

CAPITULO I: EL PROYECTO

I. Proyecto

I. 1 IDENTIFICACIÓN DEL PROYECTO

1.1.1 Título del Proyecto: Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites para Ordenamiento Territorial **“G.I.P.S.E.T”**.

1.1.2 Carrera: Ingeniería Informática

1.1.3 Facultad: Ciencias y Tecnología

1.1.4 Duración del Proyecto: 8 Meses

1.1.5 Área/línea de investigación priorizada: Tecnologías de la información y comunicación / Desarrollo de sistemas y software.

1. 2. Personal Vinculado al Proyecto

1.2.1 Director de Proyecto

Jallaza Apellido Paterno	Garnica Apellido Materno	Irka María Nombre	5044883 C.I.
Ingeniería Informática Carrera		Ciencias y Tecnología Facultad:	
66-38524 Teléf. Domicilio	----- Celular	akri_elisha_5@hotmail.co m Correo electrónico	Firma

Tabla No. 1: Director de Proyecto

1.2 .2 Participantes equipo de trabajo (señale categoría: investigador, asesor, etc.)

Categoría	Nombres y Apellidos	Carrera/Profesión	C.I.	Firma
Investigador(1)	Univ. Irka María Jallaza Garnica	Ingeniería Informática	5044883	
Asesor	Ing. Silvana Paz	Ingeniera	1860481	

		Informática		
Asesor	Lic. Efraín Torreón	Lic. En Informática	1337531	
Asesor	Ing. Omar Choque	Ing. Informático		

Tabla No. 2: Participantes Equipo de Trabajo

1.2.3 Equipo de trabajo de: Empresas/Instituciones/Organizaciones participantes/cooperantes

Nombre: Ordenamiento Territorial.			
Dirección: Calle Juan Misael Saracho No. E-0659		Teléf. Oficina: 66-43208	
Nombre y Apellidos	Cargo	C.I.	Firma
Lic. Silvia Yebara	Encargada de proyectos y lotes de Ordenamiento Territorial		

Tabla No. 3: Empresas/Instituciones/Organizaciones participantes/cooperantes

1.2.4 Actividades previstas para los integrantes del equipo de investigación

Responsable *	Actividades
Director	▪ Organizar el equipo de trabajo. ▪ Planificar y llevar el control del cronograma del Proyecto. ▪ Asignar los recursos y gestionar las prioridades del Proyecto. ▪ Coordinar la interacción con los usuarios internos y externos. ▪ Establecer prácticas para asegurar la calidad y la integridad de los artefactos del proyecto. ▪ Realizar el seguimiento de todas las etapas del proyecto.

	▪ Mantener al equipo del proyecto enfocado en los objetivos. ▪ Gestionar los riesgos del Proyecto. ▪ Elaboración de informes parciales del trabajo de investigación. ▪ Elaboración del informe final de la investigación. ▪ Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. ▪ Elaborar el Modelo de Análisis y Diseño del Sistema. ▪ Participar en la organización y sistematización de datos. ▪ Llevar el Desarrollo del Sistema. ▪ Ejecutar la capacitación y difusión del sistema. ▪ Investigar aspectos científicos y tecnológicos. ▪ Coadyuvar en el Diseño y Desarrollo del Sistema. ▪ Apoyar a la difusión y capacitación del Sistema. ▪ Elaborar la documentación del sistema. ▪ Colaborar en la ejecución de la capacitación y difusión del sistema. ▪ Organización de la presentación final del sistema. ▪ Colaborar con un buen asesoramiento durante todo el calendario del proyecto.
--	--

Tabla No. 4: Actividades de los integrantes del equipo de investigación

1.2. Descripción del Proyecto

1.2.1. Resumen Ejecutivo del Proyecto

Ordenamiento Territorial es una institución sin fines de lucro que se encarga de:

- Planificar el desarrollo urbano de la ciudad de Tarija y la provincia cercado
- Identificar, recuperar, administrarlas áreas de propiedad municipal destinadas a áreas verdes, equipamiento, infraestructura, banco de tierras y del patrimonio histórico.
- Controlar el crecimiento urbano, las urbanizaciones, loteamientos, y las construcciones. Sancionar y tomar medidas coercitivas para el cumplimiento de las normas y reglamentos de uso del suelo y de las construcciones.
- Formular el plan maestro de desarrollo urbano.
- Aprobar las Urbanizaciones y Loteamientos en base al plan maestro de crecimiento urbano.
- Demoler las construcciones que no cumplan con las normativas del uso del suelo, urbanística, de saneamiento básico y normas administrativas especiales. (Ver Anexos)

El Proyecto está dirigido fundamentalmente al área de Proyectos y lotes de Ordenamiento Territorial, siendo la misma área la beneficiada con este proyecto. Además de que el Proyecto va a contribuir a mejorar cualitativa y cuantitativamente la Gestión de Información en el Proceso de Seguimiento de Trámites, introduciendo criterios para la Información, aprovechando las nuevas tecnologías lo cual va permitir en un corto y mediano plazo, el incremento del Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites y por ende, en un largo plazo una mejor calidad de vida para los habitantes de la ciudad de Tarija.

Por el impacto que ha tenido el desarrollo de proyectos informáticos con el fin de automatizar la información para facilitar el trabajo, reducir el tiempo de acceso a la

información, aumentar la eficiencia, etc. a tenido mucho éxito en los lugares donde fueron desarrollados y a los problemas identificados como pérdida de tiempo, acceso a la información se vio por conveniente desarrollar un proyecto informático para el Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites para Ordenamiento Territorial. El cual va contar con dos componentes:

- I. El componente 1 (Sistema informático para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial se encargara de registrar los datos de los propietarios y el de sus respectivos trámites, ver el estado en el que se encuentra dicho trámite, además de generar reportes de acuerdo a los requerimientos del usuario.

- II. El componente 2 (Taller de capacitación sobre el uso de las TIC, orientado al Personal de Ordenamiento Territorial, específicamente va dirigida al área de Proyectos y Lotes y el manejo del Sistema Informático para la gestión de información en el proceso de seguimiento de trámites (“G.I.P.S.E.T”) sobre la importancia de la implementación de los recursos TICs en el contexto y el manejo correspondiente del sistema informático para la gestión de información en el proceso de seguimiento de trámites, a los beneficiados con este, para que conozcan, manejen, manipulen las herramientas tecnológicas que van a facilitar su vida labor y su desempeño en el cargo. Además se otorgara certificados de participación a cada uno de los asistentes.
Para llevar a cabo la capacitación, se utilizará toda la tecnología moderna posible necesaria para llegar con gran facilidad, agrado y detalle de comprensión a los usuarios finales.

El proyecto tiene como estrategia utilizar la metodología Rational Unified Process (RUP) asociado con UML así mismo, el desarrollo del Sistema contemplará lenguajes de programación, así como el gestor de la base de datos será

seleccionado con sumo cuidado para que el rendimiento del mismo sea óptimo y propicio a las exigencias tecnológicas actuales de acuerdo a los Requerimientos.

1.2.2 Descripción, Fundamentación y Justificación del Proyecto (qué y por qué)

Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites para Ordenamiento Territorial **“G.I.P.S.E.T”**.

Después de realizar la ingeniería de requerimientos en de Ordenamiento Territorial, se pudo identificar problema en el manejo de la información, la pérdida de tiempo al momento de acceder a ella.

Por lo Tanto el Proceso de Seguimiento de Trámites en Ordenamiento Territorial, es precaria porque se realiza manualmente y utilizando Excel para registrar los datos del propietario, la fecha de ingreso, que observaciones tiene, hacia qué departamento de la institución fue, cuando regreso de ese lugar y con qué observación, y otros.

La realización de los reportes se los realiza de manera poco óptima utilizando mediante hojas de cálculo Excel, por lo que corre serios riesgos de perder información o modificarse, ya que esta modalidad de elaboración de reportes es sumamente vulnerable de fácil acceso a ser modificado y hasta corre riesgos de ser borrado por completo.(ver Anexos)

Esta modalidad presenta tediosas demoras en su elaboración ya que se las realiza de tres maneras preliminares, acumulativas y final de los trámites atendidos en el área de Proyectos y lotes, por lo que resulta tediosa esta tarea por no contar con una base de datos para el manejo de información.

En resumen cualquier operación en cualquiera de estos registros por hoja de cálculo, conlleva a diferentes fallas de simple transcripción, modificación, reemplazo u otra aplicación, para que posteriormente tener que rehacer, en caso de realizar una eliminación accidental de cualquiera de estas hojas de cálculo. Por lo que se puede determinar que este tipo de aplicaciones y métodos, no son seguros ni se encuentran de acuerdo a las épocas y circunstancias de las que nos presenta y favorece la tecnología actual.

Los beneficios de la utilización de nuevas tecnologías de comunicación como ser el Internet nos aseguran la integridad, veracidad de nuestra información.

Por ello el presente proyecto pretende contribuir al Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites para mejorar la calidad de vida de los habitantes de la provincia cercado y del departamento de Tarija.

Además de contribuir a lograr el objetivo específico que tiene Ordenamiento Territorial que es el de planificar el desarrollo urbano, identificar, recuperar, administrar las áreas de propiedad municipal destinadas a áreas verdes, controlar el crecimiento urbano, las urbanizaciones, loteamientos y construcciones.

A través de un Sistema informático el cual será de escritorio y entre algunos de los aspectos más comunes se tienen: Adicionar datos de los propietarios y sus trámites, además de ver en qué estado se encuentran, información acerca de sus observaciones, Administración, que se imparten en dicha institución y otros aspectos de vital importancia para el Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites. Los Sistemas pueden ser utilizados como una vía directa para la Información.

Esto va a contribuir a mejorar la calidad de vida de los Habitantes del departamento de Tarija.

1.2.3. Objetivos

1.2.3.1 Objetivo General

La gestión de Información en el proceso de seguimiento de trámites para Ordenamiento Territorial del departamento de Tarija, mejorada.

1.2.3.2 Objetivos Específicos

- Sistema informático para la gestión de en el proceso de seguimiento de trámites para Ordenamiento Territorial, desarrollado y en funcionamiento
- Realizar un taller de capacitación sobre el uso de las TIC, orientado al personal de La Ordenamiento Territorial, implementado.

1.2.4 Metodología

Referente al Desarrollo del Proyecto:

Para el presente proyecto Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites para Ordenamiento Territorial “**G.I.P.S.E.T**”.

Se aplicará la metodología Rup para la ingeniería de requerimientos y UML para el diseño del sistema como: Los modelos de objeto del negocio, casos de uso del sistema, diagrama de actividades, diagramas de secuencia, diagramas de colaboración y además de los diagramas de clases.

Por con lo que se pretende establecer un ciclo de vida al proyecto, compuesto de fases que se van dando de manera secuencial:

Fase de Planificación: En donde se va a realizar la idea del proyecto, presentado el perfil del proyecto usando el Sistema de Marco Lógico para presentar de forma sistemática y lógica los objetivos del proyecto y sus relaciones de causalidad además de mejorar la planificación, el seguimiento y la evaluación del proyecto. Incluyendo un calendario de actividades y un cronograma de control de las actividades.

Fase de Ejecución: La recopilación de información se realizara por medio de entrevistas y reuniones con los integrantes al proyecto de Ordenamiento Territorial .

Esta información será utilizada para desarrollar un sistema de gestión, en el diseño de las aplicaciones mediante modelos, como modelo de negocios, casos de uso, diagramas de clases, etc.

Metodología para el desarrollo de aplicaciones: La metodología se va a realizar para optimizar los tiempos y resultados, ejecutadas por el siguiente equipo de trabajo multidisciplinario.

Especialistas como los administradores del departamento de Proyectos y Lotes, técnicos, el proceso de seguimiento de trámites para definir los requerimientos del sistema.

Metodología para el desarrollo de las aplicaciones:

Se utilizará la metodología RUP (Racional Unified Process), que mejora considerablemente la calidad de desarrollo del sistema, ya que la misma utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software.

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

El flujo de trabajo fundamental tiene los siguientes pasos:

- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Programación e Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Requerimientos.

En base a las entrevistas se obtendrá la información que refleje las necesidades de los involucrados para la determinación de requerimientos.

Análisis y diseño:

En base a la determinación de requerimientos, se estructurará las diferentes vistas (Diagramas, base de datos, Pantallas) de la aplicación, tomando en cuenta metodologías de desarrollo de software.

Programación e Implementación:

La programación será modular y orientada a objetos, se utilizarán tecnologías de punta contando con el apoyo de programadores experimentados (estudiantes), creando la aplicación informática que tenga el comportamiento deseado.

Pruebas y validación:

Antes de desarrollar las pruebas se procederá a la introducción de datos.

Introducida esta información al sistema se dará inicio a la fase de pruebas de desarrollo que serán mediante casos de prueba y se realizarán los ajustes necesarios para una correcta validación.

Este proceso se torna repetitivo si se detectan inconsistencias en el sistema implicando el retorno de cualquiera de las fases anteriores para su corrección.

Referente a la Transferencia de Resultados:

- **Metodología de Enseñanza Socializada**

Se pretende ejecutar una metodología de enseñanza socializada, dirigida al personal de Ordenamiento Territorial con lo que se busca una integración social sin descuidar la individualización, aplicando métodos de asamblea y Panel.[3]

- **Técnica Expositiva**

Consiste en la exposición oral, por parte del capacitador; en la que se estimulará la participación del personal de Ordenamiento Territorial en los trabajos que se realicen, este requiere una buena motivación para atraer la atención de los educandos.[3]

- **Técnica de la Experiencia**

La experiencia es un procedimiento eminentemente activo y que procura:

- Exponer de forma presencial los pasos a seguir para realizar alguna acción.
- Explicar el funcionamiento de algo mientras es observado.

- Comprobar, con razones lo que va a suceder, partiendo de experiencias.
- Conferir confianza para actuar en el terreno de la realidad de manera lógica
- Convencer a cerca de la veracidad de la ley de causa y efecto.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.[3]

- **Formas de Socialización**

Para el conocimiento de la población en general y del propio personal de Ordenamiento Territorial se pretende llevar a cabo la socialización de los componentes producidos por el proyecto mediante [4]:

- Afiches
- Trípticos
- Propaganda

1.2.5 Resultados esperados

- El proyecto tendrá un impacto importante en el desempeño de ORDENAMIENTO TERRITORIAL en lo referente a la mayor flexibilidad de los procesos administrativos y de trámites, ya que tendrán a su disposición una herramienta que les permita adaptarse a las nuevas técnicas para llevar a cabo el registro de propietarios, observaciones, ingresos, salidas y otros procesos de gestión, para de ésta manera poder atender mejor a la población tarijeña.
- El desarrollo de un Sistema Informático para la gestión de la información en proceso de seguimiento de trámites, el cual accederá a crear usuarios del sistema, asignarle roles con restricciones de acceso, en donde podrá realizar la administración de personal, propietarios, administrar observaciones correspondientes a cada trámite, ingresos salidas, envíos y recepción de cada trámite.

Se llevará a cabo la planificación y la implementación de un Estrategias de socialización del proyecto la cual estará dirigida la personal de Ordenamiento Territorial, en particular al área de Proyetos y Lotes y a la población tarijeña, transfiriendo de esta forma los beneficios otorgados por el proyecto, este componente es un aspecto muy necesario para el éxito del proyecto, por ello se pretende realizar la publicidad del mismo por los medios más populares de difusión: entre ellos la propaganda la elaboración de afiches y trípticos o folletos que identifiquen el proyecto, sus objetivos y otros; todo esto con el objeto de aumentar la probabilidad de éxito del proyecto y su sostenibilidad.

1.2.6 Transferencia de resultados

a) Medios y estrategias para la transferencia de resultados.

- Presentación final del sistema informático a las autoridades universitarias correspondientes y a Ordenamiento Territorial de la ciudad de Tarija.
- Entrega de instaladores del sistema informático y la documentación desarrollada en el proyecto.
- Diseño de Afiches que forman parte de las estrategias de socialización del proyecto.
- Capacitación a Ordenamiento Territorial acerca de las TIC y especialmente en el manejo del sistema desarrollado, haciendo uso de métodos de enseñanza expositiva y técnicas basadas en la experiencia
- Socialización del producto final con las institucion involucradas.

b) Grupo de beneficiarios de los resultados

Los beneficiarios del proyecto son:

- Ordenamiento Territorial.
- La sociedad en general del departamento de Tarija.
- Área de Proyectos y Lotes, Ordenamiento Territorial.

1.2.7 Cronograma de Actividades

Nº	Actividad	Nº días	M1	M2	M3	M4	M5	M6	M7	M8
1	Especificación de requerimientos En la presente actividad se realiza el análisis de todas las necesidades de los usuarios, respecto a la problemática del proyecto.	14	X							
2	Elaboración de los diagramas UML del Sistema En esta actividad se realiza el análisis y el diseño del sistema y se representa el resultado con el lenguaje de modelado unificado.	21	X							
3	Diseño de la Base de Datos del Sistema. Se analiza y se diseña la	20		X	X					

	estructura en la cual la información del sistema será almacenada.(Se realiza el modelo de la Base de Datos)									
4	Diseño del prototipo del Sistema Se realiza un bosquejo de la distribución de interfaces de usuario según la funcionalidad del sistema.	14				X				
5	Programación del Sistema Web. Se realiza la codificación y construcción del Sistema final.	84					X	X		
6	Validación del Software Se realiza las pruebas necesarias de calidad y de rendimiento del sistema.	21								X

7	Elaboración del Informe final sobre el Desarrollo del Sistema Web. Se realiza el documento final que describe todas las etapas de desarrollo del Sistema.	21								X
8	Planificación del contenido temático del taller de capacitación. Se realiza la Planificación programada, según contenido establecido.	21							X	
9	Definición de metodologías de enseñanza Se realiza la definición de la metodología y del material utilizado durante la capacitación del proyecto	7							X	

10	Realizar las Estrategias de socialización y conjuntamente se realiza la capacitación del proyecto al personal de “D.D.U”.	2								X
----	---	---	--	--	--	--	--	--	--	---

Tabla N° 5: Cronograma de Actividades

1.2.8 Marco Lógico del Proyecto

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir a mejorar la calidad de vida de los habitantes de la ciudad de Tarija y la provincia Cercado.	Al final del 2º año de ejecución del proyecto, la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial, se incrementa en un 30% en comparación al año base (2011).	Informes oficiales avaladas por el Director de la Dirección de, Ordenamiento Territorial con estadísticas del mejoramiento del proceso de seguimiento de trámites de los habitantes de la ciudad de Tarija	Sostenibilidad • La Sociedad tarijeña hace uso de los servicios que le brinda el proyecto. • Aplicación del sistema por parte de las autoridades involucradas.
Objetivo General (Propósito) La gestión de Información en el proceso de seguimiento de trámites para, Ordenamiento	Al finalizar el proyecto la gestión de Información en el proceso de seguimiento de trámites para	Informes oficiales elaborados por Ordenamiento Territorial, sobre el mejoramiento de	Propósito a Fin • Los habitantes de la ciudad de Tarija tienen el interés en realizar sus trámites

Territorial del departamento de Tarija, mejorada.	Ordenamiento Territorial se ve fortalecida en 15% con un entorno tecnológico y metodológico.	la gestión de Información en el proceso de seguimiento de trámites.	en, Ordenamiento Territorial. - Recursos dotados por parte de La Oficialía Mayor de Planificación, Ordenamiento Territorial y Gestión Urbana - Las autoridades de la región impulsan el de la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial.
---	--	--	--

Objetivos Específicos			Componentes a Propósito
1. Sistema informático para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial , desarrollado y en funcionamiento	Al finalizar el proyecto, y en base a la ingeniería de requerimientos, el sistema para la gestión de Información en el proceso de seguimiento de trámites para, Ordenamiento Territorial , está desarrollado en un 100%.	Informe avalado por el Director de Ordenamiento Territorial expresando su conformidad con el sistema. Documento de Análisis y diseño del Proyecto presentado a los Docentes de Taller III. Informe de satisfacción del sistema informático elaborado por los involucrados al proyecto, por Ordenamiento Territorial.	• El personal de La Oficialía Mayor de Planificación, Ordenamiento Territorial y Gestión Urbana brindan la información necesaria. • La institución mantiene el interés en fortalecer la gestión de información en el proceso de seguimiento de trámites. • Se cuenta con los recursos económicos y humanos necesarios para llevar a cabo el desarrollo de los componentes del proyecto.

2. Estrategias de socialización del sistema dirigido a la población tarijeña, implementadas.	Al final de la ejecución del proyecto se realiza el diseño de afiches y trípticos destinados a la publicación y promoción del sistema de Escritorio. Elaboración de afiches y folletos destinada a la promoción del sistema de Información al final del séptimo mes. Al finalizar el proyecto, al menos un taller de capacitación se realizó en un 80% de 10 personas estimadas.	Los diseños de los afiches serán presentados en el Documento. Listas de Asistencia debidamente firmadas por los asistentes al taller de capacitación.	• El personal capacitado mantiene el interés en continuar explorando los beneficios que le brinda el sistema. • La campaña publicitaria es exitosa. • Ordenamiento Territorial está de acuerdo en participar en las capacitaciones impartidas. • El personal de Ordenamiento Territorial tienen interés en ser capacitados.
---	---	---	--

Actividades			Actividades a Componente
1. Sistema informático para la gestión de información en el proceso de seguimiento de trámites Ordenamiento Territorial , desarrollado y en funcionamiento 1.1 Ingeniería de requerimientos 1.2 Diseño 1.3 Instrumentación 1.4 Instalación 2. Estrategias de Socialización Definir las estrategias para la socialización del proyecto. Llevar a cabo las estrategias de socialización	Presupuesto por partidas(ver punto 3 Presupuesto / Justificación)	Informe de ejecución presupuestaria avalado por el Director del proyecto. Manual de Usuario Manual de instalación	• Los desembolsos se realizan de acuerdo al cronograma • Disponibilidad de Equipos de computación.

del producto final. Taller de capacitación sobre el uso de las TIC, orientado al Personal de Ordenamiento Territorial ha sido implementado 2.1 Definición de metodologías de enseñanza. 2.2 Planificación del contenido temático del taller de capacitación. 2.3 Implementación del taller 2.4 Definir los medios y llevar a cabo las estrategias de Socialización del producto final.			
---	--	--	--

Tabla N° 6: Matriz del Marco Lógico

1.3 Presupuesto / Justificación

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs)
20000	SERVICIOS PERSONALES NO			
	21000. Servicios Básicos			320
	22000. Servicios de transporte			540
	23000. Alquileres			240
	24000. Mantenimiento y reparación			
	25000. Servicios Profesionales y Comerciales			4684
	Sub total rubro			5974
30000	MATERIALES SUMINISTROS Y			
	31000. Alimentos y Productos Forestales			2460
	32000. Productos de Papel, Cartón e Impresos			920
	33000. Textiles y Vestuario.			

	34000. Productos Químicos, Combustibles y Lubricantes			
	39000. Productos Varios.			60
	Sub total rubro			3440
40000	ACTIVOS REALES			
	43000. Maquinaria y Equipo.			1000
	46000. Descripción de estudios y proyectos para inversión			
	49000. Otros Activos			
	Sub total rubro			1000
	TOTAL			10224

Tabla N° 7: Presupuesto/Justificación

1) GRUPO 20000. SERVICIOS NO PERSONALES

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

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación			
21200	Energía Eléctrica	40	8	320
21300	Agua			
21400	Servicios Telefónicos			

Total				320

Tabla N° 8: Servicios No Personales

- 2) * 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

Partida	Personal	Lugar	N° de viajes	Costo unitario *	Costo total
22100	Pasajes				
	Pasaje Director	Dpto. de Investigación Ciencia y Tecnología	270	2	540
Total					540

Tabla N° 9: Descripción de los gastos de viajes y transporte de personal

- 3) * En el caso de pasajes debe indicarse el costo de ida y vuelta (costo unitario), indicando el número de viajes.

Partida	Personal	Lugar	Duración (días)	Costo unitario *	Costo total
22200	Viáticos				
22300	Fletes y Almacenamientos				
22600	Transporte de Personal				
Total					

Total sub grupo 22000			
-----------------------	--	--	--

Tabla N° 10: Indicarse el costo de ida y vuelta

* 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

Partida	Alquiler de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
23100	Alquiler de Edificios			
23200	Alquiler de Equipos y Maquinaria 20 Equipos de computación	6	2 días	240
23300	Alquiler de Tierras y Terrenos			
Total				240

Tabla N° 11: Descripción de los gastos

* 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

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

Tabla N° 12: Descripción mantenimiento y reparación

* 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

Partida	Tipo de servicio profesional y comercial *	Cantidad	Costo unitario	Tiempo mes	Costo total
25500	Publicidad				
25600	Imprenta				
	□ Certificados de participación	20	5		100
	□ Empastados	6	100		600
25700	Capacitación de Personal				
25800	Estudios e Investigaciones Para Proyectos de Inversión				
25810	Consultores por Producto				
	Consultora Informática en programación				
	Consultor diseño grafico				
	Consultor Agrónomo				
25820	Consultores en Línea				
	Director	1	442.6	9	3984
	Total				4684

Tabla N° 13: Descripción de los gastos en servicios profesionales

* 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.

1) 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	820	3	2460

	varios			
31200	Alimento para Animales			
31300	Productos Agroforestales y Pecuarios			
Total				2460

Tabla N° 14: 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

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio ☐ Resma de hojas bond	10	45	450
32200	Productos de Artes Graficas, Papel y Cartón ☐ Tinta negra ☐ Tinta a colores	4 6	35 55	140 330
32300	Libros y Revistas			
32400	Textos de Enseñanza			
Total				920

Tabla N° 15: Descripción del gasto de Productos de Papel, Cartón e Impresos.

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

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

Partida	Combustibles, Productos Químicos, Farmacéuticos y Otros	Cantidad	Costo/Unitario	Total

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			
34700	Minerales			
34800	Herramientas Menores			
Total				

Tabla N° 16: Descripción del gasto de Productos de Papel, Cartón e Impresos.

2) GRUPO 40000. ACTIVOS REALES

1) SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo

Partida	Tipos de productos	Cantidad	Costo/Unitario	Total
43100	Equipo de Oficina y Muebles □ Cámara Digital	1	1000	1000
43200	Maquinaria y Equipo de Producción			
43300	Equipos de Transporte, Tracción y Elevación			
43400	Equipo Médico y de Laboratorio			
43700	Otra Maquinaria y Equipo			

Total			1000
-------	--	--	------

Tabla N° 17: Descripción del gasto de Productos

* 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.

1.4 Curriculum Vitae (del Director y Equipo de Trabajo: llenar una ficha para cada uno de ellos)

DIRECTOR DEL PROYECTO

1.4.1 Antecedentes personales

Jallaza Apellido Paterno	Garnica Apellido Materno	Irka María Nombre	5044883 C.I.
29 / 05 / 1987 Fecha de nacimiento	M..... F...X... Sexo	c. Final Ballivián Dirección	
Tarija Ciudad	66-38524 Teléfono Domicilio	----- Celular	akri_elisha_5@hotmail.com Correo electrónico

Tabla N° 18: Antecedentes personales

1.4.2 Antecedentes académicos

Carrera	Curso/Semestre	Año
Ingeniería Informática	Grupo 1 / Primer	2005
Ingeniería Informática	Grupo 1 / Segundo	2005
Ingeniería Informática	Grupo 1 / Primer	2006
Ingeniería Informática	Grupo 1 / Segundo	2006

Ingeniería Informática	Grupo 1 / Primer	2007
Ingeniería Informática	Grupo 1 / Segundo	2007
Ingeniería Informática	Grupo 1 / Primer	2008
Ingeniería Informática	Grupo 2 / Segundo	2008

Tabla Nª 19: Antecedentes académicos

1.4.3 Participación en proyectos de investigación

Título proyecto	Institución	Cargo	Año

Tabla Nª 20: Participación en proyectos de investigación

1.4.4 Publicaciones realizadas (libros, revistas, compendios y otros)

Autor	Tipo de publicación, Año, título, volumen, páginas, editorial

Tabla Nª 21: Publicaciones realizadas (libros, revistas, compendios y otros)

1.4.5 Antecedentes en ayudantías

Carrera	Semestre	Año

Tabla Nª 22: Antecedentes en ayudantías

1.5 INVESTIGADOR (1)

1.5.1 Antecedentes personales

Jallaza	Garnica	Irka María	5044883 Tja
Apellido	Apellido	Nombre	C.I.
Paterno	Materno		

29 / 05 / 1987 Fecha de nacimiento	M.... F....X... Sexo	c. Final Ballivián Dirección	
Tarija Ciudad	66-38524 Teléfono Domicilio	----- Celular	akri_elisha_5@hotmail.com Correo electrónico

Tabla N° 23: Antecedentes personales**1.5.2 Antecedentes académicos**

Carrera	Curso/Semestre	Año
Ingeniería Informática	Grupo 1 / Primer	2005
Ingeniería Informática	Grupo 1 / Segundo	2005
Ingeniería Informática	Grupo 1 / Primer	2006
Ingeniería Informática	Grupo 1 / Segundo	2006
Ingeniería Informática	Grupo 1 / Primer	2007
Ingeniería Informática	Grupo 1 / Segundo	2007
Ingeniería Informática	Grupo 1 / Primer	2008
Ingeniería Informática	Grupo 2 / Segundo	2008

Tabla N° 24: Antecedentes académicos**1.5.3 Participación en proyectos de investigación**

Título proyecto	Institución	Cargo	Año

Tabla N° 25: Participación en proyectos de investigación

1.6 Cuadro de Involucrados

GRUPO	INTERESES	PROBLEMAS PERCIBIDOS	RECURSOS Y MANDATOS
Propietario	• Obtener la mayor información posible de su trámite. • Aprobar su trámite en el menor tiempo posible.	• Conocimiento insuficientes en el área. • Construcciones clandestinas. • Demora en la obtención de observaciones • Pocos medios de publicación sobre las normas de construcción. • Escasos Recursos. • Trámites sin realizar.	• R: Disponibilidad para realizar sus trámites. • R: Disponibilidad para pagar sus multas. • R: Disponibilidad de tiempo para recabar información de su trámite en proceso.
Director de Ordenamiento Territorial.	• Mejorar el ordenamiento territorial de la ciudad de Tarija.	• Medios inadecuados para dar a conocer la información de los requisitos a seguir en los	• R: Máxima autoridad de la O.T. • R: Cuentan con el apoyo

	• Brindar el mejor servicio a la sociedad Tarijeña.	trámites. • Demoras en el manejo de la información.	de la oficialía mayor técnica. • R: Cuentan con personal calificado. • R: Aprueba y supervisa las aprobaciones de los trámites en proceso. • M: Servir a los mejores intereses de la sociedad Tarijeña. • M: Mejorar el servicio de atención a la población.
Jefes de Proyectos y Lotes	• Aprobar Trámites en el menor tiempo posible. • Dar posibles soluciones a trámites problemáticos.	• Medios de información Inadecuados para dar a conocer las observaciones. • Manejo de información inadecuado.	• R: Experiencia en el área. • R: Cuenta con la potestad de aprobar tramites. • M: Servir a los mejores intereses de los propietarios.

			• M: Mejorar el manejo de la información
Encargados de Proyectos y Lotes	• Brindar las observaciones oportunas a los propietarios. • Mejorar el manejo de la información.	• Mayor demanda de información por parte de los propietarios. • Manejo de información inadecuada. • Demora en la atención a los Propietarios debido a los procesos manuales que se realizan (anotaciones, etc.).	• R: Disponibilidad de brindar la información necesaria. • R: Contar con el acceso a la información necesaria. • M: Servir a los mejores intereses de los ciudadanos tarijeños.
Técnicos	• Mejorar la información sobre los trámites en proceso. • Realizar las observaciones	• Planos no actualizados. • Métodos Inadecuados para a conocer las observaciones realizadas a los trámites en proceso.	• R: Conocimiento y experiencia en el área. • M: Proveer de un servicio seguro y eficiente a los tramites en proceso.

	pertinentes a los trámites en proceso.		
Inspectores	• Visitar el lugar de los trámites en proceso. • Dar las observaciones del trámite en proceso oportunamente al técnico.	• Medios limitados para dar a conocer las observaciones realizadas a los trámites.	• R: Disponibilidad de tiempo para realizar las inspecciones. • M: Servir a los mejores intereses de la población en general.
Topógrafos	• Realizar las observaciones pertinentes a los trámites en proceso.	• Medios limitados para dar a conocer las observaciones realizadas a los trámites.	• R Disponibilidad de tiempo para realizar las mediciones topográficas. • R: Conocimiento en el área. • M: Contribuir a mejorar el servicio a la población en general.

Asesor Legal	• Brindar las observaciones oportunamente de los trámites en proceso.	• Documentación no actualizada.	• R Disponibilidad de tiempo para realizar las asesorías legales. • R: Conocimiento y experiencia en asesoría legal. • M: Ayudar a mejorar el servicio que brinda la O.T.
--------------	---	---	---

Tabla N° 26: Cuadro de Involucrados

1.7 Árbol de Problemas

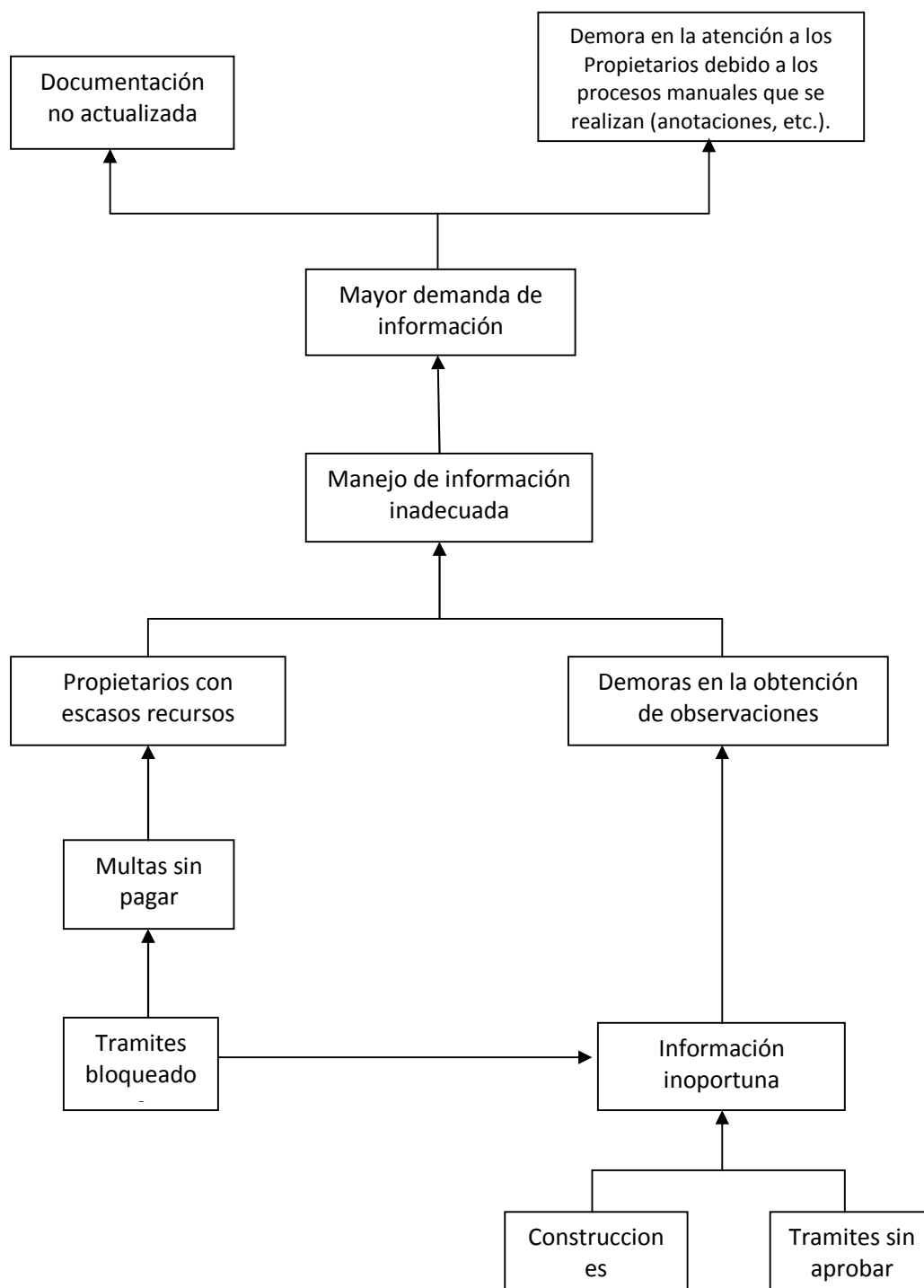


Figura 1: Árbol de Problemas

1.8 Árbol de Objetivos

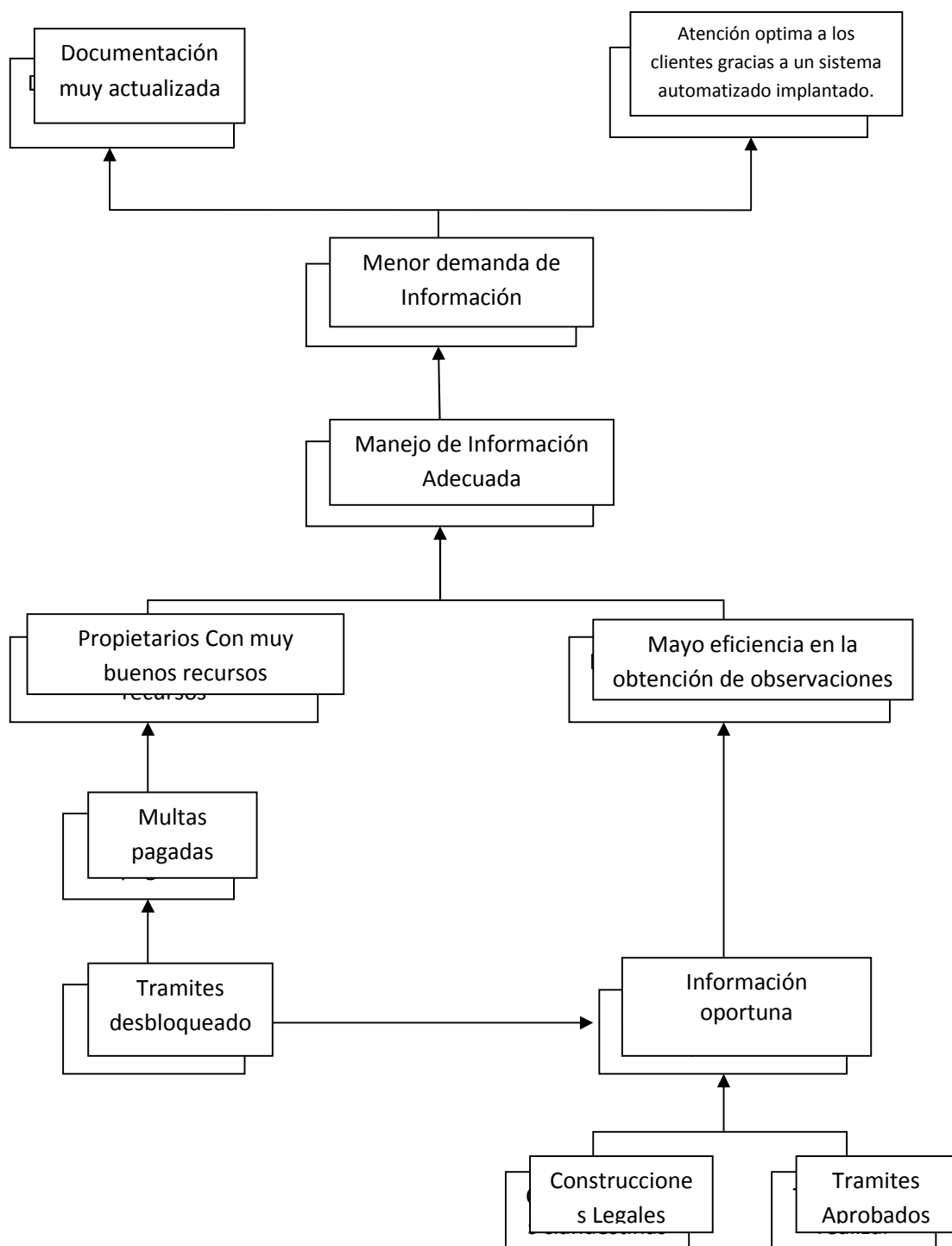


Figura 2: Árbol de Objetivos

CAPITULO II: COMPONENTES

II.1 COMPONENTE 1

**Sistema Informático para la Gestión de Información en el
Proceso de Seguimiento de Trámites para Ordenamiento
Territorial.**

INTRODUCCIÓN

INTRODUCCIÓN

II.1.1 INTRODUCCIÓN

Los sistemas de información computarizados forman hoy en día parte imprescindible de la mayoría de las organizaciones. Contar con fuentes de información confiables, veraces y oportunas que ofrezcan un criterio para la toma de decisiones acertadas en beneficio de la organización y de la sociedad.

Los sistemas de información ofrecen la posibilidad de contar con información de calidad, que genera un mejor desenvolvimiento de todos los demás componentes de la organización.

Por lo que hoy se conforma una sociedad de la información que incorpora grandes avances tecnológicos en la informática; velocidad, disponibilidad y conectividad, que juegan un papel de vital importancia en las organizaciones actuales, ya que las mismas son cada vez más complejas, cada día e incorporan nuevas tecnologías a su forma de trabajar con lo que consiguen competir en el mercado globalizado que el mundo actualmente maneja. Por ello es importante que los sistemas de información computarizada sean implementados y permitir de esta manera mejorar su forma de trabajo, logrando hacerse más competitivos.

La gran mayoría de las organizaciones están conscientes que los mejores resultados se obtienen adaptándose a las nuevas condiciones del entorno, un entorno que se ha ido haciendo cada vez más cambiante, exigente y turbulento. Los desafíos organizacionales están influenciados por la innovación de tecnologías Informáticas, la rapidez, servicio de calidad y mejora continua.

La importancia de los sistemas de información está presente en todos lados.

El presente trabajo está destinado al desarrollo de un proyecto para la gestión de información en el proceso de seguimiento de tramites aplicado para ORDENAMIENTO TERRITORIAL, apoyando las funciones que tiene esta

organización, enfocándonos al fortalecimiento de la información y rendimiento del mismo.

II.1. 2 ANTECEDENTES

II.1.2.1 Antecedentes de la Organización

Ordenamiento Territorial dependiente de la Honorable Alcaldía Municipal de la ciudad de Tarija y la provincia Cercado, se ocupa de identificar, recuperar, administrar las áreas de propiedad municipal destinadas a áreas verdes, equipamiento, infraestructura, banco de tierras y del patrimonio histórico. Además de controlar el crecimiento urbano, las urbanizaciones, loteamientos, y las construcciones.

Ordenamiento Territorial debe sancionar y tomar medidas coercitivas para el cumplimiento de las normas y reglamentos del uso del suelo y de las construcciones.

En consecuencia, su gran compromiso y misión es la de promover el fortalecimiento de la infraestructura social y el ordenamiento y mejor planificación de los centros urbanos de la ciudad de Tarija y la provincia Cercado.

Ordenamiento Territorial constituyen un factor fundamental para la reducción de la pobreza, la construcción de la paz y el mejoramiento del nivel de vida de la población. El acelerado proceso de urbanización de la ciudad de Tarija y la provincia Cercado exige cada vez más acciones concretas e integrales en materia de infraestructura social básica para la población, el estudio y reordenación de los patrones de concentración de la actividad económica en centros urbanos y la consolidación del sistema urbano-regional tarijeño.(Ver Anexos)

II.1.2.2 Organigrama

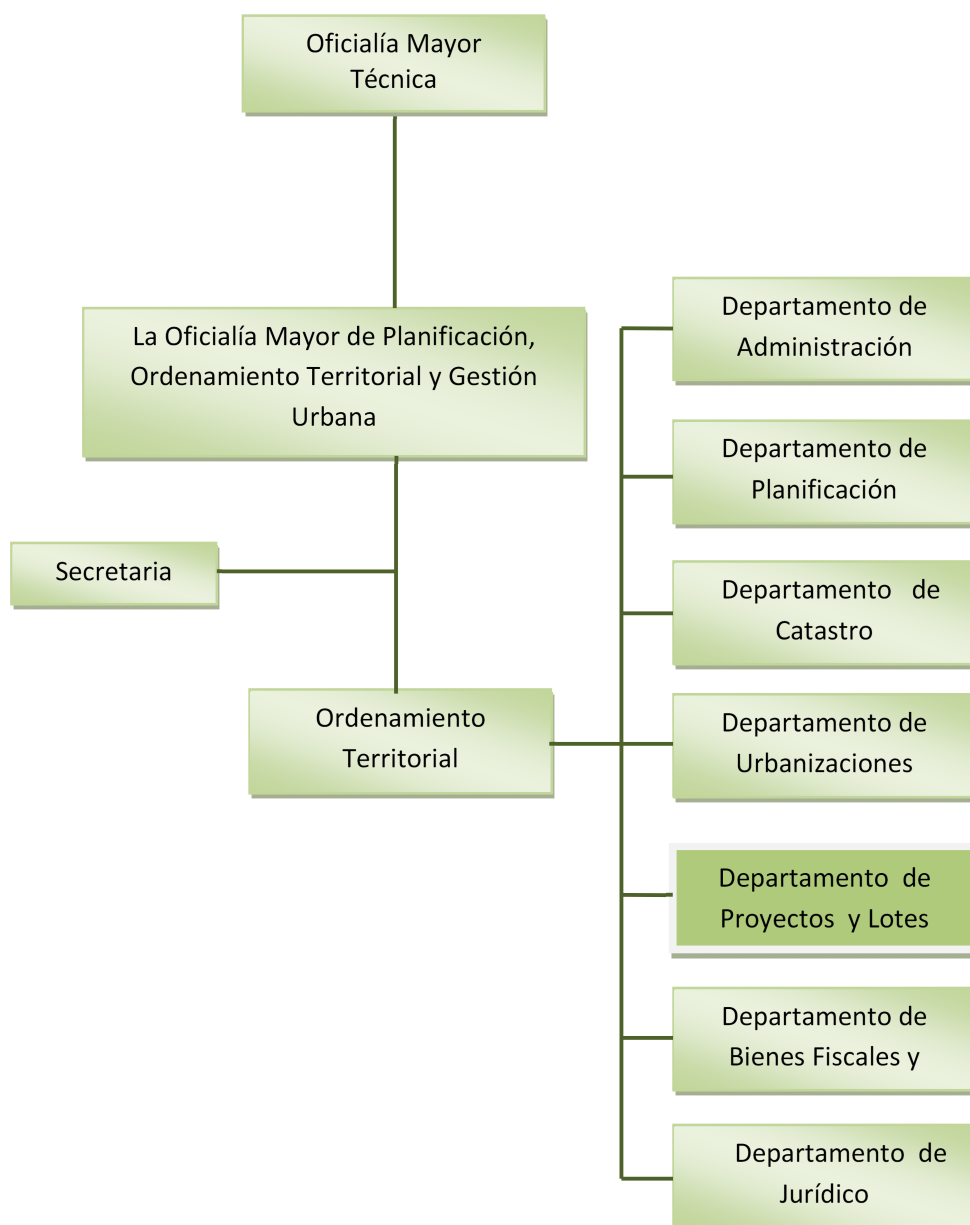


Figura 3: Organigrama de Ordenamiento Territorial

II.1.2.3 Funciones

II.2.3.1 Funciones de la Organización Ordenamiento Territorial

- Preservar el patrimonio, paisajístico así como resguardar el patrimonio de la nación existente en el municipio.
- Formular el plan maestro de desarrollo urbano.
- Formular, fiscalizar y coordinar la ejecución de los planes de ordenamiento territorial del municipio, en concordancias con las normas departamentales y nacionales y de acuerdo a criterios técnicos.
- Hacer cumplir las normas especiales nacionales y municipales de uso del suelo, subsuelo y sobresuelo.
- Demoler las construcciones que no cumplan con las normativas del uso del suelo, urbanística, de saneamiento básico y normas administrativas especiales.
- Reubicar luego de un proceso técnico-administrativo-jurídico, sin que medie expropiación ni compensación alguna, el uso de los inmuebles destinados a vivienda, comerciales, industriales o de cualquier otro carácter, que no cumplan y afecten al plan de ordenamiento urbano y territorial, la norma del uso del suelo o cuando el interés público así lo aconseje.
- Aprobar las urbanizaciones y loteamientos en base al plan maestro de crecimiento urbano.
- Preservar los bienes patrimoniales, arqueológicos, precolombinos, virreinales, republicanos históricos de la nación o los procedentes del culto religioso que se encuentren en su jurisdicción sean públicos o privados, promover el uso y goce lucrativo y restaurar los que sean de propiedad pública del municipio.
- Elaborar los planes de zonificación y evaluación zonal o Distrital, determinar tablas de valores según calidad de vías del suelo, calidad

y tipo de construcción, servicios y calzadas así como la delimitación literal de cada una de las zonas y zonas rurales detectadas en el proceso de zonificación, conforme a normas nacionales vigentes.

- Fiscalizar la administración del catastro urbano y rural, de acuerdo a las normas catastrales y técnico-tributarias emitidas por el poder ejecutivo.
- Garantizar que aquellas aéreas calificadas de riesgo para la construcción, no sean ocupadas con fines de vivienda ni equipamiento.
- Administrar el catastro urbano y rural en forma directa o darlo en concesión a terceros, previo autorización del Consejo Municipal.
- Ordenar la demolición de inmuebles que no cumplan con las normas urbanísticas y normas administrativas especiales.(Ver Anexos)

II.1.2.3.2 Funciones del Departamento de Proyectos y Lotes

- Reubicar luego de un proceso técnico – administrativo – jurídico, sin que medie expropiación ni compensación alguna, el uso de inmuebles destinados a vivienda, comerciales, industriales o de cualquier otro carácter, que no cumplan y afecten al plan de ordenamiento urbano y territorial, las normas de uso del suelo o cuando el interés público así lo aconseje.
- Ordenar las demoliciones de inmuebles que no cumplan con las normas urbanísticas.
- Aprobar planos de Lotes en base a las urbanizaciones y loteamientos aprobados por la Dirección y el Departamento de Planificación.
- Aprobar los planos de construcciones y edificaciones nuevas.
- Aprobar la subdivisión de lotes.
- Aprobar anexiones y transferencias de lotes y construcciones.

- Aprobar la partición de viviendas.
- Aprobar las remodelaciones de viviendas y edificaciones.(Ver Anexos)

II.1.3 PLANTEAMIENTO DEL PROBLEMA

En la actualidad esta Institución no cuenta con un Sistema informático para la gestión de información en el proceso de seguimiento de trámites para el Área de Proyectos y Lotes.

Por lo que Después de un estudio realizado, a Ordenamiento Territorial se pudo identificar el actual modelo de seguimiento en el proceso de trámites el Departamento de Proyecto y Lotes de Ordenamiento Territorial no es eficiente debido al tiempo de acceso de la información correcta y oportuna de esta área.

Esto se debe a:

- El seguimiento en el proceso de trámites el Departamento de Proyecto y Lotes de Ordenamiento Territorial es precaria porque se realiza manualmente.
- Los datos de los tramites atendidos en este departamento se las registra manualmente, datos de los propietarios, dirección, sus observaciones (técnicas, inspeccionales, topográficas, etc.) emitidas o algunos otros datos requeridos.
- El sistema de Informes actual es por medio de hojas de cálculo Excel, corre serios riesgos de perder información o modificarse.
- La modalidad de elaboración de los Informes es sumamente vulnerable de fácil acceso a ser modificado y hasta corre riegos de ser borrado por completo.

- Esta modalidad presenta tediosas demoras en la elaboración preliminar, acumulativa y final de cada uno de los Informes, por no contar con una base de datos para el manejo de información.
- Cualquier operación en cualquier de estos Informes por hoja de cálculo, conlleva a diferentes fallas de simple transcripción, modificación, reemplazo u otra aplicación, para que posteriormente tener que rehacer un Informe, en caso de realizar una eliminación accidental de cualquiera de estas hojas de cálculo.
- Por lo que se puede determinar que este tipo de aplicaciones y métodos, no son seguros ni se encuentran de acuerdo a las épocas y circunstancias de las que nos presenta y favorece la tecnología actual.

Por las razones mencionadas se ve por conveniente llevar a cabo un proyecto informático para desarrollar un Sistema para la gestión de información en el proceso de seguimiento de trámites para, Ordenamiento Territorial.

II.1.4 OBJETIVOS

II.1.4.1 Objetivo General

Diseñar e Implementar un Sistema informático para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial, el cual contará con información específica y actualizada de los Trámites en proceso.

II.1.4.2 Objetivos Específicos

- Analizar y Diseñar la aplicación propuesta.
- Implementar la aplicación como así también documentar la misma.
- Aplicar el Lenguaje de Modelado Unificado (UML) en toda la fase del desarrollo.
- Aplicar la Metodología de desarrollo RUP.
- Instrumentar el sistema

- Diseñar una interfaz gráfica y amigable para el usuario.
- *Proveer mayor seguridad en el manejo de la información:* El hecho de asignar roles a cada actor (usuarios) que interviene dentro del sistema nos permitirá tener cierto grado de seguridad en el manejo y acceso a la información dentro de nuestro sistema.

II.1.5. JUSTIFICACIÓN

Tal como se pudo apreciar durante la elaboración del presente documento, se ha procurado dar más conocer de cómo se realiza actualmente el proceso de tratamiento de toda la información que se produce a consecuencia del proceso de seguimiento de trámites en, Ordenamiento Territorial, como ser los datos de los propietarios, de los tramites, el estado en que se encuentran, y otros.

Por lo que en este apartado se tratará de exponer las razones que avalen la elección y justificación de elegir este tema en esta oportunidad como una tesis de grado.

II.1.5.1 Justificación Social

Uno de los fines de la” Universidad Autónoma Juan Misael Saracho “, es “ Formar profesionales idóneos, en todas las áreas del conocimiento científico, tecnológico y cultural, que responda a las necesidades del desarrollo nacional y regional”.

Mostrando en primera instancia el presente trabajo, que será un beneficio directo que obtendrán las personas que trabajen en el sector administrativo e indirectamente a la población tarijeña en su conjunto.

Debido a un mejor manejo de la información y al contar con una institución más eficiente, como así también a otras instituciones que realicen trabajos similares.

II.1.5.2 Justificación Económica

Para el desarrollo del presente Trabajo, se cuenta con todos los medios y recursos necesarios y suficientes por lo que se puede asegurar que el mismo es económicamente factible.

II.1.1.5.3 Justificación Académica

Académicamente el presente trabajo tiene como objetivo la aplicación de todos los conocimientos adquiridos durante la formación universitaria.

II.1.6. Alcances y Limitaciones

II.1.6.1 Alcances

El Alcance General del Presente Proyecto es implementar Sistema informático para la gestión de información en el proceso de seguimiento de trámites para La Ordenamiento Territorial que proporcione al usuario una interfaz amigable para que este pueda tener información actualizada y acorde a sus necesidades y requerimientos.

Entre los alcances del mismo tenemos los siguientes:

- Mantener informado tanto al propietario como al personal sobre eestado en que se encuentran los procesos en trámite.
- Un entorno fácil de usar para todos los usuarios del sistema, los que podrán tener acceso a la información de manera pertinente.
- Mantener actualizada la información sobre los requisitos a seguir.
- Automatización de las observaciones de cada trámite en proceso.
- Generar informes automatizados de los trámites en proceso.
- Satisfacer las necesidades de los usuarios del sistema.
- Introducir información acerca de los trámites en proceso de aprobación.

(Lotes, Levantamiento de Viviendas y Construcción) de Ordenamiento Territorial.

- Tiempo de duración de trámite.
- Contar con una base de datos que contenga :
 - Datos de los propietarios
 - Datos del personal administrativo
 - Datos sobre lotes, levantamiento de viviendas y construcción.
 - Datos sobre los trámites en proceso.

Para lo cual se determinó los siguientes módulos que están de acorde a las necesidades del mismo:

- *Módulo Administración:* En este módulo se lleva un registro general del personal vinculado en el área, con sus respectivos roles, el estado en que se encuentran, por lo que este módulo involucra los siguientes administrar:
 - *Administrar Personal:* Se puede adicionar un nuevo personal además de asignarle su rol y su login y clave, también se puede modificar y ver los datos introducidos.
 - *Administrar Personal de Baja:* Se puede Dar de Alta a un personal y se puede Ver su Historial de cuando a sido dado de Baja o Alta.
 - *Administrar Roles:* Se puede adicionar un nuevo rol, modificar, eliminar y ver los datos introducidos del nuevo rol.
 - *Administrar Requisitos:* Se puede adicionar un nuevo Requisito, modificar, eliminar y ver los datos introducidos del nuevo Requisito.
 - *Administrar Tipo Trámite:* Se puede adicionar un nuevo Tipo Tramite, modificar, eliminar y ver los datos introducidos del nuevo Tipo Tramite.
- *Modulo área de Proyectos:* En este módulo se lleva un registro general de los propietarios, con sus respectivos proyectistas, personal involucrado, roles, por lo que este modulo involucra los siguientes administrar:
 - *Administrar Propietario:* Se puede adicionar un Propietario con su respectivo proyectista y apoderado, modificar los datos del Propietario y ver sus datos.
 - *Administrar Ingresos Salidas:* Se puede seleccionar a un Propietario y registrar su Ingreso y si por algún motivo el Trámite tiene que salir para poder subsanar sus observaciones se registra su Salida y se puede ver los Ingresos y Salidas que tuvo

ese Trámite.

