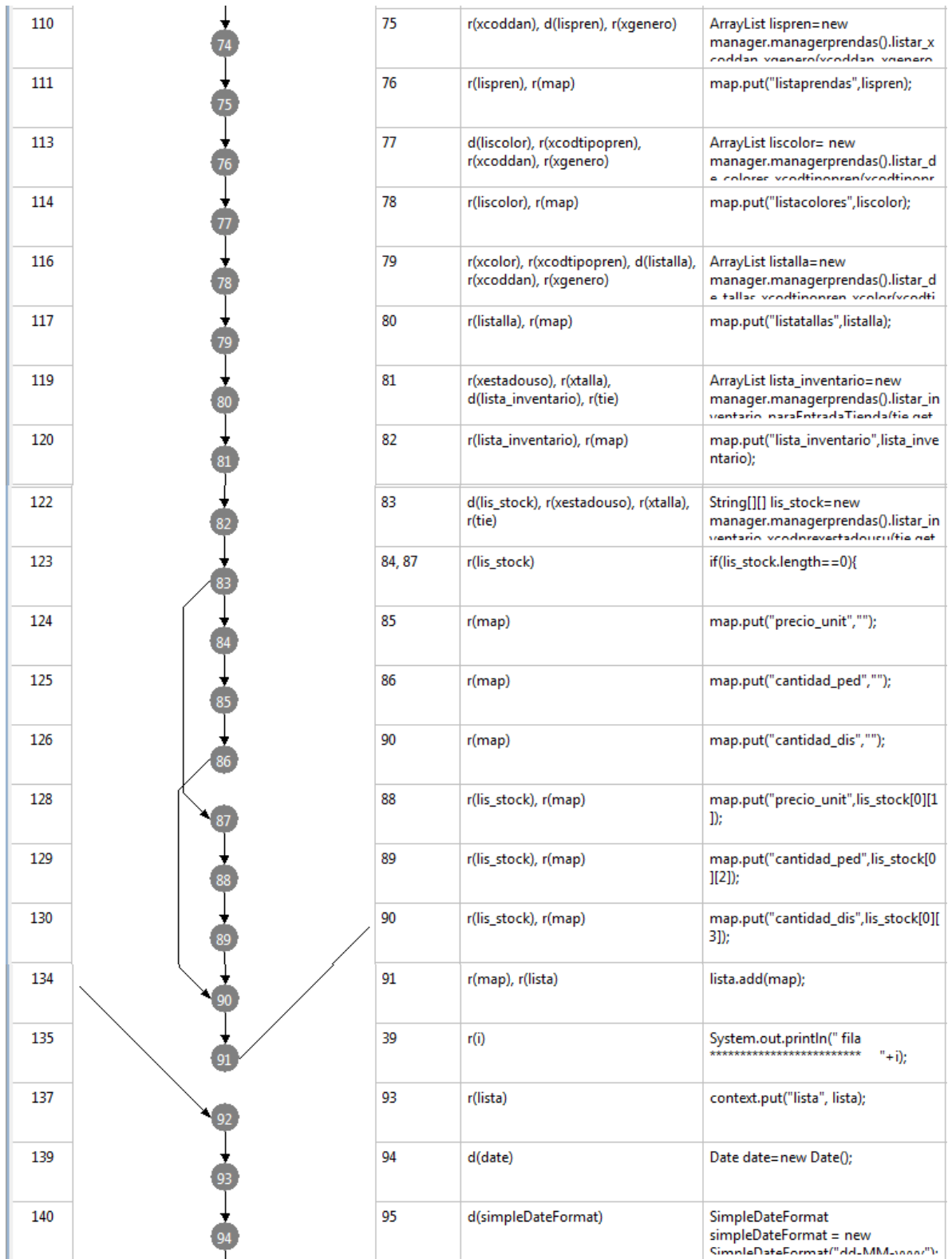
33	d(lizzona)	ArrayList lizzona=new manager.managerzonas().listaxesta
34	r(lizzona)	context.put("listazonas",lizzona);
35	d(lisdan), r(xzona)	ArrayList lisdan = new manager.managerdanzas().listanxona
36	r(lisdan)	context.put("listadanzas",lisdan);
37	d(lista)	ArrayList lista = new ArrayList();
38	d(i)	for (int i = 1; i <= Integer.parseInt(n_cantidad); i++) {
40, 92	r(i), r(n_cantidad)	for (int i = 1; i <= Integer.parseInt(n_cantidad); i++) {
38	d(i)	for (int i = 1; i <= Integer.parseInt(n_cantidad); i++) {
41	r(i)	System.out.println(" inicio fila ***** "+i);
42	d(map)	Map map = new HashMap();
43	d(xgenero), r(i)	String xgenero=request.getParameter("xgenero"+i);
44, 45	r(xgenero)	if (xgenero == null)xgenero = "femenino";
45	d(xgenero)	if (xgenero == null)xgenero = "femenino";
46	r(map), r(xgenero)	map.put("genero",xgenero);
47	d(xcoddan), r(i)	String xcoddan=request.getParameter("xcoddan"+i);
48, 49	r(xcoddan)	if (xcoddan == null)xcoddan = "0";
49	d(xcoddan)	if (xcoddan == null)xcoddan = "0";
50	r(xcoddan), r(map)	map.put("coddan",xcoddan);
51	d(xcodtipopren), r(i)	String xcodtipopren=request.getParameter("xcodtipopren"+i);
52, 53	r(xcodtipopren)	if (xcodtipopren == null)xcodtipopren = "0";
53	d(xcodtipopren)	if (xcodtipopren == null)xcodtipopren = "0";
54	r(xcodtipopren), r(map)	map.put("codtipopren",xcodtipopren);