- *Administrar Envíos Recepción:* Se puede seleccionar a un Propietario y registrar su *Envíos* si por algún motivo tiene que ir a otro departamento y cuando vuelve se registra su *Recepción* y se puede ver a donde fue *Enviado* y por quien fue Recepcionado dicho Trámite.
 - *Asignación:* Se Asigna al personal que va a estar encargado de revisar dicho Trámite.
 - *Reasignación:* Se Reasigna a un nuevo personal a un Trámite para que este a su cargo y pueda dar las observaciones.
-
- *Módulo Administrar Observaciones:* Este módulo lleva un registro general de los propietarios, con sus respectivos Observaciones, esto es dependiendo del Usuario que ingrese al sistema puede ser un técnico, topógrafo o un asesor legal.
 - *La parte de Cambiar Contraseña:* Este parte carga el Usuario y la Contraseña y se puede cambiar la contraseña y se la debe verificar antes de aceptar.
 - *La parte de Resguardar la Base de Datos:* Se puede examinar el lugar en donde se va a guardar el resguardo de la Base de Datos
 - *La parte de Restaurar la Base de Datos:* Se debe examinar el lugar en donde se va a Restaurar de la Base de Datos.

Por otro lado se realizó una propuesta de implementación de un Módulo SIG para Gestionar los datos de los predios de los propietarios y generar mapas generalizados con los respectivos datos de los propietarios.

El sistema pretende llegar a cubrir las necesidades básicas de la institución y en particular del Área de Proyectos y Lotes de ORDENAMIENTO TERRITORIAL.

II.1.7. Suposiciones y Restricciones

Las suposiciones son:

II.1.7.1 Las suposiciones son:

- La institución cuenta con equipos en los cuales se podrá implementar el sistema.
- Cuenta con un sistema de cableado de Área local.
- Tiene un personal capacitado para el manejo de equipos de computación.
- Se trabajara con los mejores equipos óptimos, capaces, y con las mejores herramientas para el desarrollo de software.
- Si se quiere implementar un módulo SIG ver propuesta de solución de implementación de un módulo SIG.(Ver Anexo - Propuesta).
- Información actualizada.
- Obtención de información, ágil y rápida.
- Verificación y disponibilidad de datos.
- Reducción y detección de errores en cuanto a la información.
- No existirá pérdida de información de acuerdo al control y seguridad que requiera el software.

Las restricciones del proyecto son:

II.1.7.2 Las restricciones son:

- No se controlara la asistencia del personal administrativo ni de servicio.
- No se retendrá información de la cantidad de personas que visitan diariamente Ordenamiento Territorial.
- No se podrá visualizar los planos, los lotes, la vivienda ni la construcción.
- El sistema no soporta un módulo contable.
- El sistema requiere capacitación y mantenimiento.
- El sistema será implementado en las plataformas Windows o Linux.

- No contempla servicio de conexión en red.
- No contempla un módulo SIG.
- El sistema será diseñado según los requerimientos de Ordenamiento Territorial.

II.1.8. 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.

Todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración del proyecto está enfocado a conseguir un cierto grado de completitud y estabilidad de los artefactos.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.) que 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 empleando estereotipos específicos para este modelo.

3) Modelo de Objetos del Negocio

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

negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

4) Requerimientos Funcionales

En este apartado se describen los requerimientos funcionales del sistema, la funcionalidad o los servicios que se espera que éste provea.

5) Glosario

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

6) 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.

7) Modelo de Casos de Uso

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

8) Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa), se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales

asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

9) Modelo de Análisis y Diseño

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

10) Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos orientada a objetos, este modelo describe un Diagrama de Clases de Objetos y luego su correspondiente implementación en el lenguaje utilizado para el desarrollo del proyecto el cual es JAVA.

11) 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.

12) Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. Refinándose el modelo de acuerdo al avance del proyecto.

13) Casos de Prueba

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

14) Material de Apoyo al Usuario Final

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

15) Producto

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación.

II.1.8.1. Organización del Proyecto

II.1.8.1.1. Participantes en el Proyecto

Jefe de Proyecto.

Universitaria Irka Maria Jallaza Garnica, alumna del último curso de la carrera de Ingeniería Informática en la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Con una experiencia modesta en metodologías de desarrollo, herramientas CASE y notaciones, en particular la notación UML y el proceso de desarrollo RUP.

Analista de Sistemas.

Universitaria de Ingeniería Informática con conocimientos de UML, con experiencia en sistemas afines a la línea del proyecto, labor que llevará a cabo Irka Maria Jallaza Garnica.

Analistas - Programadores.

Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final. Este trabajo ha sido encomendado a Irka Maria Jallaza Garnica.

Ingeniero de Software.

El perfil establecido estará a cargo de la estudiante de Informática que

participará realizando labores de Gestión de requisitos, Gestión de configuración, documentación y diseño de datos. Encargada de las pruebas funcionales del sistema, realizará la labor Irka Maria Jallaza Garnica.

II.1.8.1.2. Interfaces Externas

El director de Ordenamiento Territorial, definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada subsistema y según el plan establecido.

El equipo de desarrollo interactuará activamente con el personal del Área de Proyectos y Lotes de Ordenamiento Territorial para especificación y validación de los artefactos generados.

II.1.8.1.3. Roles y Responsabilidades

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

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.

Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

Tabla N° 27: Roles y Responsabilidades.

II.1.8.2. Gestión del Proceso

II.1.8.2.1. Estimaciones del Proyecto

II.1.8.2.1.1. Presupuesto / Justificación

Se desea estimar los costos que tendrá el sistema de Informático para el Proceso de Seguimiento de Trámites para Ordenamiento Territorial.

Utilizaremos el método **COCOMO** para estimación de costos y para esto tenemos los siguientes datos: El sistema estima producir 3,5 **MIFE** (miles de instrucciones fuentes) se pretende que el sistema tenga una complejidad media y para esto se contratará analistas y programadores que tengan conocimientos medios en este tipo de programación del sistema (cobrando alrededor de 400 US\$) pretende brindar una confiabilidad alta utilizar recursos de base de datos muy altos y una complejidad normal.

Método Cocomo Básico

Desarrollo:

Datos: Tamaño=3.5 MIFE Tarifa mensual= 400 US\$ COSTO=EN*tarifa mensual

Para calcular el Esfuerzo Nominal vamos a utilizar la siguiente fórmula:

$$EN=2.4*(MIFE)^{1.05}$$

Remplazando datos nos dará $EN=2.4*(3,5)^{1.05}$ entonces el esfuerzo nominal será:

$$EN=9 \text{ meses de programador.}$$

Para calcular el Tiempo de Desarrollo vamos a utilizar la siguiente fórmula:

$$TDES=2.5*(EN)^{0.38}$$

Remplazando datos nos dará $TDES=2.5*(9)^{0.38}$ entonces el tiempo de desarrollo será:

$$TDES=6 \text{ meses para su desarrollo.}$$

$$COSTO=9*400= \text{US\$ costo estimado del Sistema.}$$

Factor Multiplicador	Razonamiento	Valor
Confiabilidad	Nominal	1
Base de Datos	Nominal	1
Tiempo	Nominal	1
Memoria	Alto	1.06
Maquina Virtual	Nominal	1
Tiempo de Retorno	Bajo	0.87
Analistas	Alto	0.86
Programadores	Alto	0.86
Experiencia	Nominal	1
Experiencia	Nominal	1
Experiencia	Nominal	1
Practica	Muy Alto	0.82
Herramientas	Alto	0.91
Calendario	Alto	1.04
Complejidad	Muy Alto	1.30
Factor de Ajuste		0.7

Tabla N°: 28

Método Cocomo Intermedio

Desarrollo:

Con el fin de **ajustar las estimaciones** con respecto a desviaciones del valor inicial obtenido se utiliza el factor de ajuste el cual se calcula con los datos arriba descritos y el cual nos da: **FA=0.7**

Para calcular el esfuerzo nominal ajustado vamos a utilizar la siguiente formula.

$$PM=2.4*(MIFE)^{1.05} *FA$$

Remplazando datos nos dará **PM=2.4*(3,5)^{1.05}*0.7** entonces el esfuerzo nominal

ajustado será:

PM=6 meses de programador.

Para calcular el tiempo de desarrollo ajustado vamos a utilizar la siguiente formula.

$$TDES=2.5*(PM)^{0.38}$$

Remplazando datos nos dará **TDES=2.5*(6)^{0.38}** entonces el tiempo de desarrollo ajustado será:

TDES=6 meses para su desarrollo.

COSTO=PM*costo mensual=6*400=2400 US\$ costo estimado del proyecto.

II.1.8.2.1.2. Presupuesto de Gastos

Para que el sistema logre los objetivos previstos anteriormente detallaremos el gasto que se realizara por cada mes del tiempo de desarrollo del sistema, como se realizo una estimación de costo del sistema y del tiempo que durara el sistema nos basaremos en esto para detallar el gasto del sistema por cada mes que dure el mismo. En el desarrollo de dicho sistema consideraremos lo siguientes recursos básicos como ser Costo Analistas Costo Programador Costo Hardware Costo Software. A continuación describiremos cada uno de estos costos:

Costo Analista.- Este costo se refiere al precio que pagaremos por el servicio del analista a lo largo del desarrollo del sistema.

Costo Programador.- Este costo se refiere al precio que pagaremos por el servicio del programador a lo largo del desarrollo del sistema.

Costo Hardware.- Este costo se refiere al precio que pagaremos por el alquiler del equipo como computadoras, impresoras y otros, en el tiempo que dure el desarrollo del sistema.

Costo Software.- Este costo se refiere el precio que pagaremos por las diferentes herramientas de software que ocuparemos en el desarrollo del proyecto.

MESES	COSTO HARDWARE	COSTO ANALISTA	COSTO PROGRAMADOR	COSTO * SOFTWARE	COSTO TOTALES
1	50	200	-	0	250
2	70	200	-	0	270
3	70	100	100	0	270
4	70	100	400	0	570
5	70	50	400	0	520
6	70	50	400	0	520
TOTAL	400	700	1300	0	2400 U\$\$

Tabla N° 29

II.1.8.2.2. Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

II.1.8.2.2.1 Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas.

La siguiente tabla muestra una distribución de tiempo y el número de iteraciones de cada fase de forma preliminar.

Fase	Nro. Iteraciones	Duración (días)
Fase de Inicio	1	40
Fase de Elaboración	2	72

Fase de Construcción	3	115
Fase de Transición	1	39

Tabla N° 30: Cuadro de iteraciones de las fases RUP del proyecto

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

Descripción	Hito
Fase de Inicio	En esta fase desarrollarán los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del Cliente /usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera y única iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos.
Fase de	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto

Construcción	se construye en base a 1 iteración, en la cual se produce una release a la cual se le aplican las pruebas y se valida con el Cliente / usuario. Se comienza la elaboración de material de apoyo al usuario.
Fase de Transición	En esta fase se prepararán los releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla N° 31: Cuadro de hitos del fin de fases del proyecto.

II.1.8.2.2.3 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto, pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

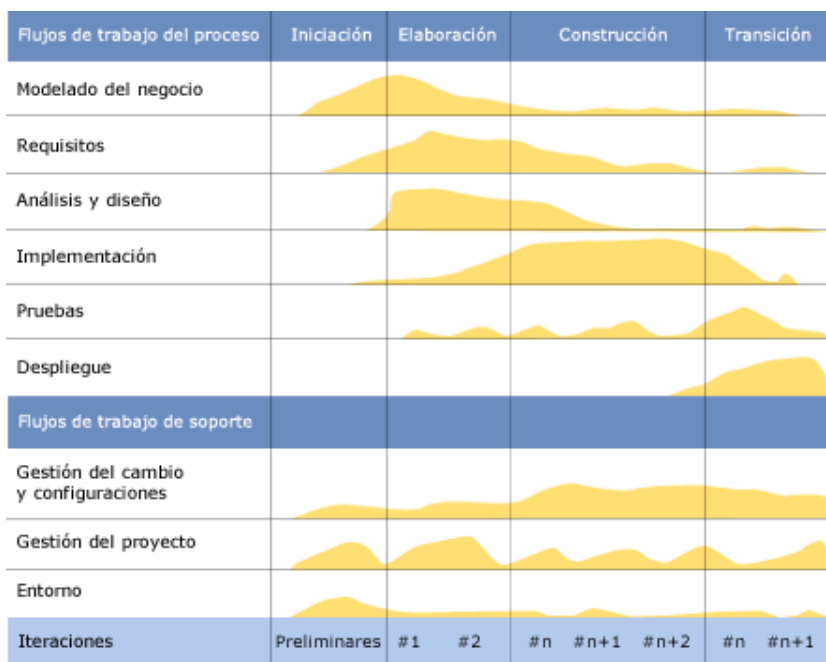


Figura N° 32: Iteraciones RUP.

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	19-04-2009	29-04-2009
Requisitos		
Glosario	15-04-2009	19-06-2009
Visión	19-04-2009	19-06-2009
Modelo de Casos de Uso	19-04-2009	19-06-2009
Análisis / Diseño		
Modelo de Análisis / Diseño	21-06-2009	24-09-2009
Modelo de Datos	21-06-2009	24-09-2009

Implementación		
Prototipos de Interfaces de Usuario	25-07-2009	24-10-2009
Modelo de Implementación	29-10-2009	25-11-2009
Pruebas		
Casos de Pruebas Funcionales	18-11-2009	28-11-2009
Despliegue		
Modelo de Despliegue	1-12-2009	23-12-2009
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de desarrollo de Software	23-03-2009	17-04-2009
Ambiente	Durante todo el proyecto	

Tabla N° 33: Calendario de fechas de los artefactos del proyecto.

II.1.8.2.2.4. Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto.

N°	Actividad	N° días	Fecha Inicio	Fecha Fin
	C1. Sistema de Escritorio			
1	Análisis y Especificación de requerimientos <i>En la presente actividad se llevará a cabo el análisis y especificación de requerimientos, mediante la realización de entrevistas, cuestionarios y plasmarlos mediante diagramas de casos de uso.</i>	14	19-04-2009	29-04-2009

2	Diseño del Sistema <i>Realizar el diseño del Sistema de Escritorio mediante diagramas UML hasta el diagrama de clases, se analizará y se diseñará la estructura en la cual la información del sistema será almacenada.(Se realiza el modelo Orientado a Objetos de la Base de Datos).</i>	21	30-04-2009	05-06-2009
3	Desarrollo del Sistema <i>Se realizará la programación y pruebas para la validación del Sistema de Escritorio.</i>	14	06-06-2009	12-11-2009
C2. Estrategias de Socialización				
4	Formación al personal en el uso de las TIC para el manejo del producto final. <i>Se realizará la formación a los usuarios finales en el uso de las TIC para el manejo del producto final, mediante un curso de capacitación previamente planificado.</i>	14	18-11-2009	08-12-2009

5	Definir los medios y llevar a cabo las estrategias de socialización del producto final. <i>Se realiza la socialización del proyecto a los gerentes o representantes y personal de las pequeñas y medianas empresas, por medio de afiches y volantes.</i>	20	09-12-2009	20-12-2009
---	--	----	------------	------------

Tabla N° 34: Calendario del Proyecto

II.1.1.5. Referencias

- Documentación de Rational Unified Process, manuals de ayuda, tutoriales, etc.
- Diseño de Aplicaciones de Escritorio usando UML, Por Miguel Reynolds
- UML y Patrones. Craig Larman
- Visual Modeling with Rational Rose and UML, Terry Quatrani. - Addison-Wesley.

II.1.9. REQUERIMIENTOS FUNCIONALES

II.1.9.1. Introducción

En este apartado se describen los requerimientos funcionales del sistema, la Funcionalidad o los servicios que se espera que éste provea.

II.1.9.1.1. Descripción Textual de los Requerimientos Funcionales

ID del Requerimiento:	REQ-M01
Nombre del Requerimiento:	Contar con un medio sencillo y económico que atienda las necesidades de gestión de Información para el Proceso de Seguimiento de Trámites para <i>Ordenamiento Territorial</i> .
Características:	El personal autorizado podrá registrarse en el sistema. Se debe tomar las precauciones debidas, como ser la validación de datos y confirmación de que el usuario que está ingresando se encuentra registrado en la Base de Datos. Una vez registrado y confirmado el usuario, podrá gestionar los procesos administrativos que él requiera de una forma asistida para facilitar su manejo.
Atributo: Prioridad • Normal	

Tabla N° 35

ID del Requerimiento:	REQ-A01
Nombre del Requerimiento:	Otorgar facilidad de ingreso al sistema minimizando los requisitos mediante la computadora.

Características:	El sistema deberá funcionar con los requisitos mínimos de software y hardware. Permitirá el acceso a los servicios que cuenta a través de la computadora con la que cuente <i>Ordenamiento Territorial</i>.
Atributo: Prioridad • Normal	

Tabla N° 36

ID del Requerimiento:	REQ-A02
Nombre del Requerimiento:	Permitir al usuario administrador crear roles de acceso.
Características:	El sistema permitirá al administrador del mismo crear roles a los que se le asignen diferentes niveles de acceso a los módulos del sistema, para otorgárselos a los otros usuarios, certificando de esta manera la seguridad de la información de <i>Ordenamiento Territorial</i> .
Atributo: Prioridad • Normal	

Tabla N° 37

ID del Requerimiento:	REQ- A03
Nombre del Requerimiento:	Administrar los datos de todos los propietarios y controlar sus ingresos salidas de sus respectivos trámites.

Características:	El usuario podrá Administrar los datos de todos los propietarios tanto los de N^a Previo, dirección, como también sus relaciones con sus apoderados y proyectistas. Deberá existir un registro de cuando ingresa un trámite como también una sección donde se pueda ver el registro del ingreso y por quien a sido registrado. Deberá existir un registro de cuando sale un trámite como también una sección donde se pueda ver el registro de la salida y por quien a sido registrado esa salida.
Atributo: Prioridad • Normal	

Tabla No. 38

ID del Requerimiento:	REQ- A04
Nombre del Requerimiento:	Dar de alta al personal mediante una ficha que permita tener un control de todos sus datos.
Características:	El usuario podrá administrar los datos del personal ya que la información sobre éstos ha cobrado mayor importancia, de los que se hace necesario conocer los datos de identificación, las características de los movimientos físicos mantenidos hasta el momento y otros.
Atributo: Prioridad • Normal	

Tabla N° 39

ID del Requerimiento:	REQ-A05
Nombre del Requerimiento:	Permitir al usuario obtener un resguardo de la Base de Datos como medida de seguridad, así como también poder restaurar la misma en el caso de que así lo requiera el usuario.
Características:	El sistema tendrá una opción para resguardar los datos, en el momento que el usuario así lo requiera. También presentará la opción de restaurar la copia de la base de datos antes resguardada.
Atributo: Prioridad • Normal	

Tabla N° 40

ID del Requerimiento:	REQ-M02
Nombre del Requerimiento:	Permitir la Administración del personal que va a interactuar con el sistema desde el registro de sus respectivos datos.
Características:	El usuario registrado podrá administrar al personal en el sistema, mediante una buena organización y sabio tratamiento de la información. Donde al momento de registrar un nuevo personal ingrese sus datos respectivos, fechas, el rol que va a desempeñar y luego se pueda ver estos datos.
Atributo: Prioridad • Normal	

Tabla N° 41

ID del Requerimiento:	REQ-M03
Nombre del Requerimiento:	Administrar Personal de Baja.
Características:	El sistema deberá permitir dar de baja a un empleado que ya no va a trabajar en la institución e ingresar el motivo y el detalle del por que está dejando de trabajar, además de poder ver los datos introducidos y el poder dar de alta a este empleado si volviera a trabajar de nuevo en esta institución.
Atributo: Prioridad • Normal	

Tabla N° 42

ID del Requerimiento:	REQ-C01
Nombre del Requerimiento:	Administrar Roles.
Características:	El sistema deberá permitir agregar nuevos roles como también así modificar un rol por si en algún momento se introdujo un dato mal, ver sus respectivos datos introducidos y también podemos eliminar un rol.
Atributo: Prioridad • Normal	

Tabla N° 43

ID del Requerimiento:	REQ-C02
Nombre del Requerimiento:	Administar los envios y recepcioneas de los Trámites
Características:	El sistema deberá llevar un registro detallado del envío y recepción de los trámites permitiendo registrar, modificar y ver cualquier envío o recepción de un trámite. Además se podrá ver los datos de que usuario autorizado fue el que registro el envío o recepción del trámite, además de ver los datos del propietario y el tipo de trámite.
Atributo: Prioridad • Normal	

Tabla N° 44

ID del Requerimiento:	REQ-C03
Nombre del Requerimiento:	Obtener reportes detallados de los usuarios, los trámites de baja o alta y otros.
Características:	El sistema deberá arrojar reportes de los usuarios, los trámites de baja o alta y otros; según lo que necesite el usuario, en forma detallada y clara.
Atributo: Prioridad • Normal	

Tabla N° 45

ID del Requerimiento:	REQ-M04
Nombre del Requerimiento:	Administrar Tipo de Trámite.
Características:	El sistema deberá permitir agregar nuevos tipos de trámites como también así modificar un tipo de trámite por si en algún momento se introdujo un dato mal, ver sus respectivos datos introducidos y también podemos eliminar un tipo de trámite.
Atributo: Prioridad • Normal	

Tabla N° 46

II.1.9.5. Referencias

Encuestas realizadas a: El personal del Área de Proyectos y Lotes de Ordenamiento Territorial.

II.1.10. GLOSARIO

II.1.10.1 Introducción

Este documento recoge términos manejados durante la elaboración del proyecto de desarrollo del sistema para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial, se trata de un diccionario informal de datos y de definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para el proyecto.

II.1.10.1.1. Propósito

El propósito de este glosario es definir la terminología manejada en el proyecto de desarrollo del sistema para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial. También sirve como guía de consulta para la clarificación de los puntos conflictivos o pocos esclarecedores del proyecto.

II.1.10.1.2. Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos por el personal del área de Proyectos y Lotes de Ordenamiento Territorial. De tal modo que la terminología empleada en la administración reflejen con claridad en este documento.

II.1.10.2 Referencias

El presente glosario hace referencia a los siguientes documentos:

- Documento plan de desarrollo de Software del Sistema de Informático desarrollado para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial.

II.1. 10.3 Organización del Glosario

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

II.1. 10.4 Definiciones

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto –Fortalecimiento de la Gestión de Información en el Proceso de Seguimiento de Trámites para Ordenamiento Territorial “G.I.P.S.E.T”||.

II.1. 10.1 Apoderado.- Persona que realiza el trámite con el consentimiento del propietario, este apoderado debe estar debidamente identificado.

II.1. 10.2 Asesor Legal.- Es un abogado que se encuentra en cada área.

II. 1.10.3 Coordinador.- Es el encargado de coordinar todas las área pertenecientes

a Ordenamiento Territorial.

II. 1.10.4 Comprobante.- Es un papel que sirve de constancia de haber pagado la multa o aprobación de trámite en el banco.

II. 1.10.5 Documento.- Se llama Documento al Fólder que ingresa por primera vez a cualquier área de Ordenamiento Territorial.

II. 1.10.6 Estaqueado de Lote.- Se refiere a la medición de Lote que realiza el Topógrafo, este es medido con estacas que muestran los límites del Lote.

II. 1.10.7 Ingreso Previo.- Es la primera vez en la que los documentos del trámite.

II. 1.10.8 Inspector.- Persona encargada de ir a verificar las anomalías que puede tener un lote o si en realidad es una construcción de vivienda, un levantamiento de vivienda, ampliación de vivienda.

II. 1.10.9 Línea Municipal.- Es la línea que se marca Desde el Lote hacia la calle, este marca el espacio que debe ceder el propietario para la calzada de la calle. Solo con después de medir esta línea el propietario puede cerrar su Lote.

II. 1.10.10 Nota.- Es un papel en el que están los datos del monto a pagar que puede ser por concepto de multa o aprobación de trámite.

II. 1.10.11 Número de Trámite.- Número que se asigna a todos los Trámites aprobados es el número oficial del trámite.

II. 1.10.12 Número Previo.- Número que se asigna a cada documento que ingresa.

II. 1.10.13 OK.- Es el término que usan para decir que el Trámite esta aprobado.

II. 1.10.14 Planeamiento.- Es el plano que se presenta además de otros requisitos; en el documento este muestra los límites del Lote del Propietario.

II. 1.10.15 Propietario.- Persona que es dueña de un lote y quiere realizar un trámite.

II. 1.10.16 Proyectista.- Es un arquitecto, quien realiza todos los trámites para el propietario, este puede trabajar en alguna consultora, etc. Este no puede recoger el trámite aprobado, ya que es un privilegio del apoderado o del propietario.

II. 1.10.17 Recepcionista.- Persona encargada de recepcionar los trámites referidos

a su área y asignar numero previo o numero de trámite.

- II. 1.1018 Revisión Previa.-** Es la revisión que se da al documento, como ser la revisión Técnica y la revisión Legal.
- II. 1.10.19 Revisión Técnica.-** Es la revisión que realizan los Técnicos de cada área a los planos de cada trámite.
- II. 1.10.20 Sobre posición.-** Cuando el propietario realiza su plano de lote o su construcción fuera de los límites de su propiedad.
- II. 1.10.21 Subdivisión de Lote.-** Este trámite se realiza cuando los existe un lote que quiere ser dividido entre varias personas. (Este trámite tiene sus requisitos específicos).
- II. 1.10.22 Técnico.-** Es el arquitecto encargado de revisar tramites precedentes y dar el visto bueno.
- II. 1.10.23 Topógrafo.-** Persona encargada ir a medir los Lotes y del estaqueado de lotes.
- II. 1.10.24 Trámite.-** Se llama trámite a la documentación en proceso.
- II. 1.10.25 Ventanilla Única.-** Es el lugar en el que los propietarios o apoderados pueden recoger sus trámites, presentando previamente sus boletas de pago ya sea de multas o solo de aprobación del plano.
- II. 1.10.26 Reporte.-** Es el documento caracterizado por contener información u otra materia reflejando el resultado de una investigación adaptado al contexto de una situación y de una audiencia dadas. Generalmente, su propósito es informar. Sin embargo pueden incluir elementos persuasivos, tales como recomendaciones, sugerencias u otras. [6]
- II. 1.10.27 Administración.-** Es el proceso de planificar, organizar, dirigir y controlar el uso de los recursos de la organización (humanos, financieros, tecnológicos, materiales, de información) y las actividades de trabajo con el propósito de lograr los objetivos o metas de la organización de manera eficiente y eficaz. [7]
- II. 1.10.28 TIC.-** Tecnologías de la Información y Comunicación

II. 1.10.29 Requerimientos funcionales.- Se refiere a la funcionalidad o los servicios que se espera que el sistema provea.

II. 1.10.30 Requerimientos no funcionales.- Los requerimientos no funcionales tienen que ver con las características que de una u otra forma puedan limitar el sistema como son: el rendimiento (en tiempo y espacio),

II.1.4. VISIÓN

II.1.4.1. Introducción

II.1.4.1.1. Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las características del Sistema de Escritorio desarrollado para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

Esta funcionalidad se basa principalmente en la administración, propietarios, ingresos, salidas, recepción y envío con los que cuenta el área de Proyectos y Lotes.

II.1.4.1.2. Alcance

El documento visión se ocupa, como ya se ha apuntado, del Sistema de Escritorio desarrollado para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial. Dicho sistema será desarrollado por la Universitaria: Irka Maria Jallaza Garnica. Alumna de quinto año de la carrera de Ingeniería Informática, facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Caracho (U. A. J. M. S.) Tarija.

El sistema permitirá al personal del Área de Proyectos y Lotes Administrar Los propietarios que realizan sus trámites desde registrar sus ingresos, salidas, envíos , recepción y también administrar su personal que trabaja en esta área.

II.1.4.2. Definiciones Acrónicas

RUP: Son las siglas de *Rational Unified Process*. Se trata de una metodología para describir el proceso de desarrollo del software y junto con el Lenguaje Unificado de Modelado UML.

II.1.4.3. Referencias

- Glosario.
- Plan de desarrollo del software.
- RUP (*Rational Unified Process*).
- Diagrama de casos de uso.

II.1.4.4. Seguimiento y Control del Proyecto

II.1.4.4.1 Gestión de Requisitos

De acuerdo al tiempo de vida de la realización del proyecto se irá analizando los siguientes requisitos.

Requisitos Organizacionales.-

El personal debe seguir como estándar la metodología RUP bajo un modelado con el lenguaje UML.

Se debe contar con personal que conozca sobre la plataforma java, programación, manejo del gestor de Datos Postgres, generación de reportes mediante IREPORT, manejo de componentes de Java.

Requisitos de Personal y Usuarios.-

El Usuario debe tener interés de participación en el desarrollo del sistema.

Los Usuarios deben contar con un nombre de Usuario y clave para poder realizar aportes de información.

Requisitos Físicos y de funcionamiento.-

Para el funcionamiento se debe contar con un equipo que soporte la base de datos y que tenga instalado la maquina virtual de java. El usuario debe contar con un equipo de Computación PIII con 256MB de RAM o posterior con conexión en red con los demás equipos de Ordenamiento Territorial.

Requisitos de Construcción.-

Se debe contar con 2 equipos PIV de 3.2GHZ con 1GB de RAM o superior.

En software se debe contar con un servidor.

Requisitos de Software.-

Maquina virtual java o JRE versión 1.5 en adelante.

II.1.4.4.2 Control de Plazos

Fase	Nº de Iteración	Inicio
Inicio	Primera	23-03-2009
Elaboración	Primera	20-05-2009
Construcción	Primera	24-07-2009

Transición	Primera	25-11-2009
------------	---------	------------

Tabla N° 47: Tabla de control de plazos.

II.1.4.4.3 Control de Calidad

Nuestro sistema de control de calidad se basa en los puntos siguientes:

- Detectar problemas.
- Delimitar el área problemática.
- Estimar factores que probablemente provoquen el problema.
- Determinar si el efecto tomado como problema es verdadero o no.
- Prevenir errores debido a omisión, rapidez o descuido.
- Confirmar los efectos de mejora.
- Detectar desfases.
- Realizar pruebas en cada versión.

Objetivos del control de calidad

- Aumentar la satisfacción del Cliente
- Equilibrar el esfuerzo en múltiples demandas
- Obtener el mejor producto
- Disfrutar de una ventaja competitiva

- Disponer de métricas objetivas de valoración
- Ahorrar tiempo y dinero

II.1.4.4.4 Gestión de Riesgos

Estos riesgos son comunes a todos los proyectos.

Relacionados con el tamaño del producto.

- Es posible no poder llegar a terminar todos los componentes del proyecto debido a que estos son muchos.

Relacionados con el impacto en la organización.

- No entregar el proyecto en el tiempo estimado.
- El posible no uso del software debido a los requerimientos de equipos.
- Poco uso del software es decir por pocas personas o instituciones.
- Gran cantidad de documentación a entregar al Cliente.
- Los límites legales y gubernamentales en cuanto al desarrollo y funciones del proyecto.
- Posibles errores en el producto y costos asociados.

Relacionados con el tipo de Cliente.

- Si se tiene la disponibilidad de tiempo para la especificación formal de requerimientos.
- Si están dispuestos a participar de las pruebas o revisiones.
- Si se relacionará de forma ágil con el grupo de desarrollo.

Relacionados con la definición del proceso de producción.

- El software a ser utilizado en el control del desarrollo del proyecto.
- Las herramientas a ser utilizadas en el análisis y diseño.

Relacionados con el entorno de desarrollo.

- Si hay herramientas de gestión de proyectos.
- Hay herramientas de prueba apropiadas.
- Generadores de código para la aplicación.

Relacionados con la tecnología.

- Es una nueva tecnología.
- El hardware con el que debe interactuar es nuevo o cumple la expectativas.
- La base de datos a ser utilizada ha sido probada y tiene la funcionalidad y rendimiento.
- Las interfaces son especializadas.
- La necesidad de nuevos componentes.

Relacionados con la experiencia y tamaño del equipo.

- Es el mejor personal disponible.
- Los miembros tienen las técnicas apropiadas.
- Hay suficiente gente disponible.
- El personal está comprometido a lo largo de desarrollo del proyecto.
- Tiene el personal las expectativas correctas del trabajo.

II.1.4.4.5 Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean

accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá una baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada sólo por una Solicitud de Cambio aprobada.

II.1.4.5 Referencias

- [1] <http://www.calidaddelsoftware.com/modules.php?name=News&file=article&sid=66;>
- [2] IBM Press Project Management with the IBM Rational Unified Process Jul.2006 eBook.
- [3] Addison Wesley Managing Iterative Software Development Projects Jun 2006 eBook
- [4] [http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm;](http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm)
- [5] [http://www.tecnyce.com.mx/ti/cursosN.php?ids=641;](http://www.tecnyce.com.mx/ti/cursosN.php?ids=641)
- [6] [http://html.rincondelvago.com/analisis-y-diseno-de-sistemas_1.html;](http://html.rincondelvago.com/analisis-y-diseno-de-sistemas_1.html)
- [7] [http://es.wikibooks.org/wiki/Inform%C3%A1tica_Educacional/Seguimiento_y_Control_del_Proyecto;](http://es.wikibooks.org/wiki/Inform%C3%A1tica_Educacional/Seguimiento_y_Control_del_Proyecto)

II.1.4.6. Descripción Global del Producto

II.1.4.6.1. Perspectiva del producto

El producto a desarrollar es Sistema de Escritorio desarrollado para la gestión de información en el proceso de seguimiento de trámites para Ordenamiento Territorial.

II.1.4.6.2. Resumen de características

A continuación se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

Beneficios directos: Mejora en la atención a los propietarios que realizan sus trámites en esta área.

Beneficios indirectos: Ordenamiento Territorial y en especial el Área de Proyectos y lotes obtendrán un mayor peso en la población y mayor impacto como institución.

CAPITULO I:

MARCO TEÓRICO

CAPITULO I: MARCO TEORICO

I.1.1 Definiciones

A continuación se presentan algunos términos que se manejarán a lo largo de todo el proyecto de desarrollo: Ver (II.1.3. GLOSARIO)

II.1.2 Sistema de Información Computarizado

Un Sistema de Información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio, para proporcionar información básica para la conducción de la empresa. Esta información hace posible que las empresas lleven a cabo sus tareas con mayor calidad y facilidad.

Un Sistema de Información realiza cuatro actividades básicas:

- **Entrada de Información:** Proceso en el cual el sistema toma los datos que requiere para procesar la información, por medio de estaciones de trabajo, teclado, diskettes, cintas magnéticas, código de barras, etc.
- **Almacenamiento de Información:** Es una de las actividades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sesión anterior.
- **Procesamiento de la información:** Esta característica de los sistemas permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general en un año base.
- **Salida de Información:** Es la capacidad de un sistema para sacar información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas, y otros.[9]

II.1.3 Sistemas de Gestión de Base de Datos

Los sistemas de gestión de base de datos (SGBD); son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Proporciona servicios para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en base de datos de forma segura y eficiente. Un SGBD actúa como un intermediario entre las aplicaciones y los datos. [10]

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Ventajas

- Proveen facilidades para la manipulación de grandes volúmenes de datos. Entre éstas:
 - ✓ Simplifican la programación de chequeos de consistencia.
 - ✓ Manejando las políticas de respaldo adecuadas garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores en el disco, o hay muchos usuarios accediendo simultáneamente a los mismos datos, o se ejecutaron programas que no terminaron su trabajo correctamente, etc.
 - ✓ Permiten realizar modificaciones en la organización de los datos con un impacto mínimo en el código de los programas.
 - ✓ Permiten implementar un manejo centralizado de la seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- Las facilidades anteriores bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Inconvenientes

1. Típicamente, es necesario disponer de una o más personas que administren de la base de datos, en la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o más personas que administren de los sistemas operativos. Esto puede llegar a incrementar los costos de

operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.

2. Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una planilla de cálculo.
3. Complejidad: los SGBD son software muy complejos y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
4. Tamaño: la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder correr.
5. Coste del hardware adicional: los requisitos de hardware para correr un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero. fecha de lanzamiento. [11]

II.1.3.1 Funciones de los Sistemas de Gestión de Base de Datos

La función que desempeñan los SGBD, y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.[12]

II.1.3.2 Arquitectura de los Sistemas de Gestión de Base de Datos

Las bases de datos respetan la arquitectura de tres niveles definida, para cualquier tipo de base de datos, por el grupo ANSI/SPARC. En esta arquitectura la base de datos se divide en los niveles externo, conceptual e interno.

1. Nivel interno: es el nivel más bajo de abstracción, y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.

2. Nivel conceptual: es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Se configura como visión organizativa total, e incluye la definición de datos y las relaciones entre ellos.
3. Nivel externo: es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios. En cierto modo, es la parte del modelo conceptual a la que tienen acceso. [13]

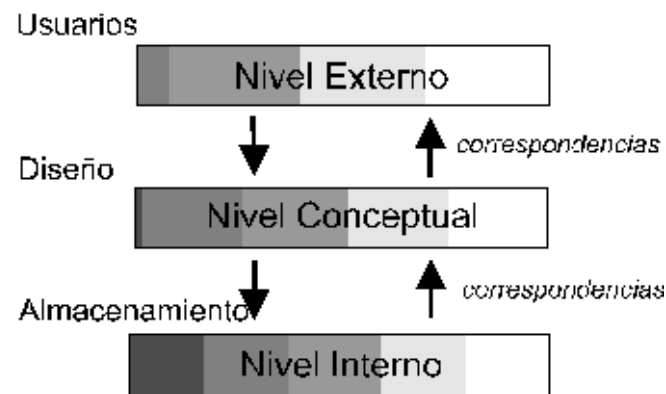


Figura 4: Niveles de la arquitectura de bases de datos.

II.1.4 Herramientas tecnológicas de Desarrollo

II.1.4.1 JAVA

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma.

Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para

distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

En aplicaciones de escritorio

Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PC de usuario 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 en las API's de desarrollo gráfico (AWT). Desde la aparición de la biblioteca Swing la situación mejoró substancialmente y posteriormente con la aparición de bibliotecas como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. sea relativamente sencillo.[14]

II.1.4.2 iReport

La herramienta iReport es un constructor y diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc.

Características de iReport

- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los Soporta internacionalización nativamente.
- Browser de la estructura del documento.
- Recopilador y exportador integrados.
- Soporta JavaBeans como orígenes de datos
- Tiene asistentes para las plantillas.
- Facilidad de instalación.

II.1.4.3 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada 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 organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Características

Algunas de sus principales características son, entre otras:

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.[15]

Otras características

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Disparadores (*triggers*): Un disparador o *trigger* se define en una acción específica basada en algo ocuriente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:
 - El nombre del disparador o *trigger*
 - El momento en que el disparador debe arrancar
 - El evento del disparador deberá activarse sobre...
 - La tabla donde el disparador se activará
 - La frecuencia de la ejecución
 - La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (*triggers*).

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

II.1.4.4 Modelo Vista Controlador (MVC)

MVC (Modelo-Vista-Controlador), en palabras simples, es la forma (Patron de Diseno) que utilizamos los programadores para implementar nuestras aplicaciones, ademas permite separar nuestra aplicacion en un modelo, una vista y un controlador. Este patron fue introducido por primera vez en el lenguaje “Smalltalk”. [16]

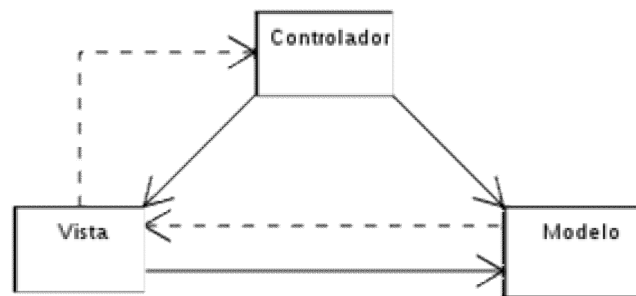


Figura 5: Modelo Vista Controlador (MVC).

Modelo: Es el encargado de administrar la logica de tu aplicacion. Tiene como finalidad servir de abstraccion de algun proceso en el mundo real, ademas tiene acceso a nuestra Base de Datos, agregando que tiene las funciones que controlan la integridad del sistema.

Vista: Sencillamente es la representacion visual del modelo. Es la encargada de representar los componentes visuales en la pantalla, Esta asociada a un Modelo, esto le permite que al momento de cambiar el Modelo, la vista redibujara la parte afectada para reflejar los cambios.

Controlador: Es el escuchador a los eventos que genere el usuario, es decir es el que permite que interactuen el usuario con el sistema. Interpreta los eventos (la entradas) atravez del teclado y/o raton.

II.1.4.4.1 Por que usar MVC?

Porque fue diseñada para reducir el esfuerzo al momento de programar. Además porque permite una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.

Permite el trabajo en equipo.

II.1.4.4.2 Java implementa MVC?

Con decir, que la mayoría de los componentes SWING, han tomado como patrón de Diseño a MVC, esto es una gran ventaja para los programadores, porque nos permite implementar nuestro PROPIO modelo de datos para cada componente swing.

Sin embargo es bueno recordar que El modelo original MVC prescribía un alto acoplamiento entre controladores y vistas. Por otro lado, mvc nos permite utilizar de la forma que queramos a cualquier componente swing con tan solo implementar nuestro propio modelo. Es decir nos ofrece la posibilidad de trabajar con componentes más eficientes.[17]

II.1.4.4.3 Responsabilidades del MVC (Modelo-Vista-Controlador)

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, un pastiche que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Elementos del patrón:

- Modelo: datos y reglas de negocio

- Vista: muestra la información del modelo al usuario
- Controlador: gestiona las entradas del usuario

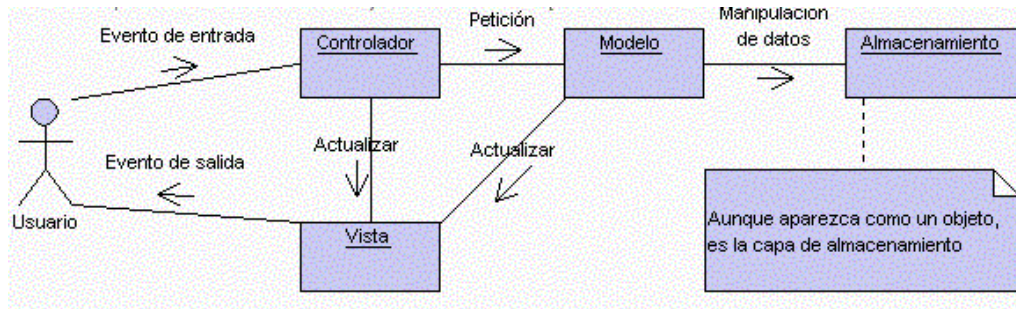


Figura 6: Elementos del patrón.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de tarta, de barras, tabular, etc. Veamos cada componente:

1. El **modelo** es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc).

2. El **controlador** es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones

al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega(nueva_orden_de_venta)".

3. Las **vistas** son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes). [18]

II.1.4.5. Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL. [19]

II.1.5 Especificación de Requerimientos

II.1.5.1 Metodologías de Desarrollo

Son métodos que indican cómo hacer más eficiente el desarrollo de sistemas de información. Para ello suelen estructurar en fases la vida de dichos sistemas con el fin de facilitar su planificación, desarrollo y mantenimiento.

Los principales objetivos de una metodología de desarrollo son:

- Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en sí.
- Satisfacer las necesidades de los usuarios del sistema.

- Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo.
- Ajustarse a los plazos y costes previstos en la planificación.
- Generar de forma adecuada la documentación asociada a los sistemas.
- Facilitar el mantenimiento posterior de los sistemas. [20]

II.1.2.2.1 Proceso de Ingeniería de Software

Ingeniería de software es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software.

- mejorar la calidad de los productos de software
- aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado. [21]

II.1.5.1 Lenguaje Unificado de Modelado UML

El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.



UML es apropiado para modelar desde sistemas de información en empresas hasta aplicaciones distribuidas basadas en la Web, e incluso para sistemas empujados de tiempo real muy exigentes. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar y luego desplegar tales sistemas. Aunque sea expresivo. UML no es difícil de aprender ni de utilizar. Aprender a aplicar UML de modo 'eficaz comienza por crear un modelo conceptual del lenguaje, lo cual requiere aprender tres elementos principales: los bloques básicos de construcción de UML, las reglas que dictan cómo pueden combinarse esos bloques y algunos mecanismos comunes que se aplican a lo largo del lenguaje.

UML es sólo un lenguaje y por tanto es tan sólo una parte de un método de desarrollo de software. UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.[22]

II.1.5.1.1 ¿QUÉ ES UML?

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (*workflow*) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.[23]

11.1.5.1.2 ¿Qué forma un modelo de UML?

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones. Veamos con mayor detalle los diagramas de UML.

Elementos

- Pueden ser estructurales, de comportamiento de agrupación o de anotación
- Elementos estructurales: actores, casos de uso, clases, objetos
- Elementos de comportamiento: mensajes
- Elementos de agrupación: paquetes

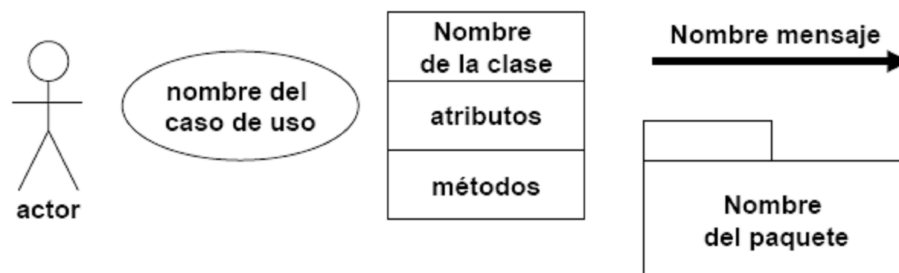


Figura 7: Los Elementos

Relaciones

Dependencia

–Es una relación semántica entre dos elementos, en la cual un cambio en un elemento puede afectar a la semántica de otro elemento

–Existen varios tipos de dependencia predefinidas que se indican mediante estereotipos <<extend>> o <<include>> para casos de uso

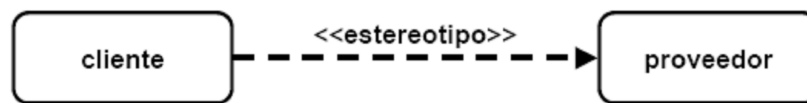


Figura 8: Asociación

Asociación

–Es una relación estructural entre dos elementos, que describen las conexiones entre ellos (suele ser bidireccional)

–Puede presentarse como agregación o composición

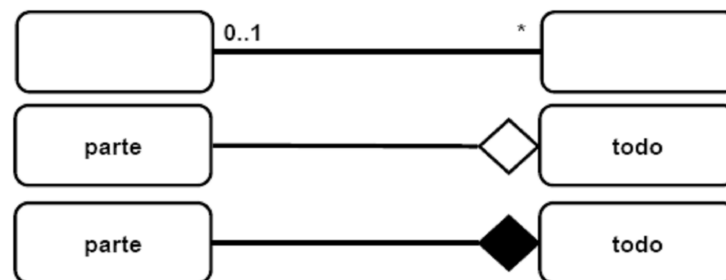


Figura 9: Agregación o Composición

Generalización

–Es una relación taxonómica entre un elemento más general (el padre) y un elemento más específico (el hijo)

–Se utiliza tanto en diagramas de clases como en diagramas de casos de uso.[24]

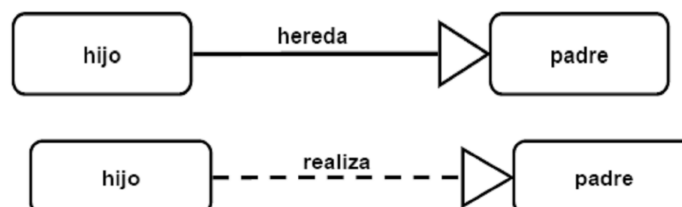


Figura 10: Generalización

II.1.5.1.3 Diagramas de UML

Los diagramas proporcionan múltiples perspectivas del sistema bajo análisis. El modelo subyacente integra estas perspectivas de forma que se puede construir un sistema auto consistente. Estos diagramas, junto con la documentación de soporte, es lo primero que ve el diseñador.

Por otro lado podemos ver el modelo de una forma estática o de una forma dinámica. Estas perspectivas nos dan la siguiente clasificación:

- Modelo estático (estructural):
 - Diagrama de despliegue
 - Diagrama de componentes
 - Diagrama de clases
 - Diagrama de objetos
- Modelo dinámico (comportamiento):
 - Diagrama de estados
 - Diagrama de actividades
 - Diagrama de secuencia
 - Diagrama de colaboración
 - Diagrama de casos de uso

II.1.5.1.3.1 Diagramas de Clases

El *diagrama de clases* muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas. [25]

II.1.5.1.3.2 Diagramas de Componentes

Un **diagrama de componentes** es un diagrama tipo del Lenguaje Unificado de Modelado.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Debido a que estos son más parecidos a los diagramas de casos de usos estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

II.1.5.1.3.3 Diagramas de Despliegue

El **Diagrama de Despliegue** es un tipo de diagrama que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Además muestra un conjunto de nodos y sus relaciones. Los diagramas de despliegue se utilizan para describir la vista de despliegue estática de un sistema. Los diagramas de despliegue se relacionan con los diagramas de componentes, ya que un nodo normalmente incluye uno o más componentes

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

II.1.5.1.3.4 Diagramas de Casos de Uso

El *diagrama de casos de usos* representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo

II.1.5.1.3.5 Diagramas de Iteración (Secuencia, Colaboración)

Los **diagramas de interacción** se dan este nombre colectivo a los diagramas de secuencia y los diagramas de colaboración. Ambos diagramas son isomorfos, es decir, se puede convertir de uno a otro sin pérdida de información.

- Un diagrama de secuencia es un diagrama de interacción que resalta la ordenación temporal de los mensajes. Un diagrama de secuencia presenta un conjunto de objetos y los mensajes enviados y recibidos por ellos. Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos.
- Un diagrama de colaboración es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra un conjunto de objetos, enlaces entre estos objetos y mensajes enviados y recibidos por estos objetos. Los objetos normalmente son instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, como colaboraciones, componentes y nodos.

II.1.5.1.3.6 Diagramas de Actividades

Un diagrama de actividades es un tipo especial de diagrama de estados que muestra el flujo secuencial o ramificado de actividades en un sistema. Conjunto de actividades, Son especialmente importantes para modelar la función del sistema, así como para resaltar el flujo de control entre objetos. [26]

II.1.6 Metodología de Desarrollo

II.1.6.1 Metodología de Desarrollo RUP (Rational Unified Process)

El **Proceso Unificado Racional** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología

estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Por lo que (RUP) describe como aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados "mejores prácticas" pues son utilizados en la industria por organizaciones exitosas.



Figura 11: Desarrollar software iterativamente

En función de la cada vez mayor complejidad solicitada para los sistemas de software, ya no es posible trabajar secuencialmente: definir primero el problema completo, luego diseñar toda la solución, construir el software y finalmente, testear el producto. Es necesario un enfoque iterativo, que permita una comprensión creciente del problema a través de refinamientos sucesivos, llegando a una solución efectiva luego de múltiples iteraciones acotadas en complejidad.

RUP utiliza y soporta este enfoque iterativo que ayuda a atacar los riesgos mediante la producción de releases ejecutables progresivos y frecuentes que permiten la opinión e involucramiento del usuario.

A través de las iteraciones que generan releases ejecutables, se logra detectar en forma temprana los desajustes e inconsistencias entre los requerimientos, el diseño, el desarrollo y la implementación del sistema, manteniendo al tema de desarrollo focalizado en producir resultados.[27]

La metodología RUP, Rational Unified Process, divide en 4 fases el desarrollo del software:

- **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**, En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transmisión**, El objetivo es llegar a obtener el reléase del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.

- Distribución: Hacer todo lo necesario para la salida del proyecto

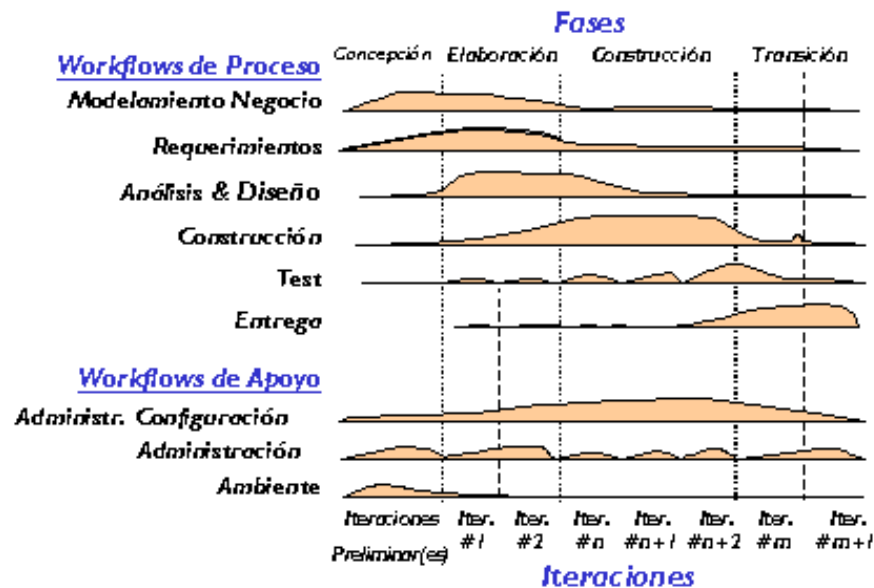


Figura 12: Fases de RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- **Actividades,** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores,** Vienen hacer las personas o entes involucrados en cada proceso.
- **Artefactos,** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.
-

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

Metodología de Desarrollo RUP (Rational Unified Process)

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).[28]

Fases

- Establece oportunidad y alcance
- Identifica las entidades externas o actores con las que se trata
- Identifica los casos de uso

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

Proceso: Las etapas de esta sección son: (Revise nuevamente la gráfica)

- Modelado de negocio
- Requisitos
- Análisis y Diseño
- Implementación

- Pruebas
- Despliegue

SopORTE: En esta parte nos encontramos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP es la que permite que éste sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las 4 fases descritas anteriormente:

- Inicio(También llamado Incepción)
- Elaboración
- Desarrollo(También llamado Implementación, Construcción)
- Cierre (También llamado Transición)

Artefactos

RUP en cada una de sus fases (pertenecientes a la estructura estática) realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema (entre otros). Estos artefactos (entre otros) son los siguientes:

Inicio:

- Documento Visión
- Especificación de Requerimientos

Elaboración:

- Diagramas de caso de uso

Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

Vista Lógica:

- Diagrama de clases
- Modelo E-R (Si el sistema así lo requiere)

Vista de Implementación:

- Diagrama de Secuencia
- Diagrama de estados
- Diagrama de Colaboración

Vista Conceptual:

- Modelo de dominio

Vista física:

- Mapa de comportamiento a nivel de hardware.

CAPITULO II:

INGENIERIA DE REQUERIMIENTOS

CAPITULO II: INGENIERIA DE REQUERIMIENTOS

II.1 DIAGRAMA DE CASOS DE USO DEL NEGOCIO

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.) permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

II.1.1 ORDENAMIENTO TERRITORIAL

Documentación

Actor	Documentación
Director O.T.	Representa a la persona encargada de Ordenamiento Territorial.
Coordinador O.T.	Representa a la persona encargada de Coordinar todo lo relacionado con Ordenamiento Territorial.

Tabla Nª 48: Actores → Director y Coordinador de la **O.T.**

Caso de Uso	Documentación
Aprobar Tramite	Representa la forma en la que el director firma un trámite.
Recibir Reporte General	Representa la forma en la que el director recibe los reportes del coordinador.
Recibir Reporte	Representa la forma en la que el coordinador recibe los reportes de los jefes de Proyectos y Lotes.

Tabla Nª 49: Casos de Uso → Aprobar Tramite, Recibir Reporte General y Recibir Reporte

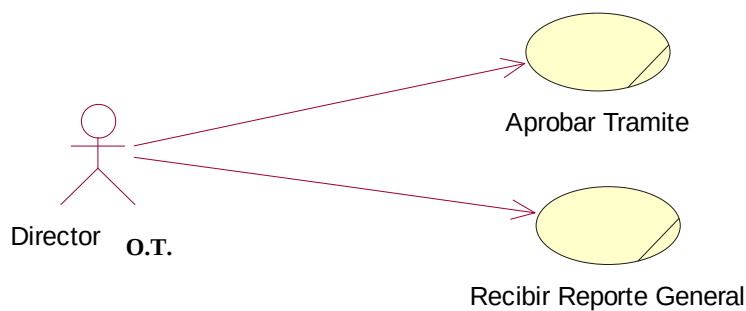


Figura 13: CASOS DE USO DEL NEGOCIO: Director O.T.



Figura 14: CASOS DE USO DEL NEGOCIO: Coordinador O.T.

II.1.2 AREA DE PROYECTOS

Documentación

Actor	Documentación
Propietario	Representa a la persona dueña de una vivienda o un lote que ingresa su documentación.
Inspector	Representa a la persona encargada de ir a verificar si de verdad existen anomalías en una construcción de vivienda, levantamiento de vivienda o en una ampliación de vivienda.
Banco	Representa a la empresa encargado de cobrar las multas por construcción clandestina o por la aprobación de un trámite.

Tabla N°50: Actores → Propietario, Inspector de la O.T. y Banco.

Caso de Uso	Documentación
Pedir Requisito	Representa la forma en la que el propietario solicita los requisitos al encargado de administración.
Entregar Documento o Tramite	Representa la forma en la que el propietario entrega su trámite al recepcionista de proyectos.
Solicitar Observación de Tramite	Representa la forma en la que el propietario solicita observaciones acerca de su trámite al encargado de proyecto y lotes.
Recibir Nota de Fecha y Numero	Representa la forma en la que el propietario recibe una nota con la fecha de ingreso o la fecha en la que debe volver a pedir sus información acerca de su tramite si esta observado o no y el numero previo o puede ser un numero de trámite.
Verificar Proyecto	Representa la forma en la que el inspector va y verifica a la construcción, levantamiento y ampliación de vivienda si en realidad existen anomalías.
Generar Informes Para Técnico	Representa la forma en la que el inspector realiza sus informes para el técnico.
Cobrar	Representa la forma en la que el banco cobra por concepto de multa por construcción clandestina o puede ser por la aprobación de trámite aprobado.
Dar Comprobante	Representa la forma en la que el banco le da al propietario el comprobante de pago por concepto de multa por construcción clandestina o puede ser por la aprobación de trámite aprobado.

Tabla N° 51: Casos de Uso → Pedir Requisito, Entregar Documento o Tramite, Solicitar Observación de Tramite, Verificar Proyecto, Generar Informes Para Técnico, Cobrar y Dar Comprobante.

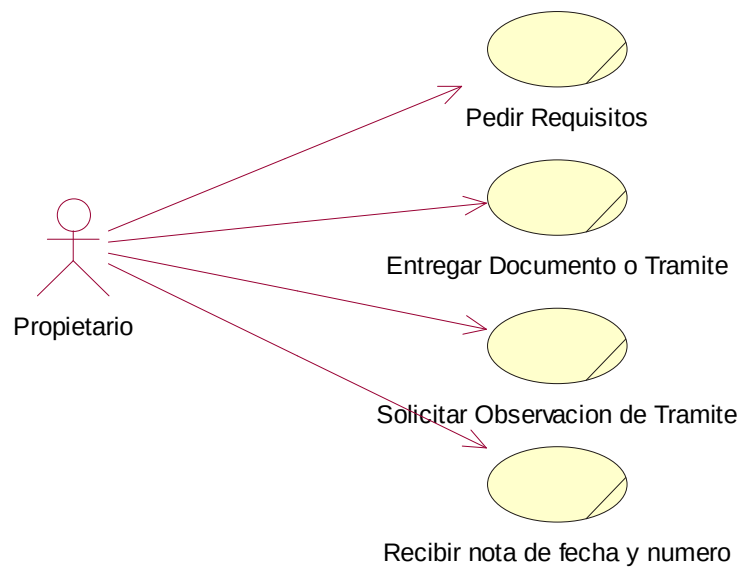


Figura 15: CASOS DE USO DEL NEGOCIO: Propietario

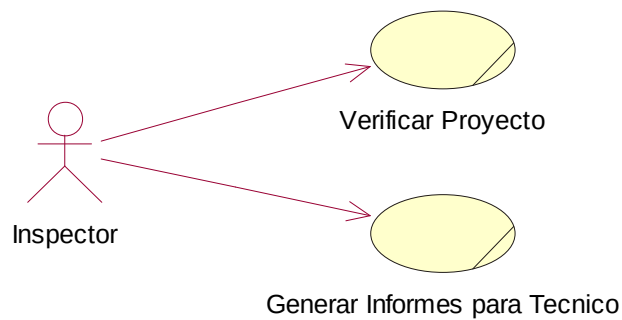


Figura 16: CASOS DE USO DEL NEGOCIO: Inspector **O.T.**

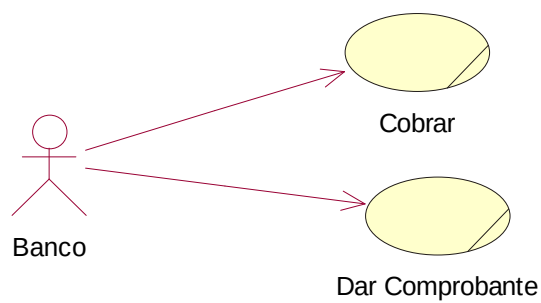


Figura 17: CASOS DE USO DEL NEGOCIO: Banco

II.1.3 AREA DE LOTES

Documentación

Actor	Documentación
Interesado	Representa a la persona a un proyectista, propietario y apoderado que ingresa la documentación.
Inspector	Representa a la persona encargada de ir a verificar si de verdad existen anomalías en lote.
Topógrafo	Representa a la persona encargada de ir a medir un lote y del estaqueado de un lote.
Banco	Representa a la empresa encargado de cobrar las multas o por la aprobación de un trámite.

Tabla N° 52: Actores → Interesado, Inspector, Topógrafo de la O.T. y Banco.

Caso de Uso	Documentación
Pedir Requisito	Representa la forma en la que el interesado solicita los requisitos al encargado de administración.
Entregar Documento o Tramite	Representa la forma en la que el interesado entrega su trámite al recepcionista de proyectos.
Solicitar Observación de Tramite	Representa la forma en la que el interesado solicita observaciones acerca de su trámite al encargado de proyecto y lotes.
Pagar Multa o Aprobación de Tramite	Representa la forma en la que el interesado va i paga al banco la multa o la aprobación de un trámite.
Recoger Tramite	Representa la forma en la que un

	propietario o un apoderado recogen su trámite aprobado.
Verificar Lote	Representa la forma en la que el inspector va y verifica un lote si en realidad existen anomalías.
Generar Informes Para Técnico	Representa la forma en la que el inspector realiza sus informes para el técnico.
Medir Lote	Representa la forma en la que topógrafo mide el lote.
Dar Comprobante	Representa la forma en la que el banco le da al interesado el comprobante de pago por concepto de multa o puede ser por la aprobación de trámite aprobado.

Tabla N° 53: Casos de Uso → Pedir Requisito, Entregar Documento o Tramite, Solicitar Observación de Tramite, Verificar Proyecto, Generar Informes Para Técnico, Medir Lote, Cobrar y Dar Comprobante.

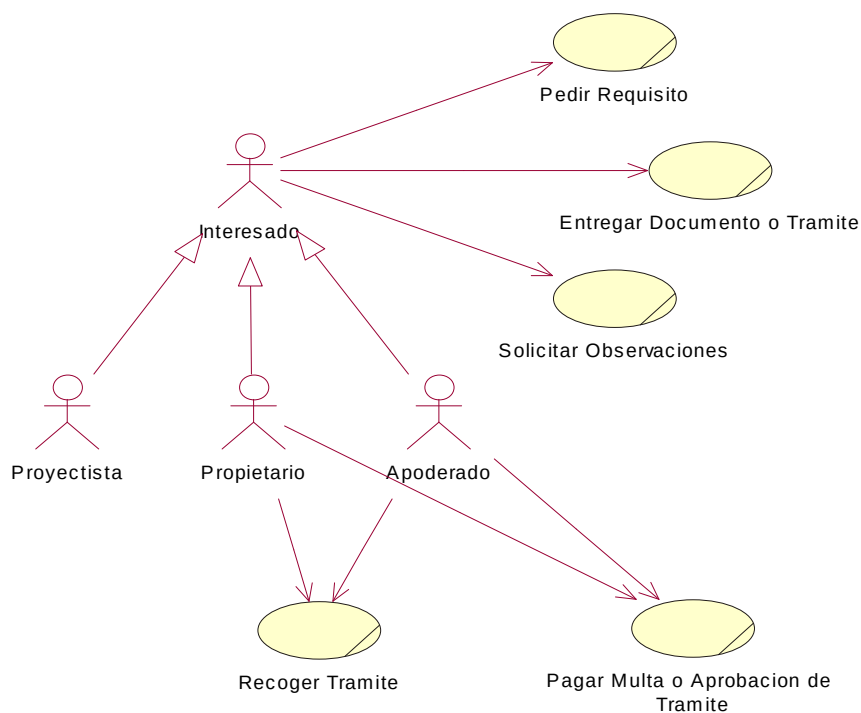


Figura 18: CASOS DE USO DEL NEGOCIO: Interesado

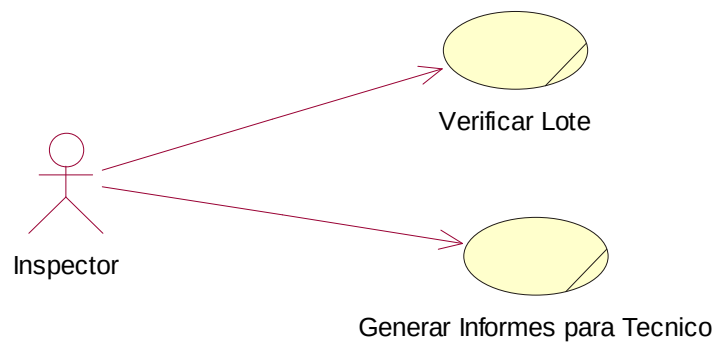


Figura 19: CASOS DE USO DEL NEGOCIO: Inspector **O.T.**



Figura 20: CASOS DE USO DEL NEGOCIO: Topógrafo **O.T.**



Figura 21: CASOS DE USO DEL NEGOCIO: Banco

II.1.4 AREA DE ADMINISTRACION

Documentación

Actor	Documentación
Encargado de Administración	Representa a la persona encargado de administración.

Tabla Nª 54: Actores → Encargado de Administración de la **O.T.**

Caso de Uso	Documentación
Dar Requisito	Representa la forma en la que el encargado de administración da los requisitos al propietario o interesado.
Informar Monto a Pagar	Representa la forma en la que el encargado de administración entrega una nota con el monto a pagar al propietario o interesado.
Entregar Tramite Aprobado	Representa la forma en la que el encargado de administración entrega un trámite aprobado a un propietario o un apoderado.

Tabla N° 55: Casos de Uso → Dar Requisito, Informar Monto a Pagar, Entregar Tramite Aprobado.

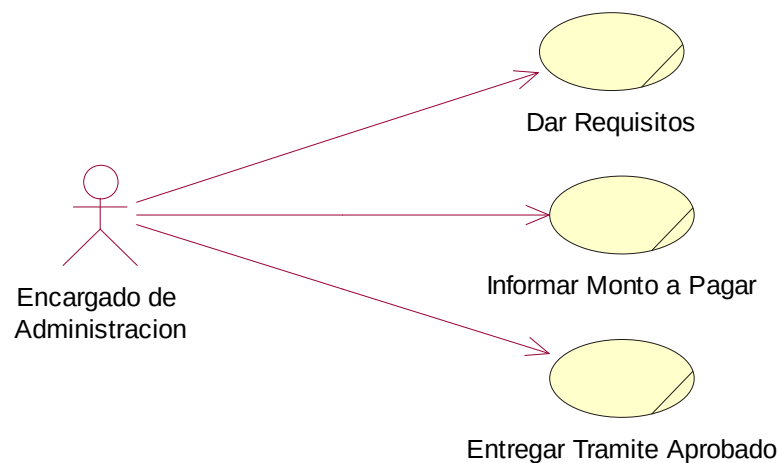


Figura 22: CASOS DE USO DEL NEGOCIO: Encargado de Administración

II.2 DIAGRAMA DE MODELO DE OBJETOS DEL NEGOCIO

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

II.2.1 Ordenamiento Territorial (O.T.)

1.-Modelo del negocio de Aprobar Trámite

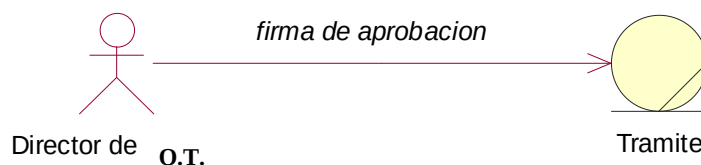


Figura 23: Modelo del negocio de Aprobar Trámite

2.-Modelo del negocio de Recibir Reportes generales

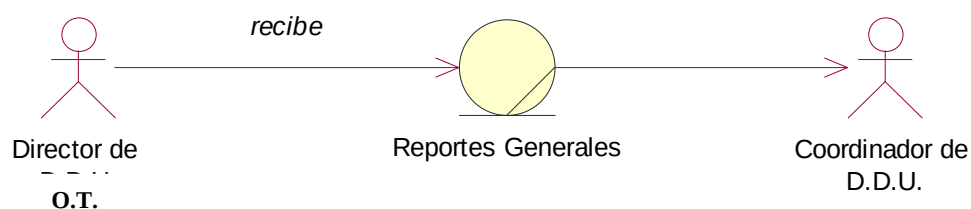


Figura 24: Modelo del negocio de Recibir Reportes generales

3.-Modelo del negocio de Recibir Reportes

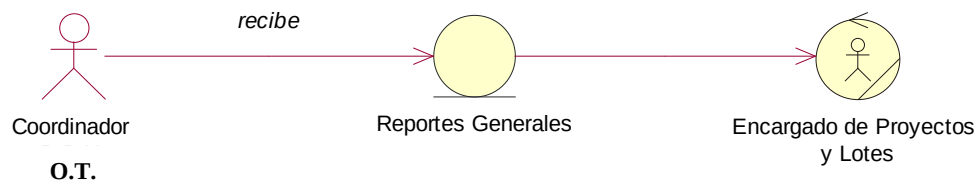


Figura 25: Modelo del negocio de Recibir Reportes

II.2.1.1 AREA DE PROYECTOS

1.-Modelo del negocio pedir requisitos

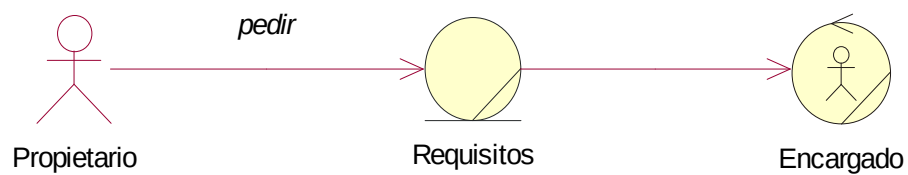


Figura 26: Modelo del negocio pedir requisitos

2.-Modelo del negocio entrega de documentación o trámite

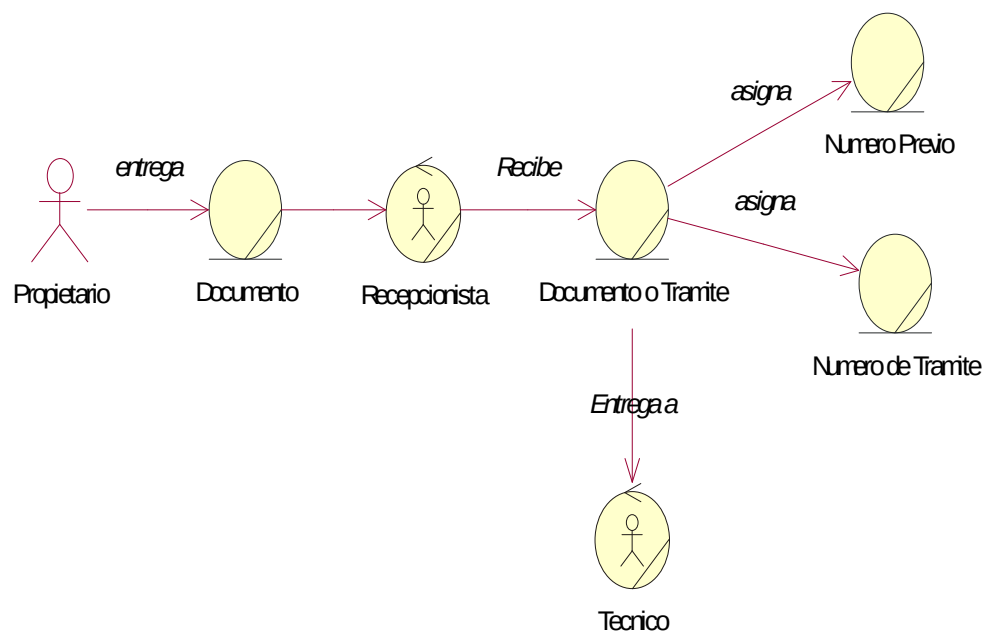


Figura 27: Modelo del negocio entrega de documentación o trámite

3.- Modelo de objeto solicitar estado tramite

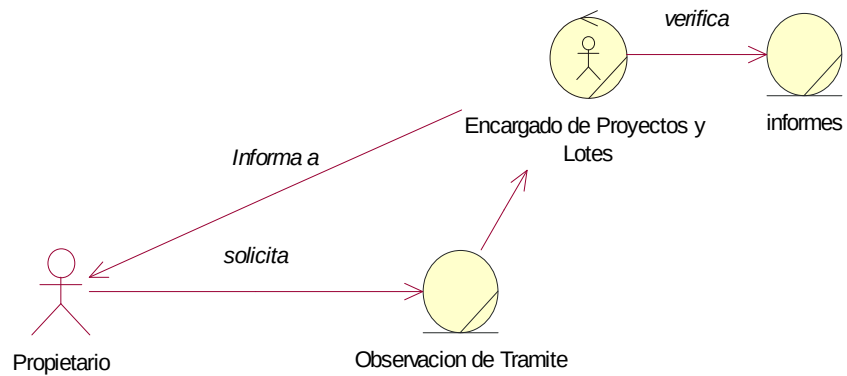


Figura 28: Modelo de objeto entrega de documentación o trámite

4.-Modelo de objeto de Recibir Nota de Fecha y Número

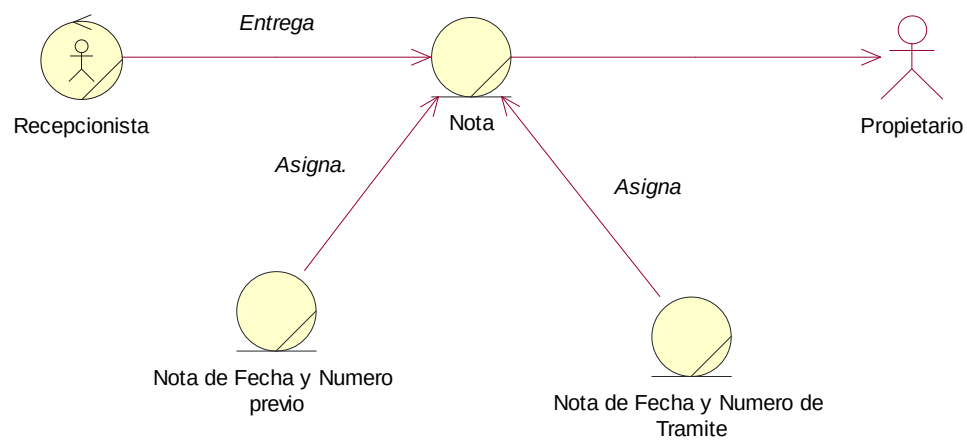


Figura 29: Modelo de objeto Recibir Nota de Fecha y Número

5.-Modelo de Objeto de Verificar Proyecto

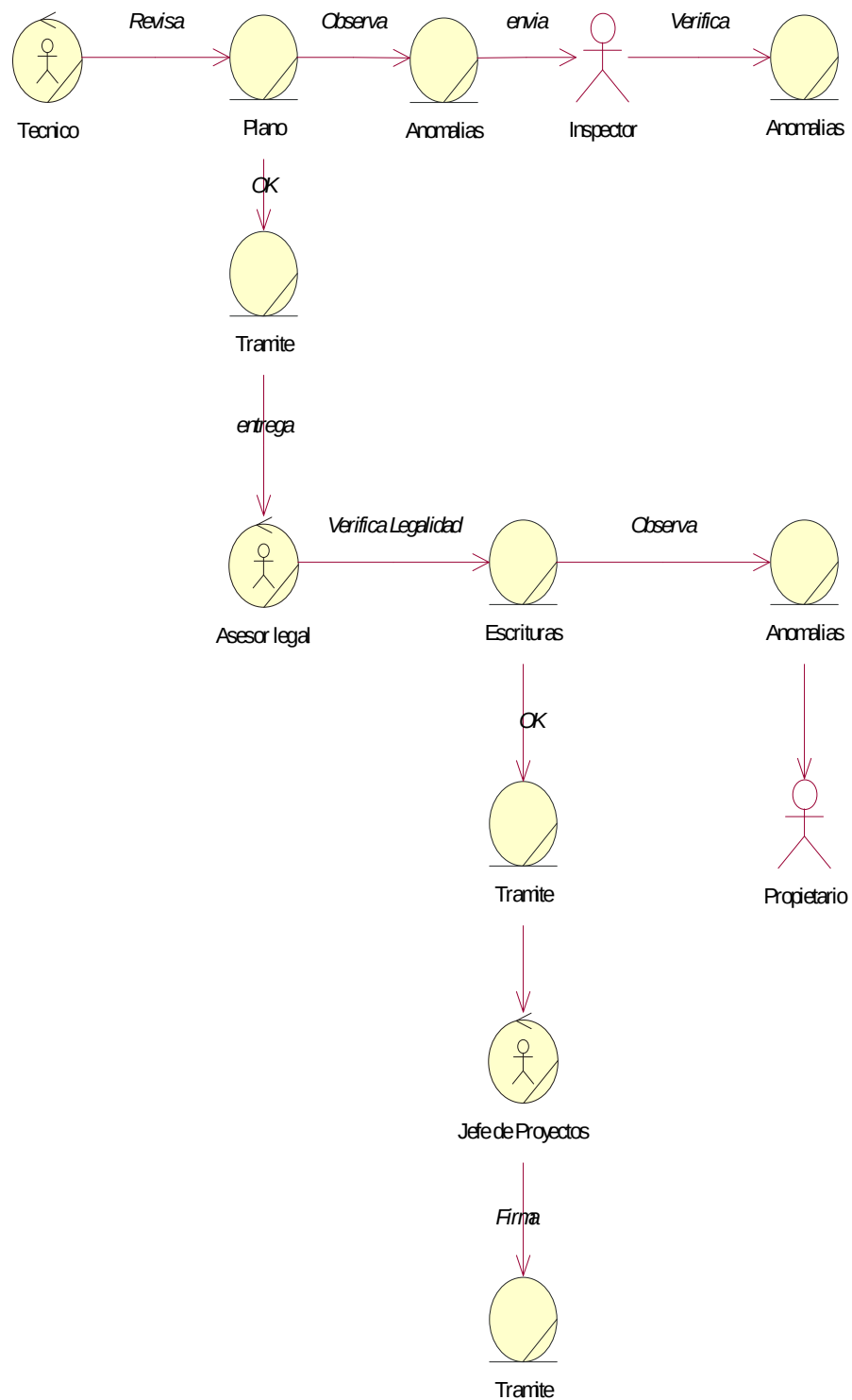


Figura 30: Modelo de Objeto de Verificar Proyecto

6.-Modelo de Objeto de Generar Informes

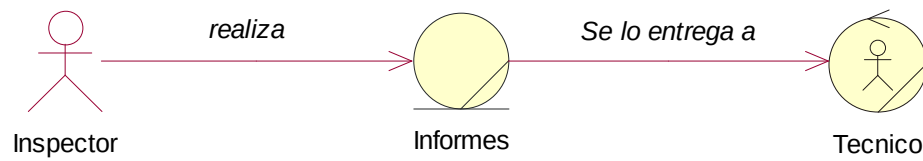


Figura 31: Modelo de Objeto de Generar Informes

7.-Modelo de Objeto de Cobrar

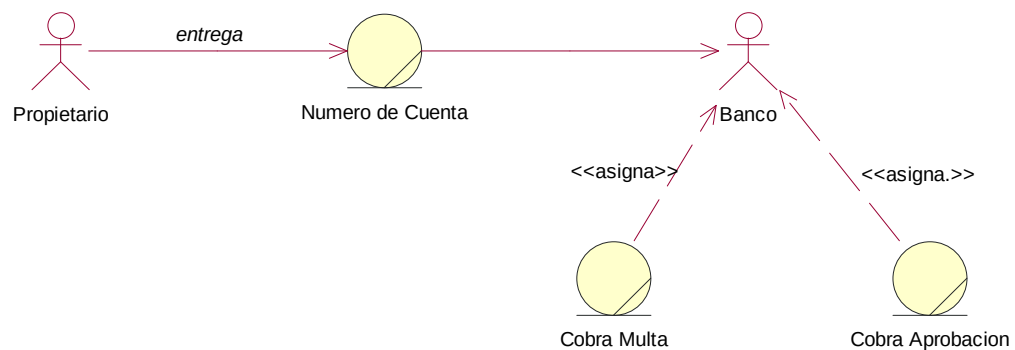


Figura 32: Modelo de Objeto de Cobrar

8.-Modelo de Objeto de Dar Comprobante

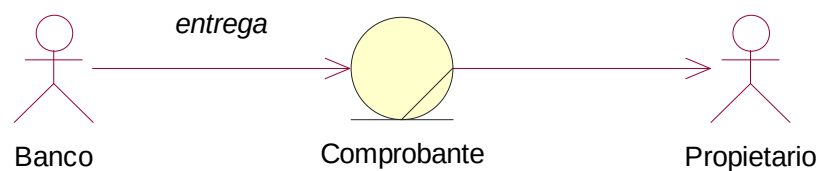


Figura 33: Modelo de Objeto Dar Comprobante

II.2.1.2 AREA DE LOTES

1.-Modelo de Objeto de Pedir Requisitos

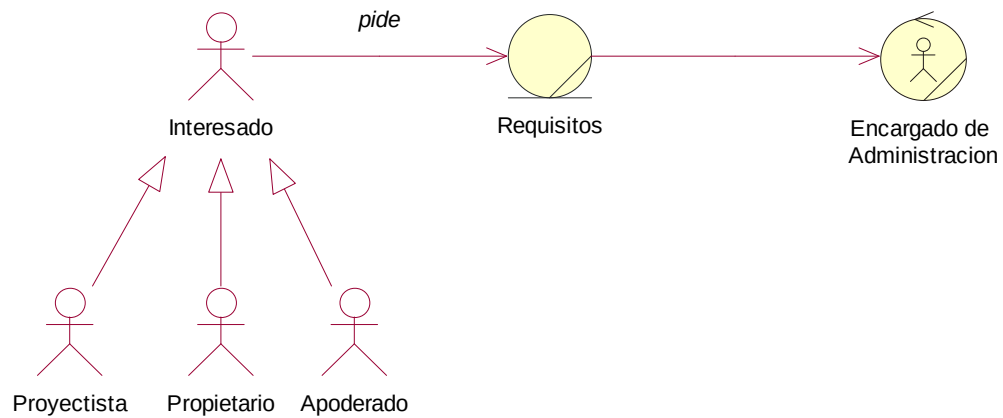


Figura 34: Modelo de Objeto de Pedir Requisitos

2.-Modelo de Objeto de entregar Documento o Trámite

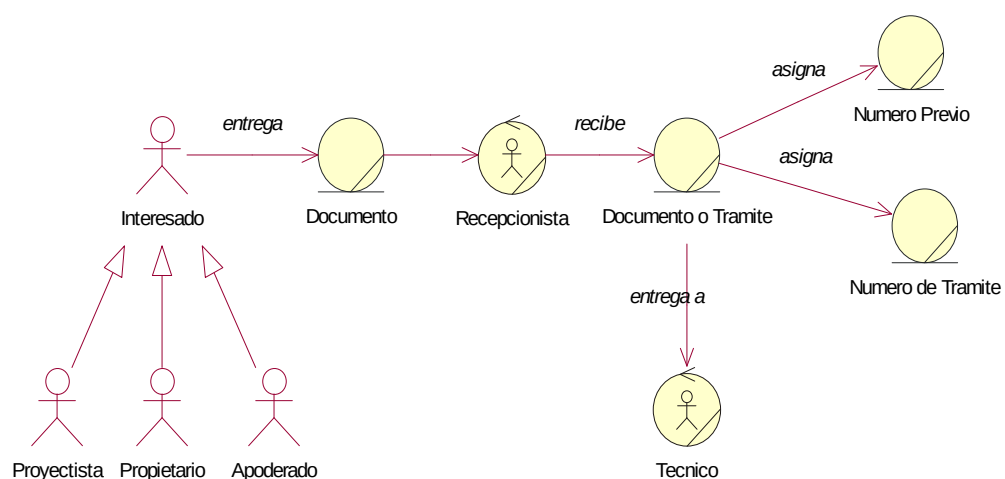


Figura 35: Modelo de Objeto de entregar Documento o Trámite

3.-Modelo de Objeto de Solicitar Observaciones

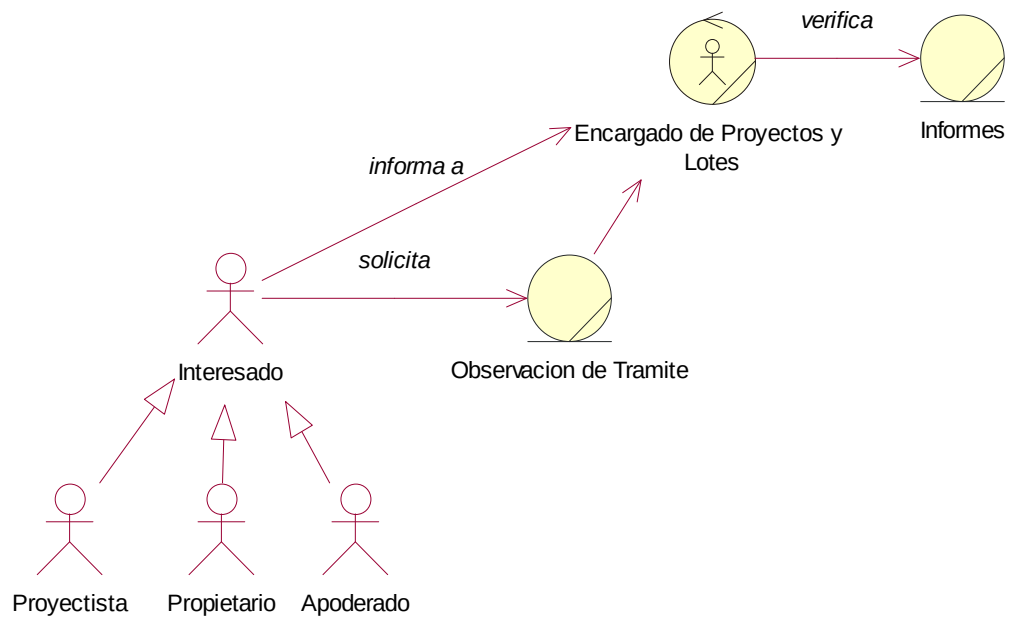


Figura 36: Modelo de Objeto de Solicitar Observaciones

4.-Modelo de Objeto de Recoger Trámite Aprobado

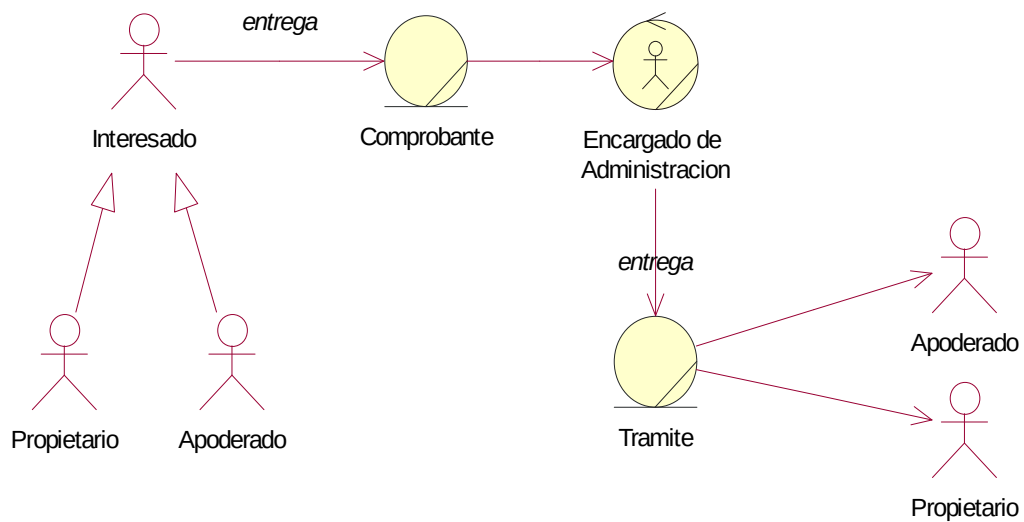


Figura 37: Modelo de Objeto de Recoger Trámite Aprobado

5.-Modelo de Objeto de Pagar Multa o Aprobación de Trámite

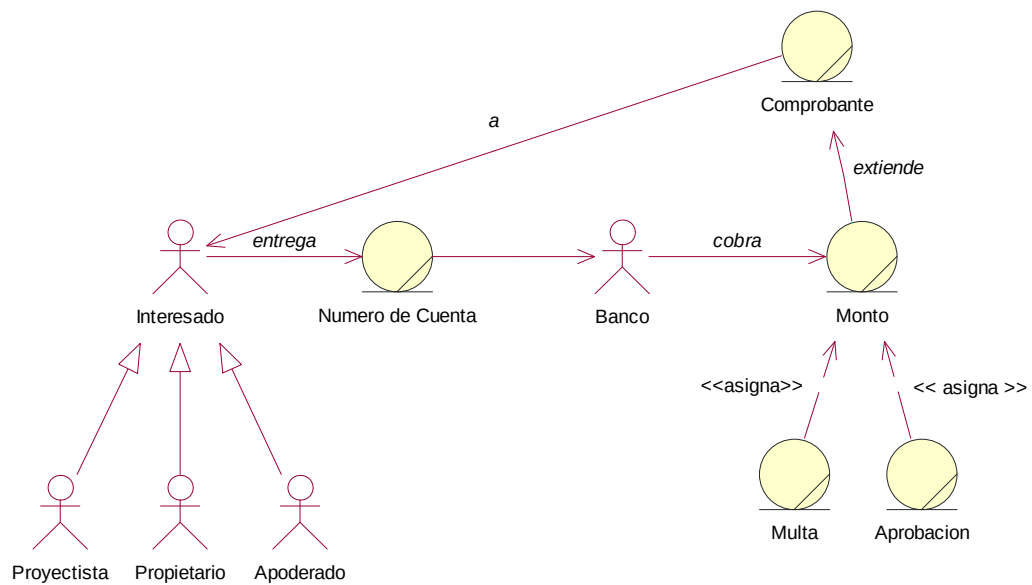


Figura 38: Modelo de Objeto de Pagar Multa o Aprobación de Trámite

6.-Modelo de Objeto de Verificar Lote

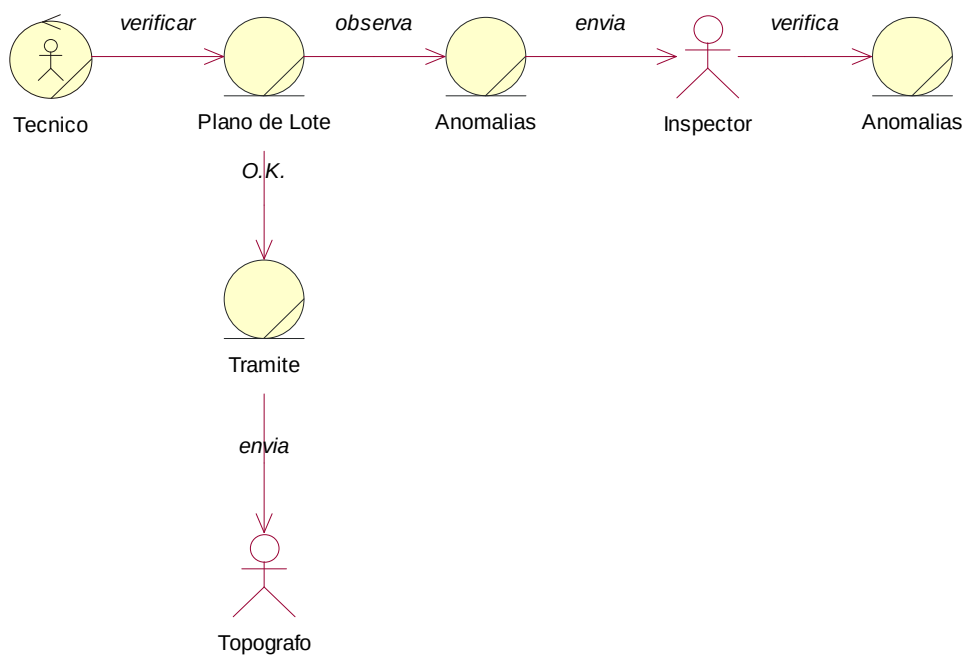


Figura 39: Modelo de Objeto de Verificar Lote

7.-Modelo de Objeto de Generar Informes para Técnico

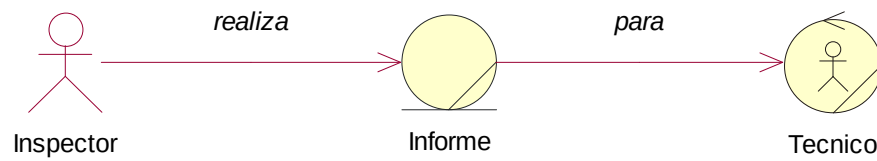


Figura 40: Modelo de Objeto de Generar Informes para Técnico

8.-Modelo de Objeto de Medir Lote

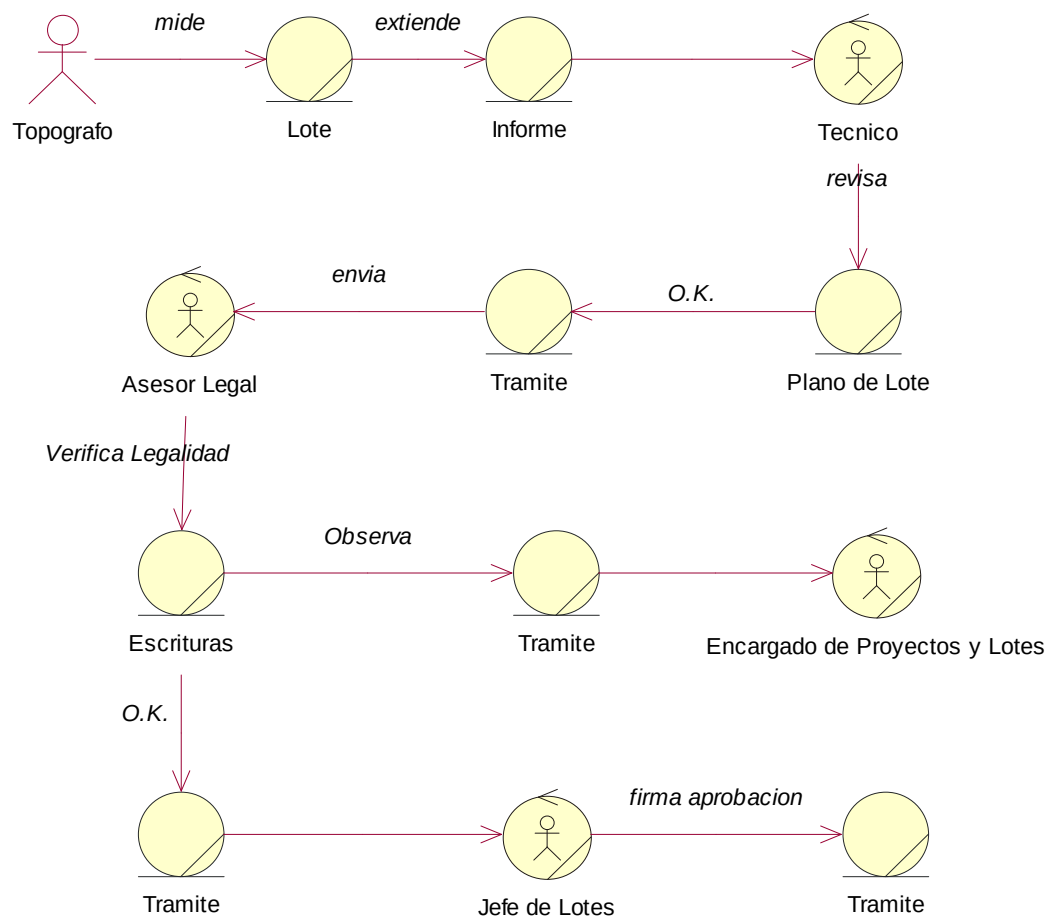


Figura 41: Modelo de Objeto de Medir Lote

9.-Modelo de Objeto de Dar Comprobante

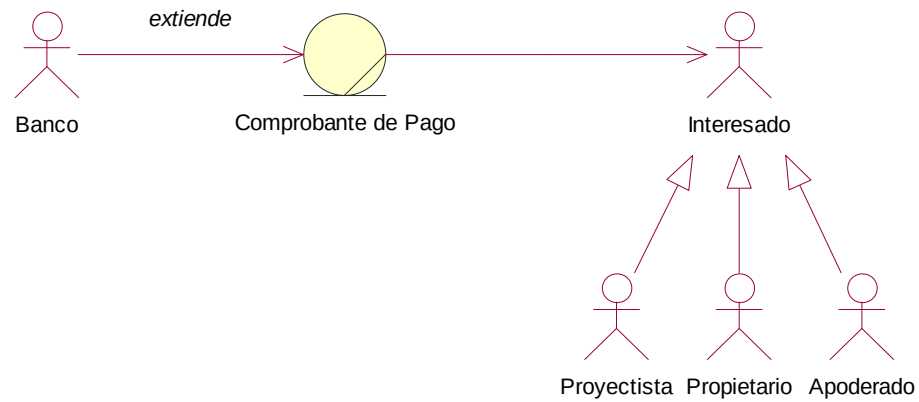


Figura 42: Modelo de Objeto de Dar Comprobante

II.2.1.3 AREA DE ADMINISTRACION

1.-Modelo de Objeto de Dar Requisitos

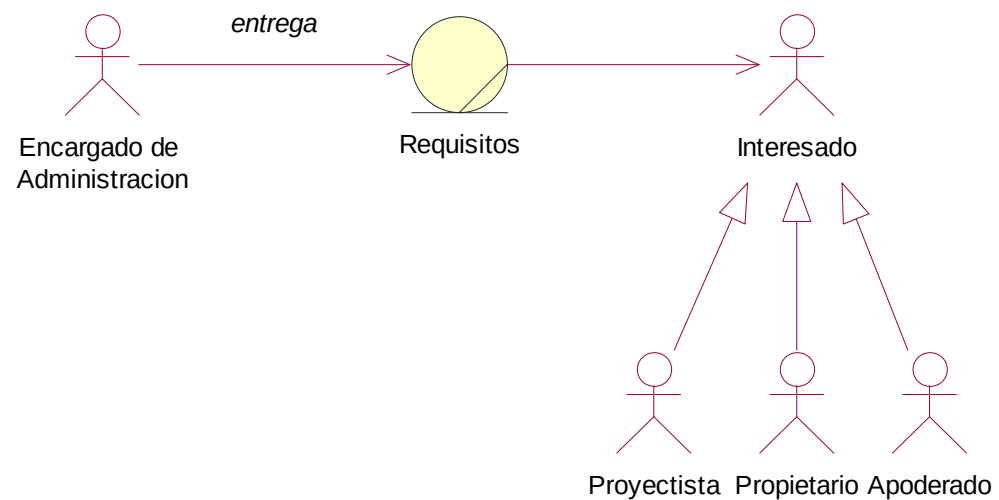


Figura 43: Modelo de Objeto de Dar Requisitos

2.-Modelo de Objeto de Informar Monto a Pagar

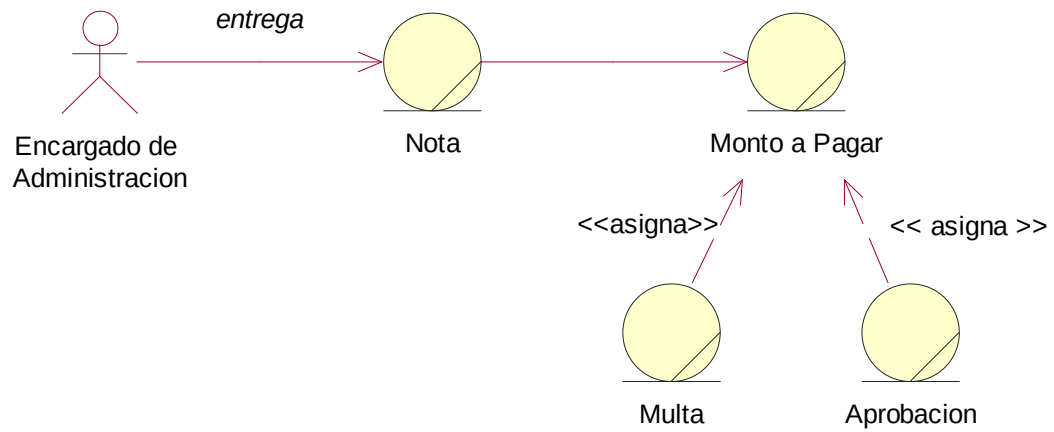


Figura 44: Modelo de Objeto de Informar Monto a Pagar

3.-Modelo de Objeto de Entregar Trámite Aprobado

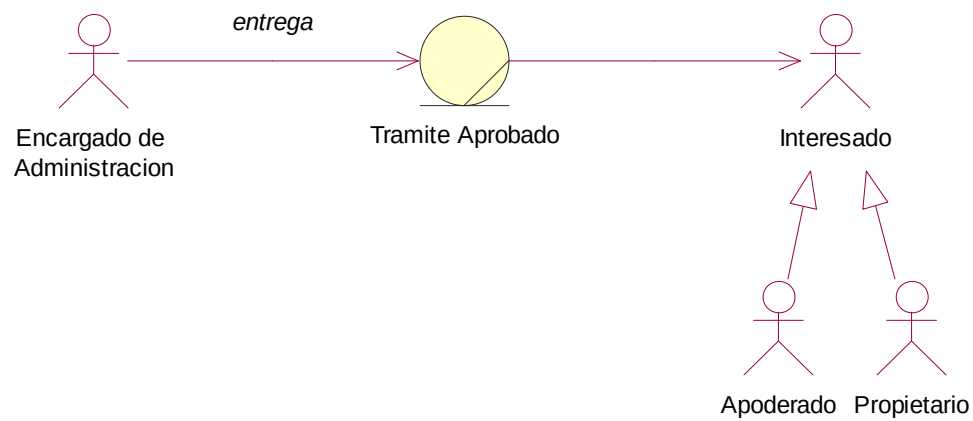


Figura 45: Modelo de Objeto de Entregar Trámite Aprobado

II.3 DIAGRAMA DE MODELO DE CASOS DE USO

II.3.1. Introducción

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

II.3.1.1. Propósito

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

II.3.1.2. Alcance

- Describe los procesos de sistema y los usuarios.
- 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.3.2 Interactuar con Recursos del Sistema