90
91
91
92
94
95
95
96
98
99
99
100
102
103
103
104
106
107
107
108



55	d(xcolor), r(i)	String xcolor=request.getParameter("xcolor");
56, 57	r(xcolor)	if (xcolor == null)xcolor = "0";
57	d(xcolor)	if (xcolor == null)xcolor = "0";
58	r(xcolor), r(map)	map.put("color",xcolor);
59	d(xtalla), r(i)	String xtalla=request.getParameter("xtalla");
60, 61	r(xtalla)	if (xtalla == null)xtalla = "0";
61	d(xtalla)	if (xtalla == null)xtalla = "0";
62	r(xtalla), r(map)	map.put("talla",xtalla);
63	d(xestadouso), r(i)	String xestadouso = request.getParameter("xestadouso");
64, 65	r(xestadouso)	if (xestadouso == null) xestadouso = "0";
65	d(xestadouso)	if (xestadouso == null) xestadouso = "0";
66	r(xestadouso), r(map)	map.put("estadouso", xestadouso);
67	d(xcantidad), r(i)	String xcantidad=request.getParameter("xcantidad");
68, 69	r(xcantidad)	if (xcantidad == null)xcantidad = "0";
69	d(xcantidad)	if (xcantidad == null)xcantidad = "0";
70	r(xcantidad), r(map)	map.put("cantidad",Integer.parseInt (xcantidad));
71	d(xpreciounitario), r(i)	String xpreciounitario=request.getParamet er("xpreciounitario");
72, 73	r(xpreciounitario)	if (xpreciounitario == null)xpreciounitario = "0";
73	d(xpreciounitario)	if (xpreciounitario == null)xpreciounitario = "0";
74	r(xpreciounitario), r(map)	map.put("preciounitario",xprecioun itario);





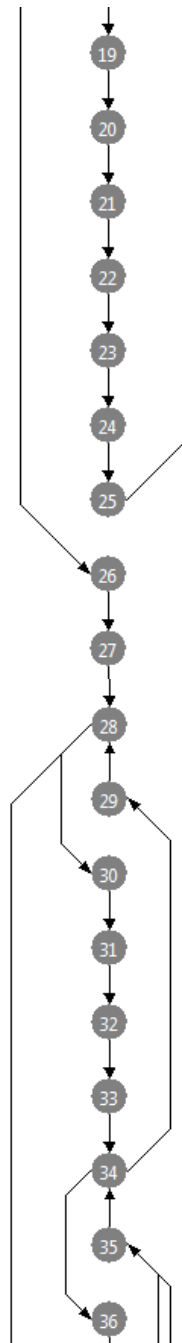
141	↓ 95	96	r(simpleDateFormat)	simpleDateFormat.setLenient(false);
142	↓ 96	97	d(fecha), r(date), r(simpleDateFormat)	String fecha = simpleDateFormat.format(date);
143	↓ 97	98	r(fecha)	context.put("hoy", fecha);
145	↓ 98	99	d(template)	template = Velocity
148	↓ 99	100		System.err.println("Exception caught: " + e.getMessage());
150	↓ 100	101	r(template)	return template;
151	↓ 101		u(template), u(lis_stock), u(xcolor), u(xpreciounitario), u(liscolor), u(xestadouso), u(xcontinuar)	}