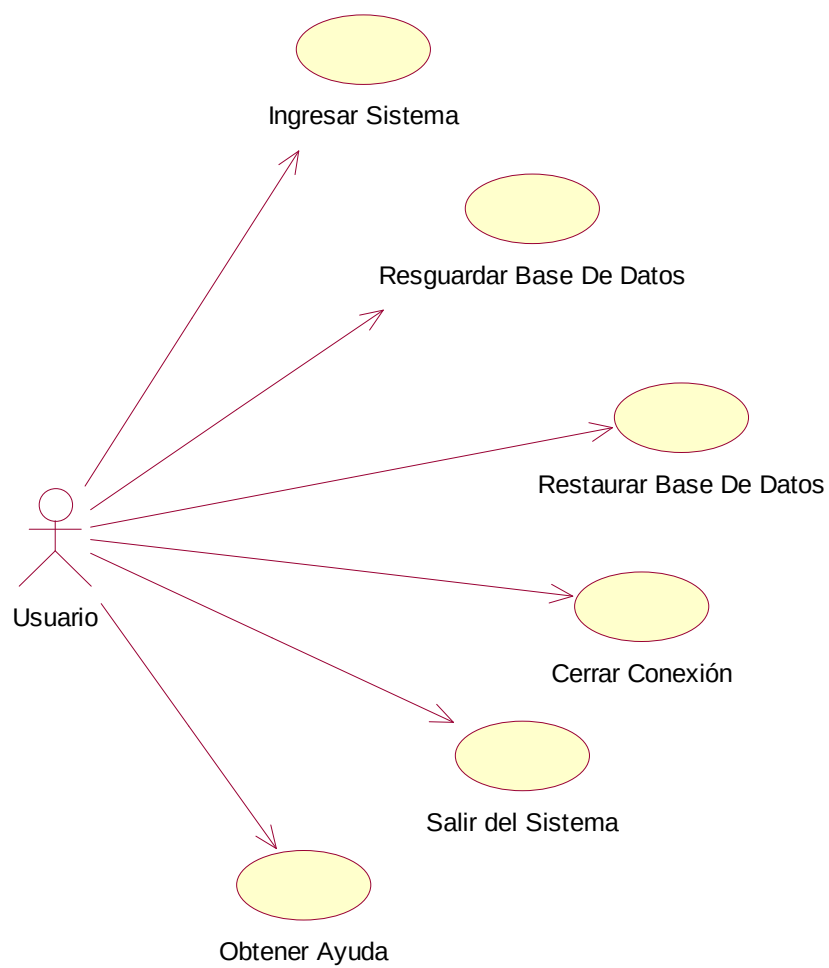


Figura 46: Caso De Uso General: Interactuar con Recursos del Sistema

II.3.3.1.1 CASOS DE USO ESPECÍFICO

1. Caso de Uso: Administrar Propietarios

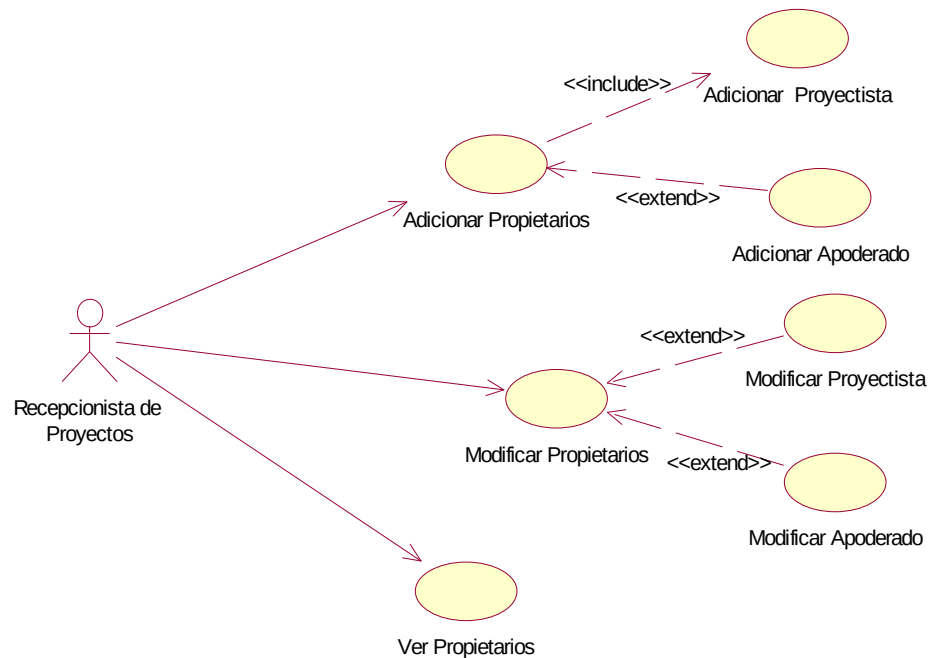


Figura 48: Caso de Uso: Administrar Propietarios

2. Caso de Uso: Administrar Observaciones Generales

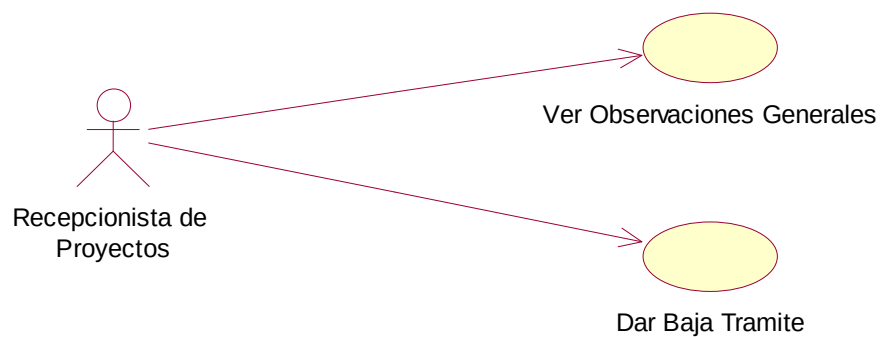


Figura 49: Caso de Uso: Administrar Observaciones Generales

3. Caso de Uso: Reasignar Trámite

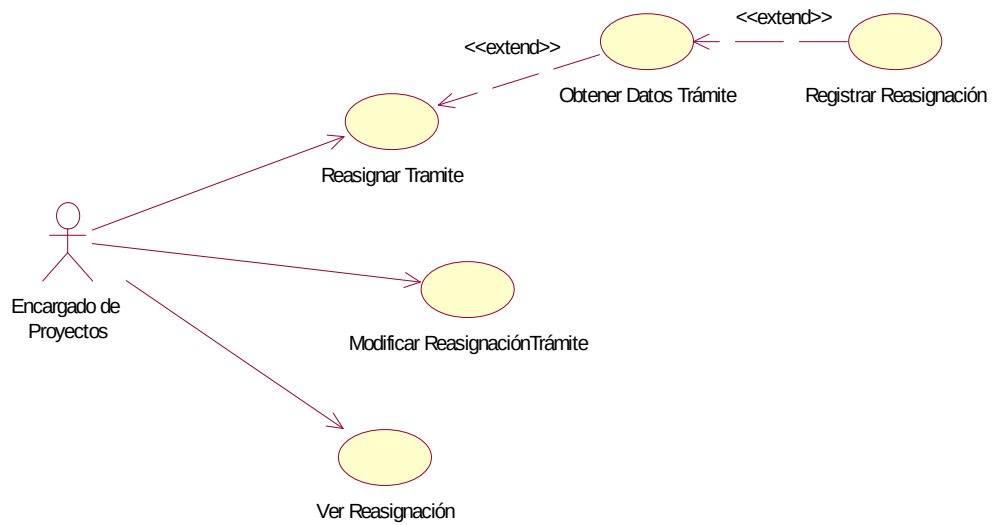


Figura 50: Caso de Uso: Reasignar Trámite

4. Caso de Uso: Ver Datos Trámite

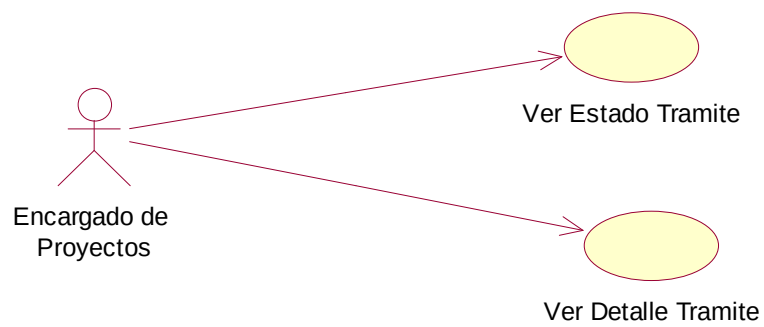


Figura 51: Caso de Uso: Ver Datos Trámite

5. Caso de Uso: Administrar Ingresos

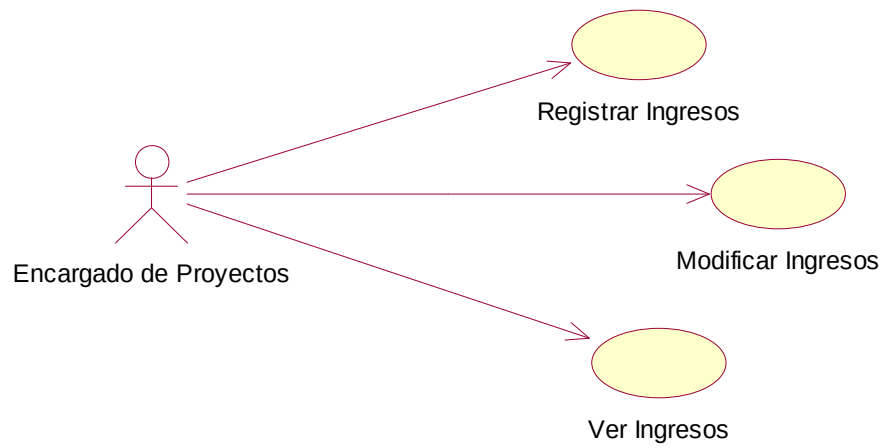


Figura 52: Caso de Uso: Administrar Ingresos

6. Caso de Uso: Administrar Salidas

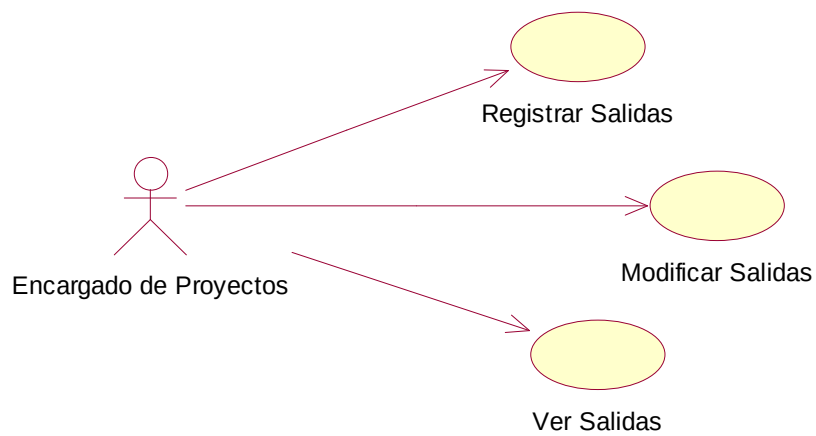


Figura 53: Caso de Uso: Administrar Salidas

7. Caso de Uso: Administrar Envíos

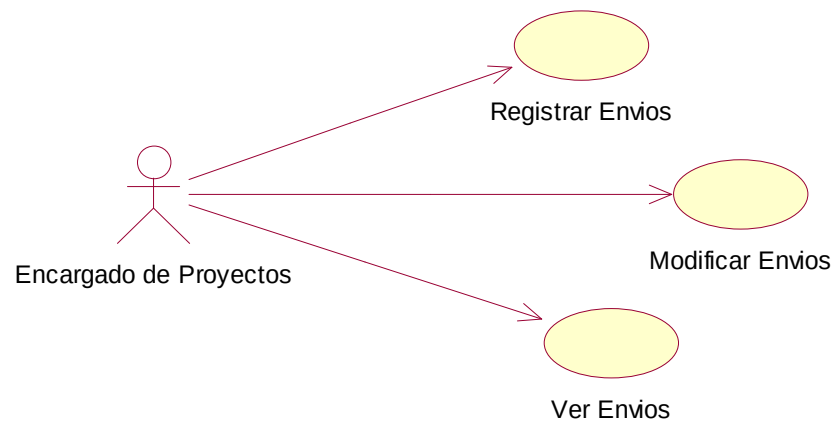


Figura 54: Caso de Uso: Administrar Envíos

8. Caso de Uso: Administrar Recepción

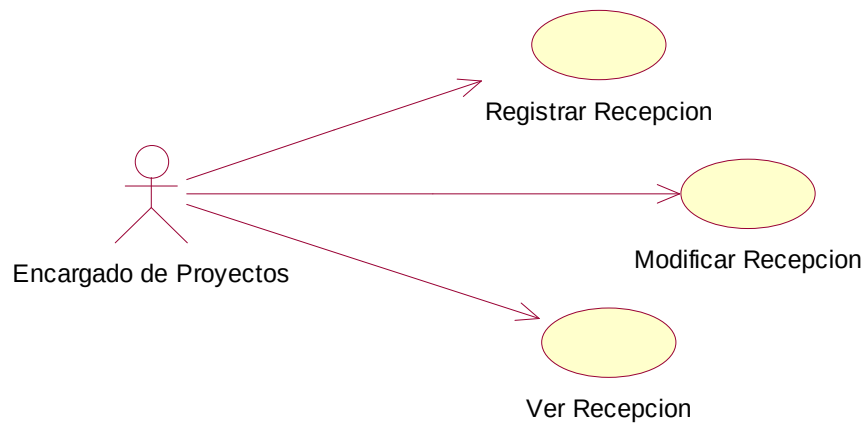


Figura 55: Caso de Uso: Administrar Recepción

9. Caso de Uso: Administrar Observaciones

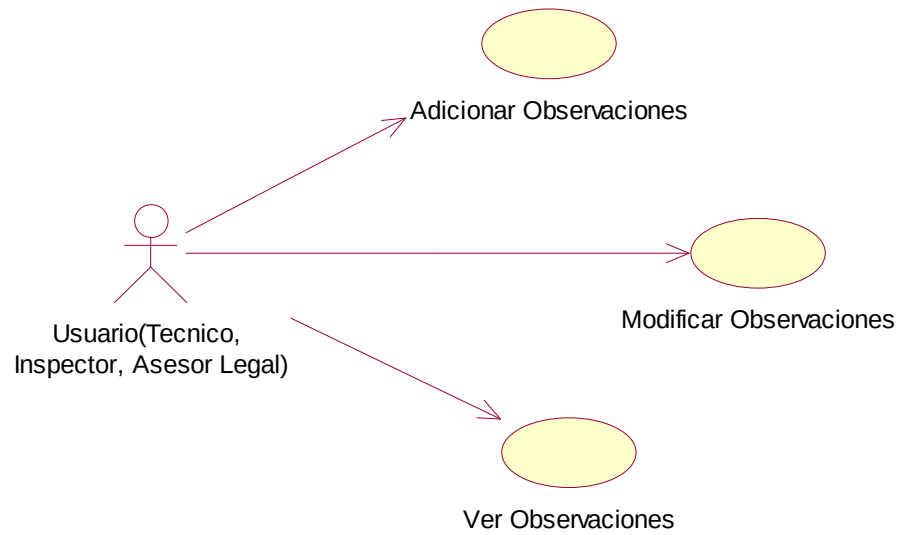


Figura 57: Caso de Uso: Administrar Observaciones

10. Caso de Uso: Administrar Observaciones Jefe Proyectos

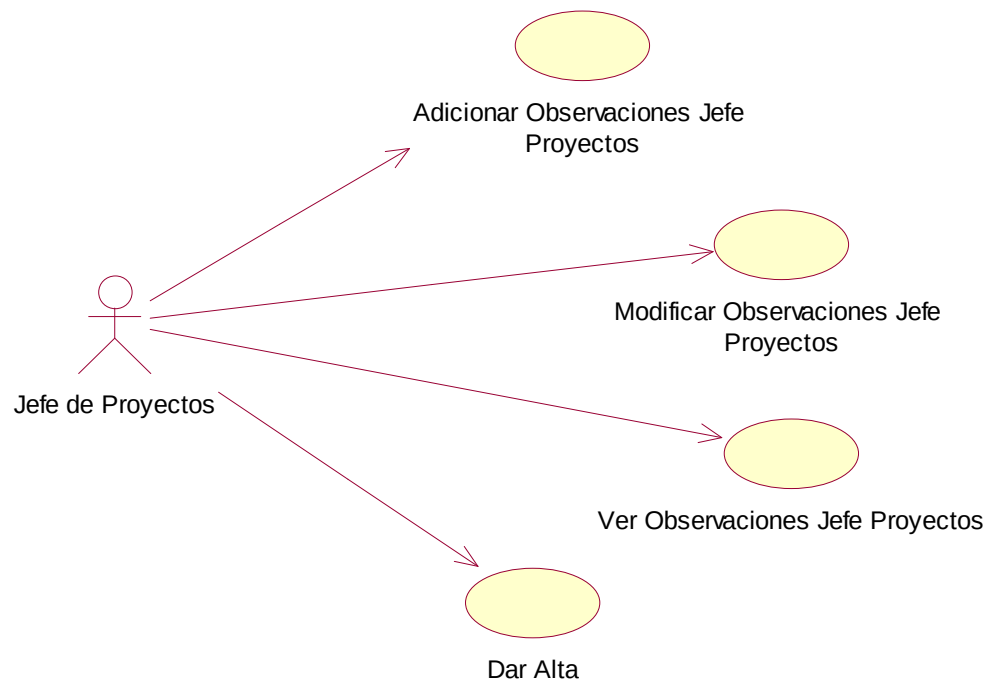


Figura 58: Caso de Uso: Administrar Observaciones Jefe Proyectos

11. Caso de Uso: Notas

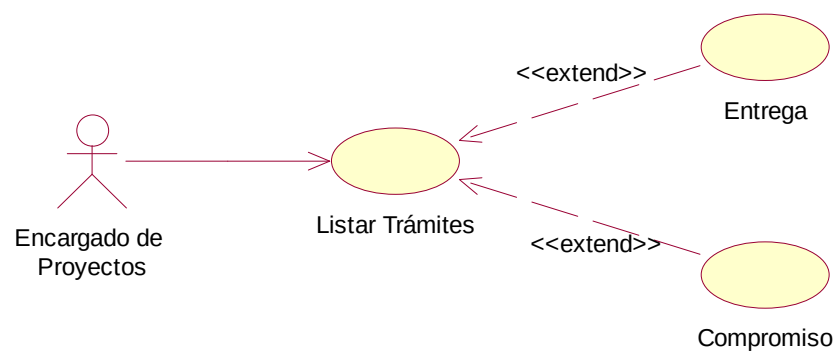


Figura 59: Caso de Uso: Notas

II.3.4 MODULO DE LOTES

II.3.4.1 CASO DE USO GENERAL

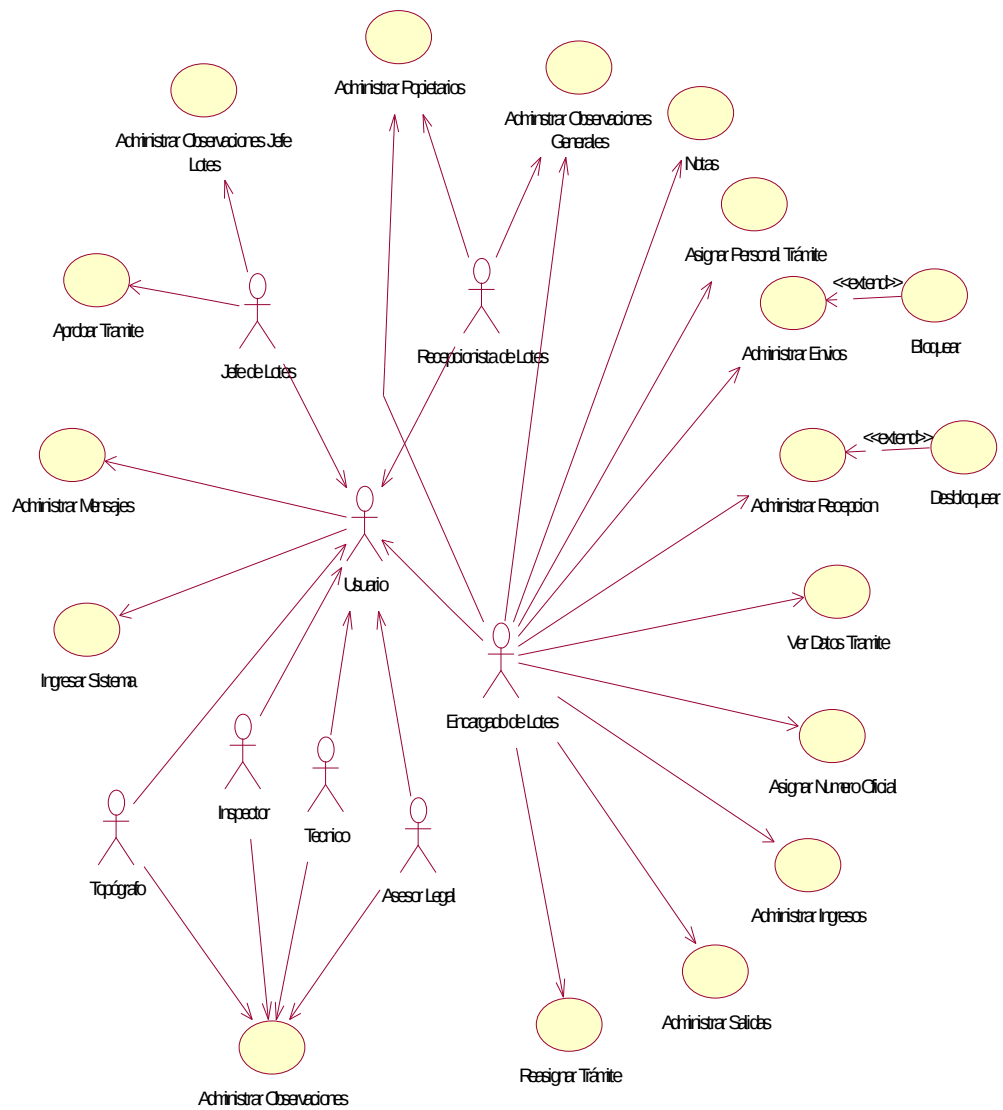


Figura 60: Caso de Uso: Caso de Uso General Área Lotes

II.3.4.1.1CASOS DE USO ESPECÍFICO

12. Caso de Uso: Administrar Propietarios

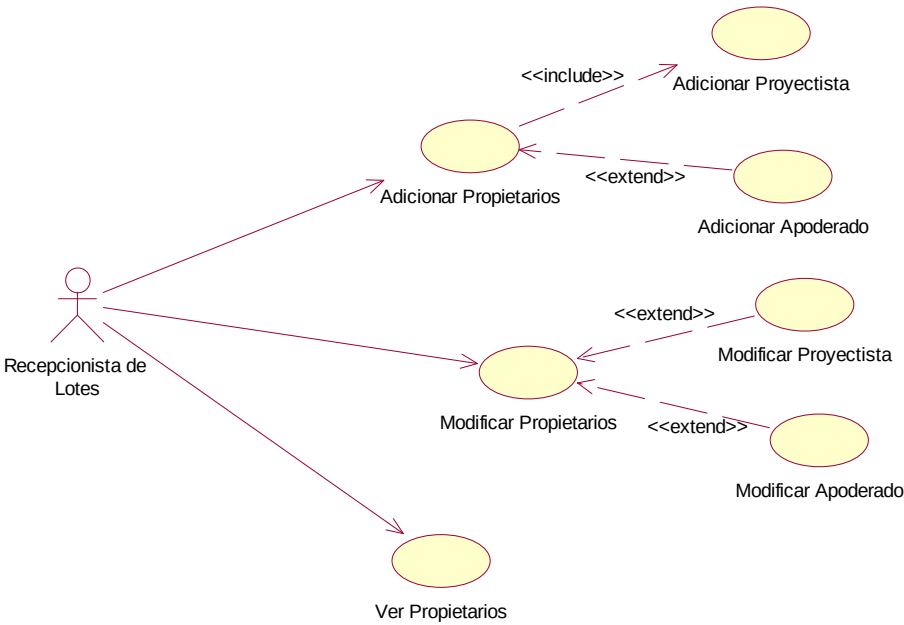


Figura 61: Caso de Uso: Administrar Propietarios

13. Caso de Uso: Administrar Observaciones Generales

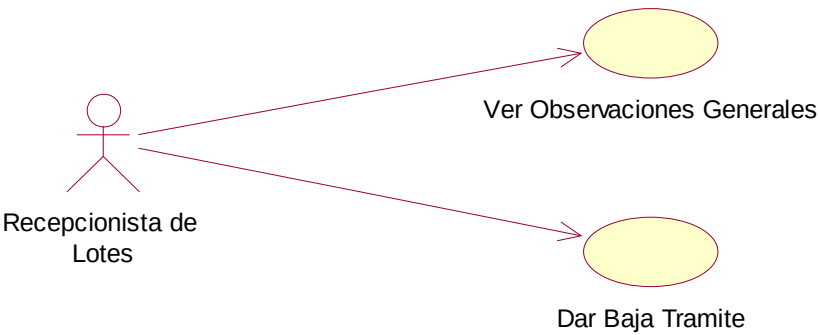


Figura 62: Caso de Uso: Administrar Observaciones Generales

14. Caso de Uso: Reasignar

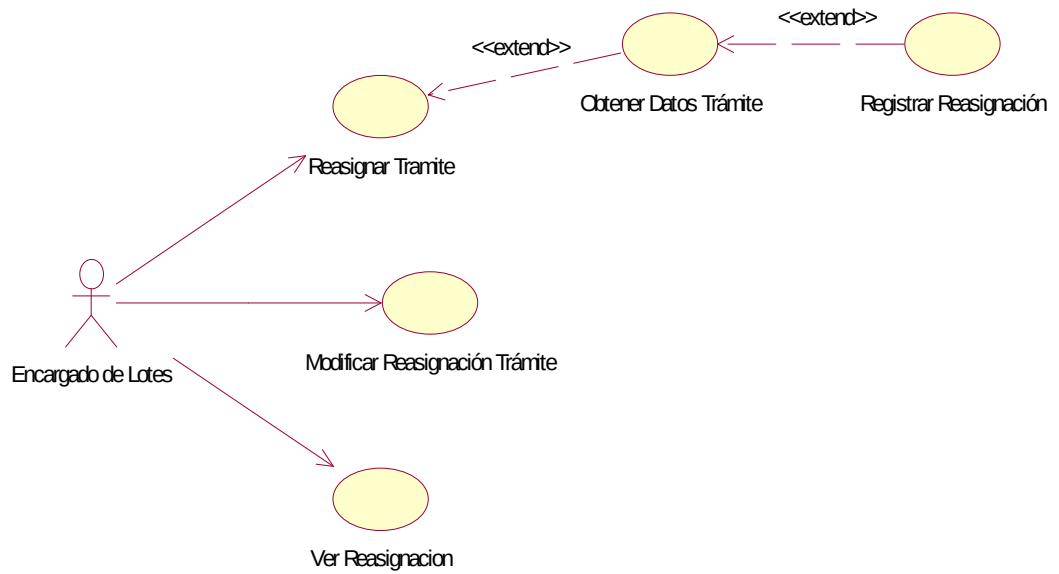


Figura 63: Caso de Uso: Reasignar

15. Caso de Uso: Ver Datos Trámite

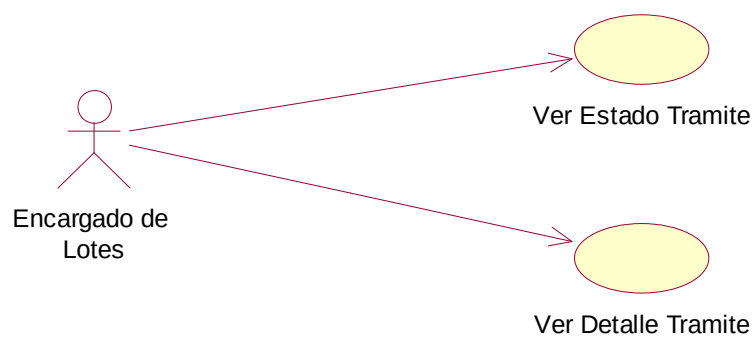


Figura 64: Caso de Uso: Ver Datos Trámite

16. Caso de Uso: Administrar Ingresos

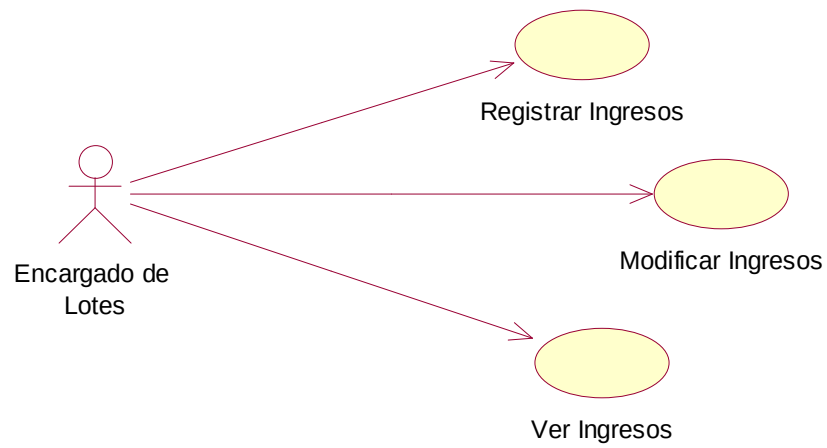


Figura 65: Caso de Uso: Administrar Ingresos

17. Caso de Uso: Administrar Salidas

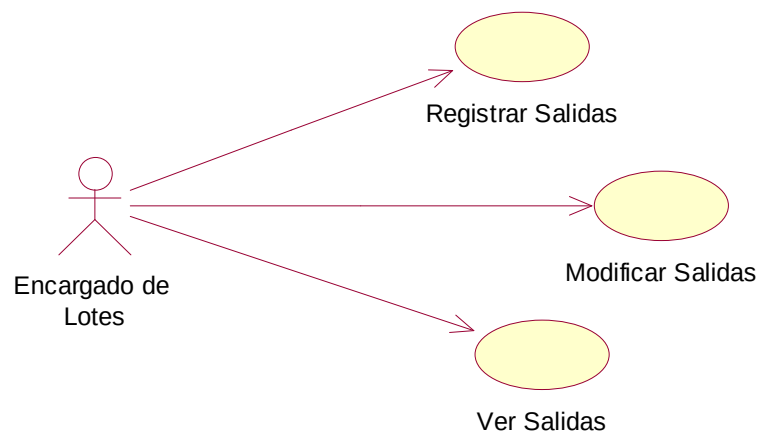
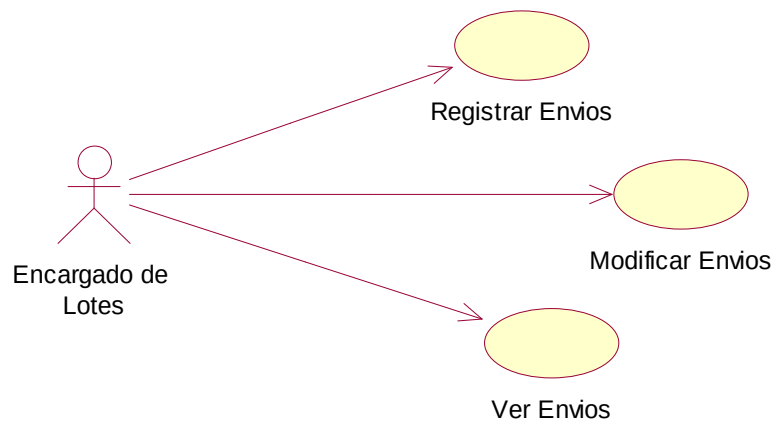
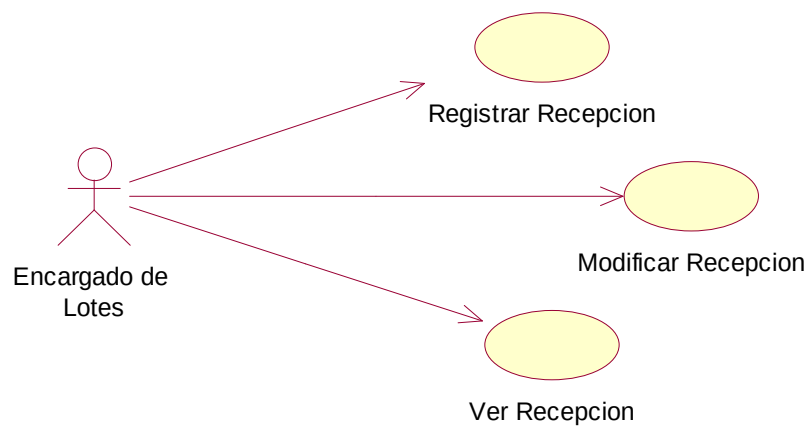


Figura 66: Caso de Uso: Administrar Salidas

18. Caso de Uso: Administrar Envíos**Figura 67: Caso de Uso: Administrar Envíos****19. Caso de Uso: Administrar Recepción****Figura 68: Caso de Uso: Administrar Recepción**

20. Caso de Uso: Administrar Observaciones

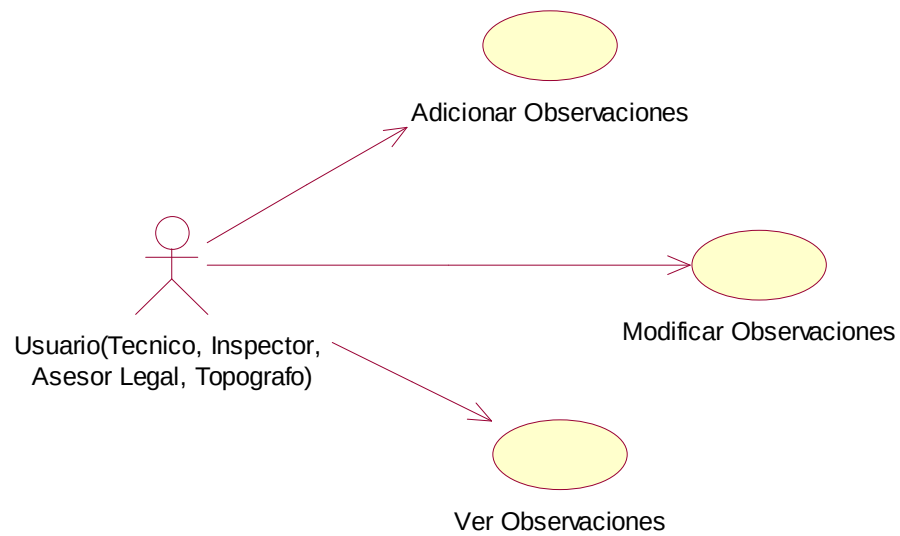


Figura 70: Caso de Uso: Administrar Observaciones

21. Caso de Uso: Administrar Observaciones Jefe Lotes

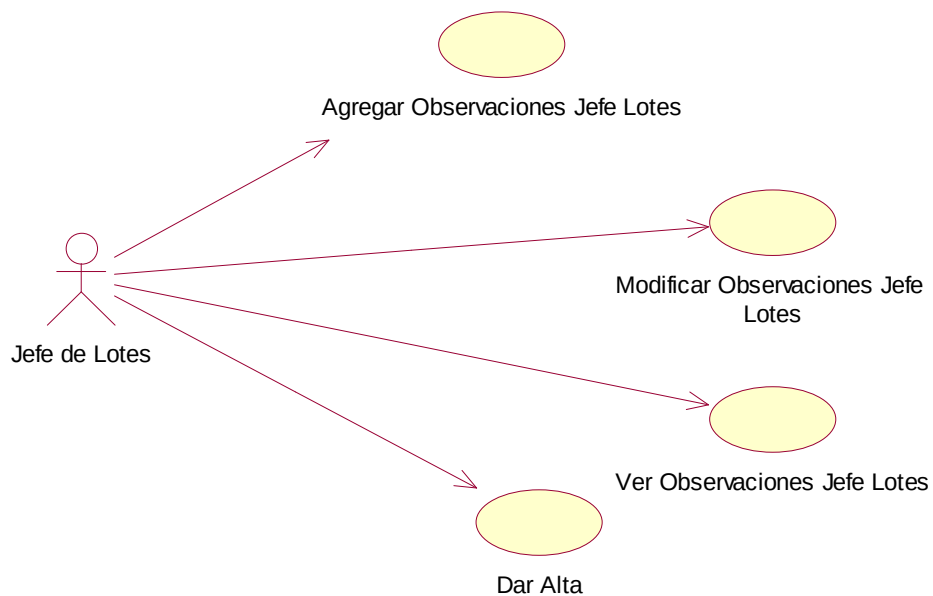


Figura 71: Caso de Uso: Administrar Observaciones Jefe Lotes.

II.3.5 MODULO DE ADMINISTRACION

II.3.5.1 CASO DE USO GENERAL

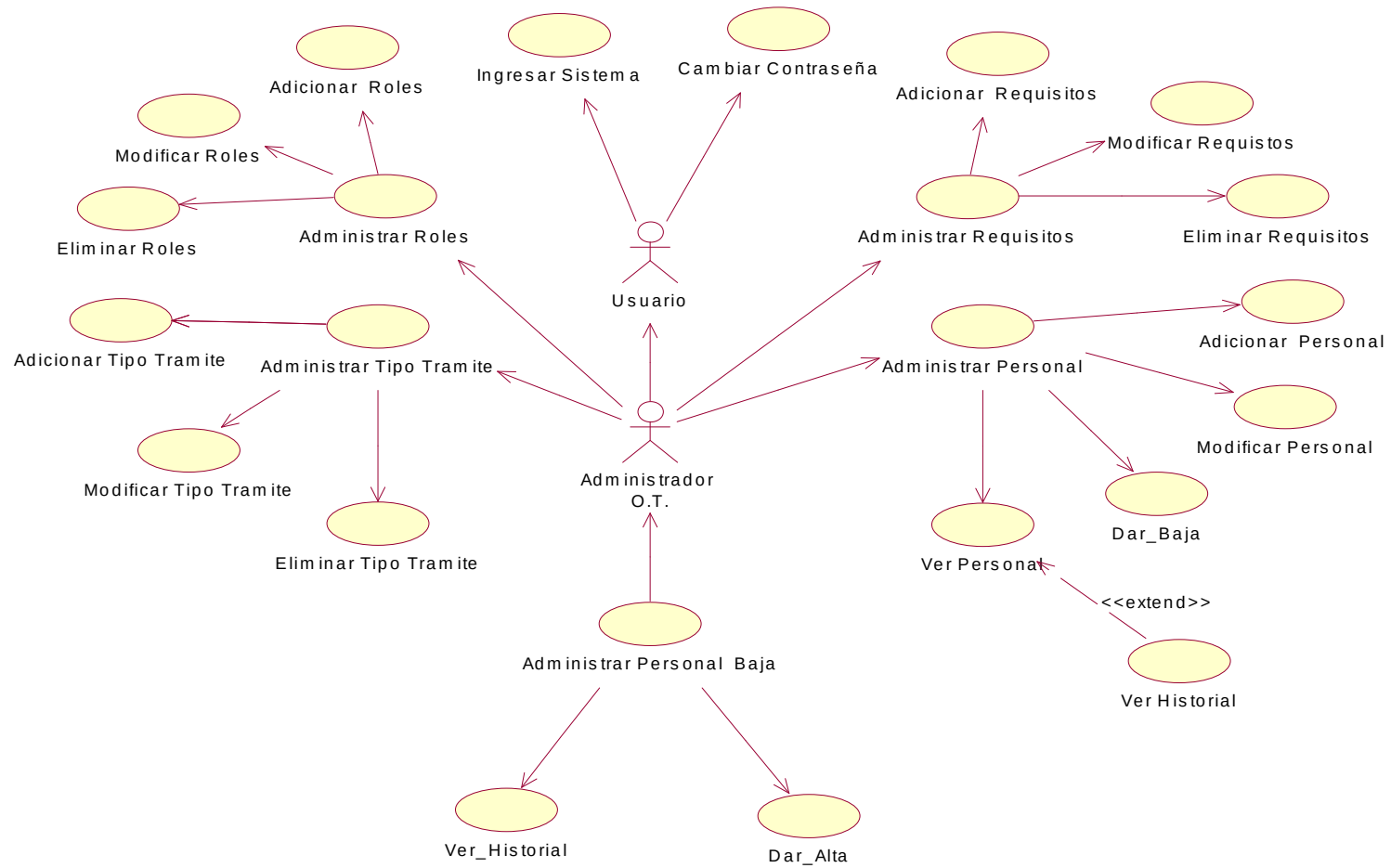


Figura 72: Caso de Uso: Caso de Uso General Modulo Administración

II.3.6 MODULO DE REPORTES

II.3.6.1 CASO DE USO ESPECIFICOS

1. Caso de Uso: Generar Reportes de Tramites

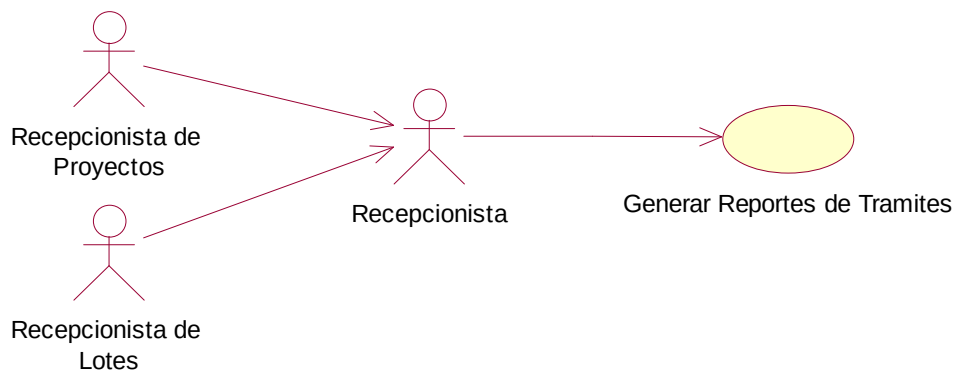


Figura 73: Caso de Uso: Generar Reportes de Tramites

2. Caso de Uso: Generar Reportes Preliminar, Acumulativos, Final

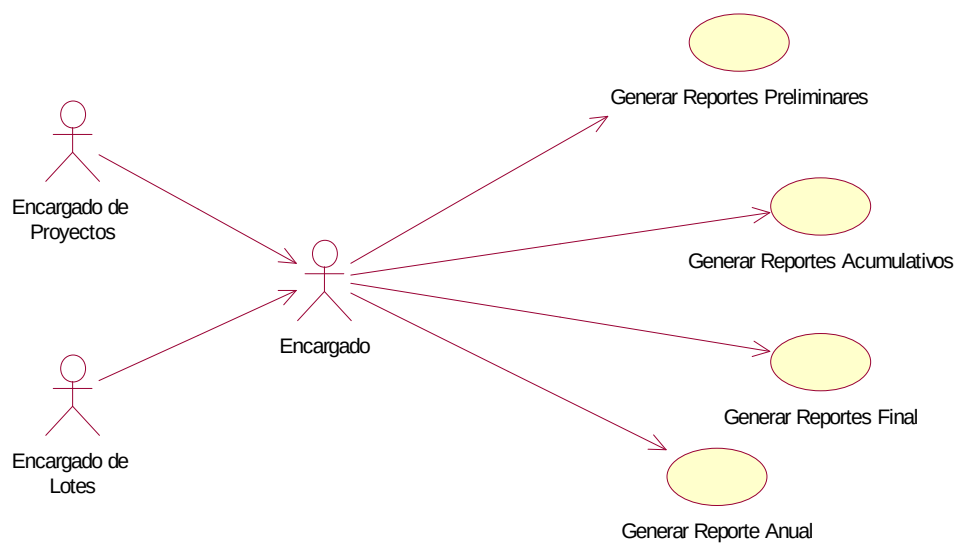


Figura 74: Caso de Uso: Generar Reportes Preliminar, Acumulativos, Final y Anual

3. Caso de Uso: Generar Reporte Técnico



Figura 75: Caso de Uso: Generar Reporte Técnico.

4. Caso de Uso: Generar Reporte de Inspección



Figura 76: Caso de Uso: Generar Reporte de Inspección.

5. Caso de Uso: Administrar Generar Reporte Legal



Figura 77: Caso de Uso: Generar Reporte Legal.

6. Caso de Uso: Generar Reporte Topográfico.



Figura 78: Caso de Uso: Generar Reporte Topográfico.

CAPITULO III:

ANÁLISIS

CAPITULO III: ANÁLISIS

III.1 MODELADO DE DIAGRAMA DE ACTIVIDADES

III.1.1. Introducción

Una vista de actividades está compuesta por los diagramas de actividades que reflejan las actividades de computación implicadas en la ejecución del sistema. Los diagramas de actividad se utilizan para modelar los aspectos dinámicos de un sistema, esto implica modelar los pasos secuenciales de un proceso.

II.1.1.1. Propósito

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

II.1.1.2. Alcance

- Describir los procesos de sistema.
- 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.

III.1.2 Diagrama de Actividades

III.1.2.1 Interactuar con Recursos del Sistema:

III.1.2.1.1 Ingresar Sistema

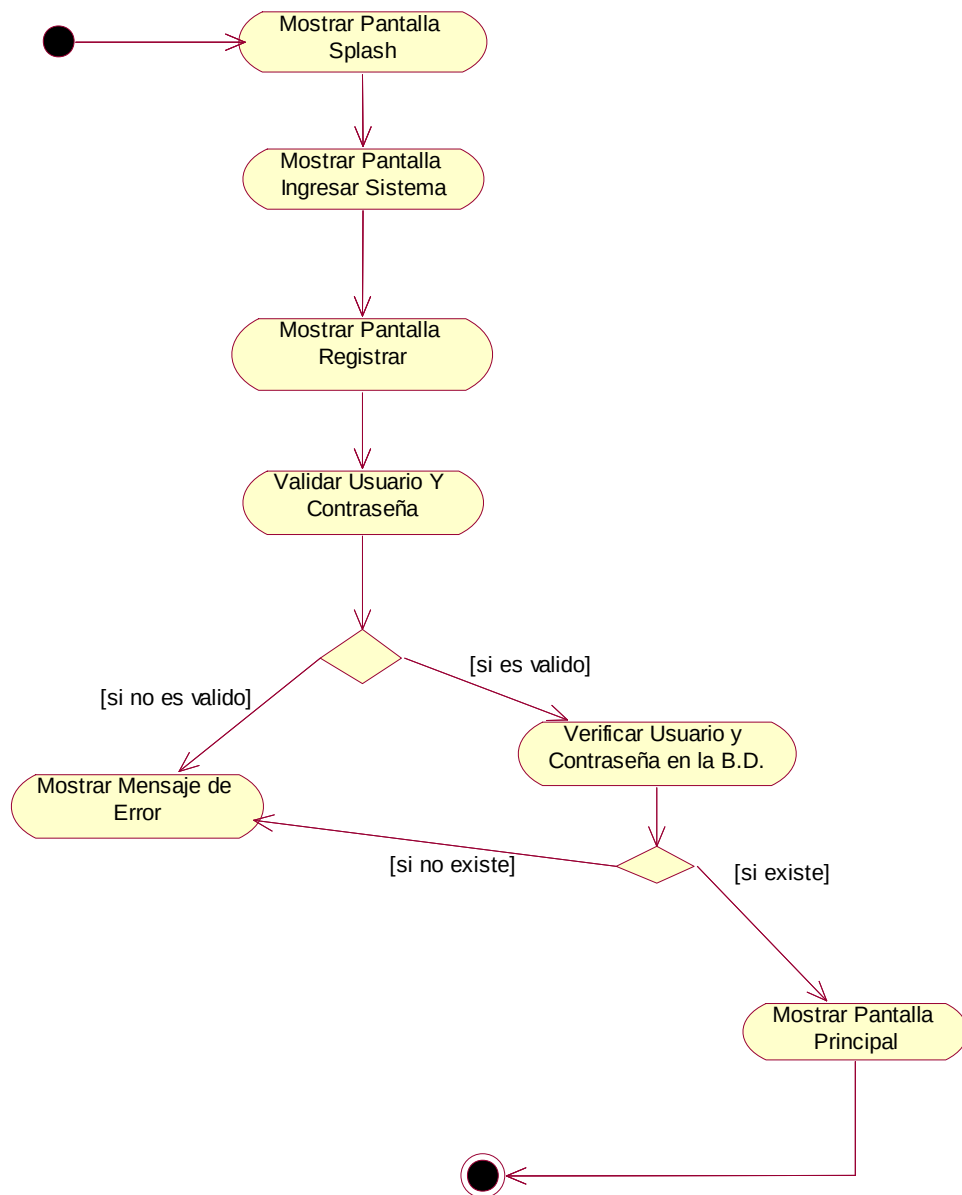


Figura 79: Diagrama de Actividad: Ingresar Sistema

III.1.2.1.1 Resguardar Base de Datos

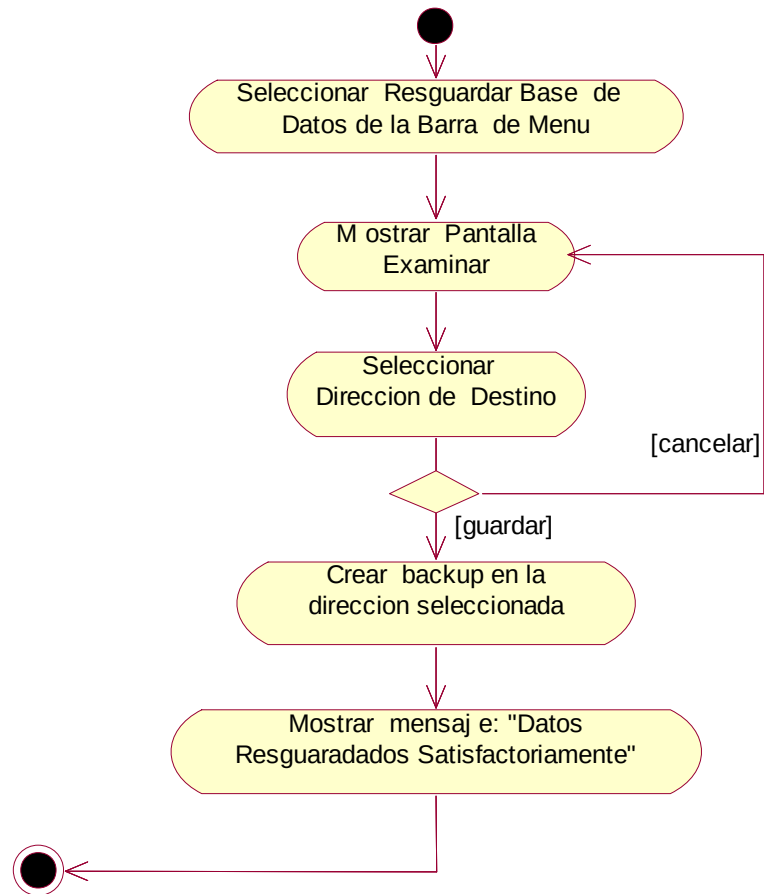


Figura 80: Diagrama de Actividad: Resguardar Base de Datos

III.1.2.1.1 Restaurar Base de Datos

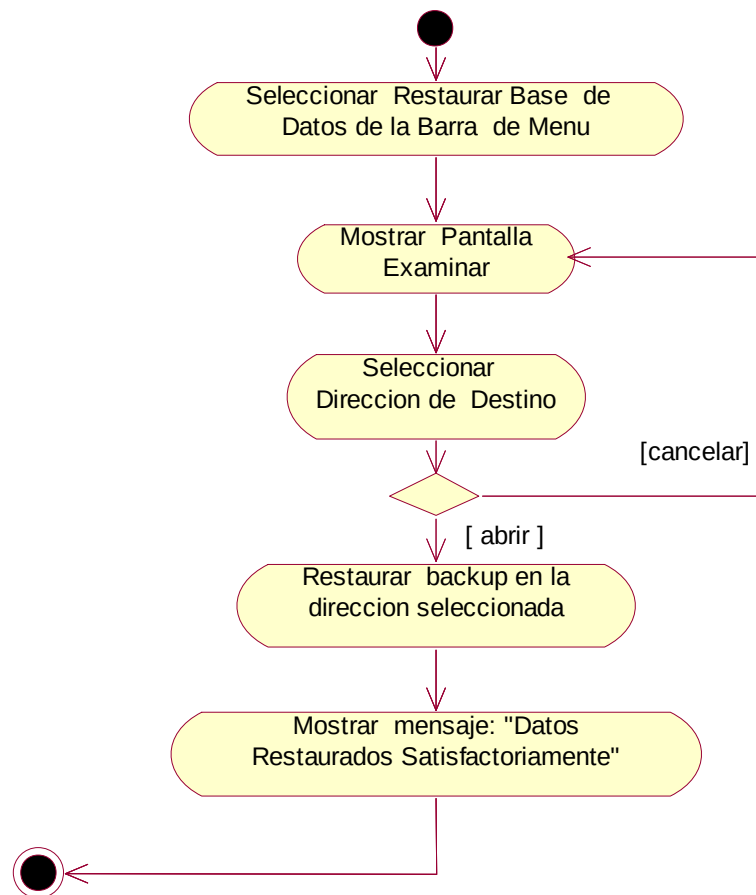


Figura 81: Diagrama de Actividad: Restaurar Base de Datos

III.1.2.1.1 Cerrar Conexión

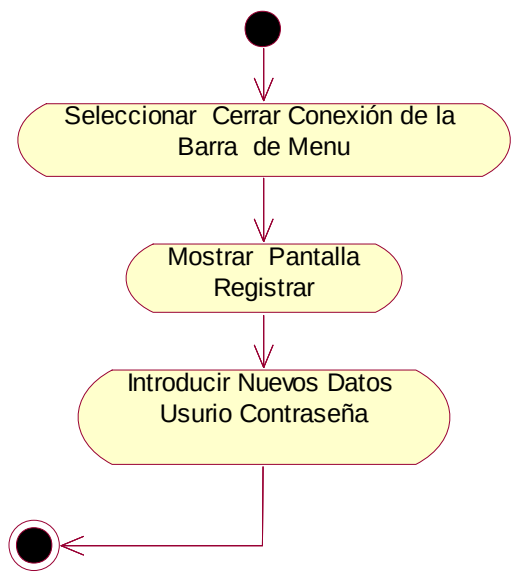


Figura 82: Diagrama de Actividad: Cerrar Conexión

III.1.2.1.1 Salir del Sistema

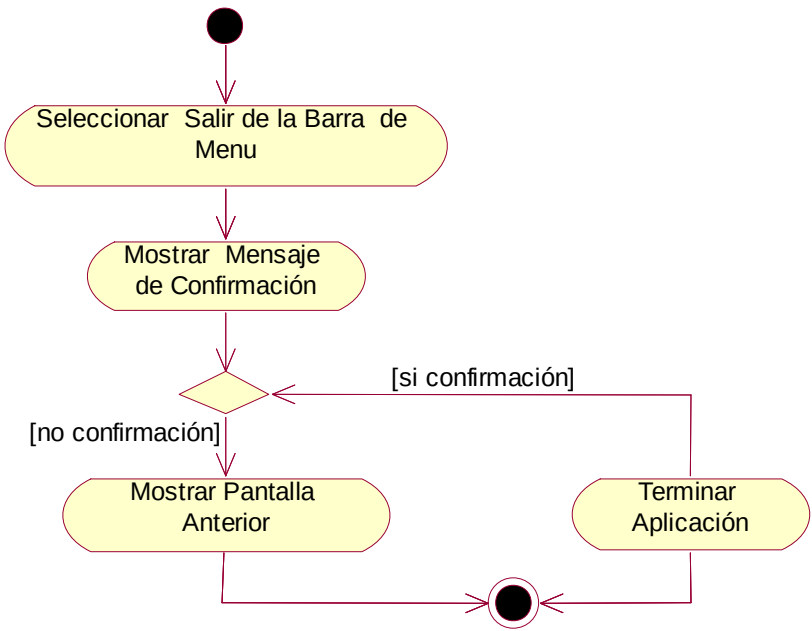


Figura 83: Diagrama de Actividad: Salir del Sistema

III.1.2.1.1 Obtener Ayuda

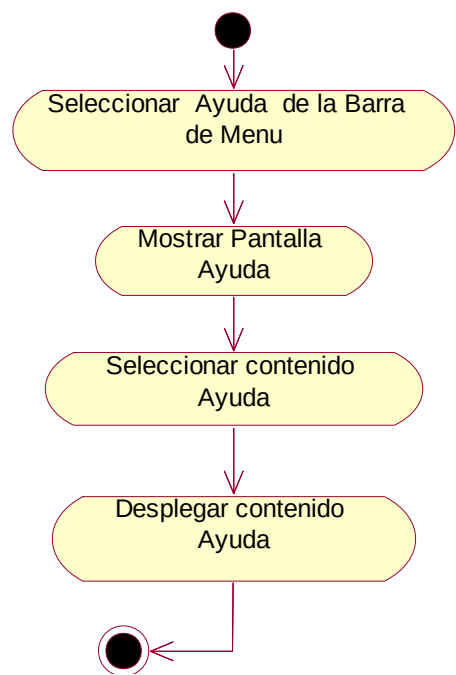


Figura 84: Diagrama de Actividad: Obtener Ayuda

III.1.2.2 MODULO DE PROYECTOS

III.1.2.2.1 Administrar Propietarios

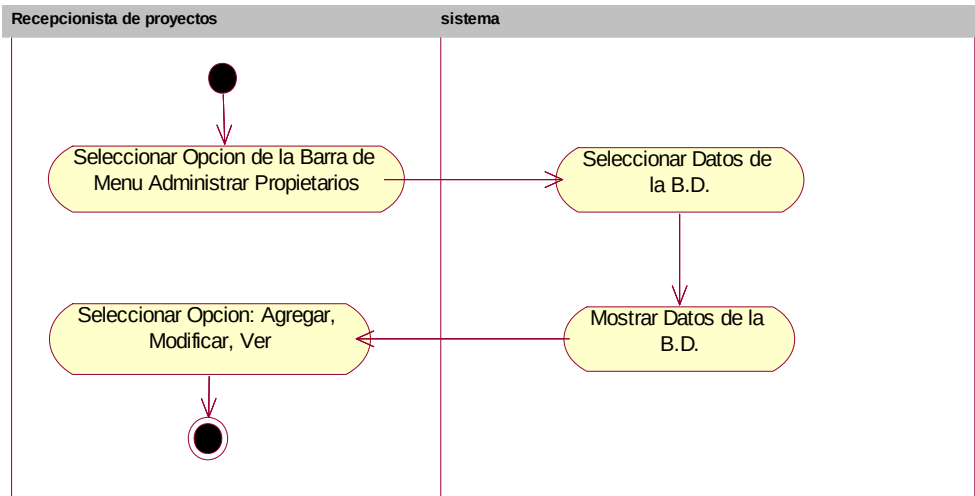


Figura 85: Diagrama de Actividad: Administrar Propietarios

III.1.2.2.2 Adicionar Propietarios

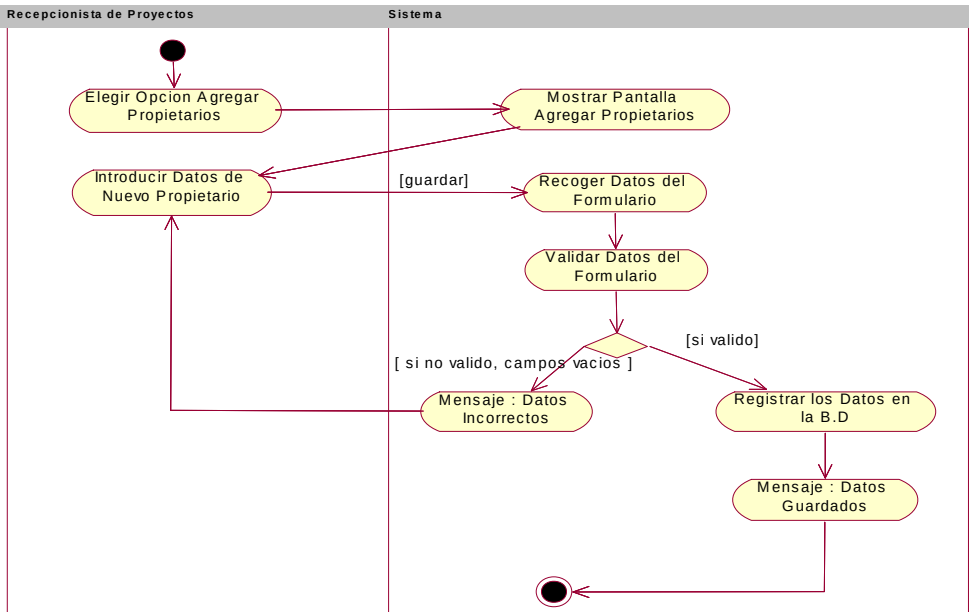


Figura 86: Diagrama de Actividad: Adicionar Propietarios

III.1.2.2.3 Adicionar Projectista

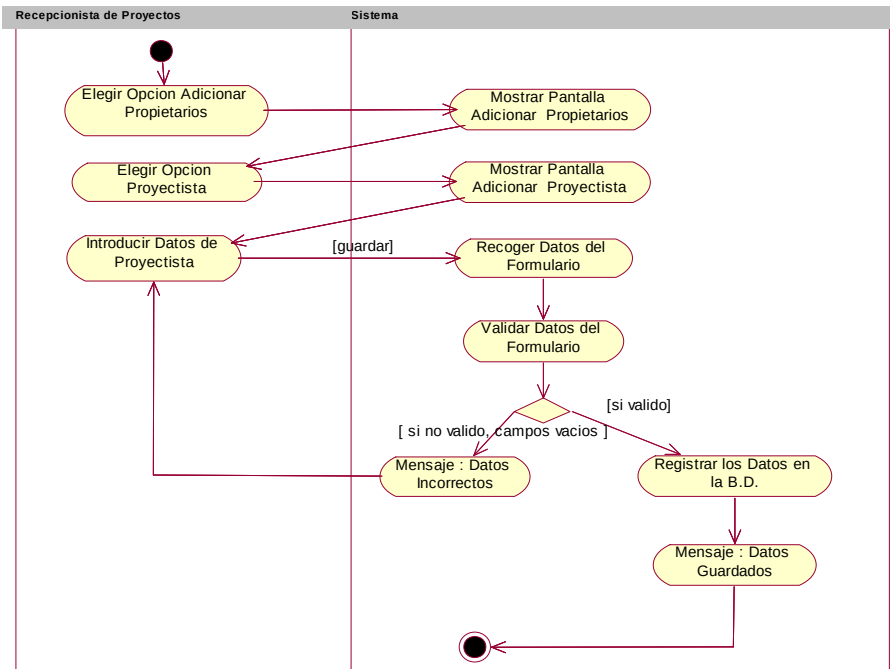


Figura 87: Diagrama de Actividad: Adicionar Projectista

III.1.2.2.4 Adicionar Apoderado

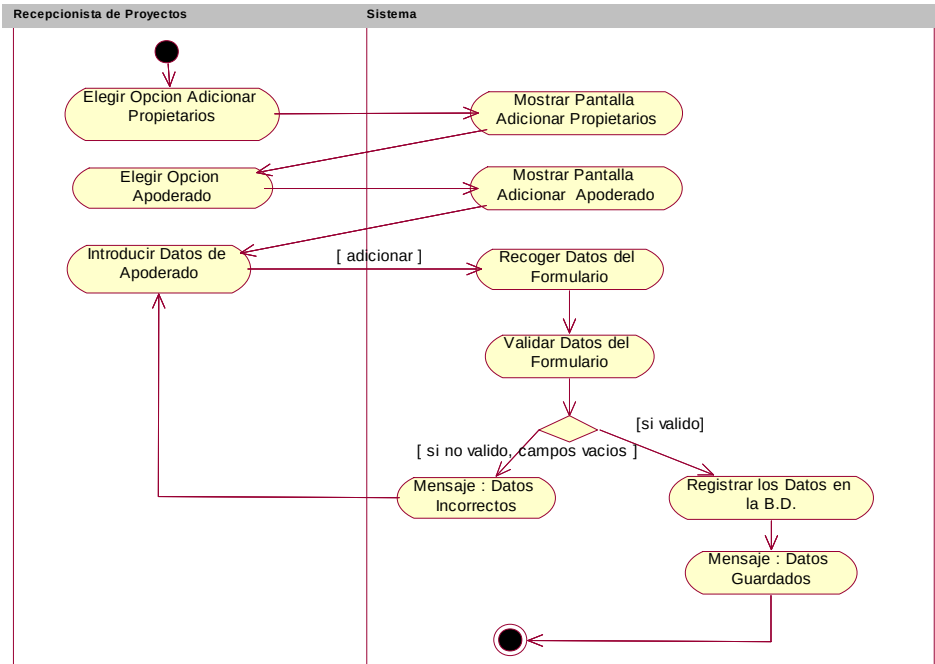


Figura 88: Diagrama de Actividad: Adicionar Apoderado

III.1.2.2.5 Modificar Propietarios

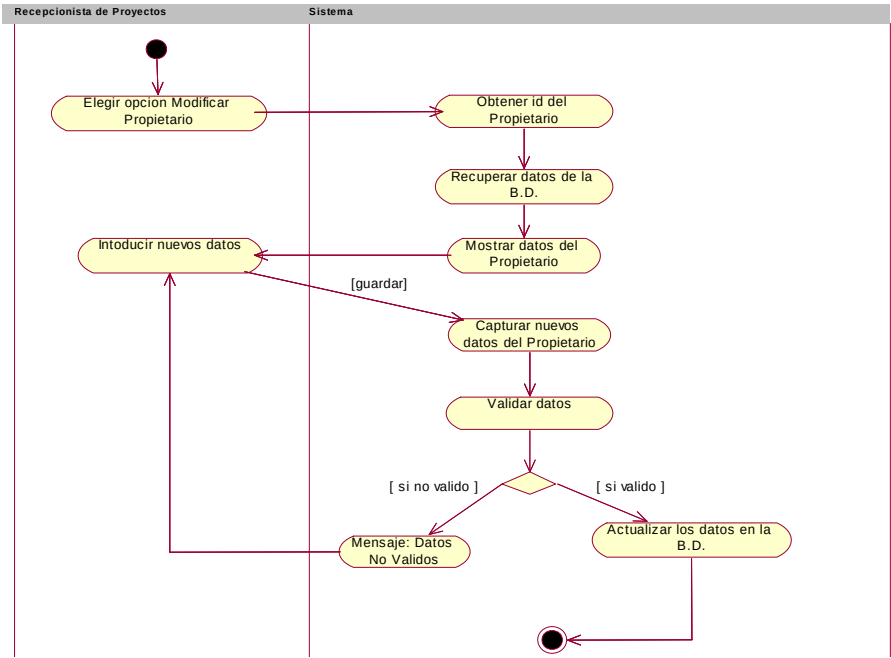


Figura 89: Diagrama de Actividad: Modificar Propietarios

III.1.2.2.6 Modificar Projectista

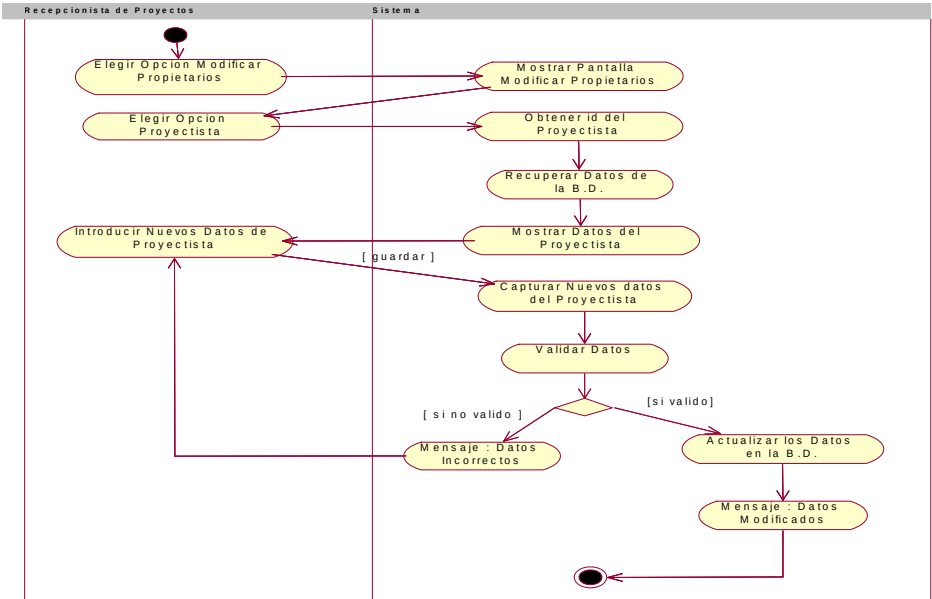


Figura 90: Diagrama de Actividad: Modificar Projectista

III.1.2.2.7 Modificar Apoderado

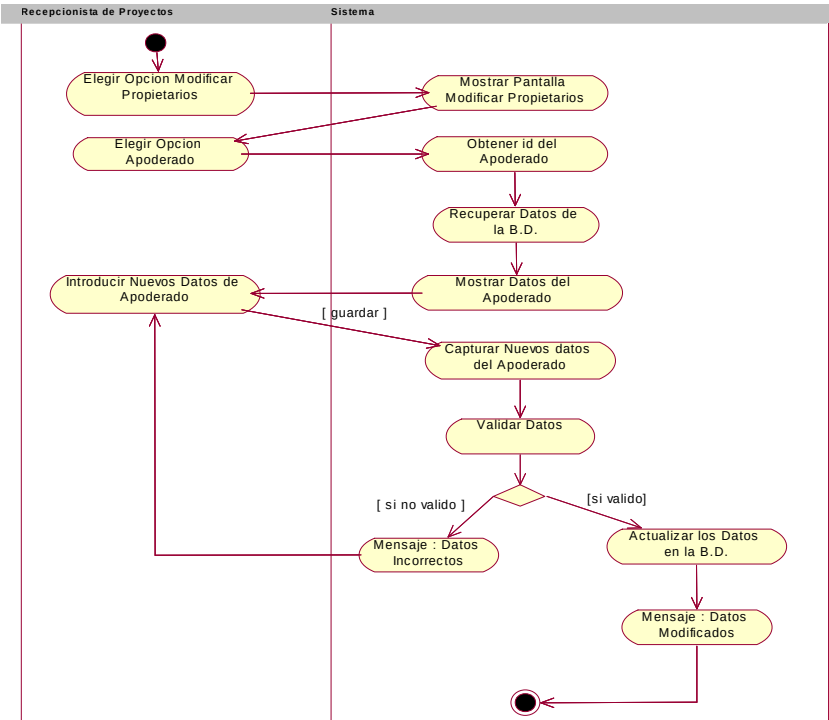


Figura 91: Diagrama de Actividad: Modificar Apoderado

III.1.2.28 Ver Propietarios

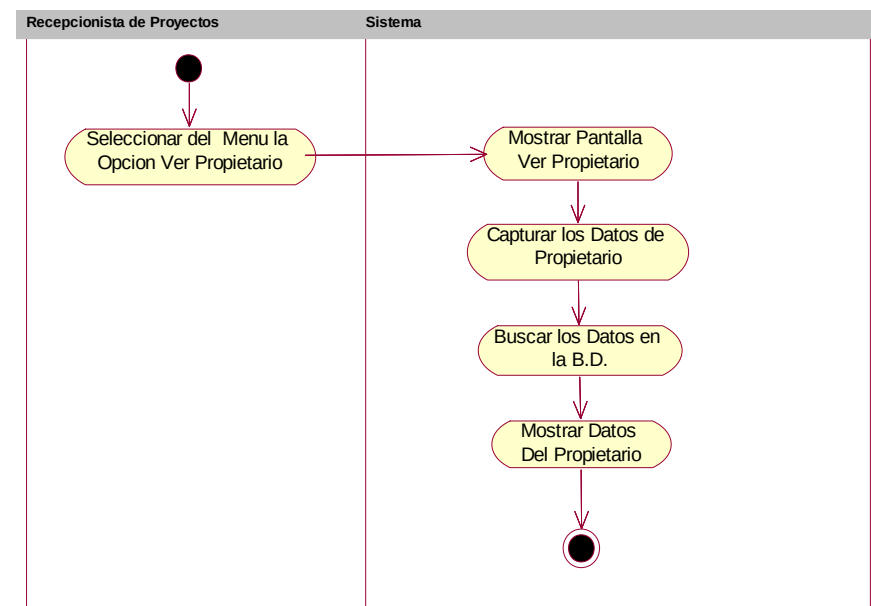


Figura 92: Diagrama de Actividad: Ver Propietarios

III.1.2.2.9 Administrar Observaciones Generales

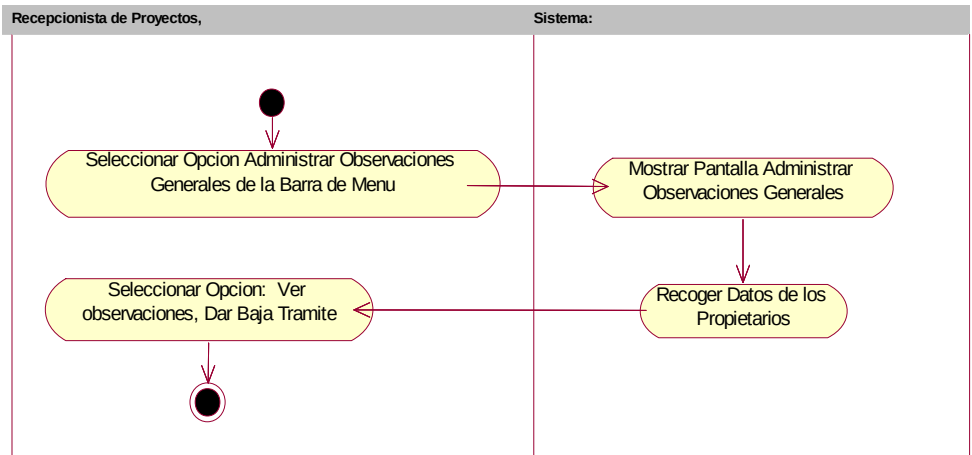


Figura 93: Diagrama de Actividad: Administrar Observaciones

III.1.2.2.10 Ver Observaciones Generales

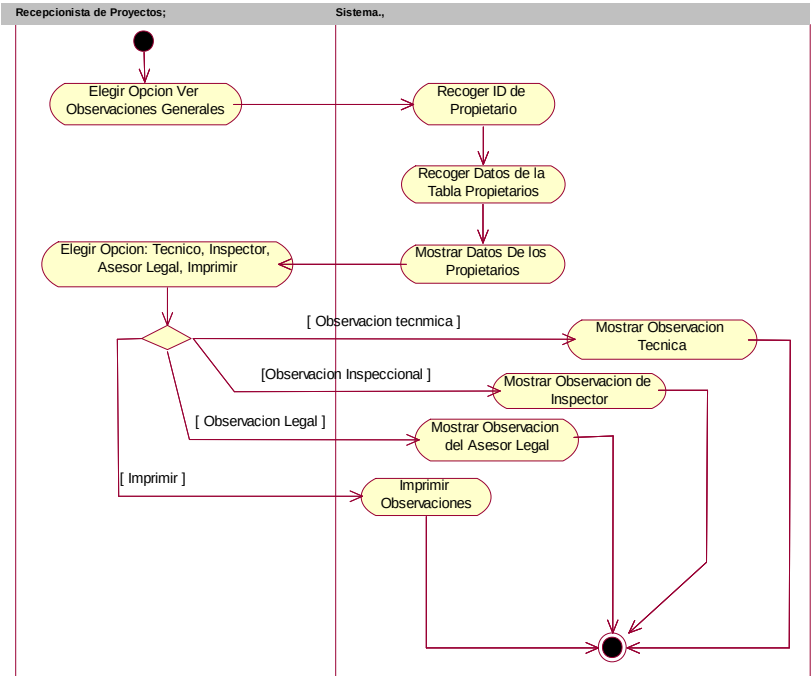


Figura 94: Diagrama de Actividad: Ver Observaciones Generales

III.1.2.2.11 Dar Baja Trámite

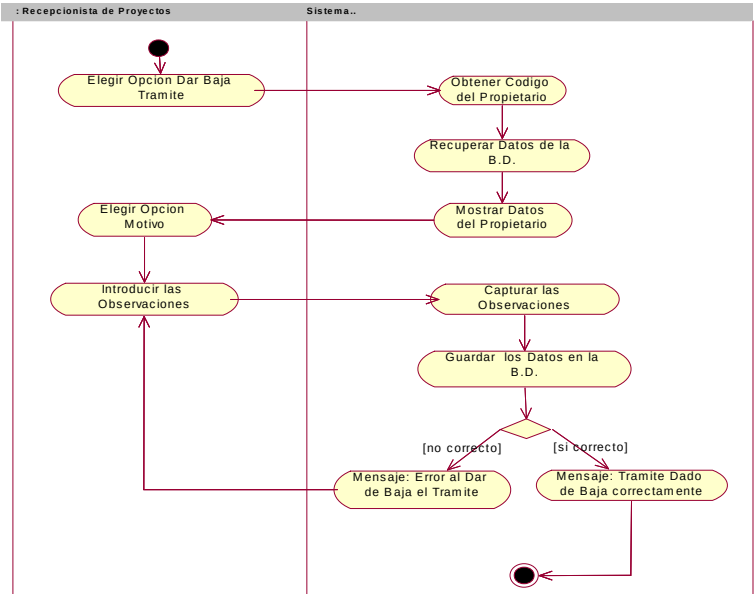


Figura 95: Diagrama de Actividad: Dar Baja Trámite

III.1.2.2.12 Reasignar Tramite

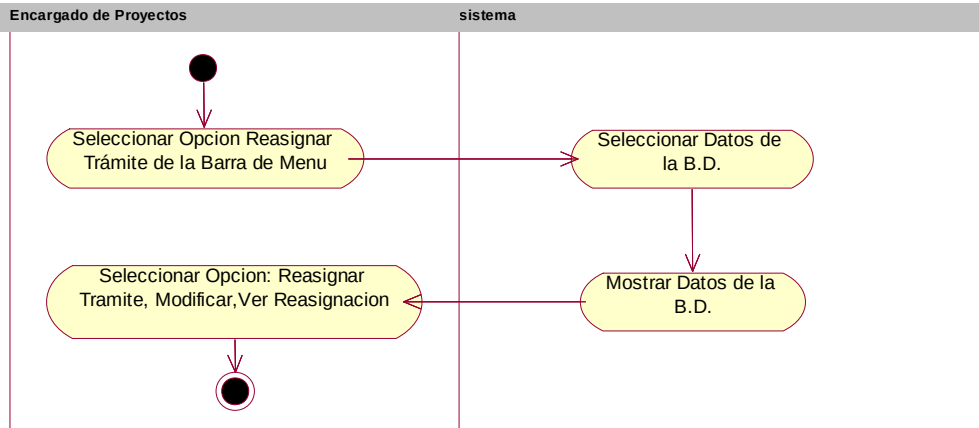


Figura 96: Diagrama de Actividad: Reasignar Trámite

III.1.2.2.13 Registrar Reasignación

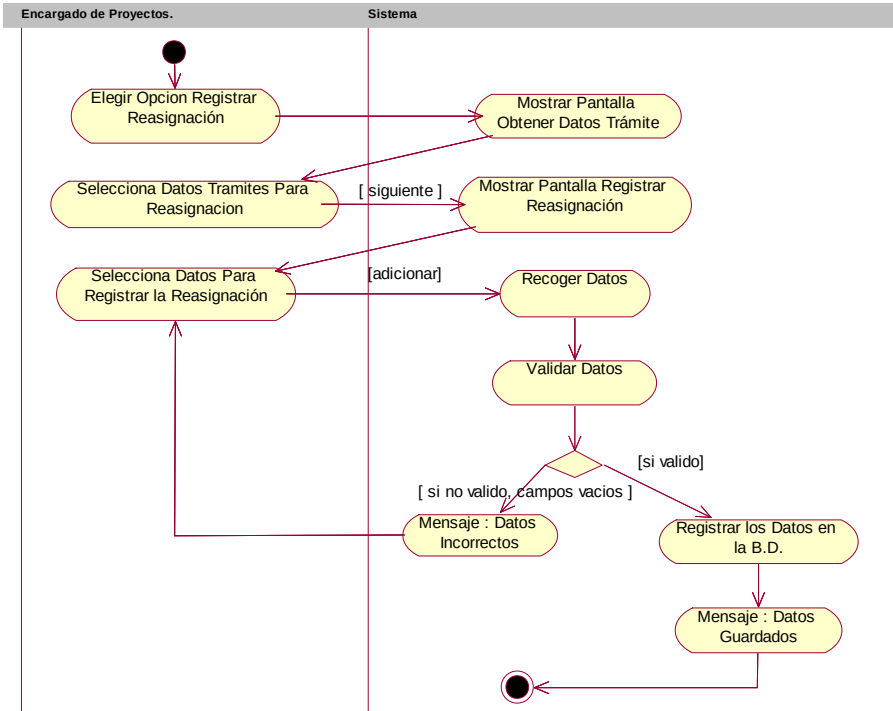


Figura 97: Diagrama de Actividad: Registrar Reasignación

III.1.2.2.14 Ver Reasignación

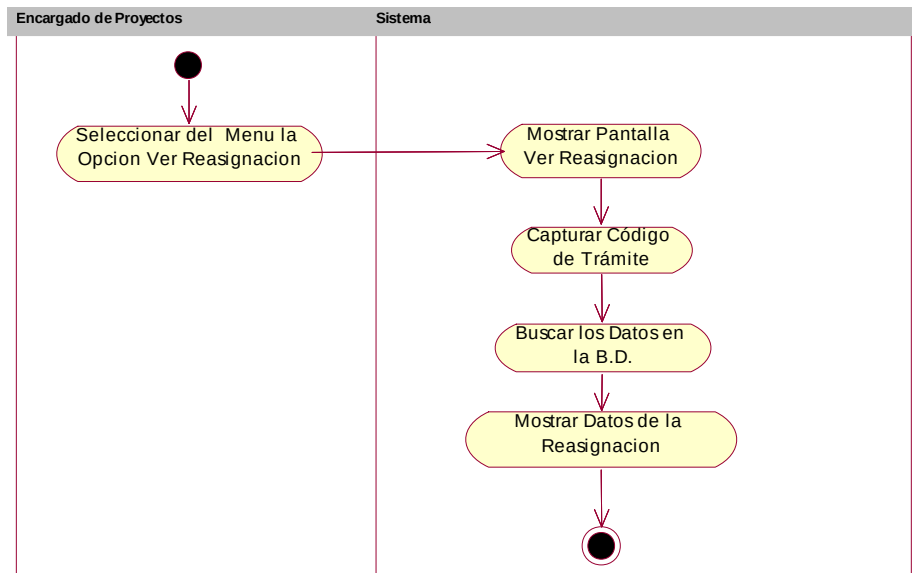


Figura 98: Diagrama de Actividad: Ver Reasignación

III.1.2.2.15 Ver Datos Trámite

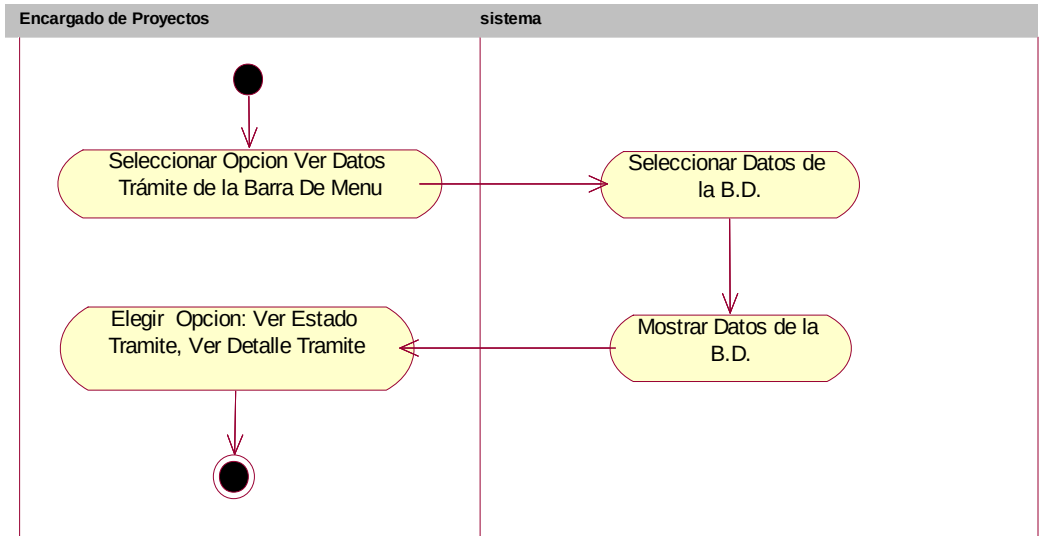


Figura 99: Diagrama de Actividad: Administrar Observaciones Generales

III.1.2.2.16 Ver Estado Tramite

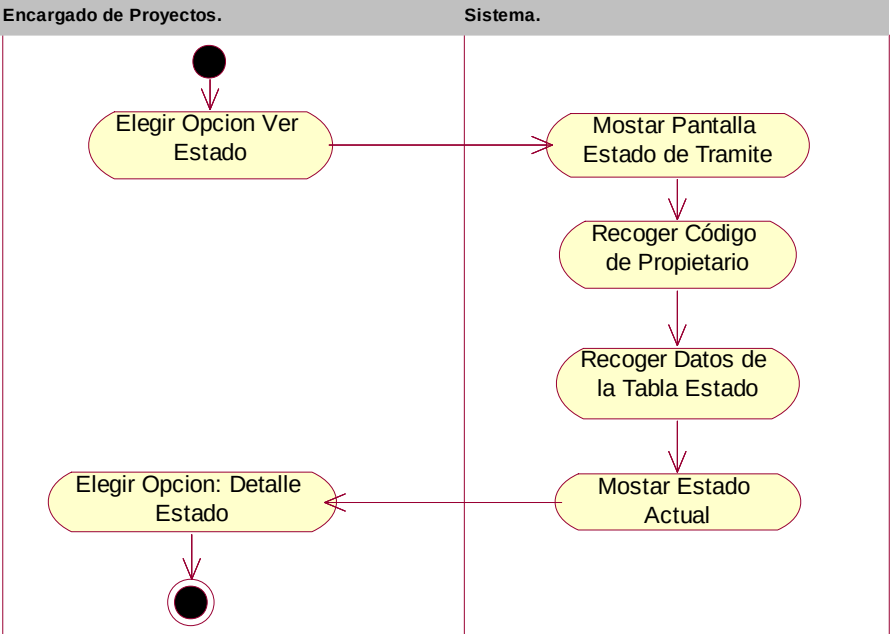


Figura 100: Diagrama de Actividad: Ver Estado Trámite

III.1.2.2.17 Ver Detalle Trámite

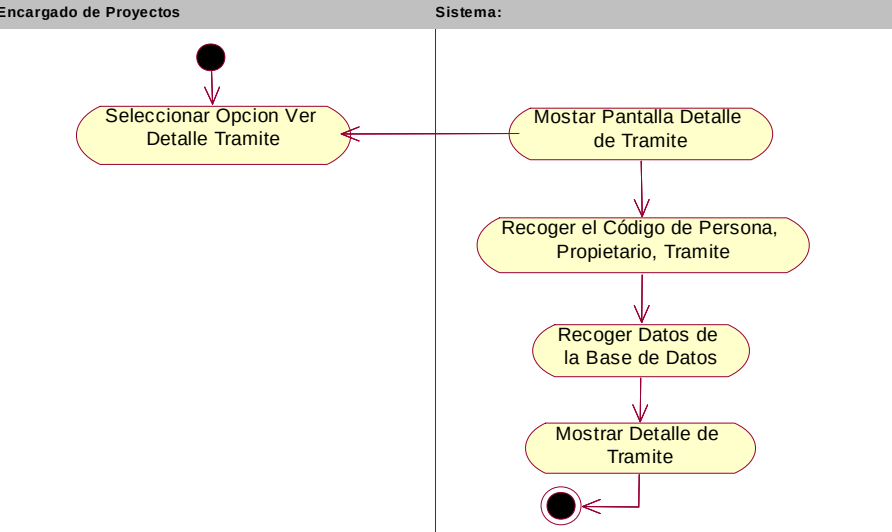


Figura 101: Diagrama de Actividad: Ver Detalle Trámite

III.1.2.2.18 Administrar Ingresos

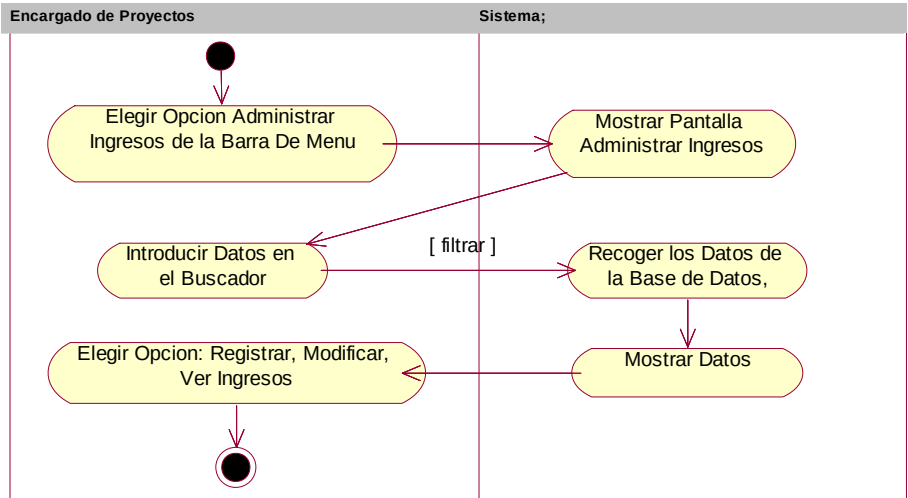


Figura 102: Diagrama de Actividad: Administrar Ingresos

III.1.2.2.19 Registrar Ingresos

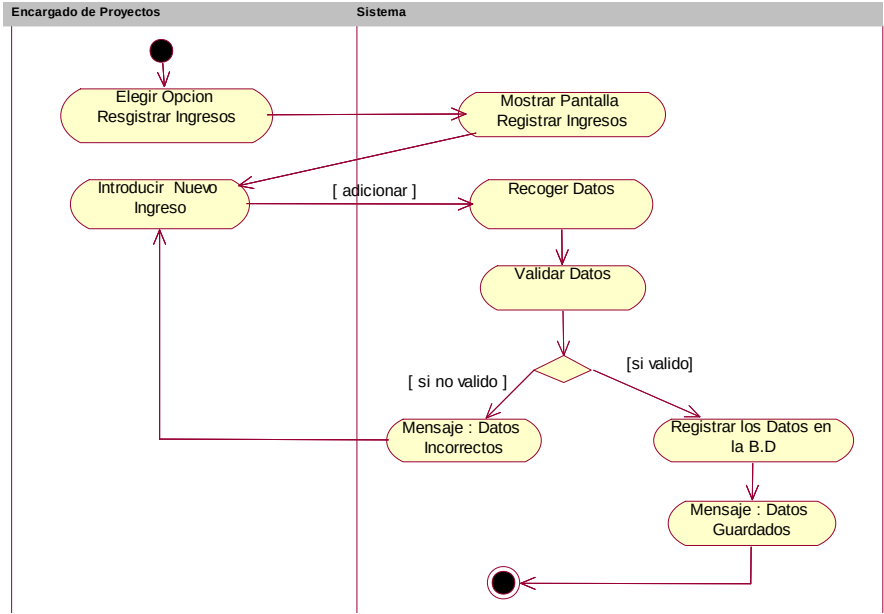


Figura 103: Diagrama de Actividad: Registrar Ingresos

III.1.2.2.20 Modificar Ingresos

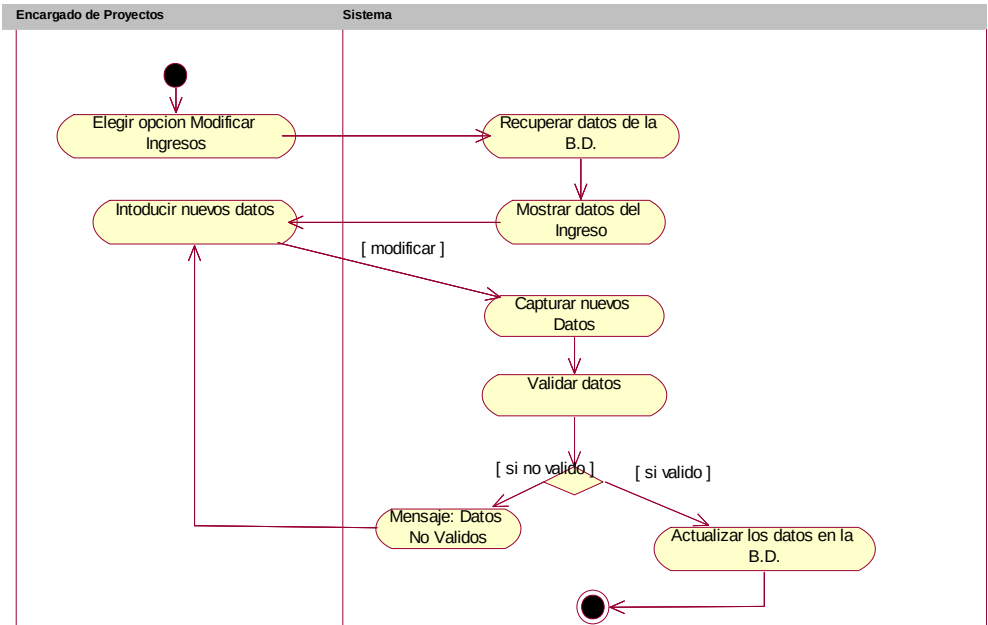


Figura 104: Diagrama de Actividad: Modificar Ingresos

III.1.2.2.21 Ver Ingresos

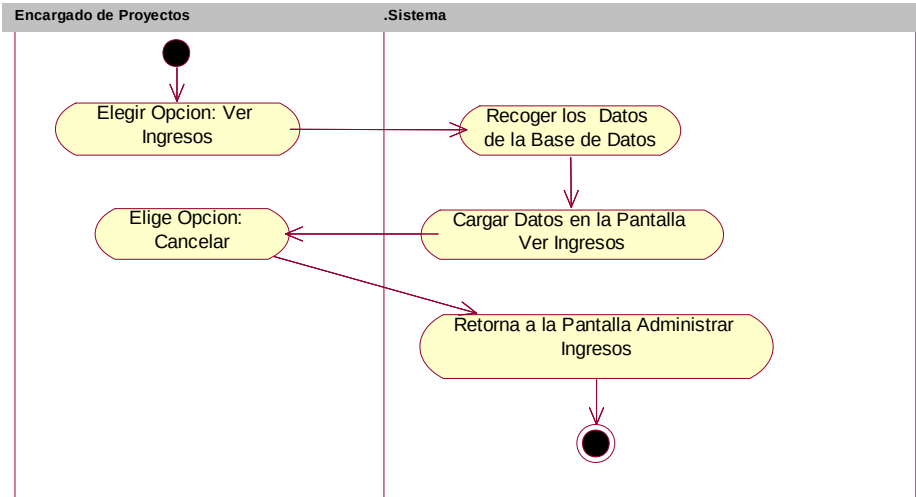


Figura 105: Diagrama de Actividad: Ver Ingresos

III.1.2.2.22 Administrar Salidas

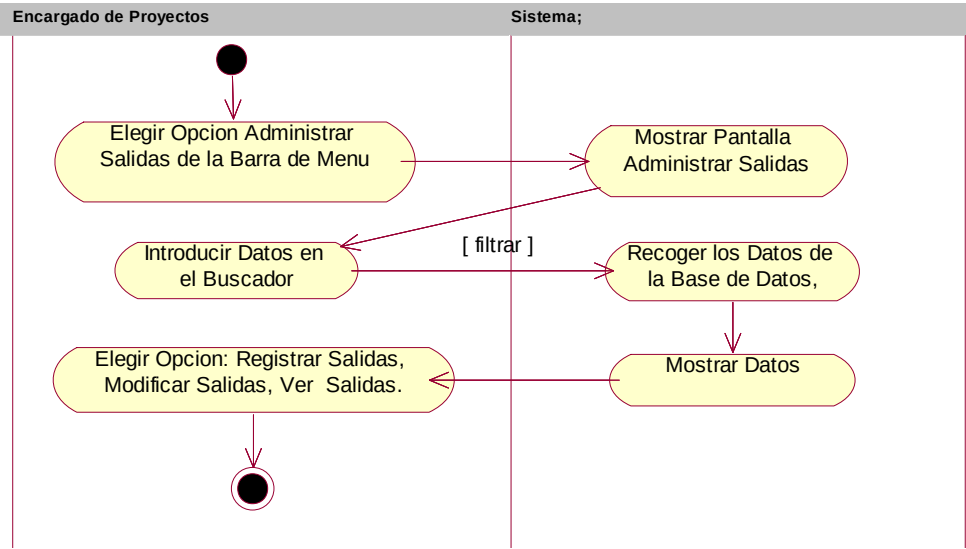


Figura 106: Diagrama de Actividad: Administrar Salidas

III.1.2.2.23 Registrar Salidas

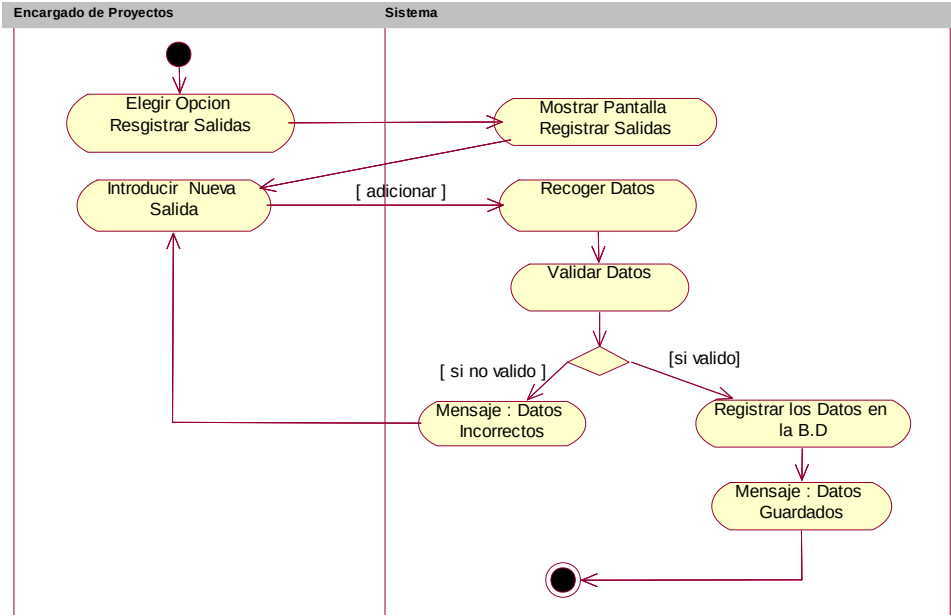


Figura 107: Diagrama de Actividad: Registrar Salidas

III.1.2.2.24 Modificar Salidas

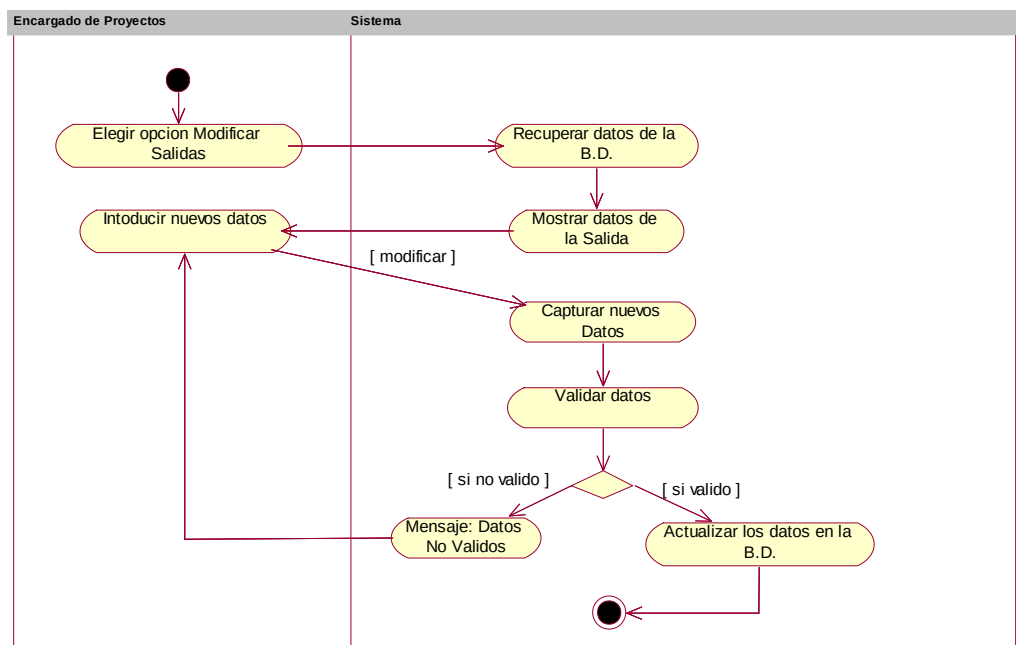


Figura 108: Diagrama de Actividad: Modificar Salidas

III.1.2.2.25 Ver Salidas

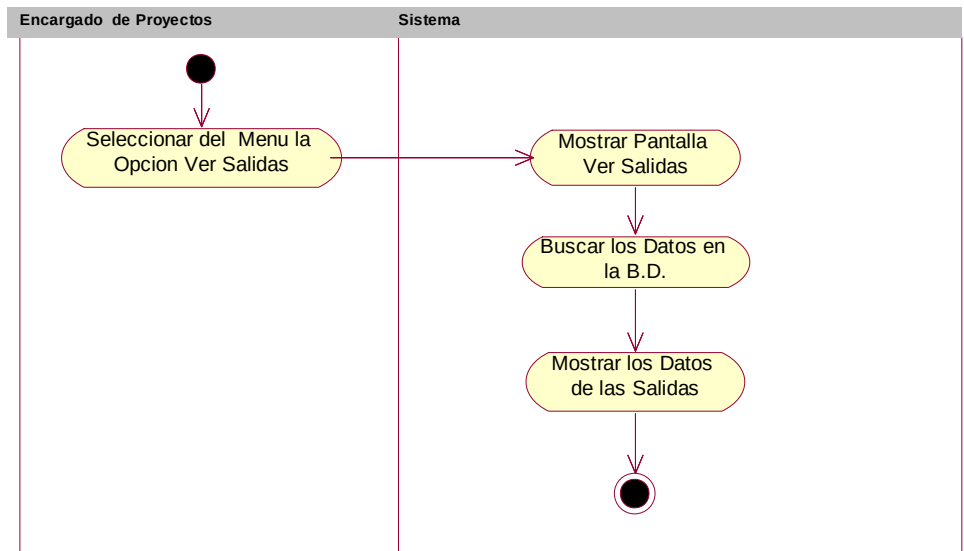


Figura 109: Diagrama de Actividad: Ver Salidas

III.1.2.2.26 Administrar Envíos

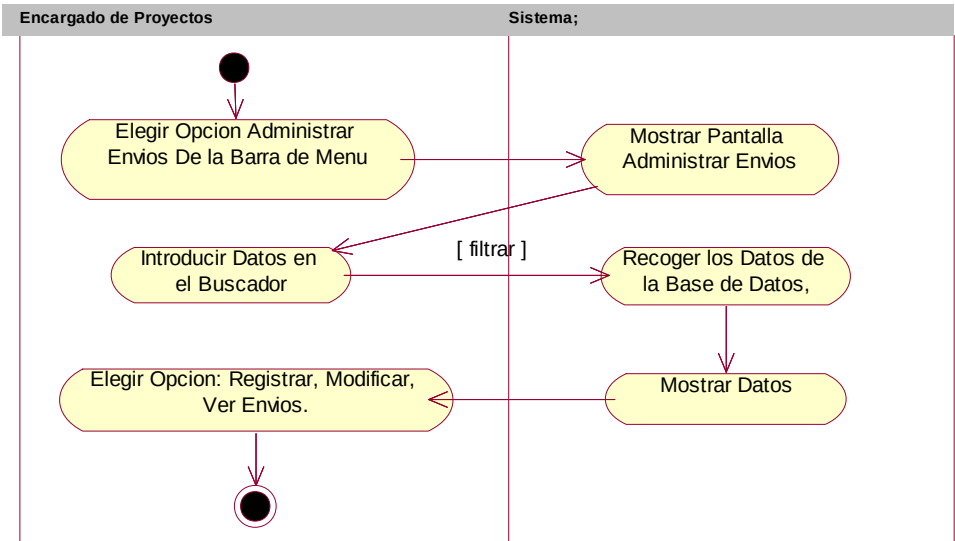


Figura 110: Diagrama de Actividad: Administrar Envíos.