## Prueba de Caja Blanca:base\_alquilerdevolucion

### Metodo:

21		1	<code>u(codinv), u(k), u(n), u(respuesta), u(y), u(x), u(k), u(codgar), u(template), u(cantidad)</code>	<code>public Template handleRequest(HttpServletRequest request</code>
21		2		<code>public Template handleRequest(HttpServletRequest request</code>
24		3	<code>d(template)</code>	<code>Template template = null;</code>
27		4	<code>d(x)</code>	<code>domain.Alquiler x=new domain.Alquiler();</code>
29		5	<code>r(x)</code>	<code>x.setCodalq(request.getParameter("x codalq"));</code>
30		6	<code>r(x)</code>	<code>x.setCi("5818447");</code>
31		7	<code>r(x)</code>	<code>x.setCodcli(request.getParameter("x codcli"));</code>
32		8	<code>r(x)</code>	<code>x.setCodtie(request.getParameter("x codtie"));</code>
33		9	<code>r(x)</code>	<code>x.setFechaalq(new Date());</code>
34		10	<code>r(x)</code>	<code>x.setFechadev(new Date());</code>
36		11	<code>d(respuesta), r(x)</code>	<code>String respuesta=new manager.manageralquileres().Adici onar();</code>
37		12	<code>r(respuesta)</code>	<code>context.put("respuesta",respuesta);</code>
41		13	<code>d(y)</code>	<code>domain.Detallealquiler y=new domain.Detallealquiler();</code>
42		14	<code>r(y)</code>	<code>y.setCodalq( new manager.manageralquileres().getult imperial());</code>
44		15	<code>d(k)</code>	<code>for(int k=1;k&lt;= Integer.parseInt(request.getParamet er("no_codgar"));k++)</code>
44		17, 26	<code>r(k)</code>	<code>for(int k=1;k&lt;= Integer.parseInt(request.getParamet er("no_codgar"));k++)</code>
44		15	<code>d(k)</code>	<code>for(int k=1;k&lt;= Integer.parseInt(request.getParamet er("no_codgar"));k++)</code>
45		18	<code>r(k), d(codgar)</code>	<code>String codgar = request.getParameter("codgar"+k);</code>
46		19, 16	<code>r(codgar)</code>	<code>if(codgar!=null){</code>

47
48
49
50
51
52
53
59
61
61
61
62
63
64
65
65
65
66



20	<code>r(k), d(valor)</code>	<code>String valor=request.getParameter("valor")</code>
21	<code>r(y), r(codgar), r(valor)</code>	<code>System.out.println(y.getCodalq()+" "+codgar+" "+valor);</code>
22	<code>d(w)</code>	<code>domain.AlquilerGarante w=new domain.AlquilerGarante();</code>
23	<code>r(y), r(w)</code>	<code>w.setCodalq(y.getCodalq());</code>
24	<code>r(codgar), r(w)</code>	<code>w.setCodgar(codgar);</code>
25	<code>r(w), r(valor)</code>	<code>w.setValor(valor);</code>
16	<code>r(w)</code>	<code>new manager.managergarantias().Adicio nar(w);</code>
27	<code>d(n)</code>	<code>int n=Integer.parseInt(request.getPara meter("n_cantidad"));</code>
28	<code>d(i)</code>	<code>for(int i=1;i&lt;= n;i++){</code>
30, 42	<code>r(n), r(i)</code>	<code>for(int i=1;i&lt;= n;i++){</code>
28	<code>d(i)</code>	<code>for(int i=1;i&lt;= n;i++){</code>
31	<code>d(xcantidad), r(i)</code>	<code>String xcantidad = request.getParameter("xcantidad_di c")+String.valueOf(i);</code>
32	<code>d(xprecio_unit), r(i)</code>	<code>String xprecio_unit = request.getParameter("xprecio_unit ")+String.valueOf(i);</code>
33	<code>d(xestadouso), r(i)</code>	<code>String xestadouso = request.getParameter("xestadouso" ")+String.valueOf(i);</code>
34	<code>d(k)</code>	<code>for(int k=1;k&lt;= Integer.parseInt(xcantidad);k++){</code>
36, 29	<code>r(k), r(xcantidad)</code>	<code>for(int k=1;k&lt;= Integer.parseInt(xcantidad);k++){</code>
34	<code>d(k)</code>	<code>for(int k=1;k&lt;= Integer.parseInt(xcantidad);k++){</code>
37	<code>d(codinv), r(k), r(i)</code>	<code>String codinv = request.getParameter("codinv_" + i + " "+String.valueOf(k));</code>