III.1.2.2.27 Registrar Envíos

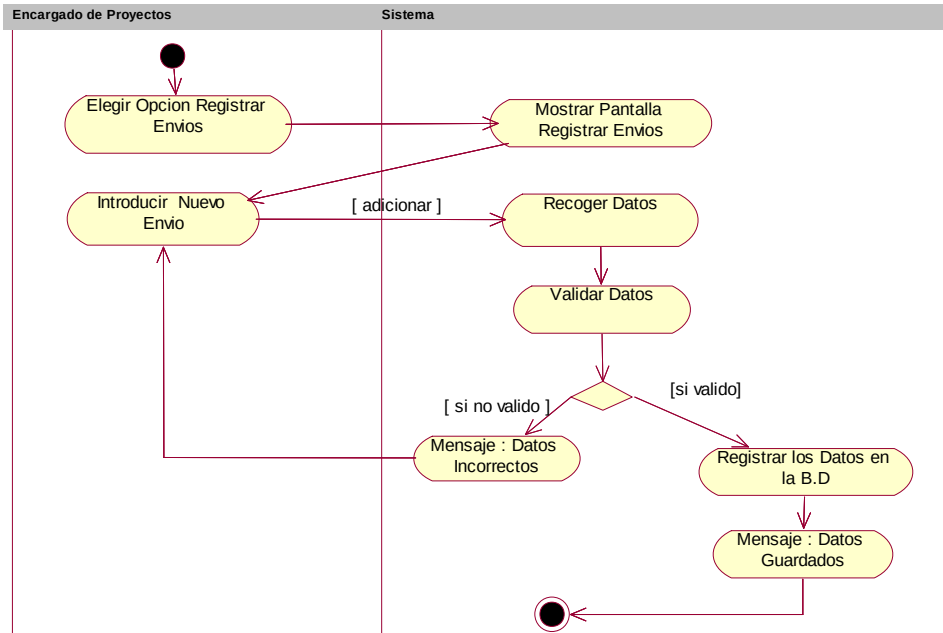


Figura 111: Diagrama de Actividad: Registrar Envíos

III.1.2.2.28 Modificar Envíos

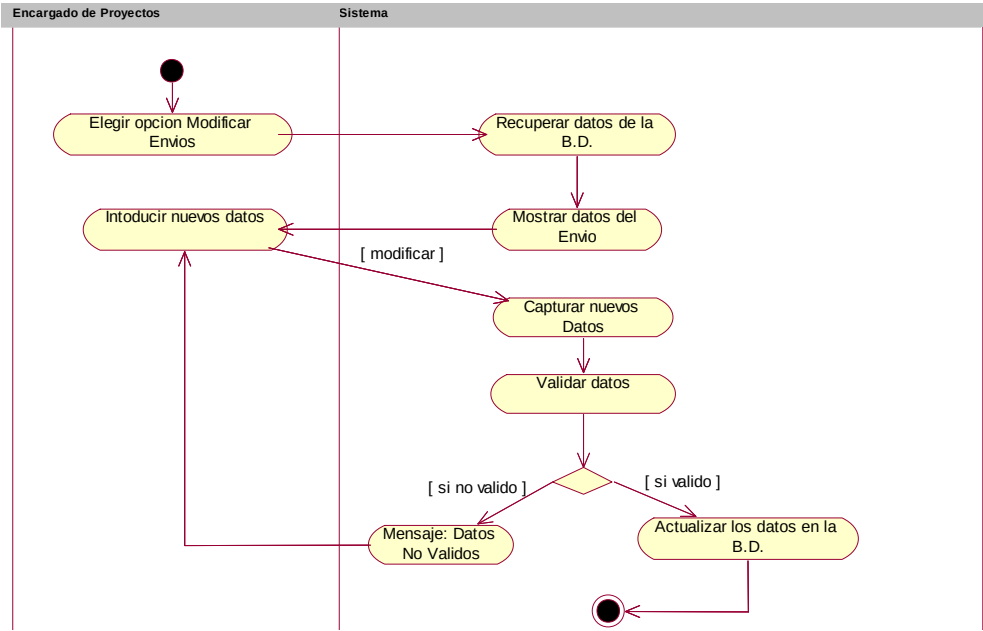


Figura 112: Diagrama de Actividad: Modificar Envíos

III.1.2.2.29 Ver Envíos

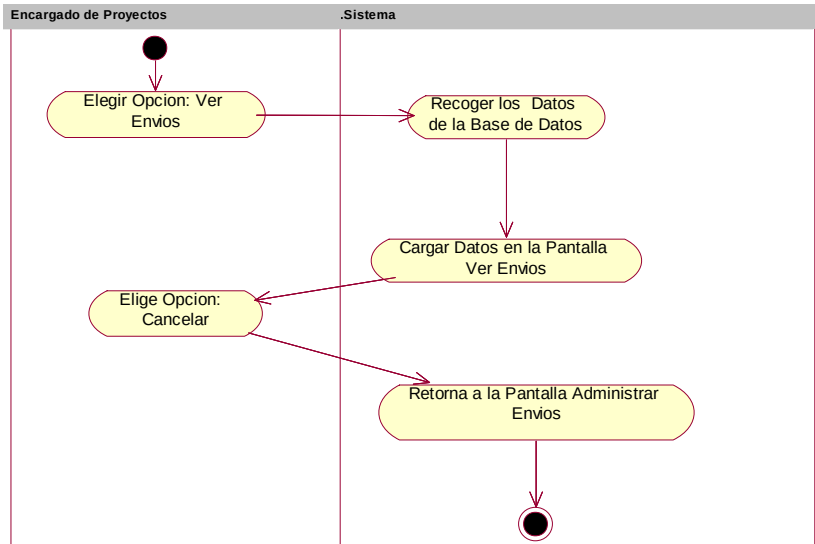


Figura 113: Diagrama de Actividad: Ver Envíos

III.1.2.2.30 Administrar Recepción

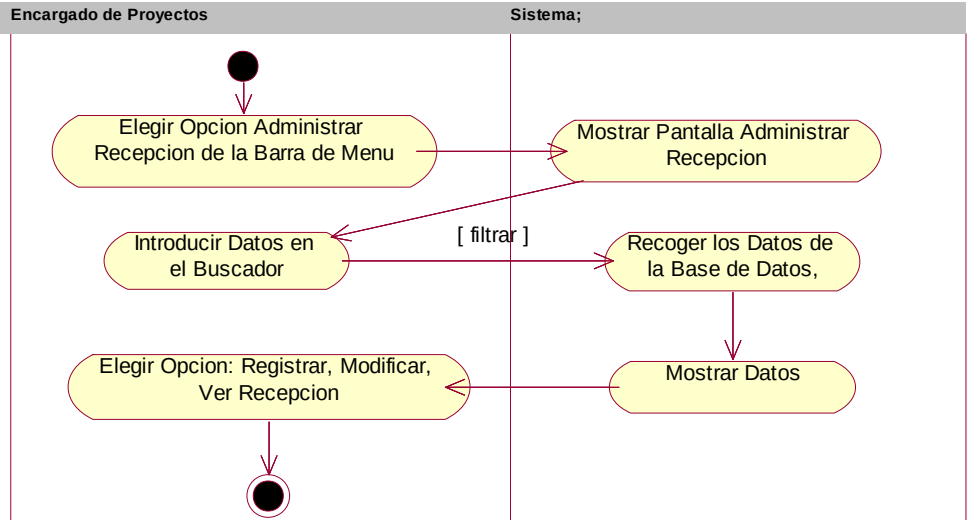


Figura 114: Diagrama de Actividad: Administrar Recepción

III.1.2.2.31 Registrar Recepción

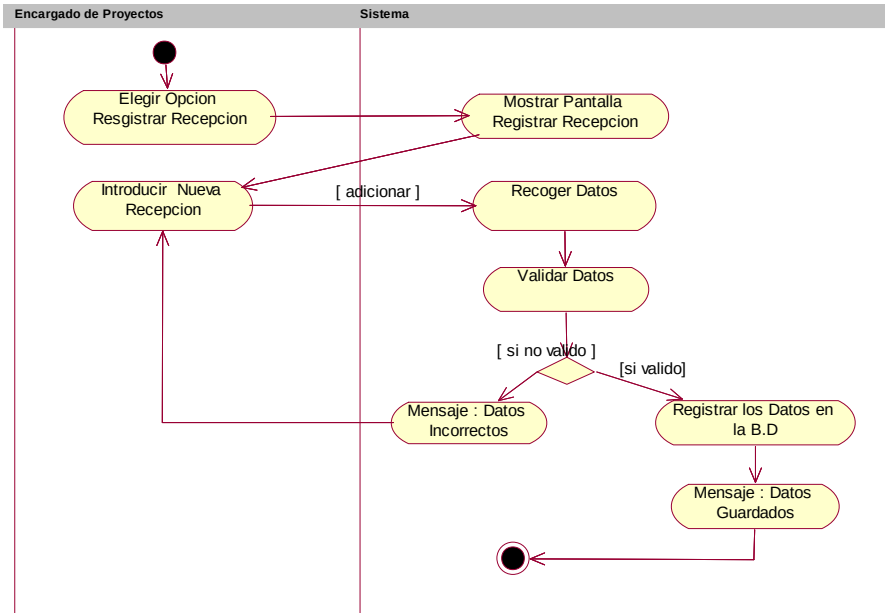


Figura 114: Diagrama de Actividad: Registrar Recepción

III.1.2.2.32 Modificar Recepción

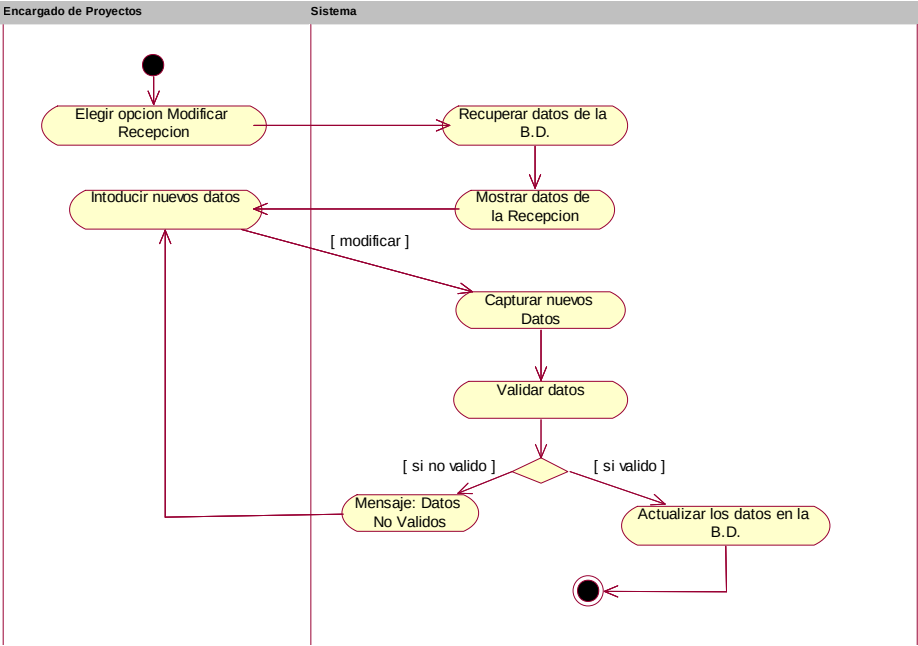


Figura 115: Diagrama de Actividad: Modificar Recepción

III.1.2.2.33 Ver Recepción

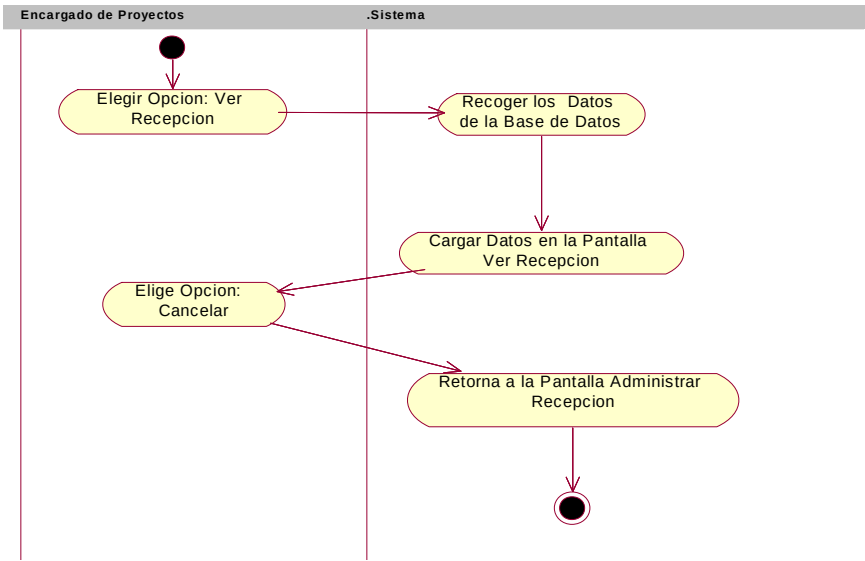


Figura 116: Diagrama de Actividad: Ver Recepción

III.1.2.2.40 Administrar Observaciones

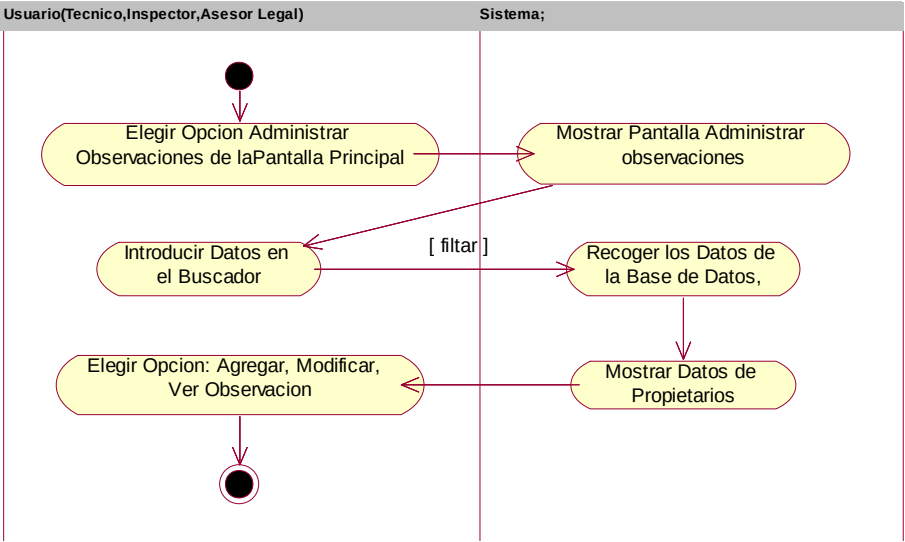


Figura 123: Diagrama de Actividad: Administrar Observaciones

III.1.2.2.41 Adicionar Observaciones

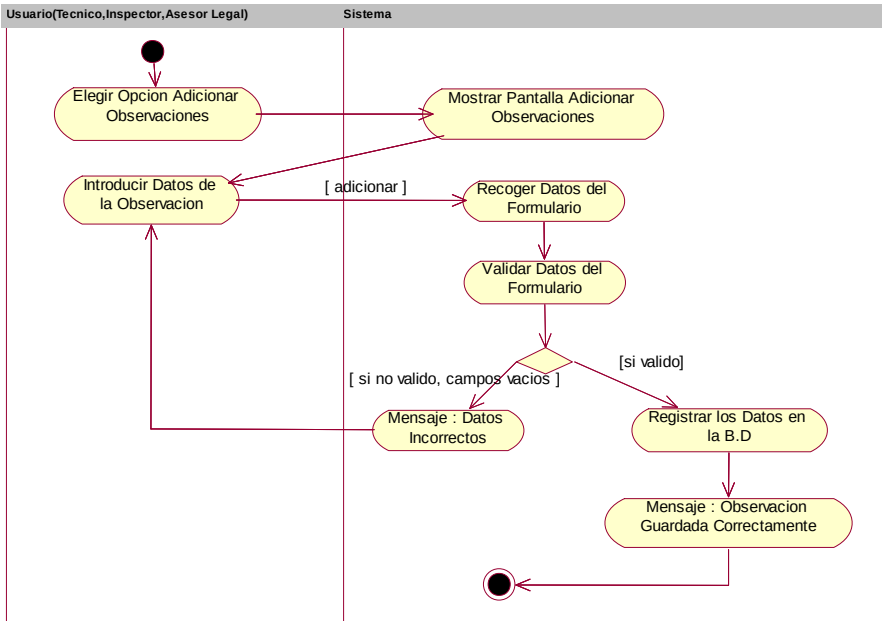


Figura 124: Diagrama de Actividad: Agregar Observaciones

III.1.2.2.42 Modificar Observaciones

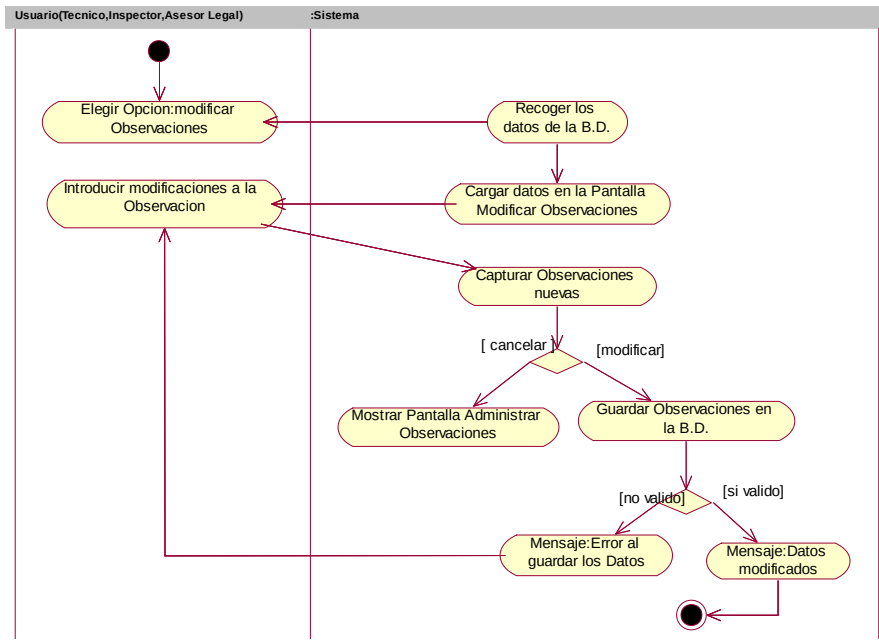


Figura 125: Diagrama de Actividad: Modificar Observaciones

III.1.2.2.44 Ver Observaciones

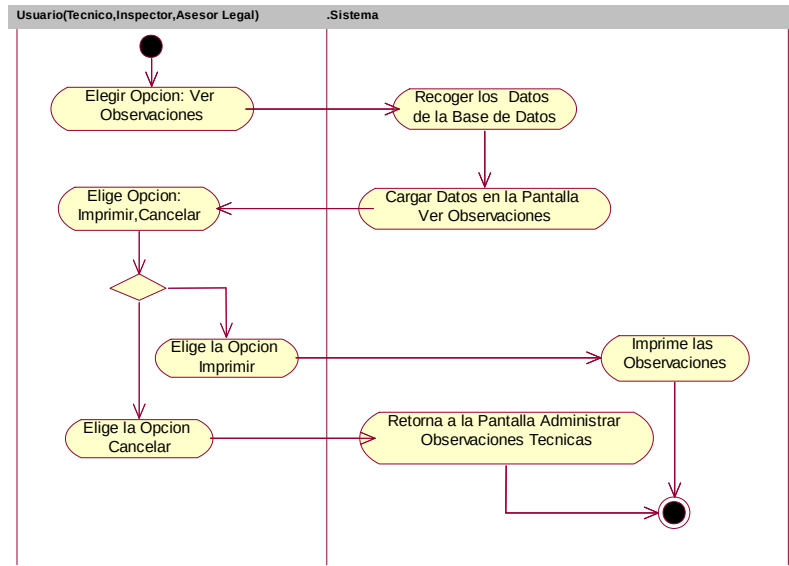


Figura 126: Diagrama de Actividad: Ver Observaciones

III.1.2.2.55 Administrar Observaciones Jefe Proyectos

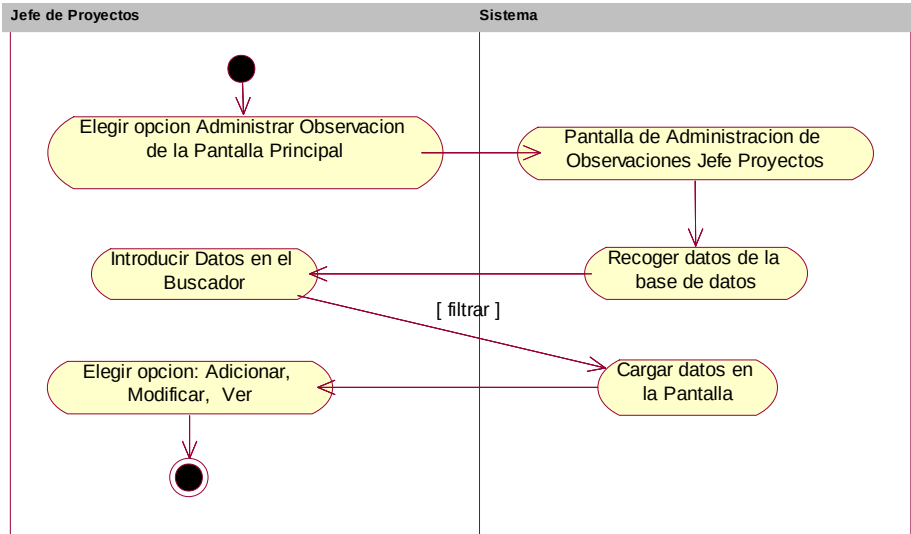


Figura 127: Diagrama de Actividad: Administrar Observaciones Jefe Proyectos

III.1.2.2.56 Adicionar Observaciones Jefe Proyectos

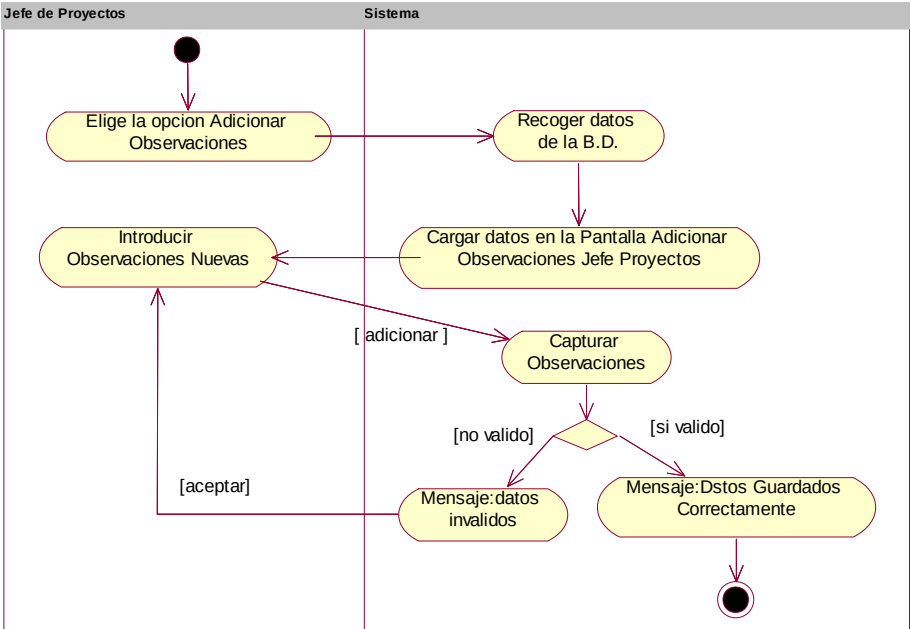


Figura 128: Diagrama de Actividad: Adicionar Observaciones Jefe Proyectos

III.1.2.2.57 Modificar Observaciones Jefe Proyectos

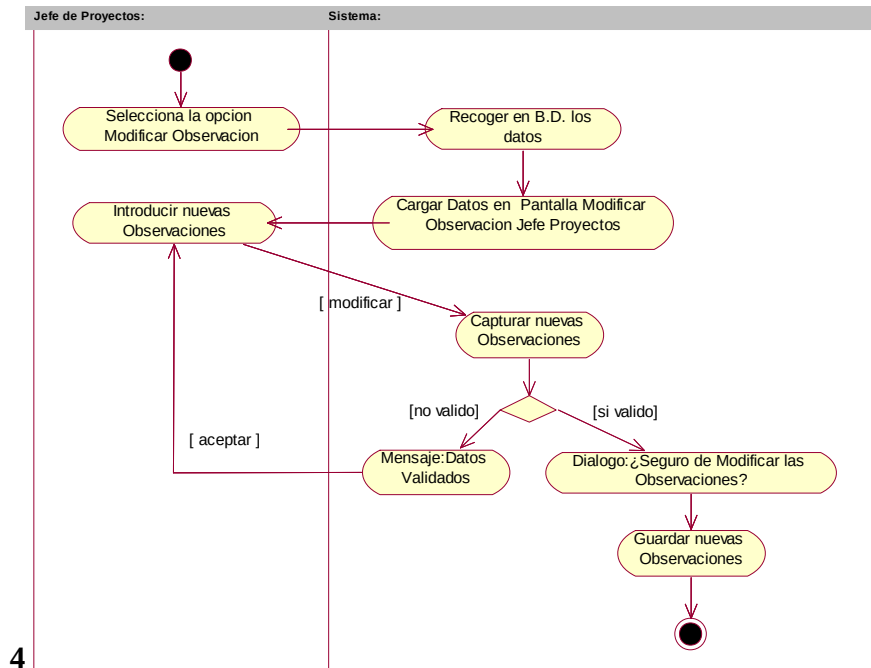


Figura 129: Diagrama de Actividad: Modificar Observaciones Jefe Proyectos

III.1.2.2.59 Ver Observaciones Jefe Proyectos

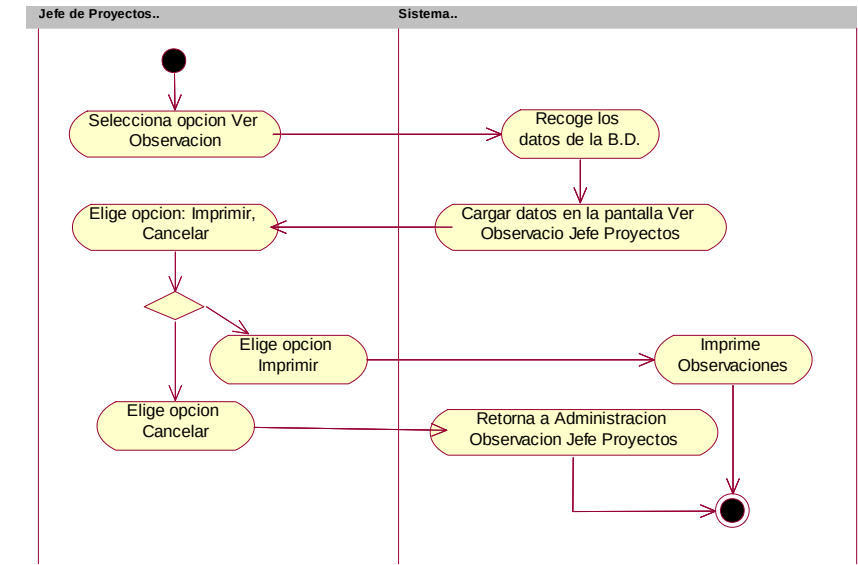


Figura 130: Diagrama de Actividad: Ver Observaciones Jefe Proyectos

III.1.2.2.60 Dar Alta

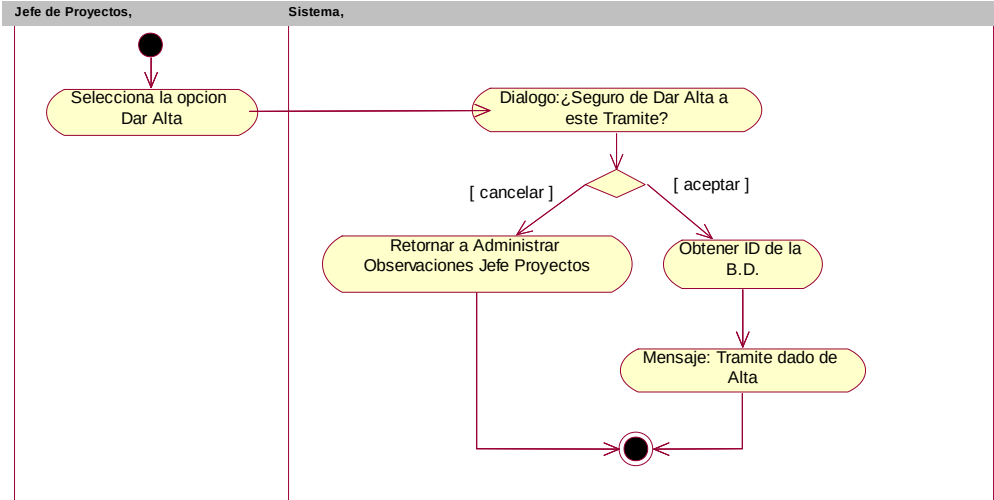


Figura 131: Diagrama de Actividad: Dar Alta

III.1.2.2.61 Aprobar Tramite

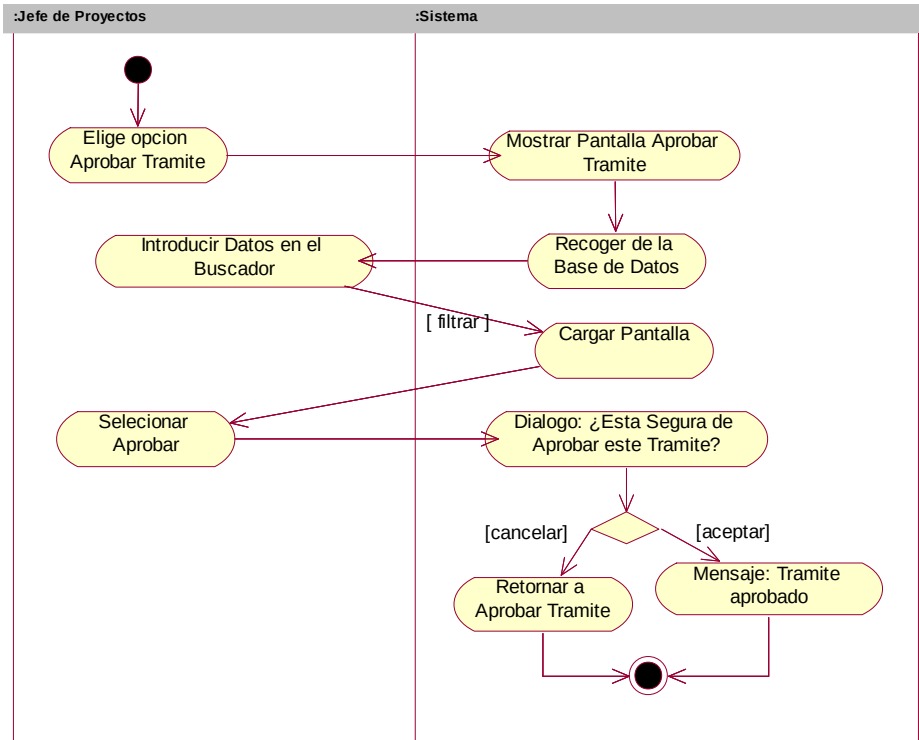


Figura 132: Diagrama de Actividad: Aprobar Tramite

III.1.2.3 MODULO DE LOTES

III.1.2.3.1 Administrar Propietarios

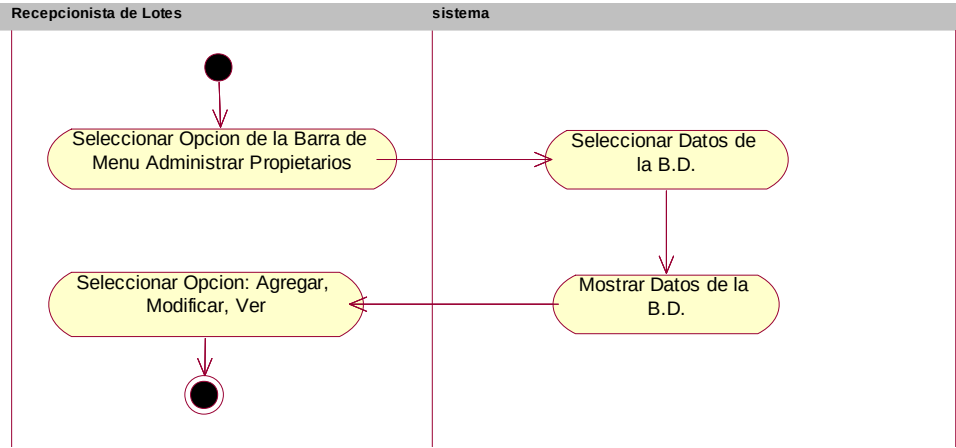


Figura 133: Diagrama de Actividad: Administrar Propietarios

III.1.2.3.2 Adicionar Propietarios

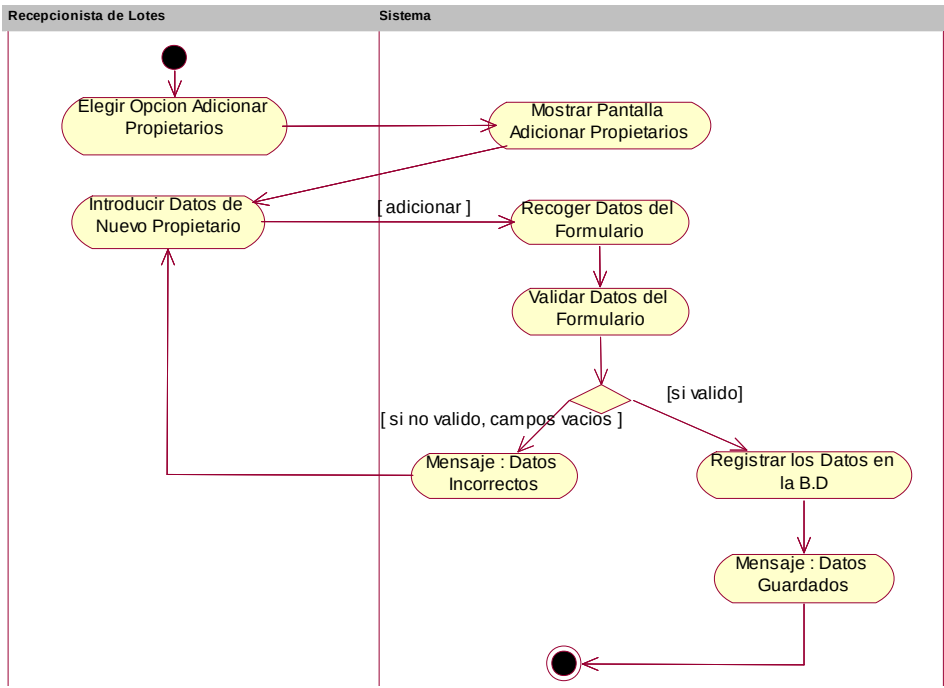


Figura 134: Diagrama de Actividad: Adicionar Propietarios

III.1.2.3.3 Adicionar Proyectista

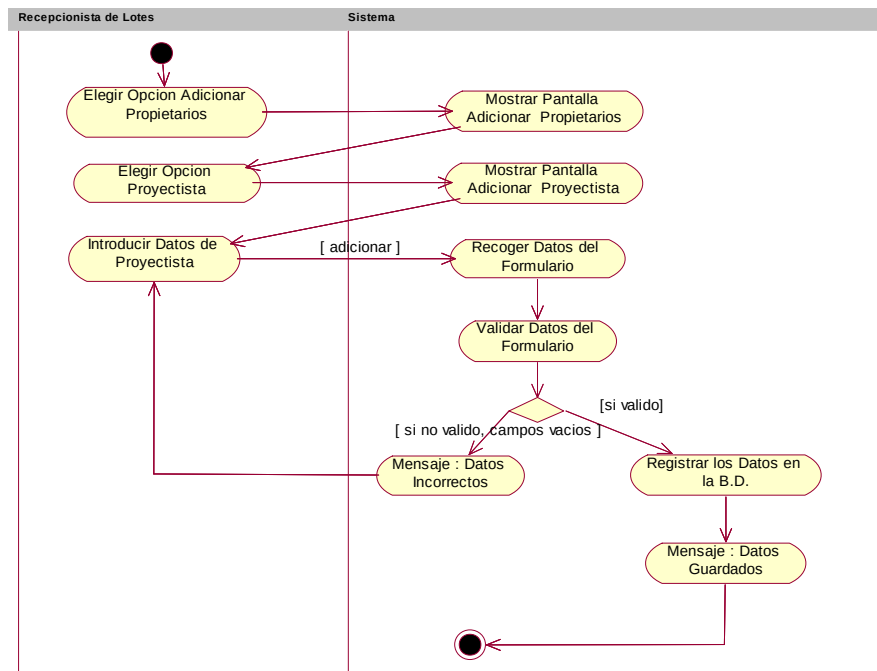


Figura 135: Diagrama de Actividad: Adicionar Proyectista

III.1.2.3.4 Adicionar Apoderado

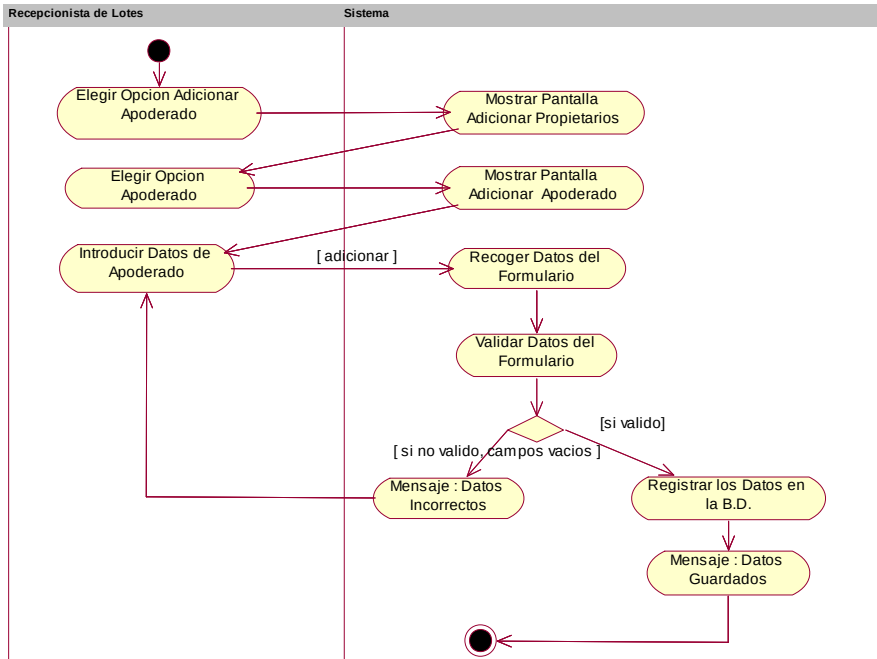


Figura 136: Diagrama de Actividad: Adicionar Apoderado

III.1.2.3.5 Modificar Propietarios

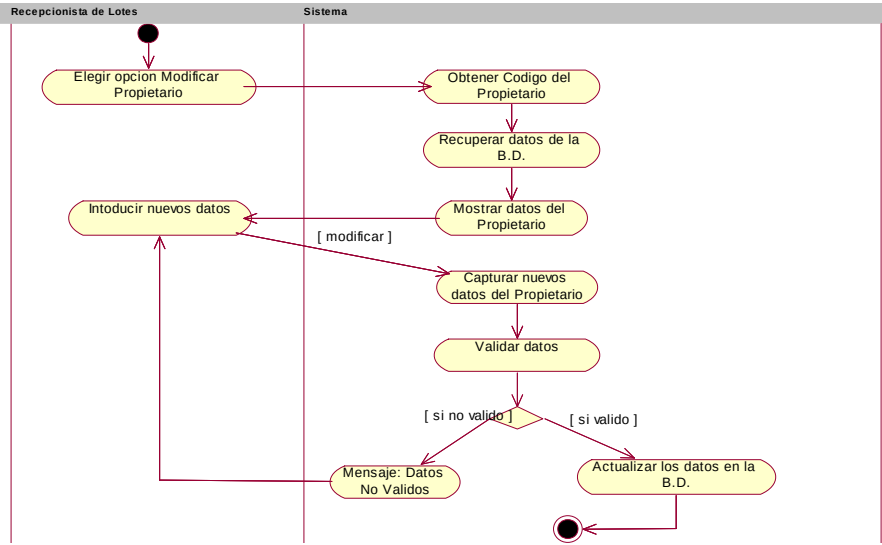


Figura 137: Diagrama de Actividad: Modificar Propietarios

III.1.2.3.6 Modificar Proyectista

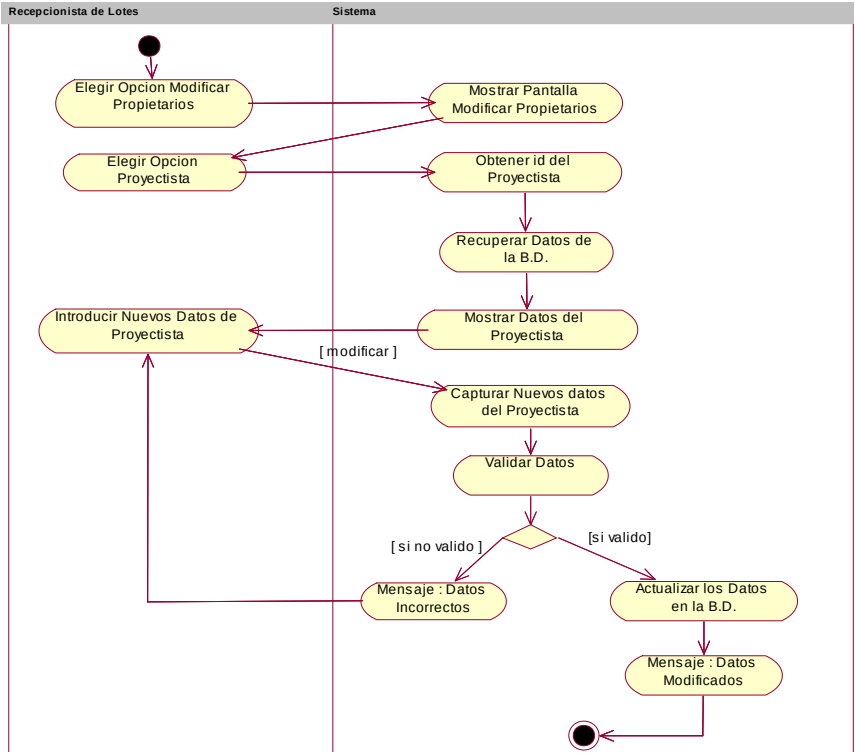


Figura 138: Diagrama de Actividad: Modificar Proyectista

III.1.2.3.7 Modificar Apoderado

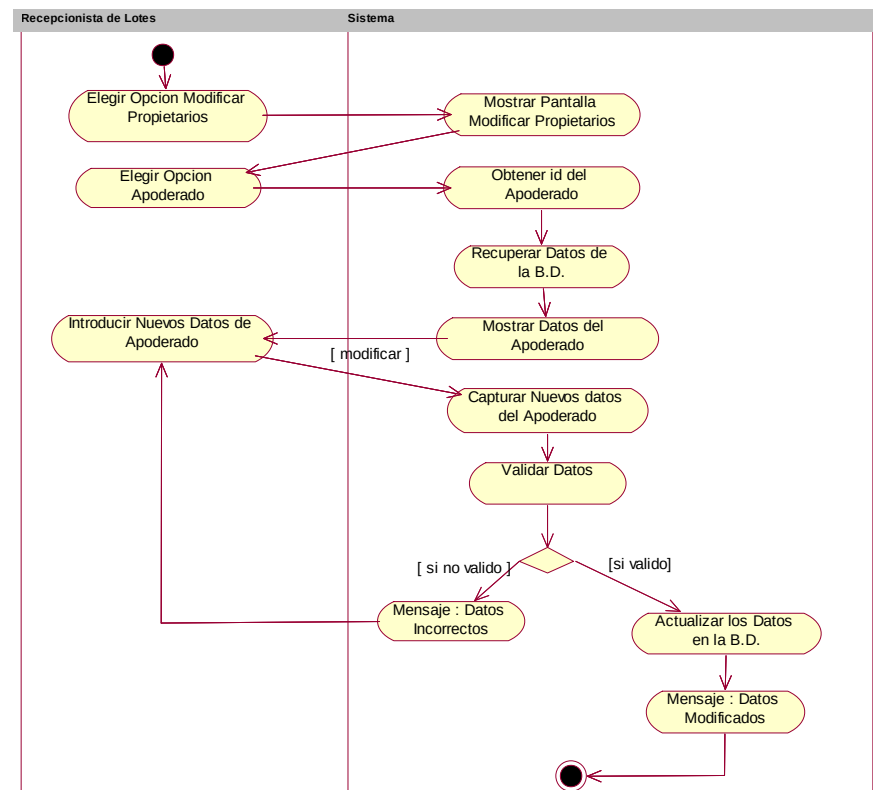


Figura 139: Diagrama de Actividad: Modificar Apoderado

III.1.2.3.8 Ver Propietarios

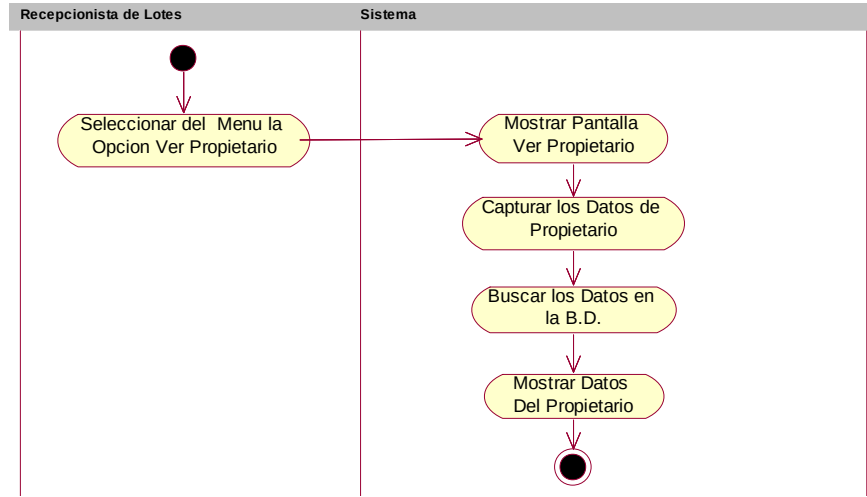


Figura 140: Diagrama de Actividad: Ver Propietarios

III.1.2.3.9 Administrar Observaciones Generales

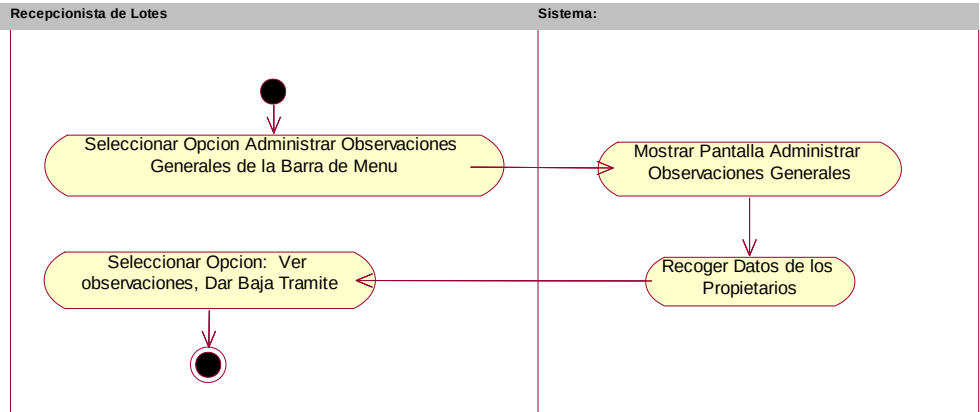


Figura 141: Diagrama de Actividad: Administrar Observaciones

III.1.2.3.10 Ver Observaciones Generales

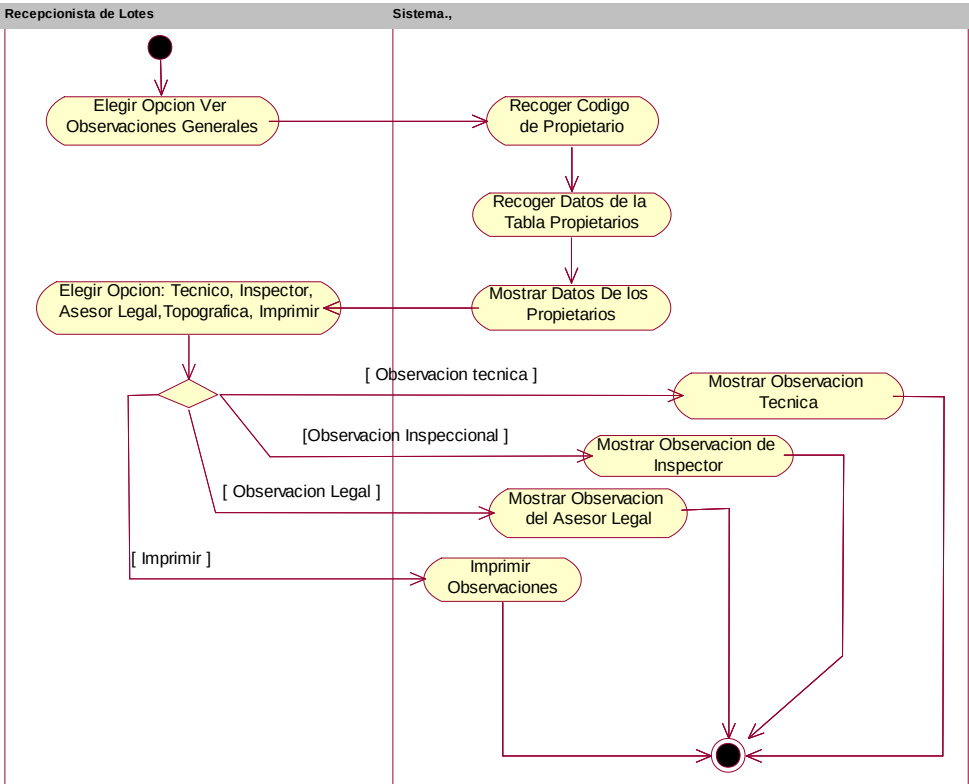


Figura 142: Diagrama de Actividad: Ver Observaciones Generales

III.1.2.3.11 Dar Baja Trámite

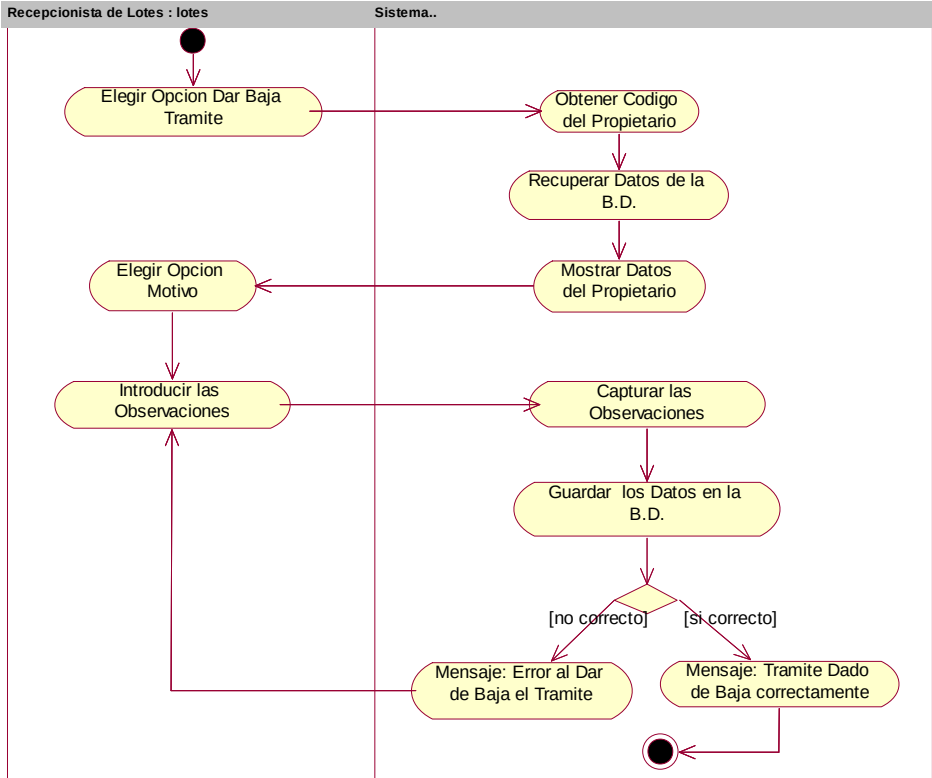


Figura 143: Diagrama de Actividad: Dar Baja Trámite

III.1.2.3.12 Reasignar Tramite

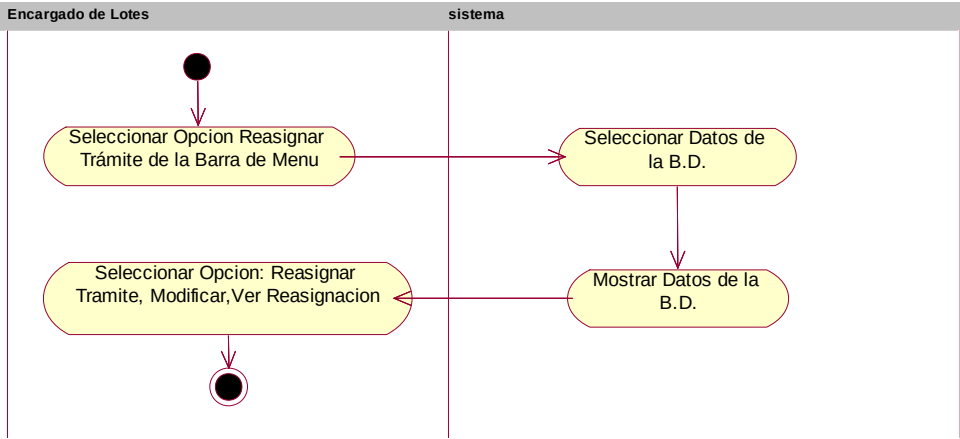


Figura 144: Diagrama de Actividad: Reasignar Tramite

III.1.2.3.13 Registrar Reasignación

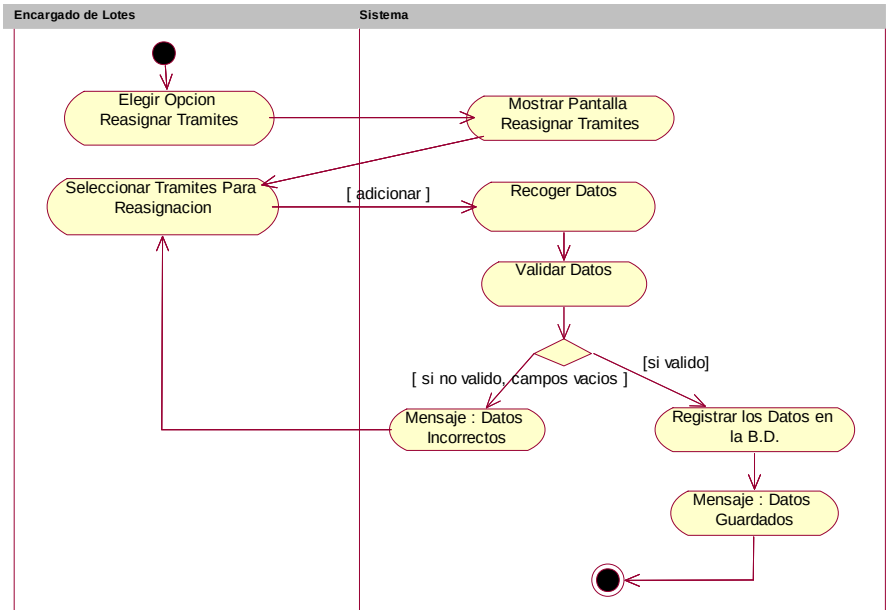


Figura 145: Diagrama de Actividad: Registrar Reasignación

III.1.2.3.14 Ver Reasignación

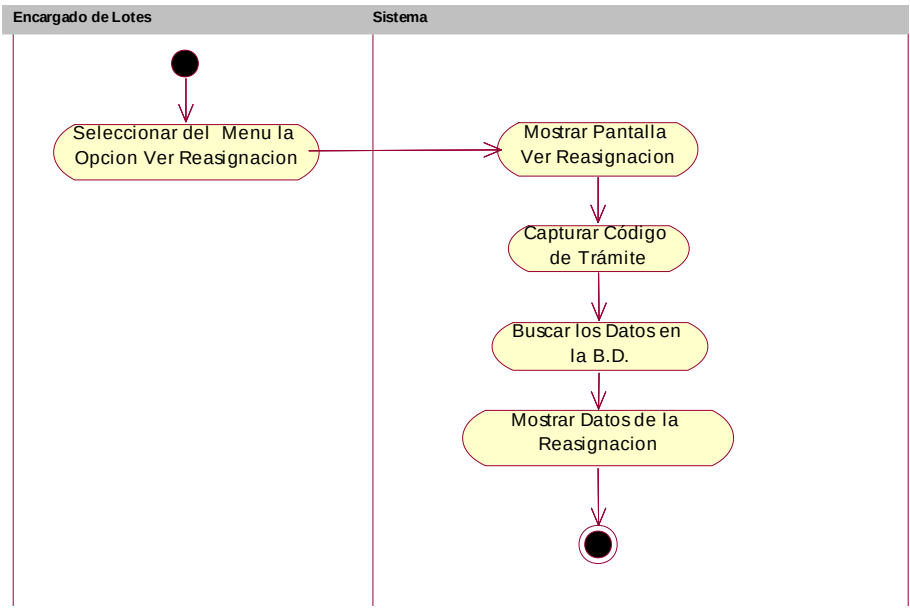


Figura 146: Diagrama de Actividad: Ver Reasignación

III.1.2.3.15 Ver Datos Trámite

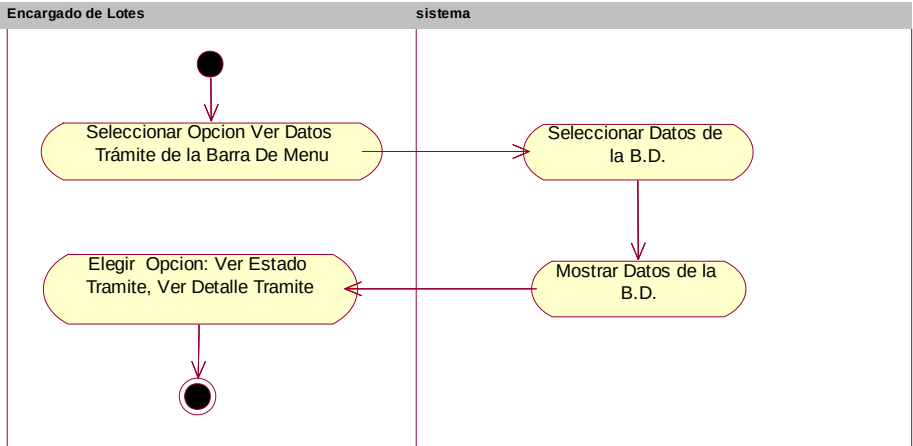


Figura 147: Diagrama de Actividad: Administrar Observaciones Generales

III.1.2.3.16 Ver Estado Trámite

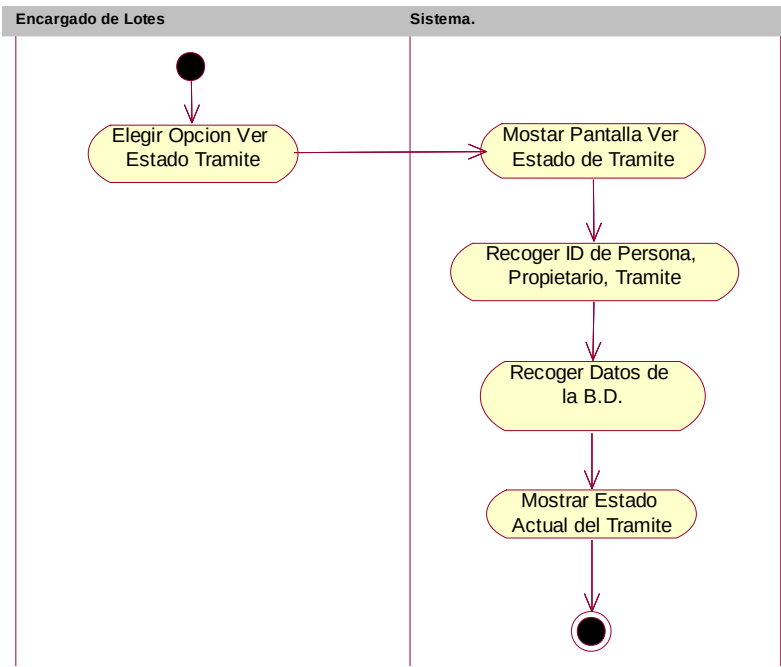


Figura 147: Diagrama de Actividad: Ver Estado Trámite

III.1.2.3.17 Ver Detalle Observación

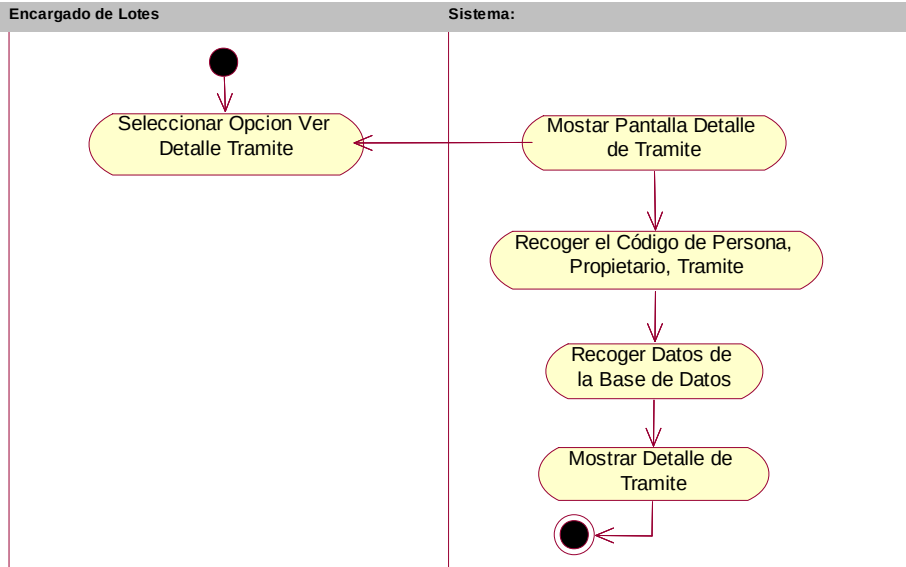


Figura 148: Diagrama de Actividad: Ver Detalle Observación

III.1.2.3.18 Administrar Ingresos

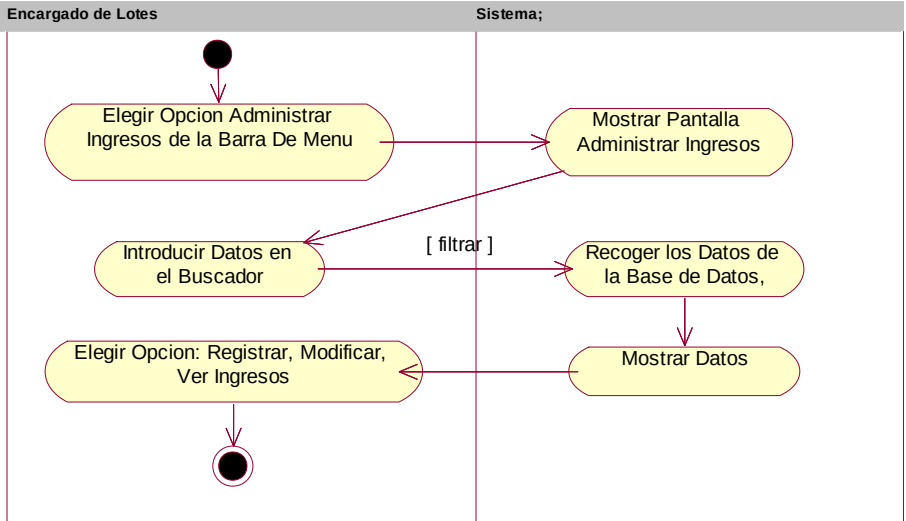


Figura 149: Diagrama de Actividad: Administrar Ingresos

III.1.2.3.19 Registrar Ingresos

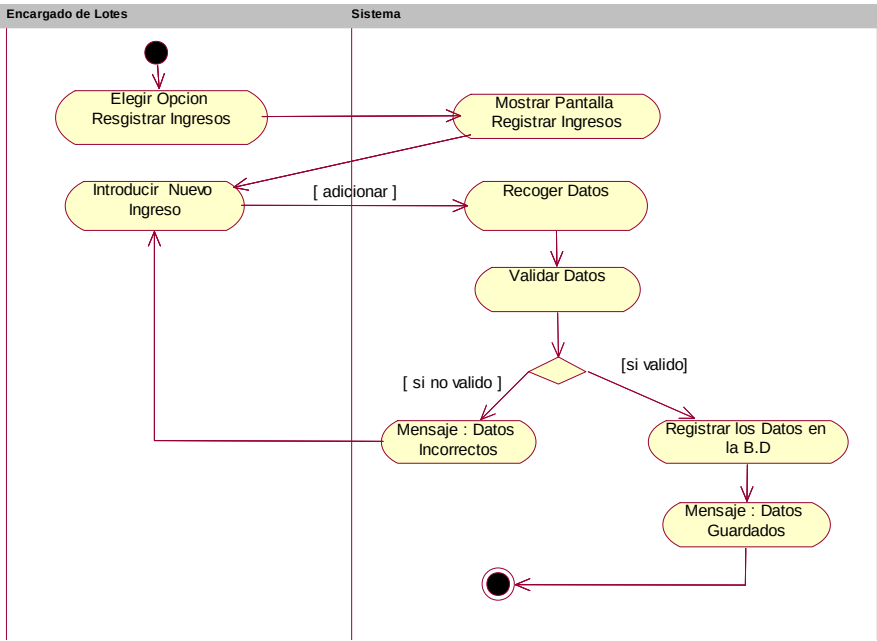


Figura 150: Diagrama de Actividad: Registrar Ingresos

III.1.2.3.20 Modificar Ingresos

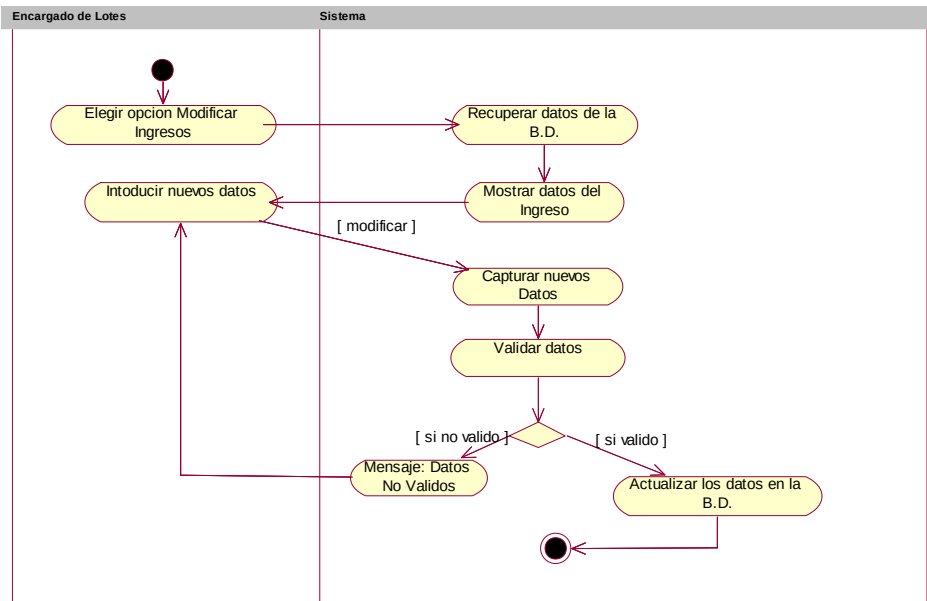


Figura 151: Diagrama de Actividad: Modificar Ingresos

III.1.2.3.21 Ver Ingresos

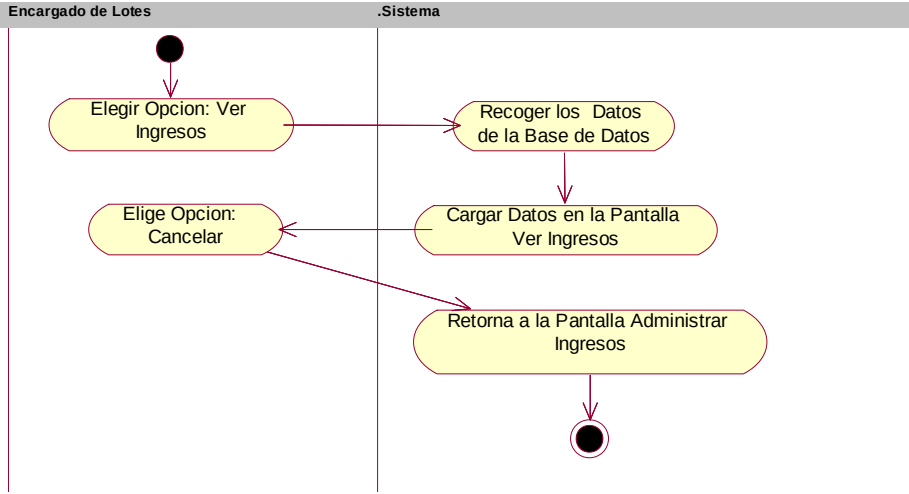


Figura 152: Diagrama de Actividad: Ver Ingresos

III.1.2.3.22 Administrar Salidas

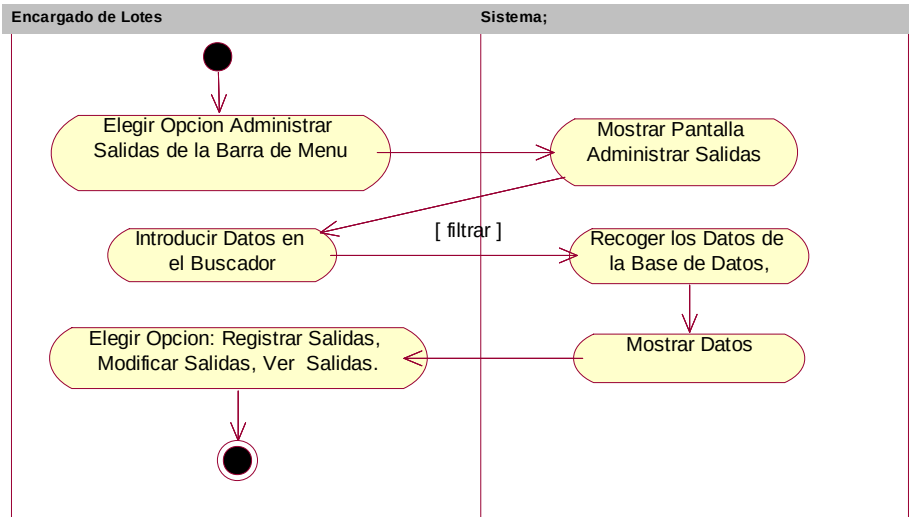


Figura 153: Diagrama de Actividad: Administrar Salidas

III.1.2.3.23 Registrar Salidas

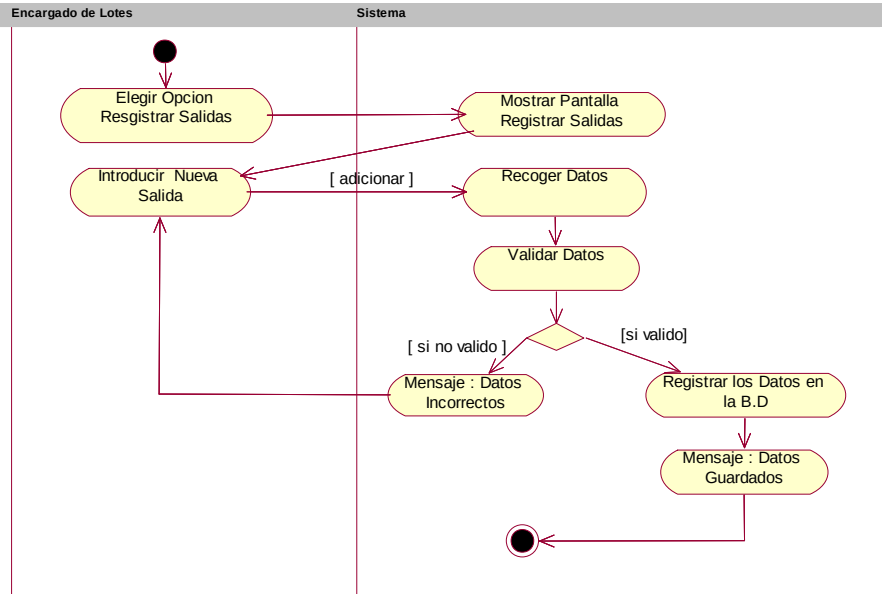


Figura 154: Diagrama de Actividad: Registrar Salidas

III.1.2.3.24 Modificar Salidas

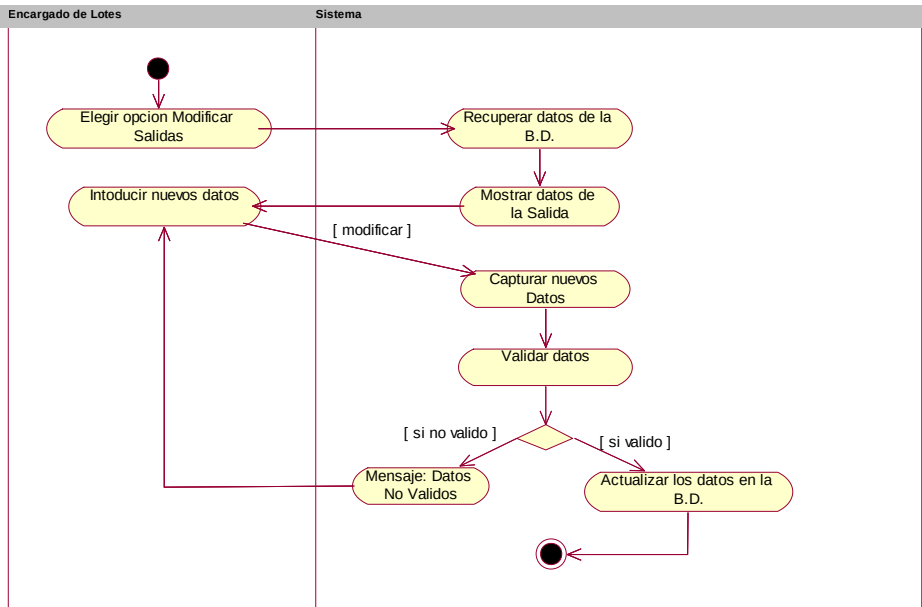


Figura 155: Diagrama de Actividad: Modificar Salidas

III.1.2.3.25 Ver Salidas

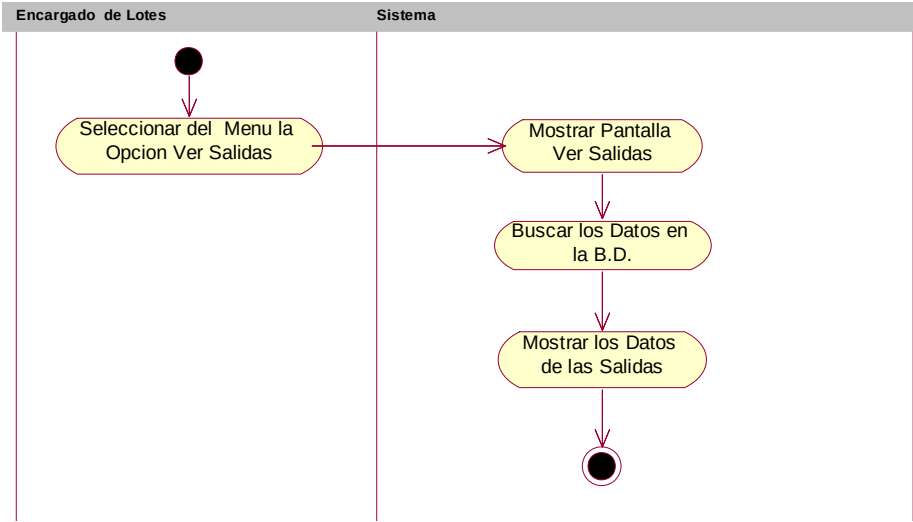


Figura 156: Diagrama de Actividad: Ver Salidas

III.1.2.3.26 Administrar Envíos

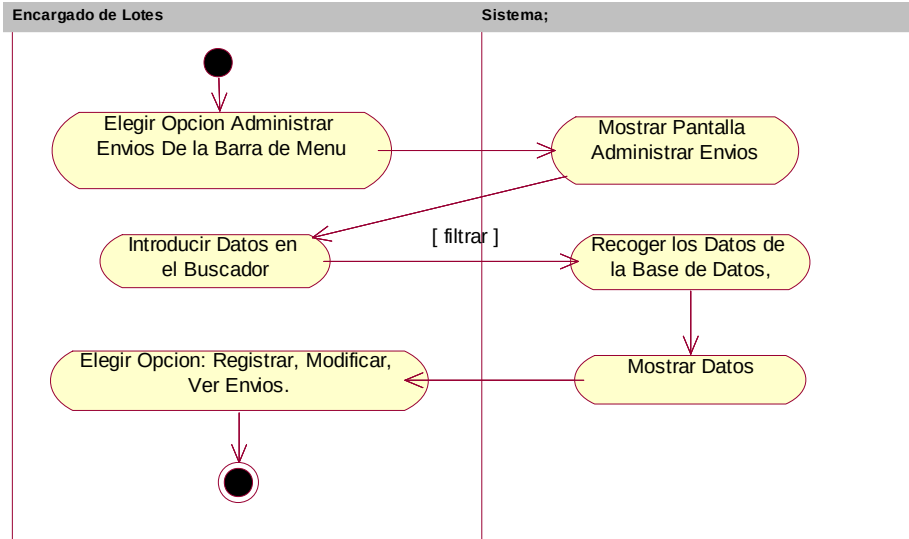


Figura 157: Diagrama de Actividad: Administrar Envíos

III.1.2.3.27 Registrar Envíos

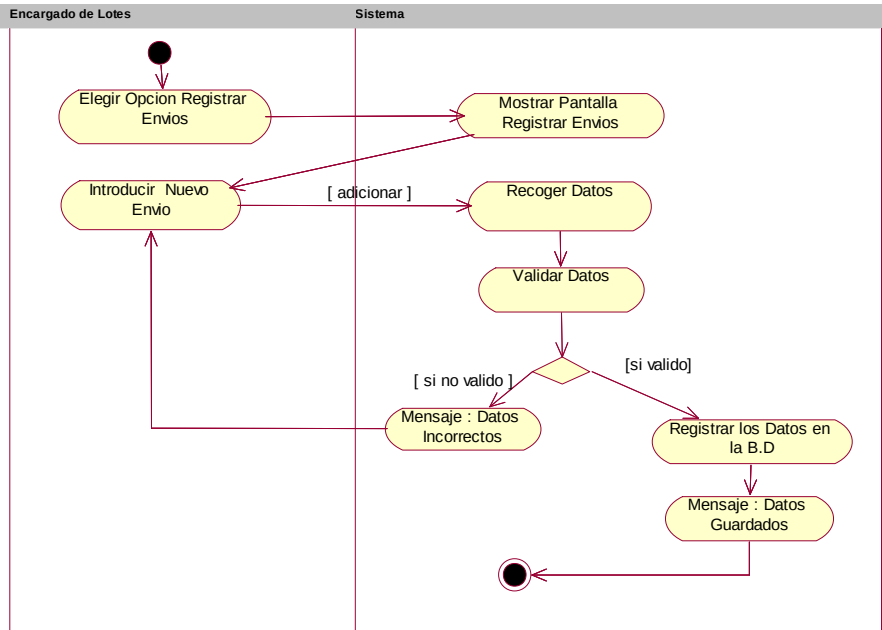


Figura 158: Diagrama de Actividad: Registrar Envíos

III.1.2.3.28 Modificar Envíos

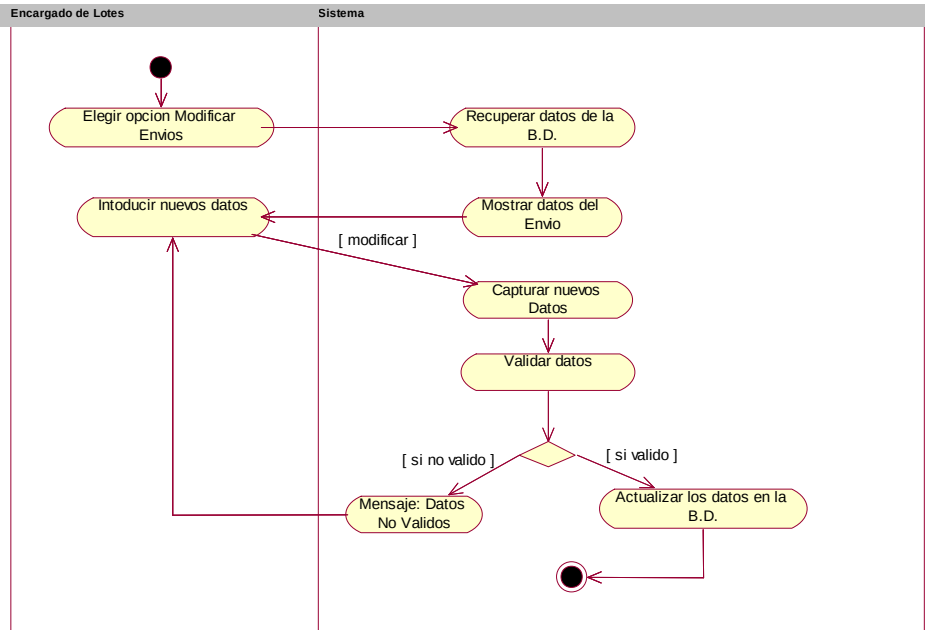


Figura 159: Diagrama de Actividad: Modificar Envíos

III.1.2.3.29 Ver Envíos

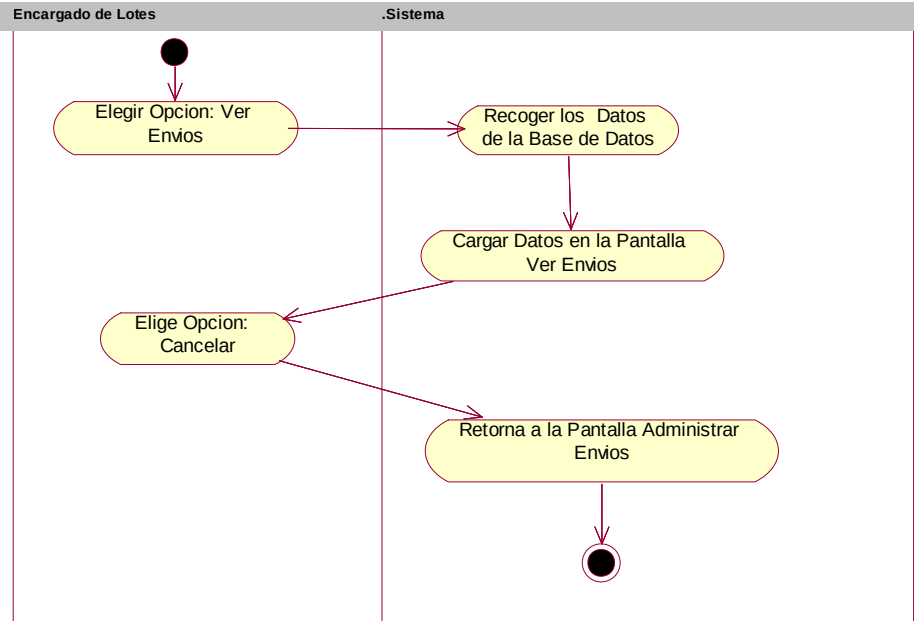


Figura 160: Diagrama de Actividad: Ver Envíos

III.1.2.3.30 Administrar Recepción

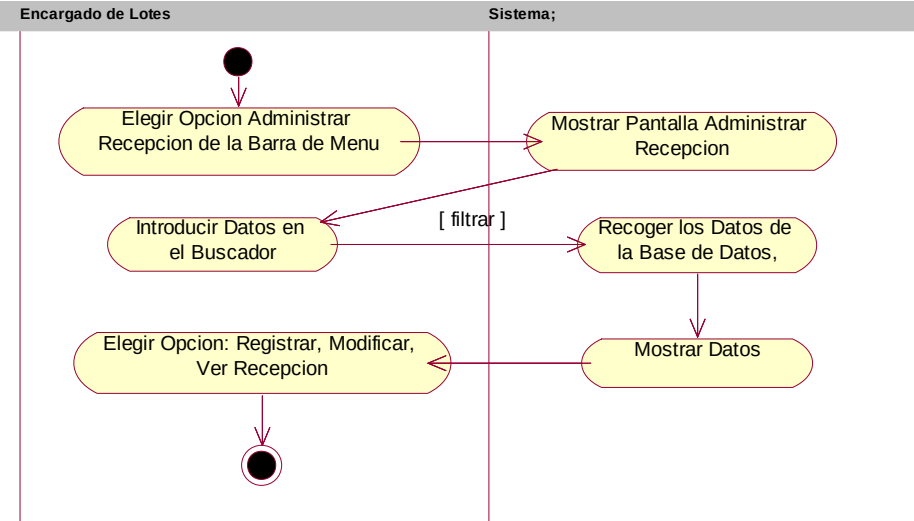


Figura 161: Diagrama de Actividad: Administrar Recepción

III.1.2.3.31 Registrar Recepción

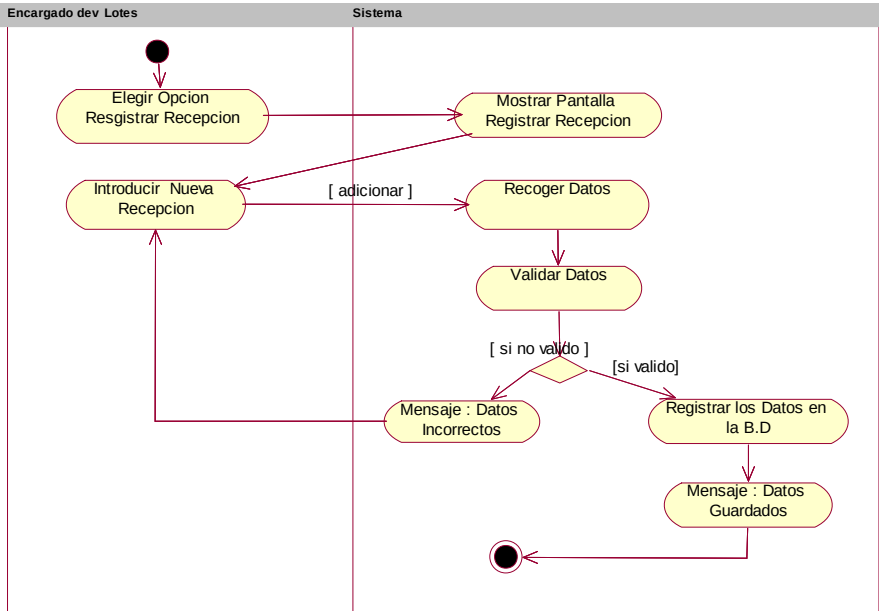


Figura 162: Diagrama de Actividad: Registrar Recepción

III.1.2.3.32 Modificar Recepción

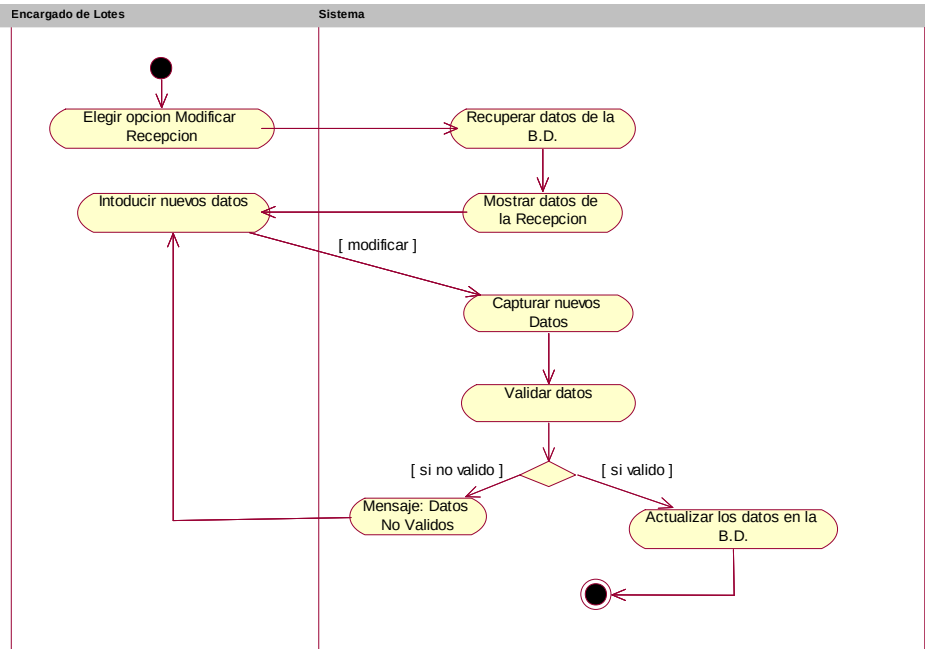


Figura 163: Diagrama de Actividad: Modificar Recepción

III.1.2.3.33 Ver Recepción

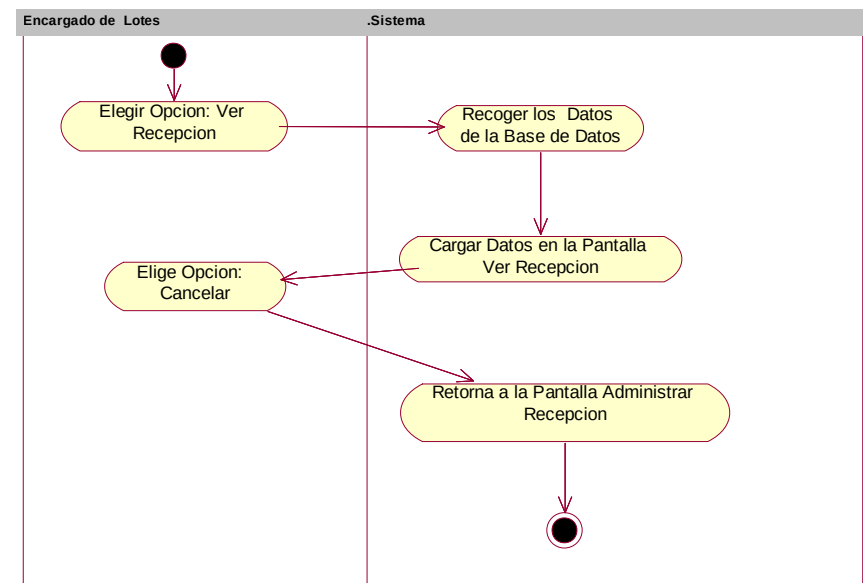


Figura 164: Diagrama de Actividad: Ver Recepción

III.1.2.3.40 Administrar Observaciones

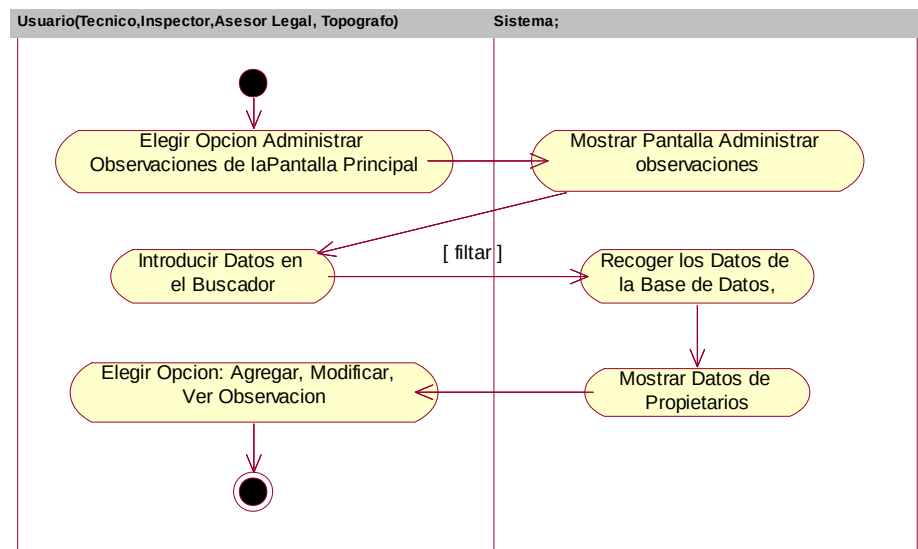


Figura 171: Diagrama de Actividad: Administrar Observaciones

III.1.2.3.41 Adicionar Observaciones

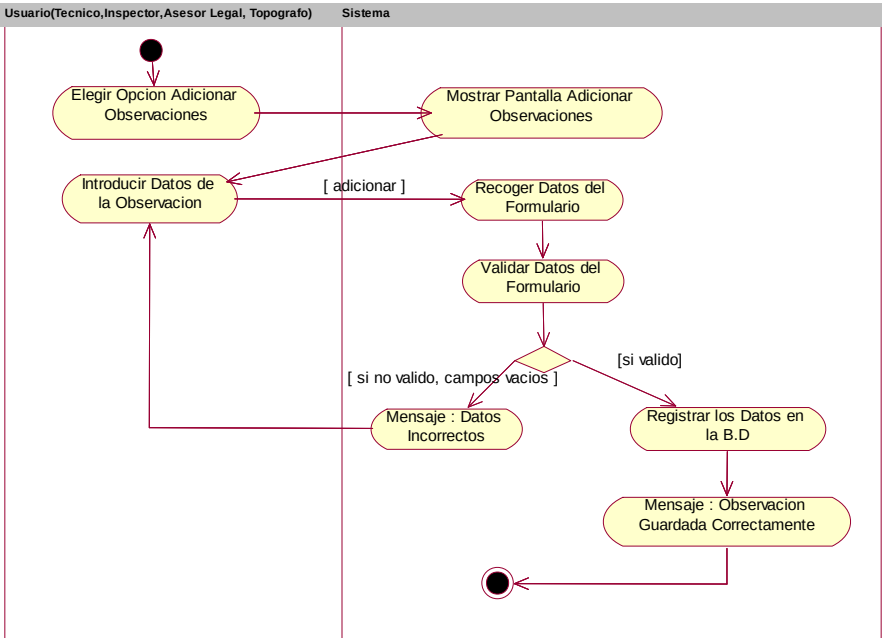


Figura 172: Diagrama de Actividad: Adicionar Observaciones

III.1.2.3.42 Modificar Observaciones

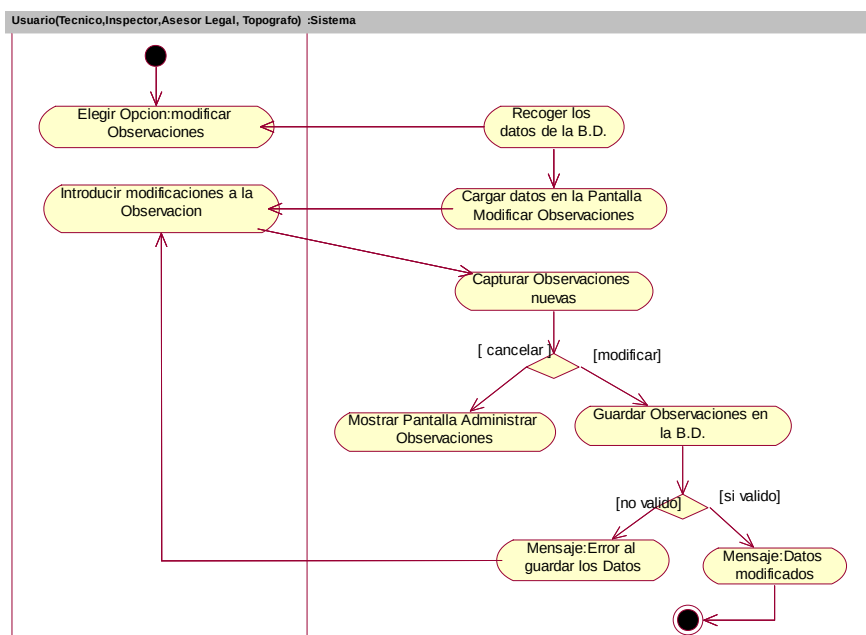


Figura 173: Diagrama de Actividad: Modificar Observaciones

III.1.2.3.44 Ver Observaciones

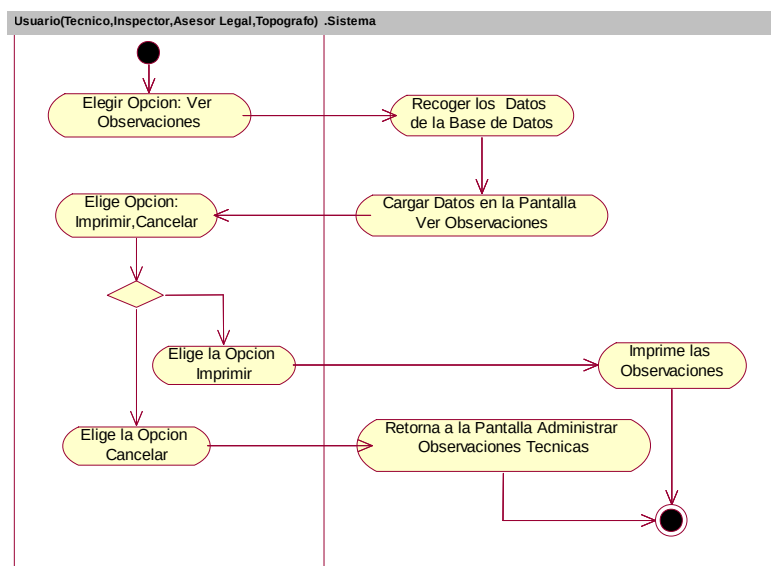


Figura 174: Diagrama de Actividad: Ver Observaciones

III.1.2.3.60 Administrar Observaciones Jefe Lotes

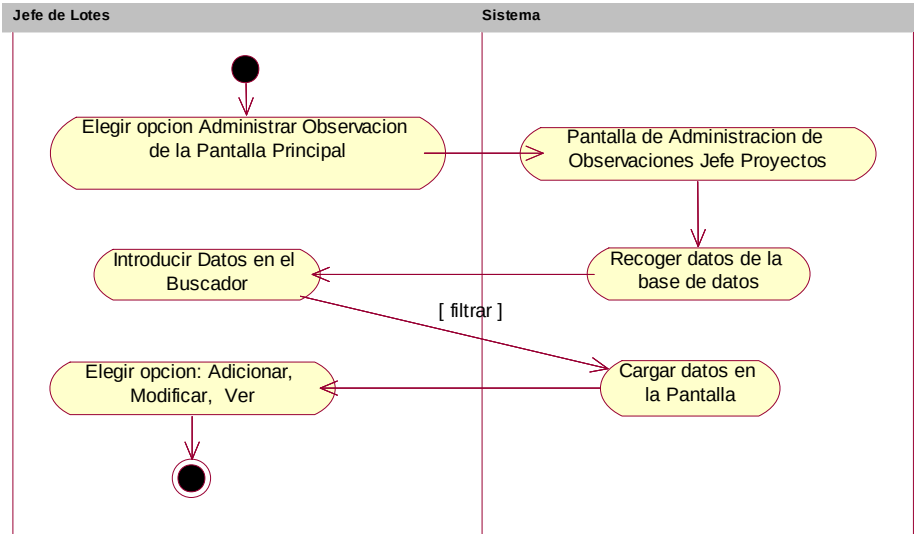


Figura 175: Diagrama de Actividad: Administrar Observaciones Jefe Lotes

III.1.2.3.61 Adicionar Observaciones Jefe Lotes

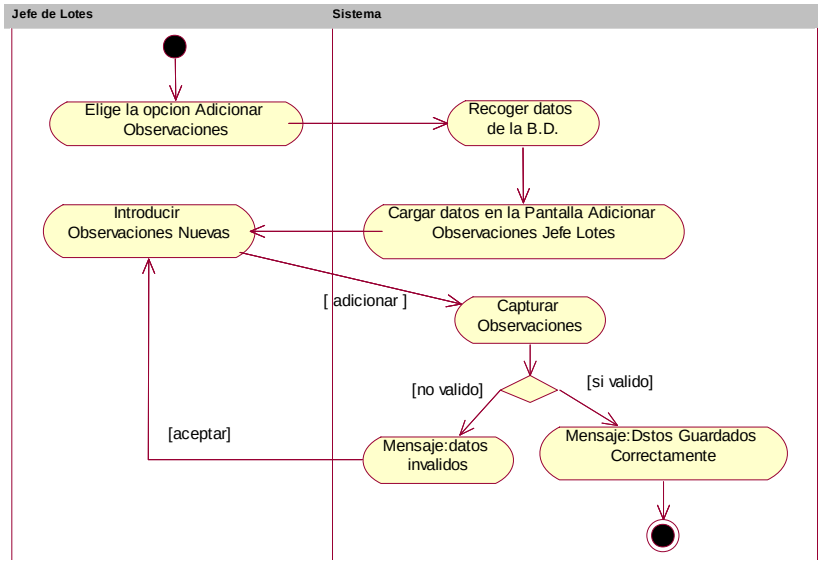


Figura 176: Diagrama de Actividad: Adicionar Observaciones Jefe Proyectos

III.1.2.3.62 Modificar Observaciones Jefe Lotes

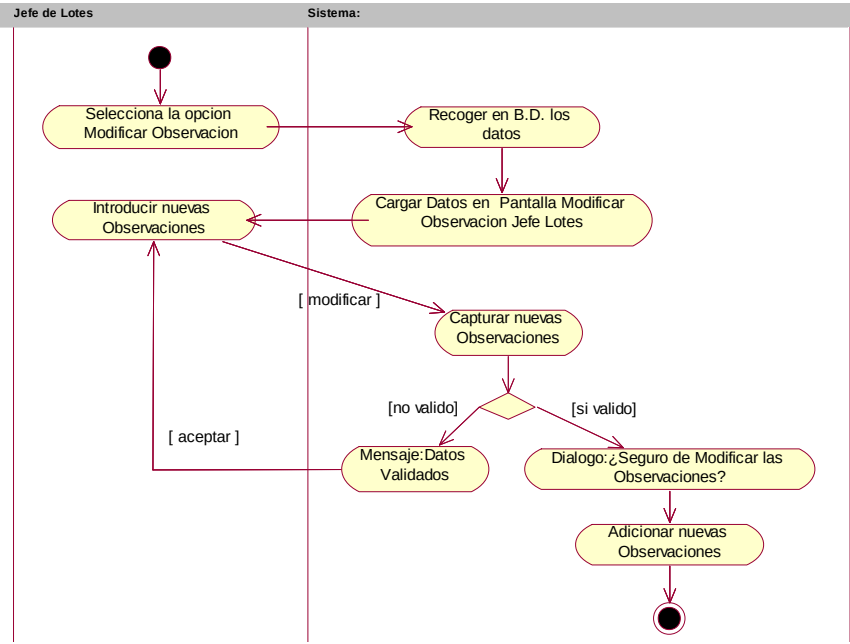


Figura 177: Diagrama de Actividad: Modificar Observaciones Jefe Proyectos

III.1.2.3.64 Ver Observaciones Jefe Lotes

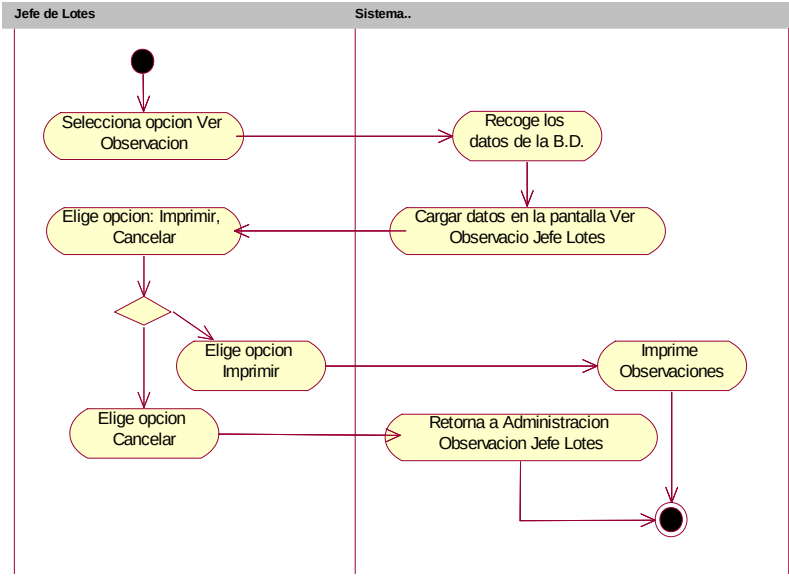


Figura 178: Diagrama de Actividad: Ver Observaciones Jefe Proyectos

III.1.2.3.65 Dar Alta

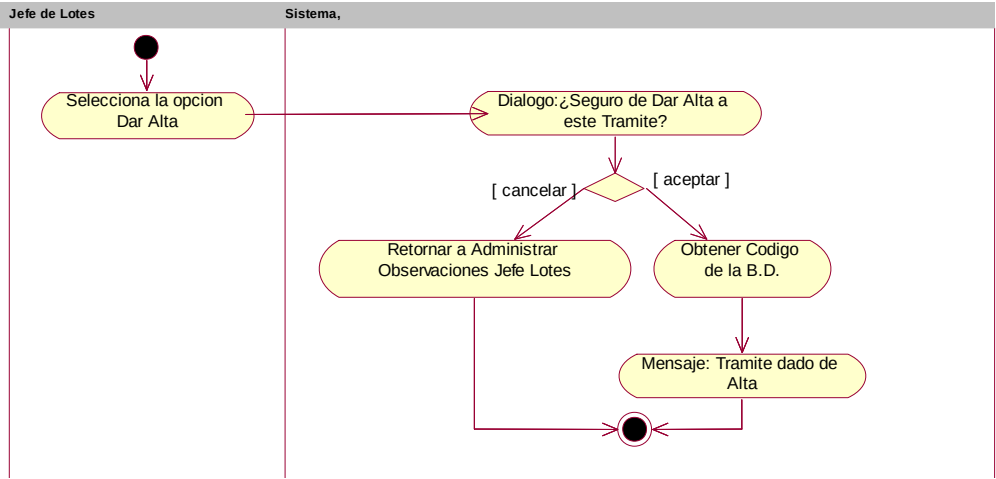


Figura 179: Diagrama de Actividad: Dar Alta

III.1.2.3.66 Aprobar Tramite

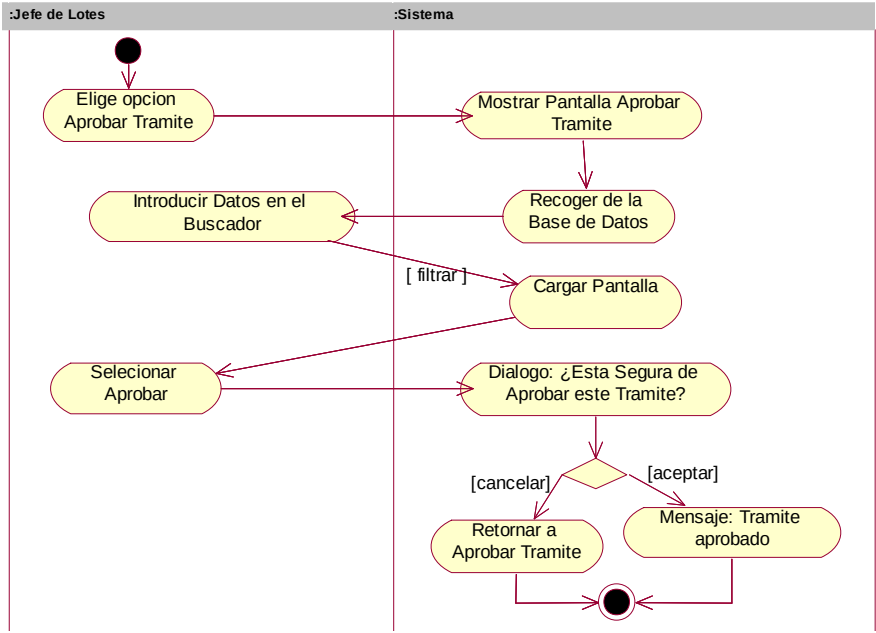


Figura 180: Diagrama de Actividad: Aprobar Tramite

III.1.2.4 MODULO ADMINISTRACION

III.1.2.4.1 Administrar Personal

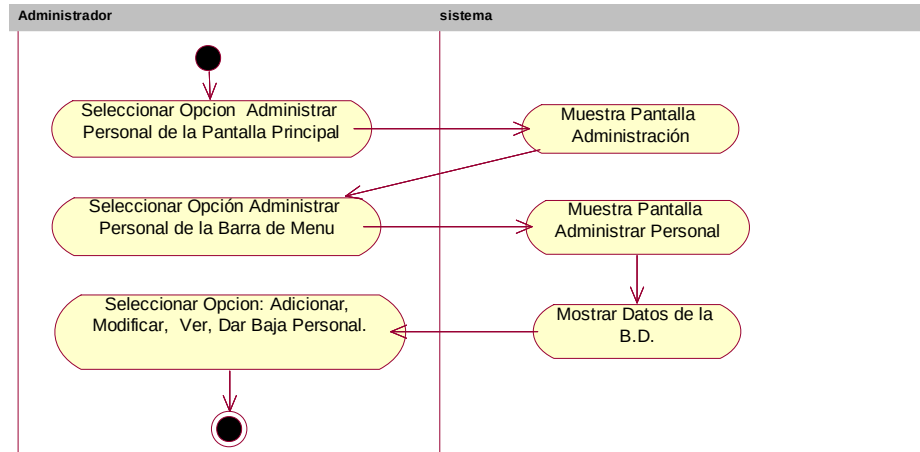


Figura 181: Diagrama de Actividad: Administrar Personal

III.1.2.4.2 Adicionar Personal

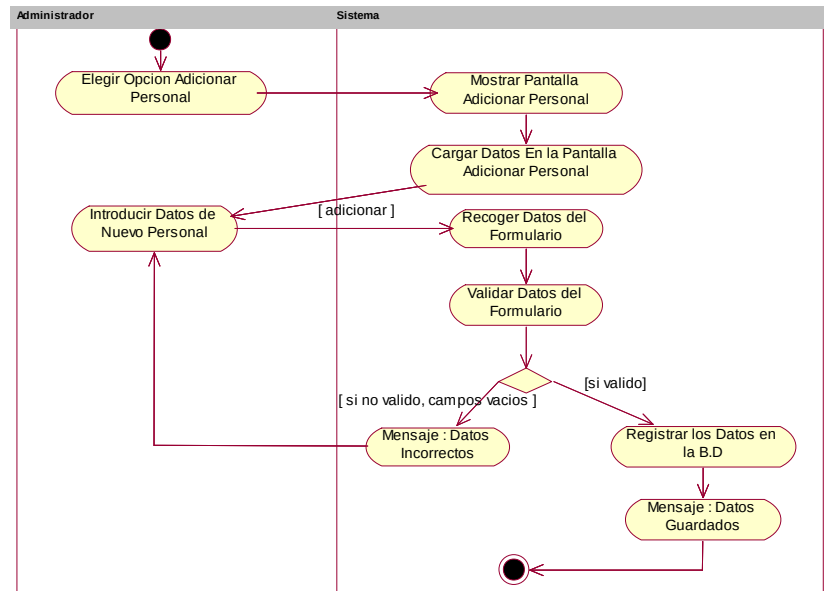


Figura 182: Diagrama de Actividad: Agregar Personal

III.1.2.4.3 Modificar Personal

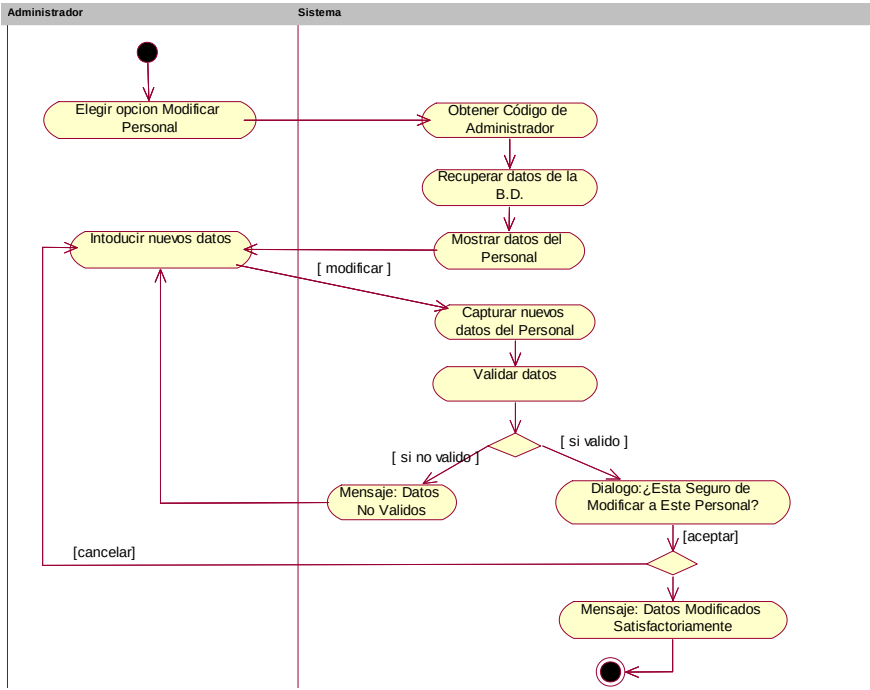


Figura 183: Diagrama de Actividad: Modificar Personal

III.1.2.4.5 Ver Personal

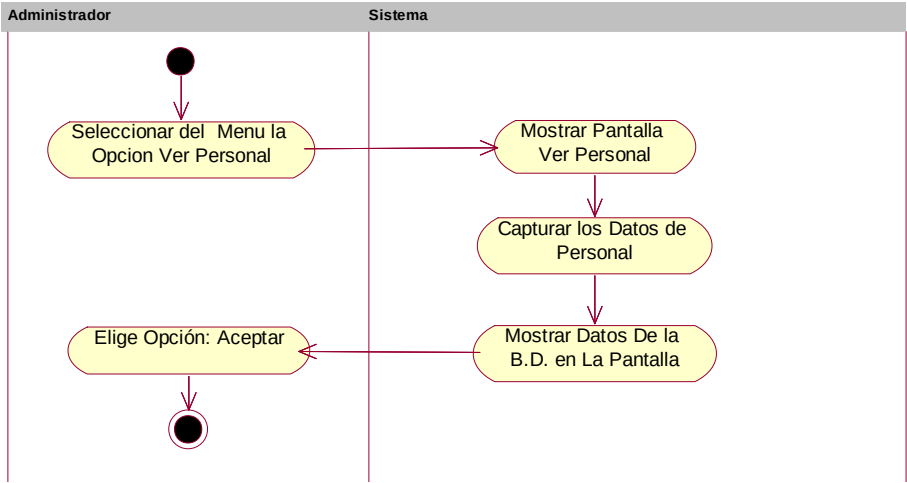


Figura 184: Diagrama de Actividad: Ver Personal

III.1.2.4.6 Administrar Roles

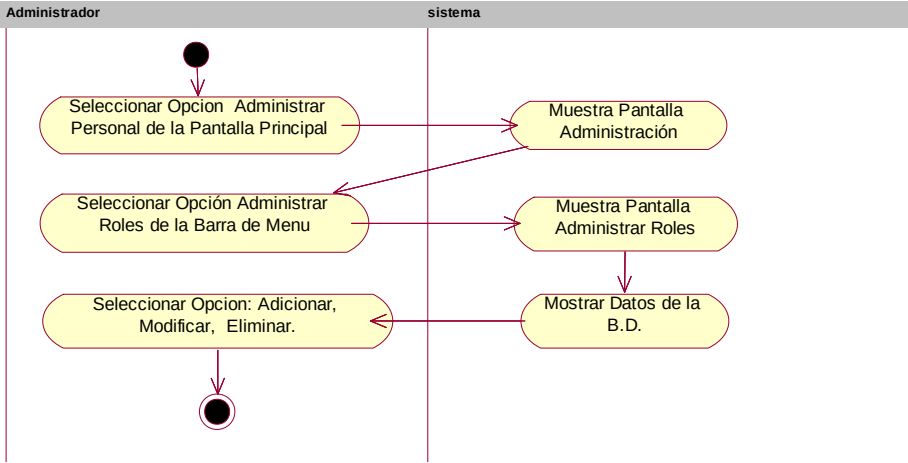


Figura 185: Diagrama de Actividad: Administrar Roles

III.1.2.4.7 Agregar Roles

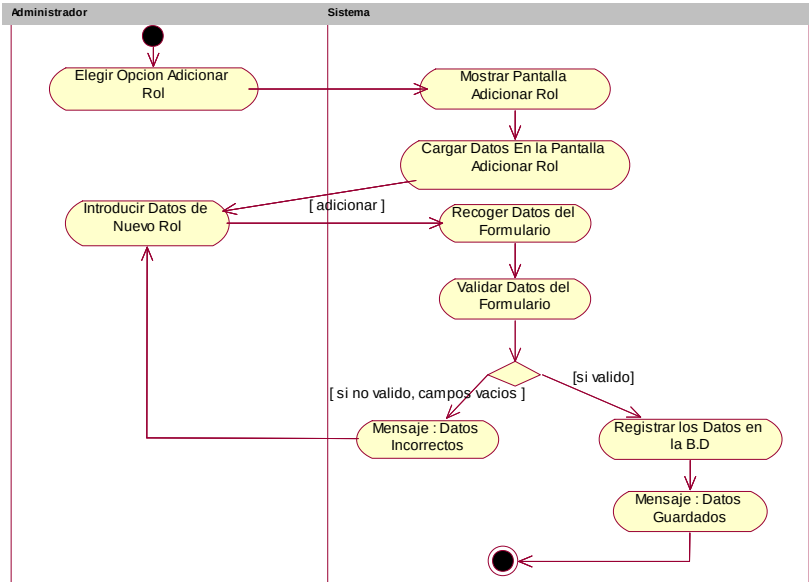


Figura 186: Diagrama de Actividad: Agregar Roles

III.1.2.4.8 Modificar Roles

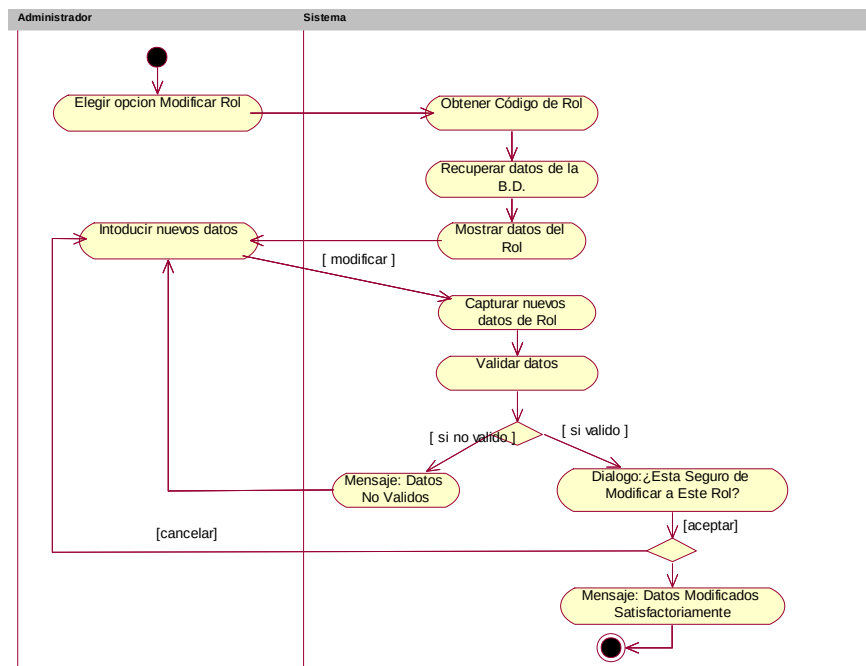


Figura 187: Diagrama de Actividad: Modificar Roles

III.1.2.4.9 Eliminar Roles

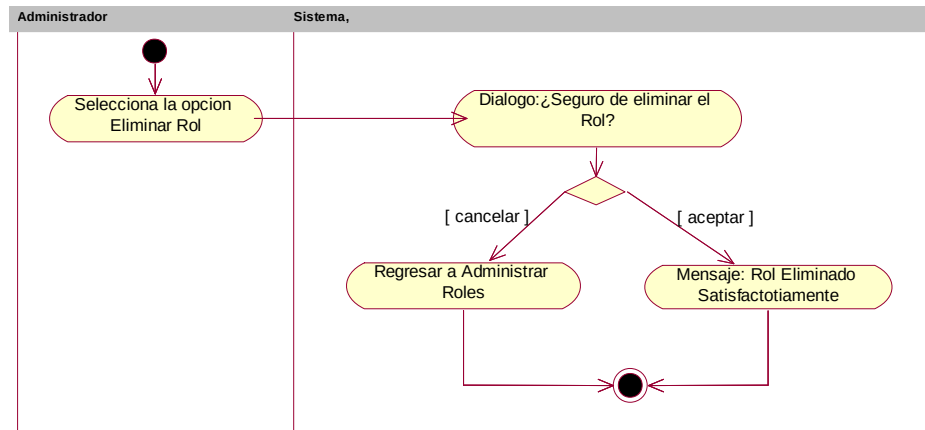


Figura 188: Diagrama de Actividad: Eliminar Roles

III.1.2.4.10 Cambiar Contraseña

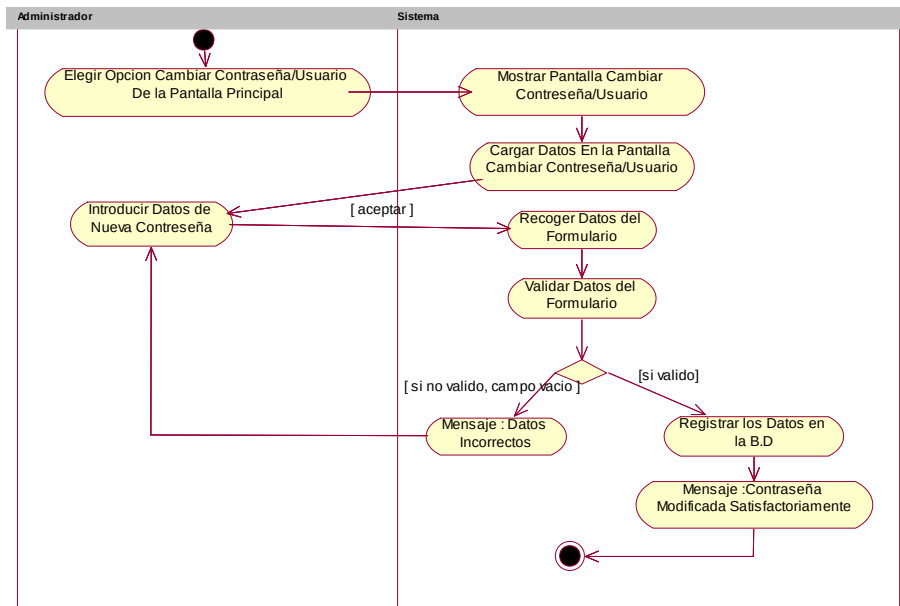


Figura 189: Diagrama de Actividad: Cambiar Contraseña

III.1.2.5 MODULO DE REPORTES

III.1.2.5.1 Generar Reportes de Tramite

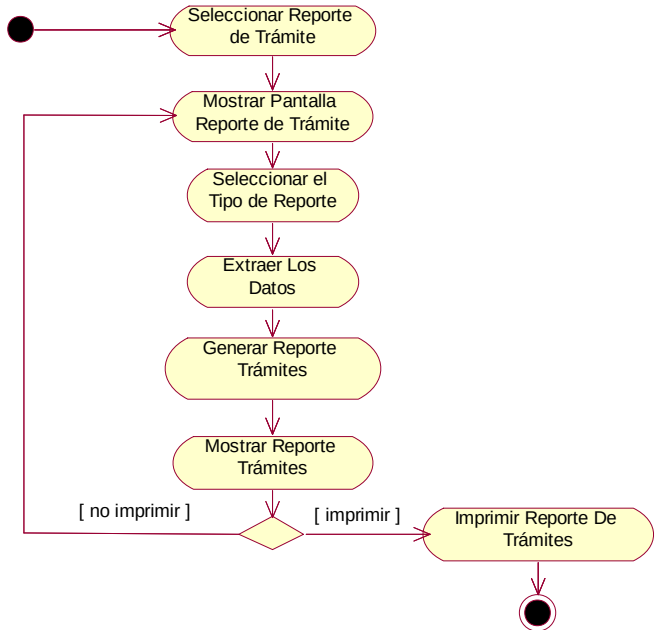


Figura 190: Diagrama de Actividad: Generar Reportes de Tramite

III.1.2.5.2 Generar Reporte Estado Tramite Baja

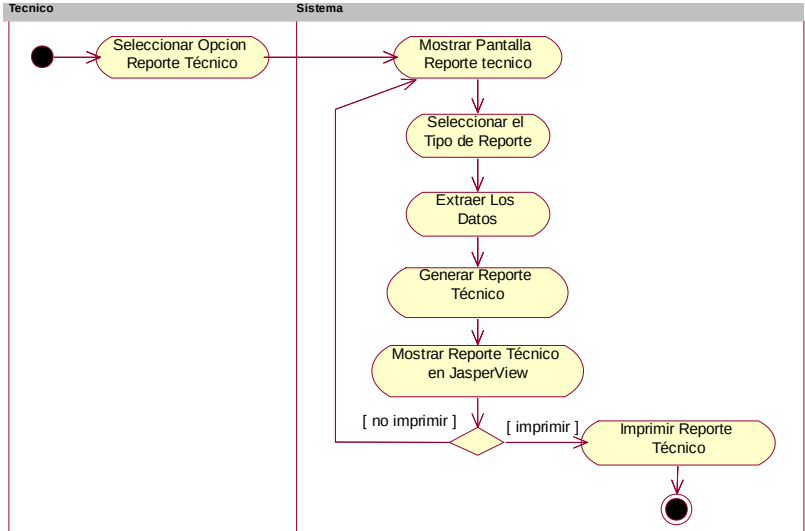


Figura 191: Diagrama de Actividad: Generar Reporte Técnico

III.1.2.5.3 Generar Reporte Estado Tramite Alta

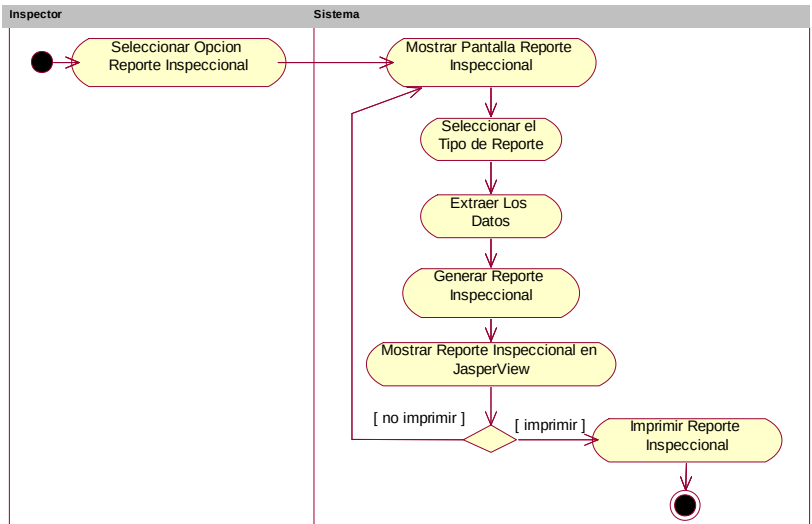


Figura192: Diagrama de Actividad: Generar Reporte de Inspección

III.2 MODELADO DE DIAGRAMA DE INTERACCIÓN

III.2.1 Introducción

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

III.2.1.1. Propósito

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