67			
68			
69			
70			
71			
77			
80			
82			
83			

37			
38			
39			
40			
41			
42			
43			
44			
45			

38, 35	r(codinv)	if(codinv!=null){
39	r(codinv), r(y)	y.setCodinv(codinv);
40	r(y), r(xprecio_unit)	y.setPrecio_unit(xprecio_unit);
41	r(y), r(xestadouso)	y.setEstadouso(xestadouso);
35	d(respuesta), r(y)	respuesta=new manager.managerdetallealquileres() Adicionar(y);
43	d(template)	template = Velocity.getTemplate("templates/al quiler/hacer_nuevoalquiler.vm");
44		System.err.println("Exception caught: " + e.getMessage());
45	r(template)	return template;
	u(codinv), u(k), u(n), u(respuesta), u(y), u(x), u(k), u(codgar), u(template), u(cantidad)	}

## Prueba de Caja Blanca: alquilerdevolucion

Método: `handleRequest`

21		0	1	<code>u(i), u(template), u(ci), u(sesion), u(map), u(lista_inventario), u(licencia), u(a), u(lista), u(x)</code>	<code>public Template handleRequest(HttpServletRequest request)</code>
21		1	2		<code>public Template handleRequest(HttpServletRequest request)</code>
24		2	3	<code>d(sesion)</code>	<code>HttpSession sesion=request.getSession(true);</code>
25		3	4	<code>d(ci), r(sesion)</code>	<code>String ci=(String) sesion.getAttribute("ci");</code>
26		4	5	<code>d(template)</code>	<code>Template template = null;</code>
28		5	6, 8	<code>r(ci)</code>	<code>if(ci==null){</code>
29		6	7		<code>System.out.println(" sesion invalida");</code>
30		7	38	<code>d(template)</code>	<code>template = Velocity.getTemplate("templates/PrerentacionWeb/index.vm");</code>
33		8	9		<code>System.out.println("Ilego al controlador");</code>
36		9	10	<code>d(a)</code>	<code>String a=request.getParameter("xcodalq");</code>
37		10	11	<code>r(a), d(x)</code>	<code>domain.Alquiler x=new manager.manageralquileres().getalquiler(a);</code>
38		11	12	<code>r(x)</code>	<code>context.put("codalq",x.getCodalq());</code>
39		12	13	<code>r(x)</code>	<code>context.put("ci",x.getCi());</code>
41		13	14	<code>r(x)</code>	<code>context.put("codcli",x.getCodcli());</code>
42		14	15	<code>r(x)</code>	<code>context.put("codtie",x.getCodtie());</code>
43		15	16	<code>r(x)</code>	<code>context.put("estado",x.getEstado());</code>
44		16	17	<code>r(x)</code>	<code>context.put("fechaalq",x.getFechaalq());</code>
45		17	18	<code>r(x)</code>	<code>context.put("fechadev",x.getFechadev());</code>

50		19	d(lispren), r(a)	String [][] lispren=new manager.managerprendas().listarPrendasDetalles(a);
52		20	d(lista)	ArrayList lista = new ArrayList();
53		21	d(i)	for (int i =0; i < lispren.length; i++) {
53		23, 36	r(i), r(lispren)	for (int i =0; i < lispren.length; i++) {
53		21	d(i)	for (int i =0; i < lispren.length; i++) {
54		24	r(i)	System.out.println(" inicio fila ***** "+i);
55		25	d(map)	Map map = new HashMap();
56		26	r(i), r(map), r(lispren)	map.put("prenda",lispren[i][1]);
57		27	r(i), r(map), r(lispren)	map.put("color",lispren[i][2]);
58		28	r(i), r(map), r(lispren)	map.put("talla",lispren[i][3]);
60		29	r(i), r(map), r(lispren)	map.put("genero",lispren[i][4]);
61		30	r(i), r(map), r(lispren)	map.put("nombredan",lispren[i][5]);
63		31	r(i), d(lista_inventario), r(lispren), r(a)	ArrayList lista_inventario=new manager.managerprendas().listarPrendasDetalles(a);
64		32	r(map), r(lista_inventario)	map.put("lista_inventario",lista_inventario);
65		33	r(i)	System.out.println(" Enviando Cantidad ***** "+i);
66		34	r(map), r(lista_inventario)	map.put("cantidad",lista_inventario.size());
68		35	r(map), r(lista)	lista.add(map);
69		22	r(i)	System.out.println(" fila ***** "+i);
71		37	r(lista)	context.put("lista", lista);
82		38	d(template)	template = Velocity.getTemplate("templates/alquiler/alquilerdeolucion.vm");
86		39		System.err.println("Exception caught: " + e.getMessage());
88		40	r(template)	return template;
89			u(i), u(template), u(ci), u(sesion), u(map), u(lista_inventario), u(lispren), u(a), u(lista), u(i)	}

## **II.2 Componente 2: Capacitación al Personal**

### **II.2.1 Introducción**

La tecnología de la información y Comunicación (TIC), ayuda a mejorar la disponibilidad de la información logrando una mayor eficiencia en el alquiler, fortaleciendo las condiciones de alquiler de trajes típicos “Ballet Sin Fronteras”.

Luego de realizar una entrevista a los trabajadores de la empresa percatamos q los mismos cuentan con poca información necesaria, ni con el equipamiento adecuado, con la implementación de las TICs se podrá disponer de información en tiempo real, además de contar con mejores condiciones gracias a las TICs y el sistema computarizado orientado a la web que ayuda en sobremanera a mejorar la forma en que los procesos de la institución son manejados, es fundamental ejecutar este componente, pues de ello significa una mejor calidad de los usuarios finales en la institución.

Este componente “Capacitación al Personal” se refiere a la preparación impartida al personal de la empresa sobre el manejo del sitio web.

Resulta importante ya que el personal de la tienda no cuenta con conocimientos amplios sobre el manejo del Sistema Web, y les será de mucha utilidad en el desempeño de sus funciones.

### **II.2.2 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.

### **II.2.3 Objetivo General**

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



## **II.2.4 Material a Impartir**

- Manual de Usuario

## **II.2.5 Metodología**

- La capacitación se la realizara mediante cursos dictados por el responsable del proyecto en los ambientes de la tienda.
- Para la capacitación se emplearán manuales de usuarios que contiene toda la información detallada sobre el manejo del sitio Web y llevarnos un registro del personal que asiste a la capacitación para que al finalizar el curso sean acreedores a un certificado avalado por el Departamento de Informática y Sistemas de la Universidad Autónoma Juan Misael Saracho.

## **II.2.6 Proceso de enseñanza aprendizaje y metodologías de capacitación**

### **El proceso de enseñanza**

#### **II.2.6.1 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.

### **II.2.6.2 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.

### **II.2.6.3 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.

### **II.2.6.4 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.

## **II.2.6.5 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**

1. El instructor diseña o selecciona la experiencia apropiada para enfatizar el tema.
2. El instructor prepara los materiales o instrumentos necesarios para la experiencia.
3. El instructor explica al grupo la mecánica de la experiencia estructurada.
4. El instructor conduce al grupo a lo largo de la experiencia.
5. Al finalizar la experiencia, solicita al grupo los comentarios y reflexiones sobre el tema.
6. El grupo destaca lo aprendido en la experiencia.
7. 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.

### **II.2.6.6 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

- Mejorar el servicio de alquiler, automatizando los procesos del mismo.
- Se concluye, con el presente proyecto, que la aplicación de las Tecnologías de la Información es indispensable para la optimización de los procesos en cualquier organización.
- Se logró brindar a los usuarios del sistema la información oportuna de alquileres, reservas, prendas, danzas, tiendas, clientes, pedidos, proveedores.
- Con los reportes generados cuenta con la información necesaria y oportuna.
- Cabe señalar que algunos procesos que no fueron requeridos por el propietario de la tienda “**Ballet Sin Fronteras**”, fueron incluidos para un mejor desempeño del sistema.
- En la capacitación se han utilizado los manuales y demostraciones del sistema la misma que hicieron más fácil el aprendizaje.
- Uno de los aspectos más sobresalientes del sistema es que permite controlar los alquileres a los clientes y controlar de forma segura los procesos y evitar el doble registro en el momento de adicionar, modificar, borrar lógicamente y asignación.

### III.2 Recomendaciones

Un Sistema Web es accesible a todo público que navega por la red por lo tanto es necesario que los usuarios que accedan al sitio estén conscientes de que la información que puedan brindar servirá de ayuda a otros usuarios y que de igual manera ellos se benefician con la información que buscan dentro del sistema.

Se realizan las siguientes recomendaciones:

- El proyecto de administración de alquileres de prendas no contempla la parte contable de la empresa, se recomienda implementarlo.
- Realizar una concientización a la sociedad y particular a los usuarios de visitar el sitio como medio de información de las actividades que se realiza por la empresa “Ballet Sin Fronteras”.
- Profundizar más aún en el desarrollo de otros requerimientos por la empresa “Ballet Sin Fronteras”.
- Se aconseja manejar los navegadores como ser: Mozilla Firefox, Google Chrome, Opera.
- Se recomienda ampliar las reservas, que cualquier tipo de clientes pueda realizar su reserva.
- Se recomienda hacer un estudio más profundo en lo que respecta al cálculo de tiempos a la hora de realizar el cronograma del proyecto.
- Se recomienda proveer algunas dificultades con la cooperación de las instituciones participantes para evitar el retraso en las diferentes etapas del proyecto.



- Se recomienda realizar campañas publicitarias con información sobre la empresa a través de medios televisivos y volantes.
- Se recomienda poder ampliar el sistema para controlar la información de la empresa en las sucursales interprovinciales.

## Bibliografía

- Lenguaje Unificado de Modelado  
<http://es.wikipedia.org/wiki/UML>
- Proceso Unificado de Rational  
<http://es.wikipedia.org/wiki/RUP>
- Sistemas de Información AdministrativosI. Programa .  
Universidad Arturo Prat, Departamento de Auditoria y Sistemas  
de Información Iquique – Chile.
- <http://www.unap.cl/-setcheve/siiqq/page34.html>
- Diccionario Informático
- Concepto y Características del Sistema de Gestión de  
Información [http://www.scribd.com/doc/12227873/Concepto-y-  
Caracteristicas-Del-Sistemas-de-Gestion-de-Informacion](http://www.scribd.com/doc/12227873/Concepto-y-<br/>Caracteristicas-Del-Sistemas-de-Gestion-de-Informacion).
- MIDEPLAN, Programas Públicos.Marco Lógico: Instrumento  
Para La Formulación de Programas.
- Trabajo de grado: APLICACION DE LA TECNOLOGÍA DE  
INFORMACIÓN PARA EL CONTROL DE INVENTARIOS  
Y VENTAS EN LA EMPRESA EXIMPORTTARIJA LTDA
- [[http://www.calidaddelsoftware.com/modules.php?name=News  
&file=article&sid=66](http://www.calidaddelsoftware.com/modules.php?name=News<br/>&file=article&sid=66)].
- IBM Press Project Management with the IBM Rational Unified  
Process Jul.2006
- Addison Wesley Managing Iterative Software development  
Projects Jun 2006 eBook.
- <http://www.tecnyce.com.mx/ti/cursosN.php?ids=641;>
- [http://html.rincondelvago.com/analisis-y-diseno-de-  
sistemas\\_1.html](http://html.rincondelvago.com/analisis-y-diseno-de-<br/>sistemas_1.html)
- [http://es.wikibooks.org/wiki/Inform%C3%A1tica\\_Educacional/  
Seguimiento\\_y\\_Control\\_del\\_Proyecto](http://es.wikibooks.org/wiki/Inform%C3%A1tica_Educacional/<br/>Seguimiento_y_Control_del_Proyecto)
- <http://www.desarrolloweb.com> > Manuales > Manual de Java

- [http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_orientada\\_a\\_objetos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos)
- [http://www.ciberaula.com/articulo/tecnologia\\_orientada\\_objetos/](http://www.ciberaula.com/articulo/tecnologia_orientada_objetos/)
- <http://www.alegsa.com.ar/Dic/plataforma.php>
- [http://es.wikipedia.org/wiki/Sistema\\_operativo](http://es.wikipedia.org/wiki/Sistema_operativo)
- <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r33018.PDF>
- <http://maromasdigitales.net/2011/05/generando-un-modelo-de-clases-en-ef-a-partir-de-una-base-datos-usando-codigo-primero/>
- <http://www.monografias.com/trabajos30/estrategias-seguridad-webmasters-I-II/estrategias-seguridad-webmasters-I-II.shtml>
- <http://www.mexcompany.com/dominios/significados.html>
- <http://www.monografias.com/trabajos17/extensiones/extensiones.shtml>
- <http://www.serviweb.net/dominios/tiposdedominios.php>
- <http://www.softonic.com/s/crear-juegos-rol>
- <http://www.eumed.net/rev/tlatemoani/01/jmil.htm>
- <http://www.softonic.com/s/crear-juegos-rol>
- <http://www.vsantivirus.com/sbam-extensiones.htm>
- <http://www.nuevafase.com/informatica/extensiones/>