III.2.1.2. Alcance

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

III.2.2 DIAGRAMAS DE SECUENCIA

III.2.2.1 Diagrama de Secuencia Ingresar Sistema

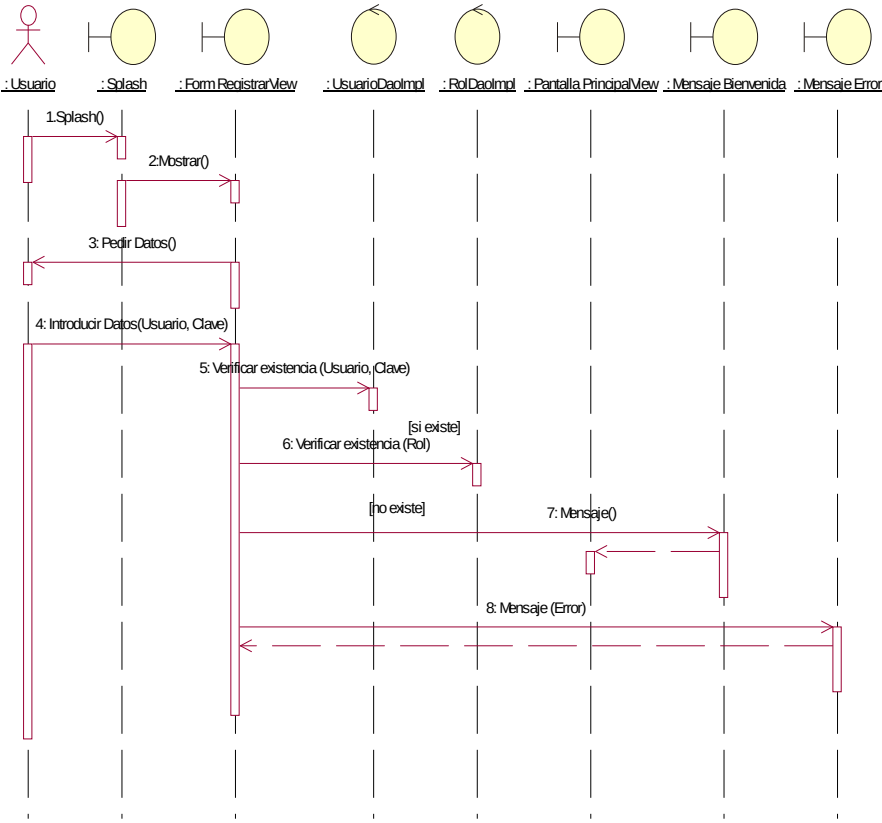


Figura 195: Diagrama de Secuencia: Ingresar Sistema

III.2.2.2 Diagrama de Secuencia Resguardar Base de Datos

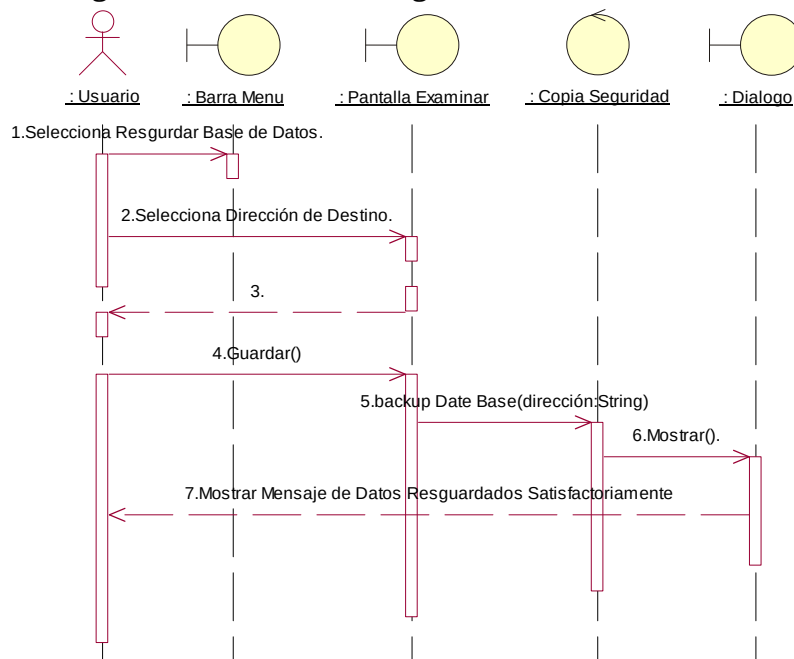


Figura 196: Diagrama de Secuencia: Resguardar Base de Datos

III.2.2.3 Diagrama de Secuencia Restaurar Base de Datos

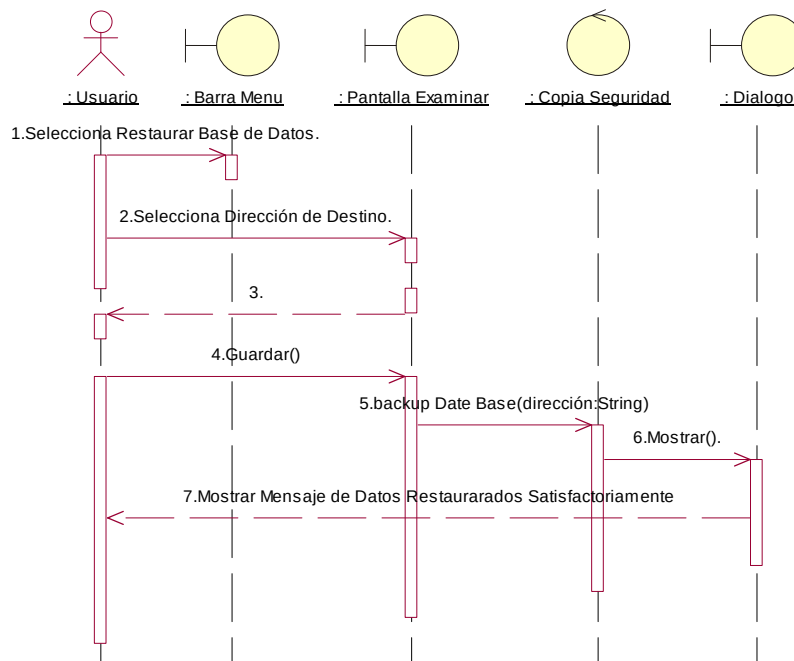


Figura 197: Diagrama de Secuencia: Restaurar Base de Datos

III.2.2.4 Diagrama de Secuencia Cerrar Conexión

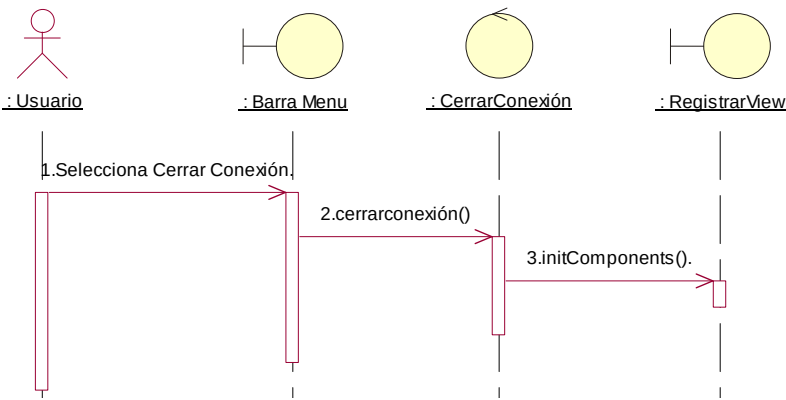


Figura 198: Diagrama de Secuencia: Cerrar Conexión

III.2.2.4 Diagrama de Secuencia Salir Sistema

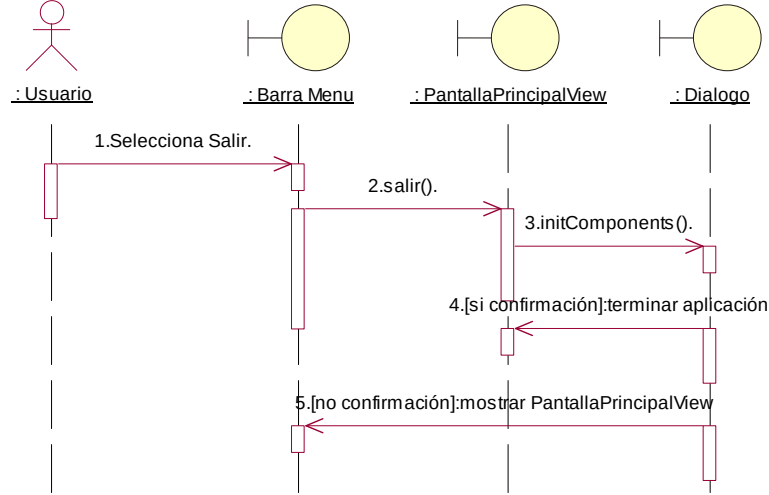


Figura 199: Diagrama de Secuencia: Salir Sistema

III.2.2.5 Diagrama de Secuencia Obtener Ayuda

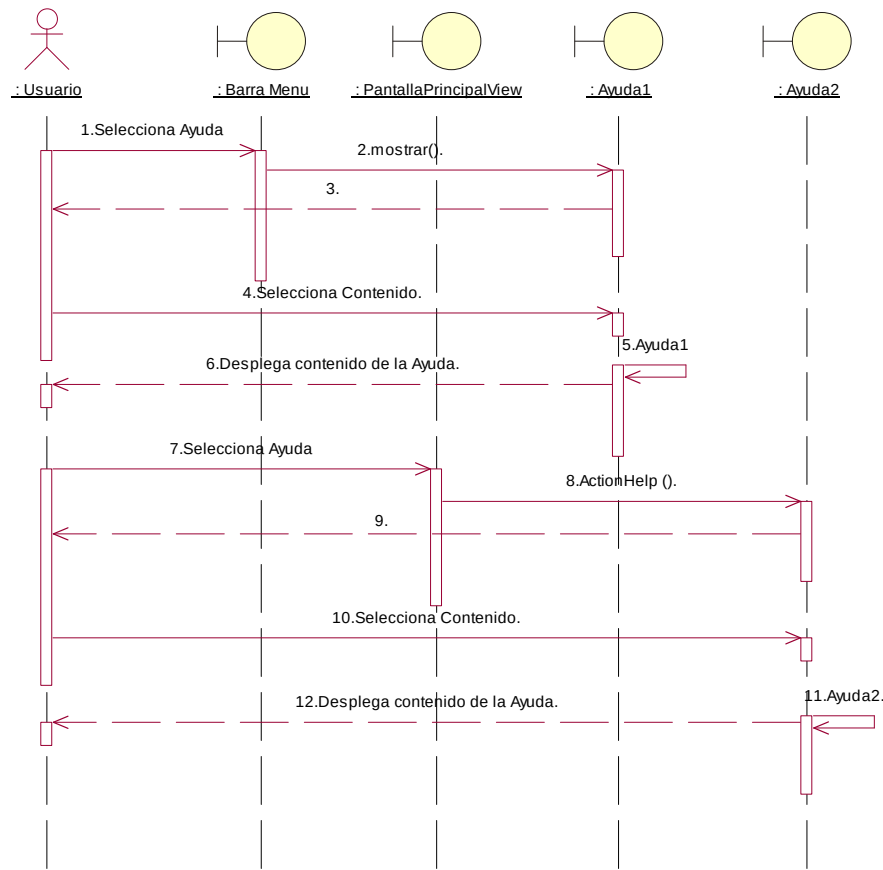


Figura 200: Diagrama de Secuencia: Obtener Ayuda

III.2.2.2 MODULO DE PROYECTOS

III.2.2.2 .1 Diagrama de Secuencia Administrar Propietarios

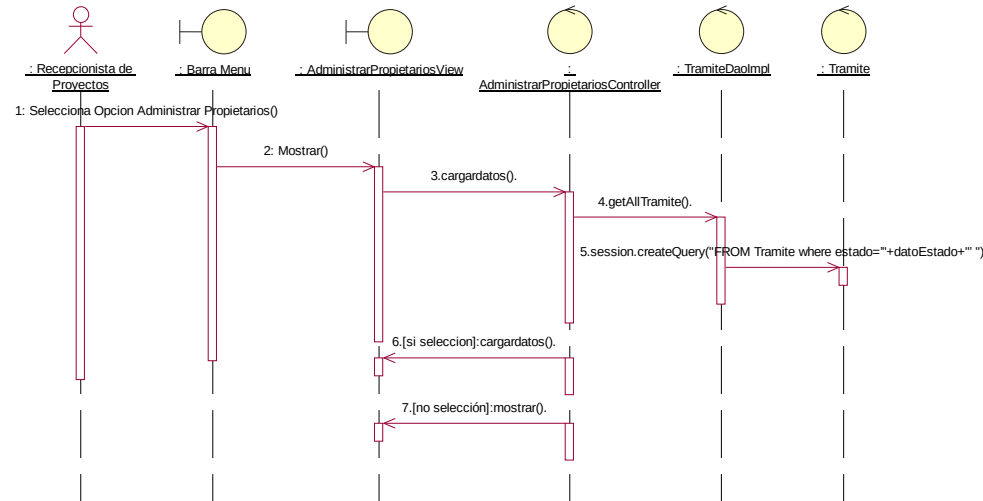


Figura 201: Diagrama de Secuencia: Administrar Propietarios

III.2.2.2 .2 Diagrama de Secuencia Adicionar Propietarios

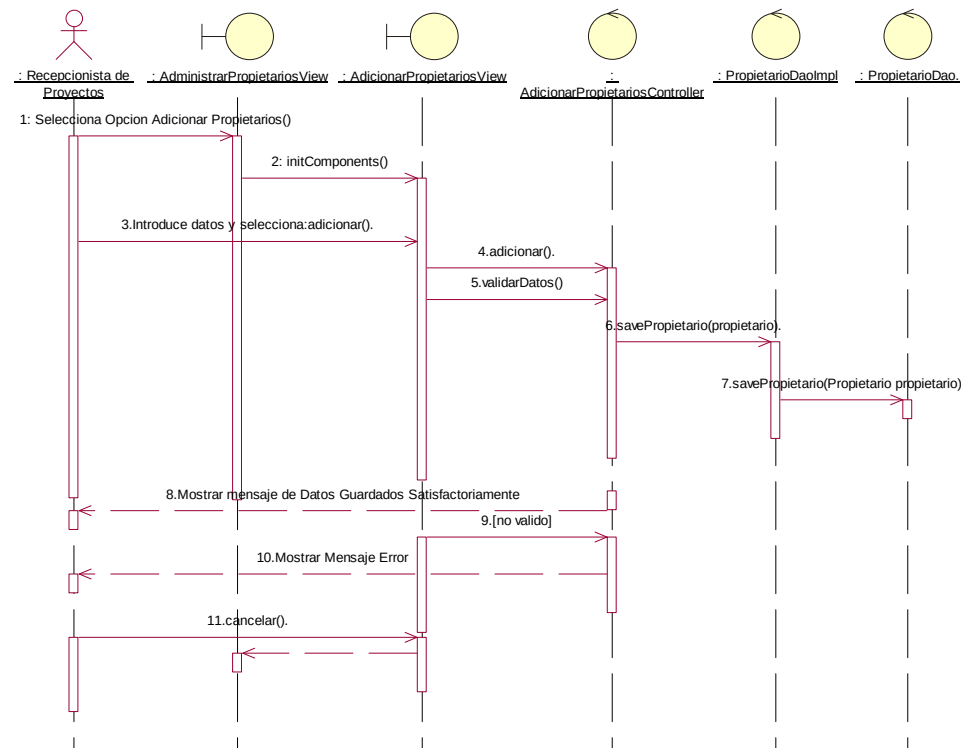


Figura 202: Diagrama de Secuencia: Adicionar Propietarios

III.2.2.2 .3 Diagrama de Secuencia Adicionar Projectista

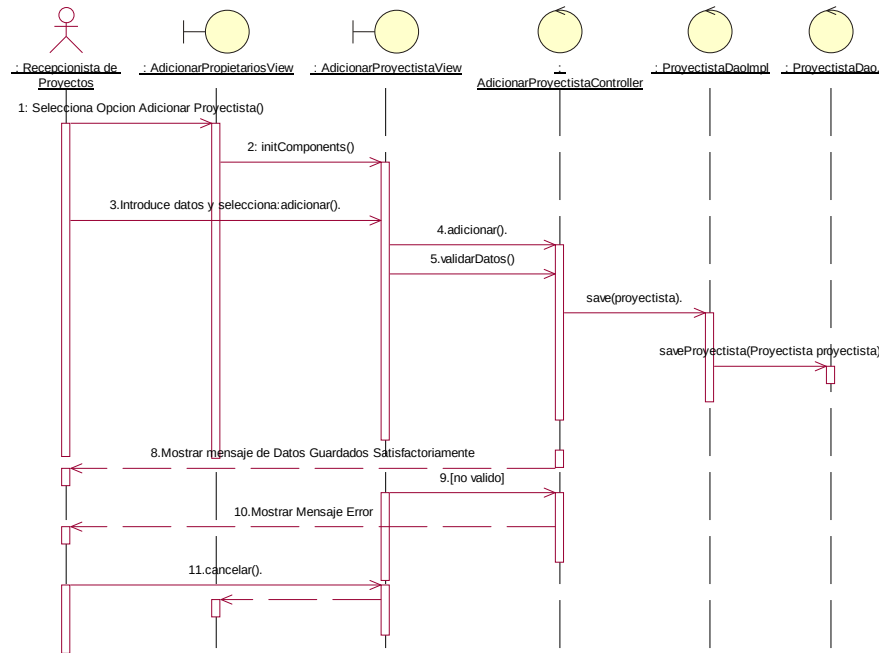


Figura 203: Diagrama de Secuencia: Adicionar Projectista

III.2.2.2 .4 Diagrama de Secuencia Adicionar Apoderado

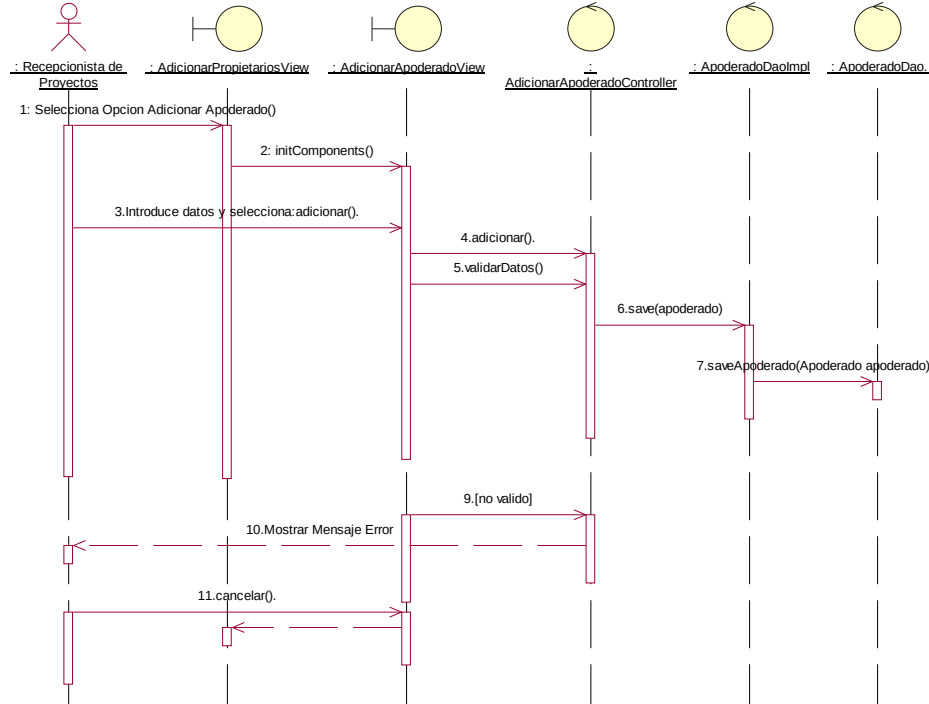


Figura 204: Diagrama de Secuencia: Adicionar Apoderado

III.2.2.2 .5 Diagrama de Secuencia Modificar Propietarios

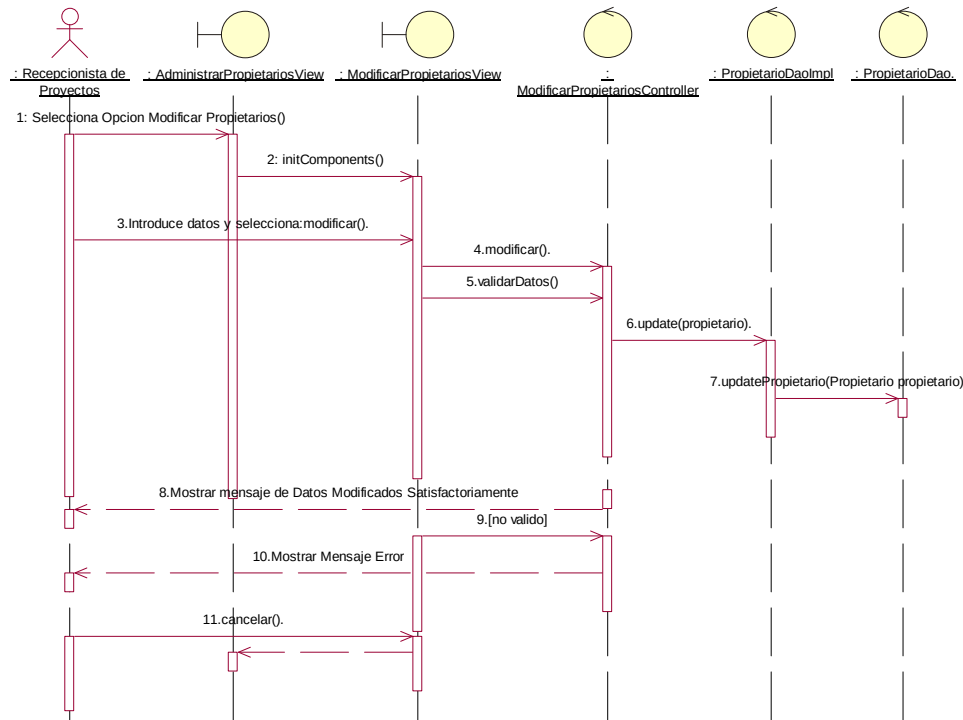


Figura 205: Diagrama de Secuencia: Modificar Propietarios

III.2.2.2 .6 Diagrama de Secuencia Modificar Proyectista

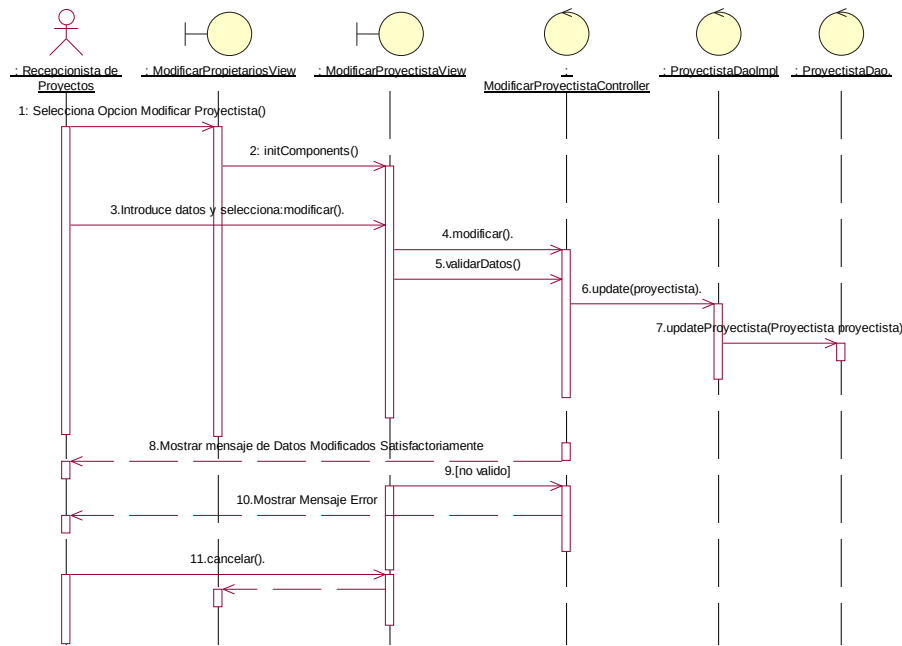


Figura 206: Diagrama de Secuencia: Modificar Proyectista

III.2.2.2 .7 Diagrama de Secuencia Modificar Apoderado

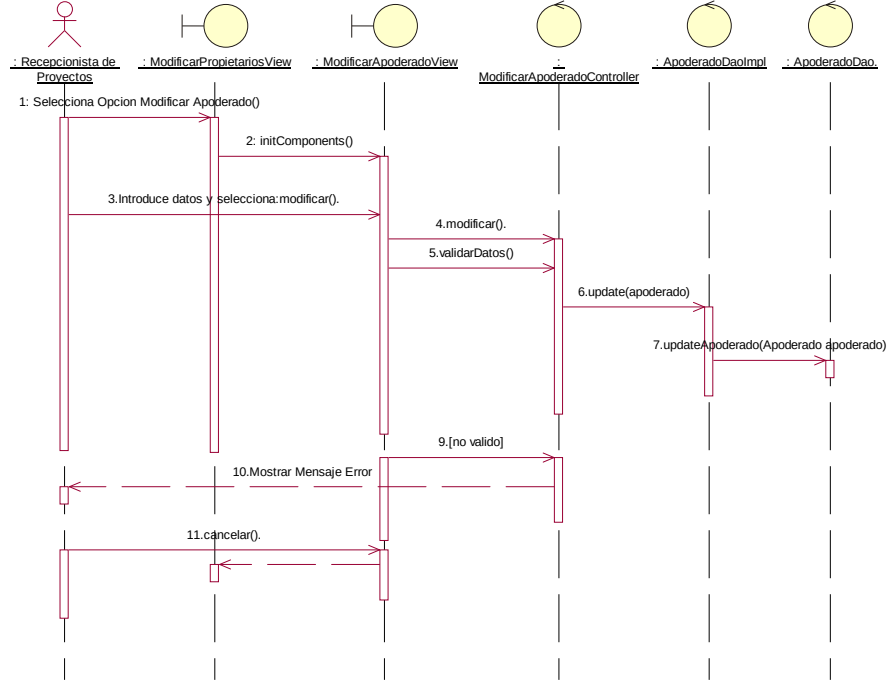


Figura 207: Diagrama de Secuencia: Modificar Apoderado

III.2.2.2 .8 Diagrama de Secuencia Ver Propietarios

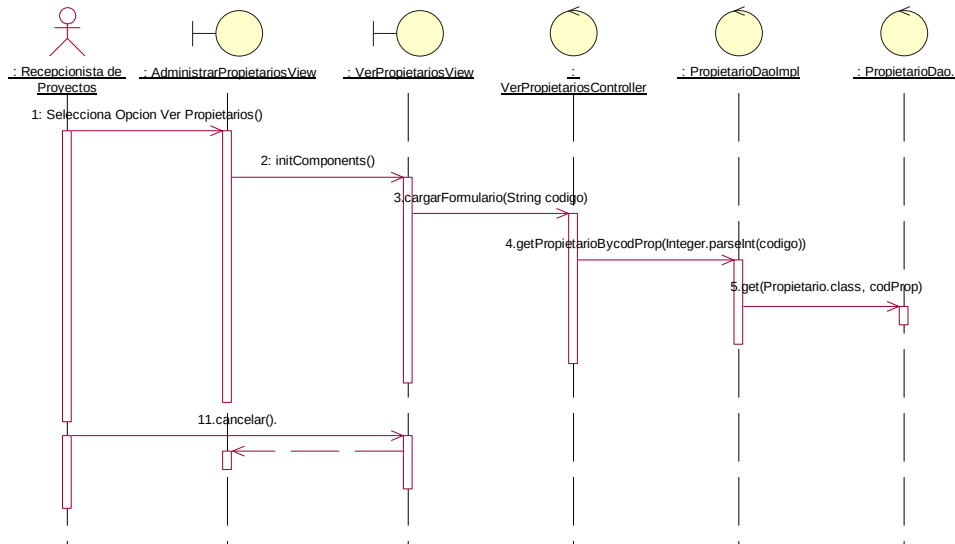


Figura 208: Diagrama de Secuencia: Ver Propietarios

III.2.2.2 .9 Diagrama de Secuencia Administrar Observaciones Generales

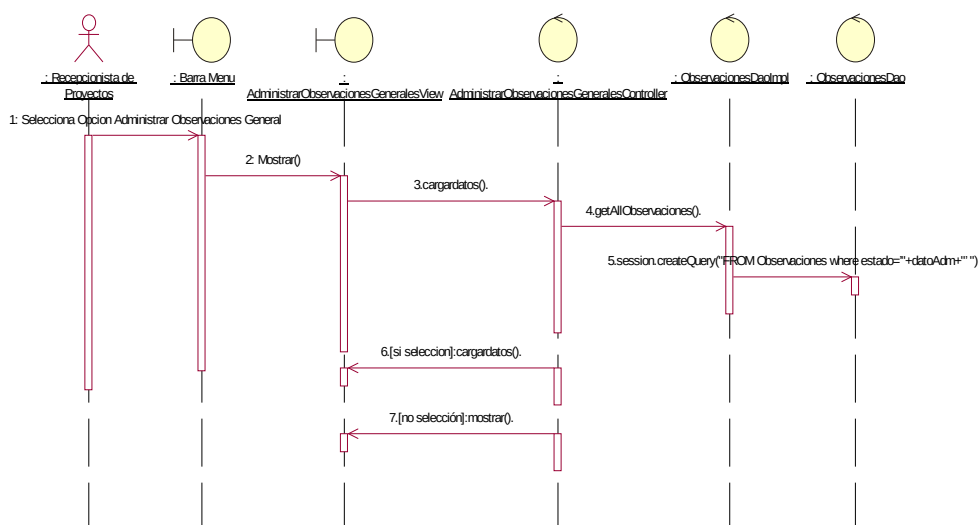


Figura 209: Diagrama de Secuencia: Administrar Observaciones Generales

III.2.2.2 .10 Diagrama de Secuencia Ver Observaciones Generales

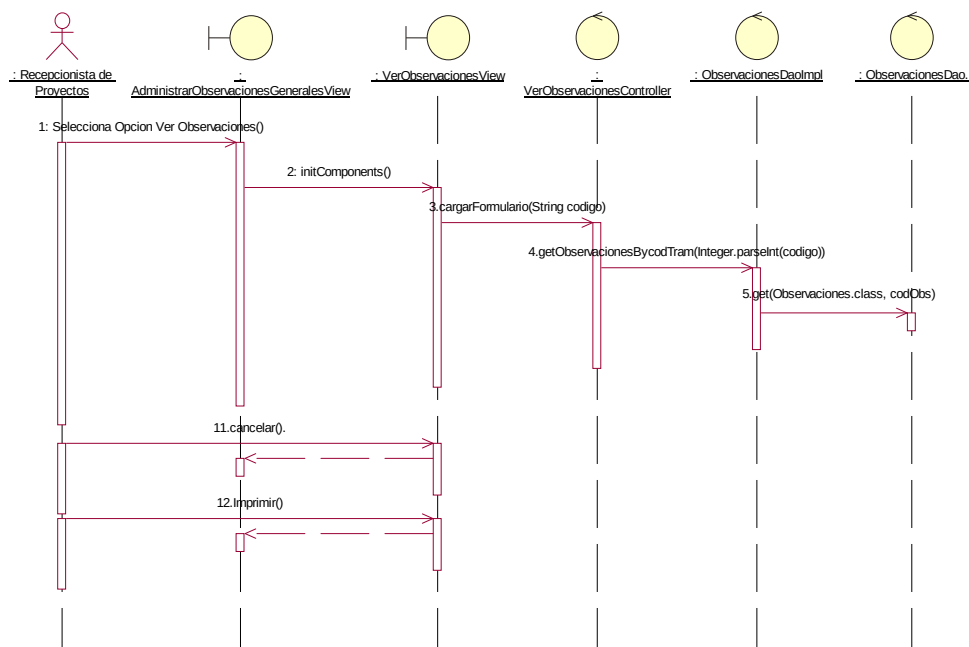


Figura 210: Diagrama de Secuencia: Ver Observaciones Generales

III.2.2.2 .11 Diagrama de Secuencia Dar Baja Trámite

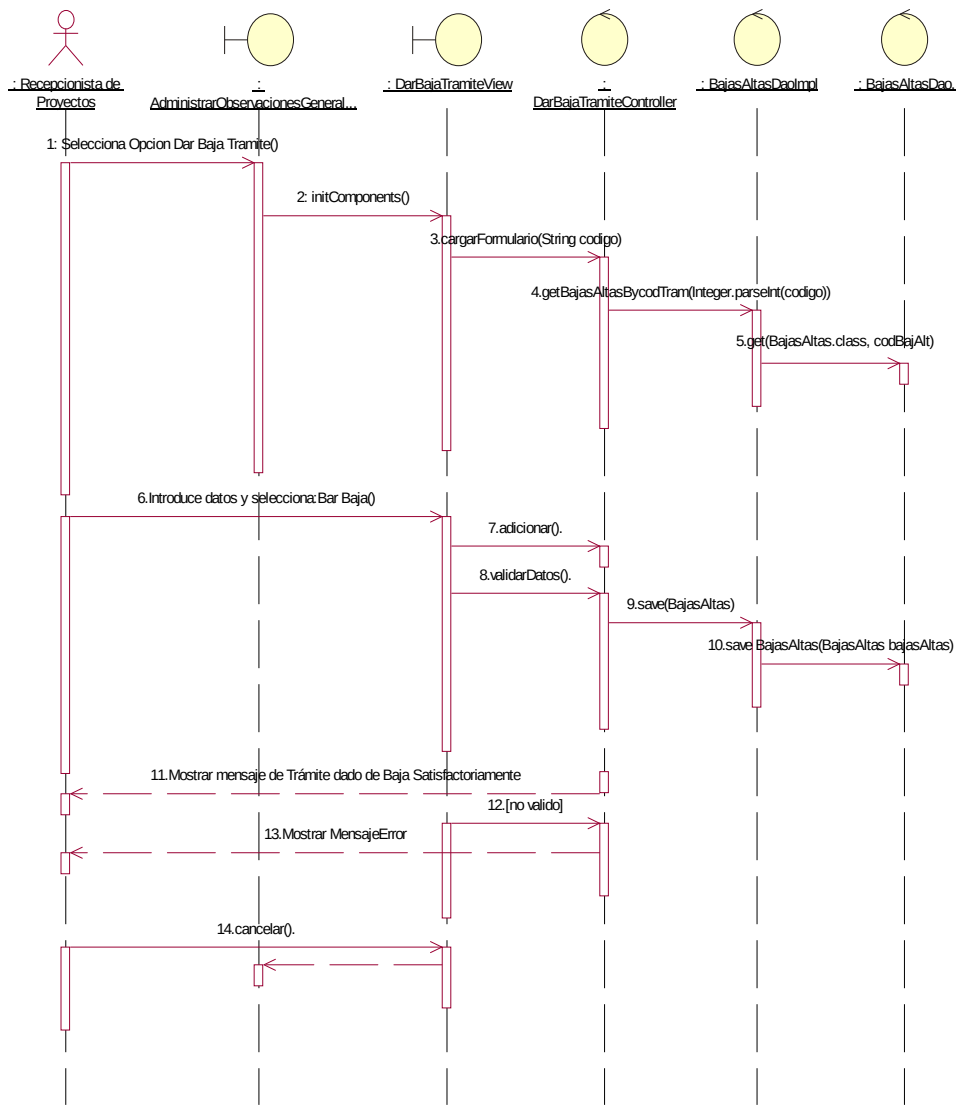


Figura 211: Diagrama de Secuencia: Dar Baja Trámite

III.2.2.2 .12 Diagrama de Secuencia Reasignar Trámite

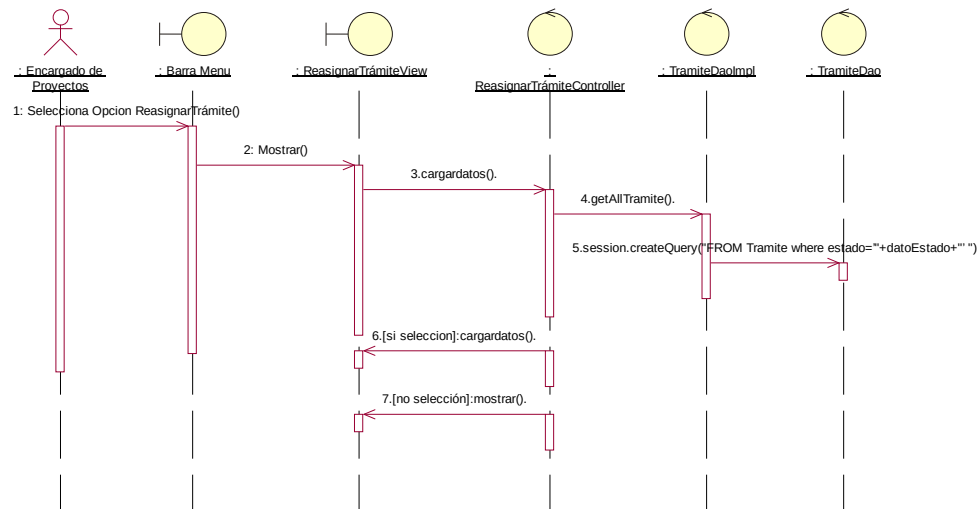


Figura 212: Diagrama de Secuencia: Reasignar Trámite

III.2.2.2 .13 Diagrama de Secuencia Registrar Reasignación

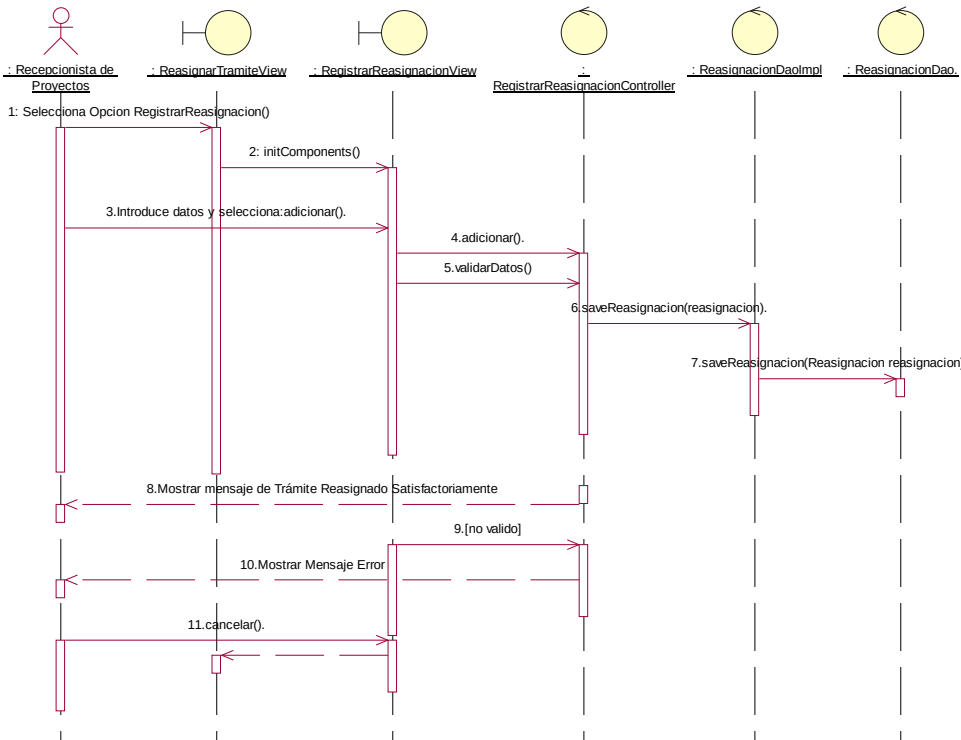


Figura 213: Diagrama de Secuencia: Registrar Reasignación

III.2.2.2 .14 Diagrama de Secuencia Ver Reasignación

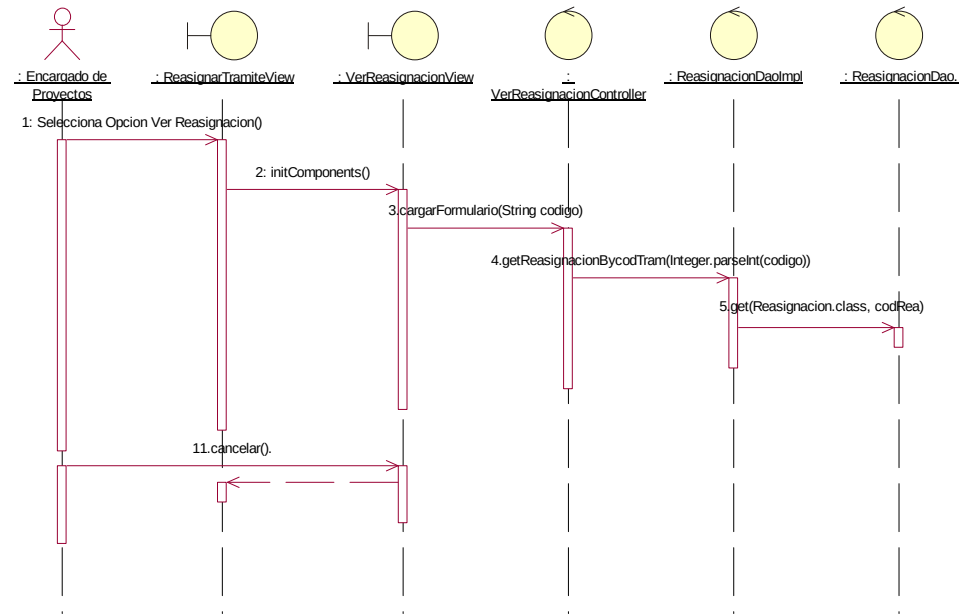


Figura 214: Diagrama de Secuencia: Ver Reasignación

III.2.2.2 .15 Diagrama de Secuencia Ver Datos Trámite

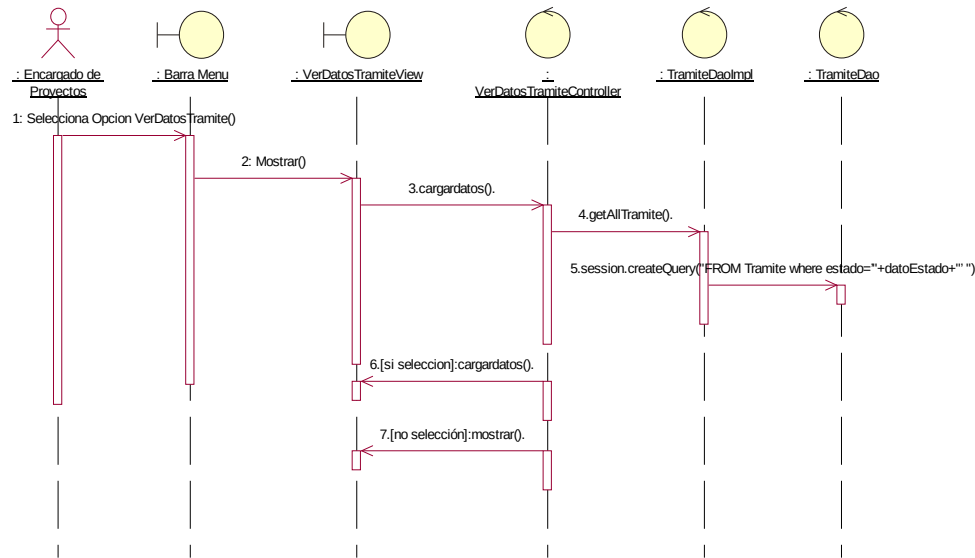


Figura 215: Diagrama de Secuencia: Ver Datos Trámite

III.2.2.2 .16 Diagrama de Secuencia Ver Estado Tramite

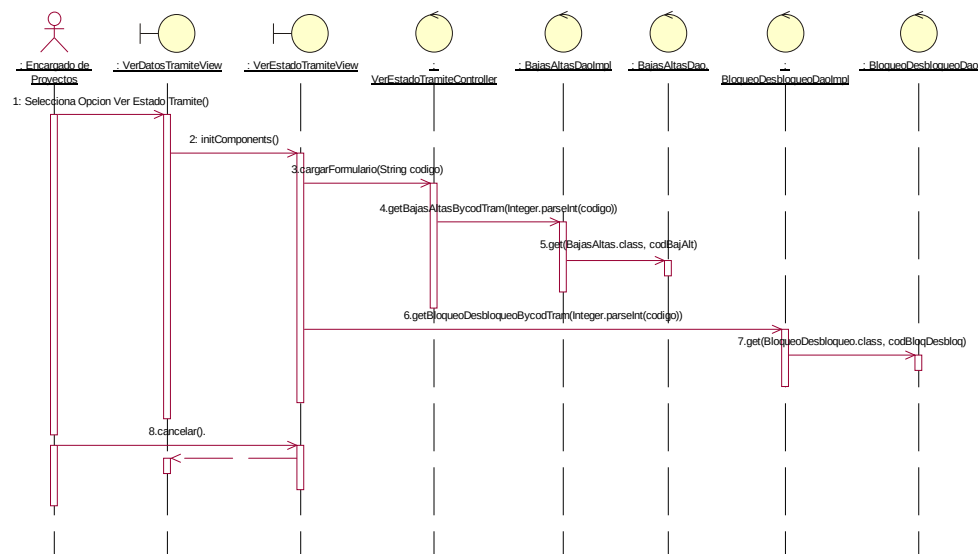


Figura 216: Diagrama de Secuencia: Ver Estado Tramite

III.2.2.2 .17 Diagrama de Secuencia Ver Detalle Trámite

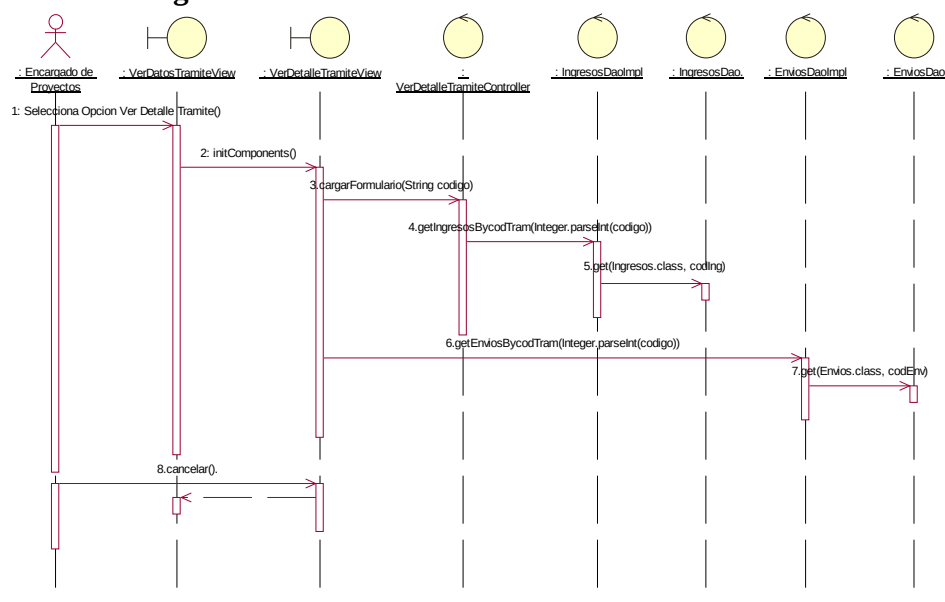


Figura 217: Diagrama de Secuencia: Ver Detalle Trámite

III.2.2.2 .18 Diagrama de Secuencia Administrar Ingresos

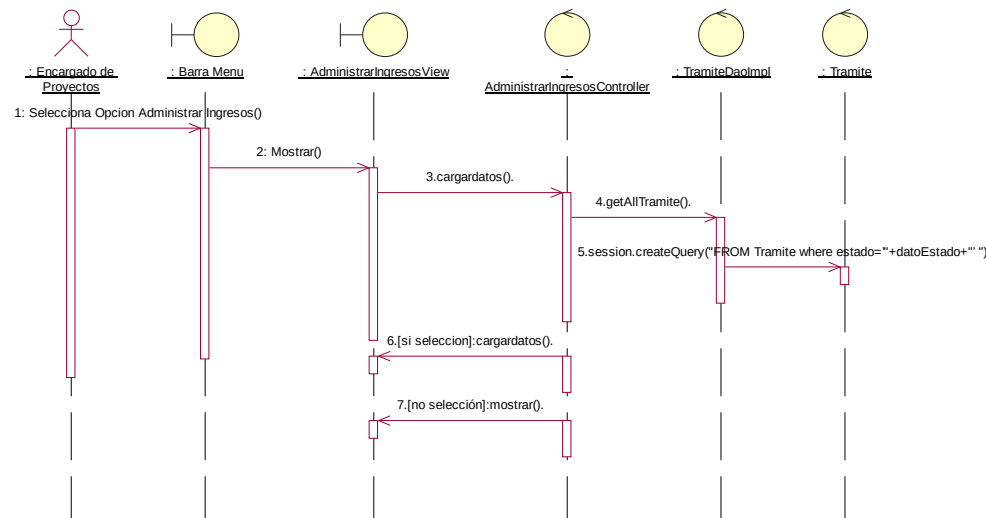


Figura 218: Diagrama de Secuencia: Administrar Ingresos

III.2.2.2 .19 Diagrama de Secuencia Registrar Ingresos

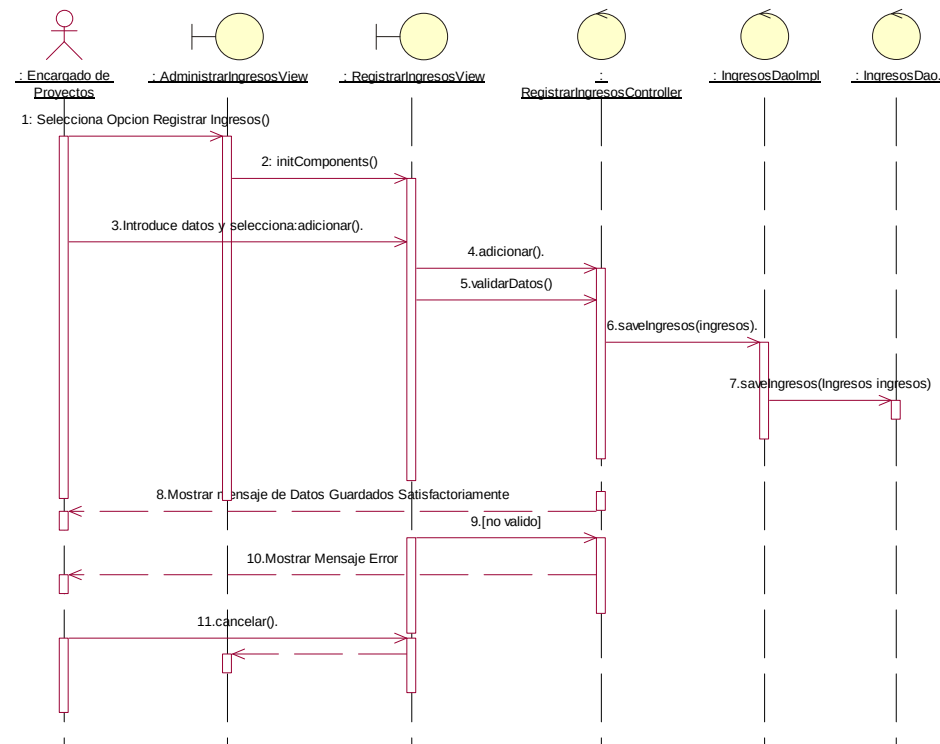


Figura 219: Diagrama de Secuencia: Registrar Ingresos

III.2.2.2 .20 Diagrama de Secuencia Modificar Ingresos

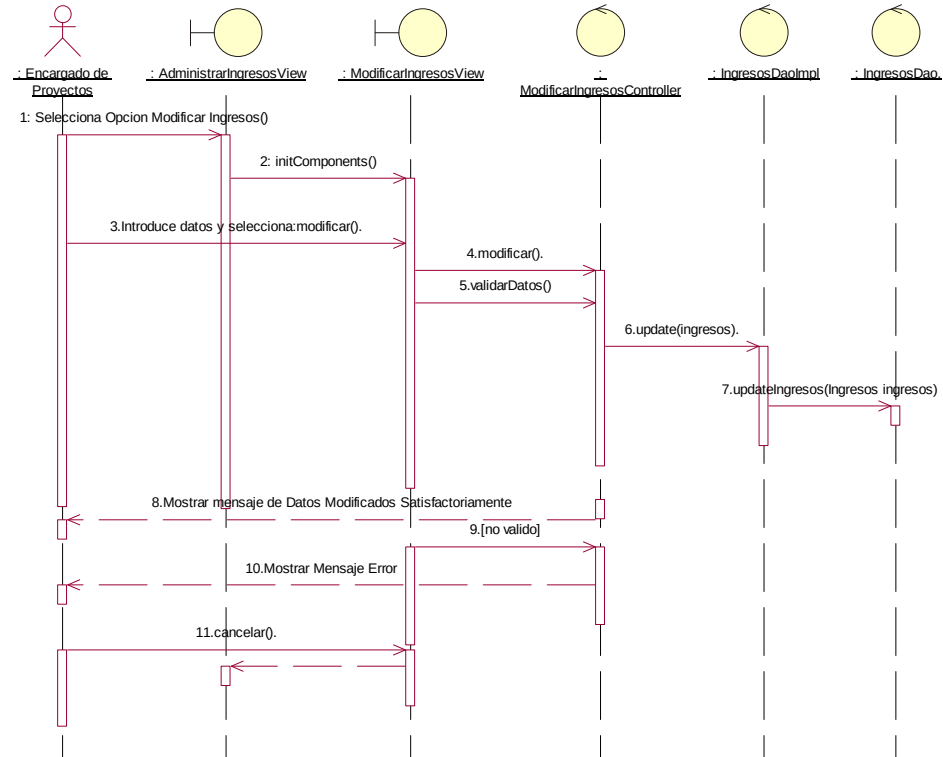


Figura 220: Diagrama de Secuencia: Modificar Ingresos

III.2.2.2 .21 Diagrama de Secuencia Ver Ingresos

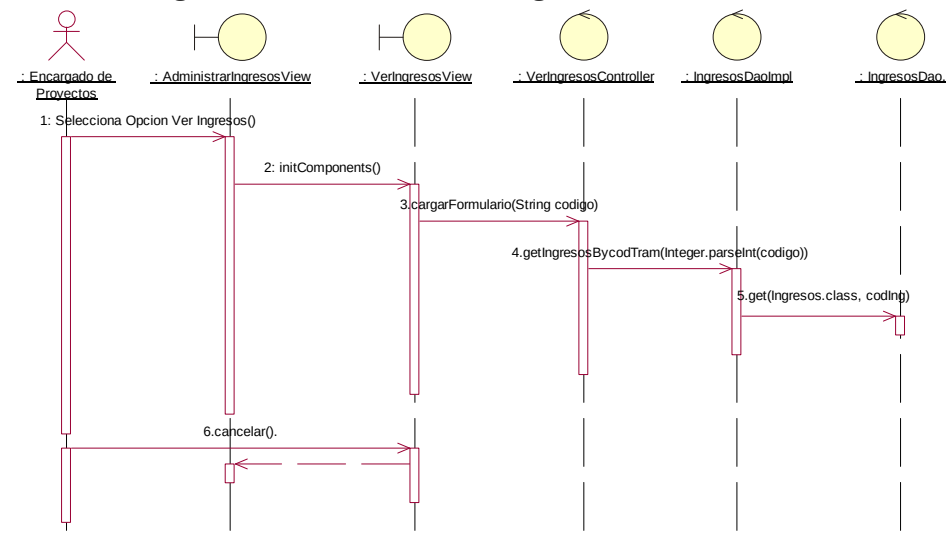


Figura 221: Diagrama de Secuencia: Ver Ingresos

III.2.2.2 .22 Diagrama de Secuencia Administrar Salidas

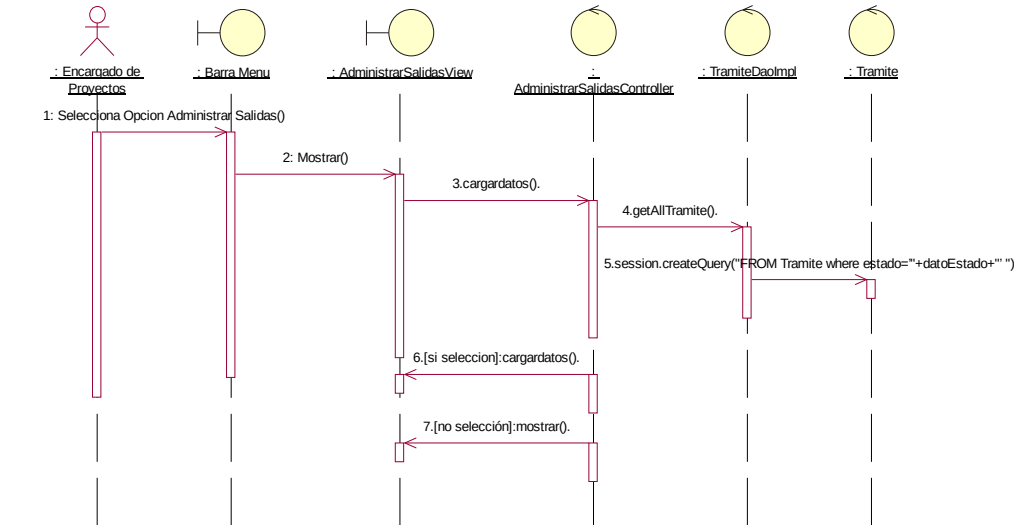


Figura 222: Diagrama de Secuencia: Administrar Salidas

III.2.2.2 .23 Diagrama de Secuencia Registrar Salidas

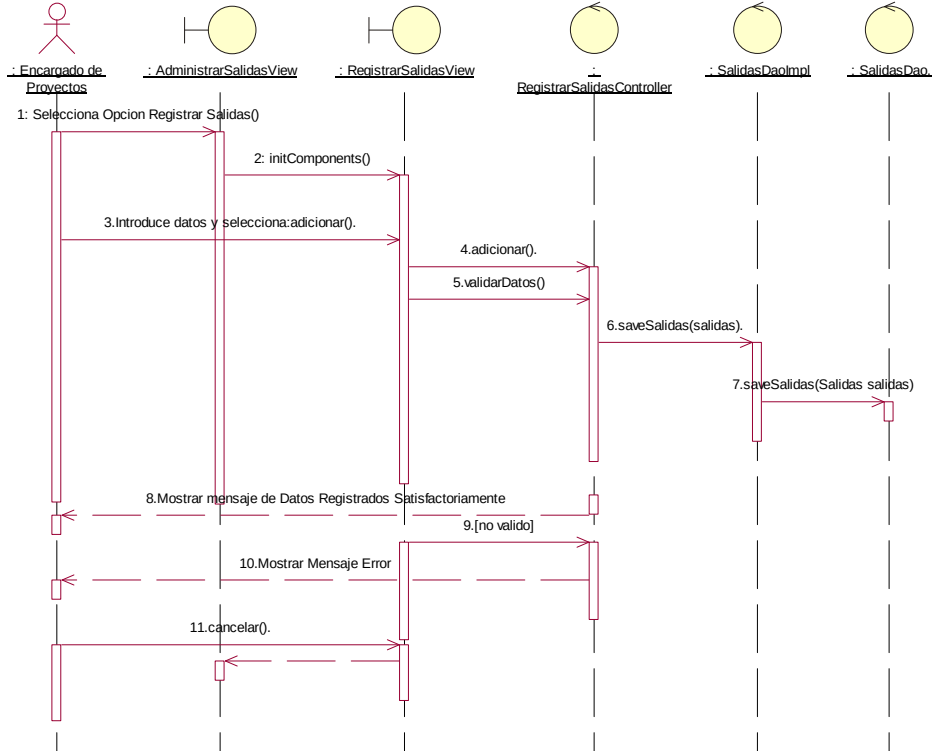


Figura 223: Diagrama de Secuencia: Registrar Salidas

III.2.2.2 .24 Diagrama de Secuencia Modificar Salidas

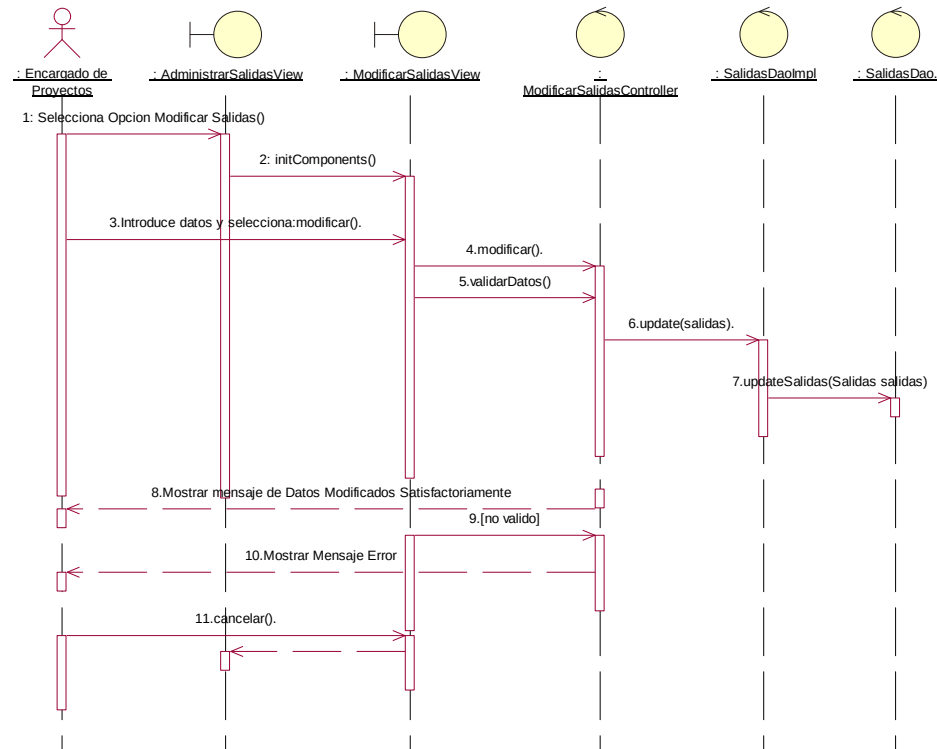


Figura 224: Diagrama de Secuencia: Modificar Salidas

III.2.2.2 .25 Diagrama de Secuencia Ver Salidas

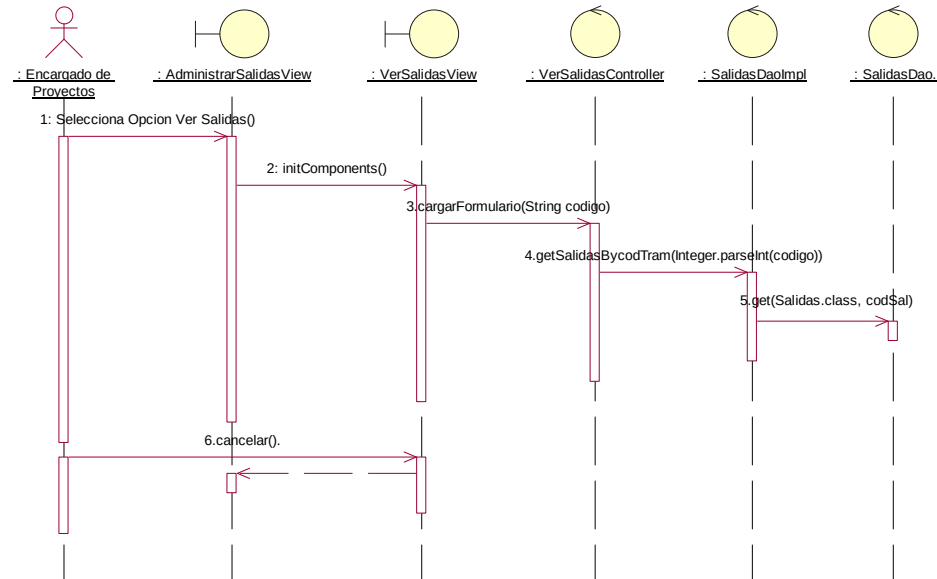


Figura 225: Diagrama de Secuencia: Ver Salidas

III.2.2.2 .26 Diagrama de Secuencia Administrar Envíos

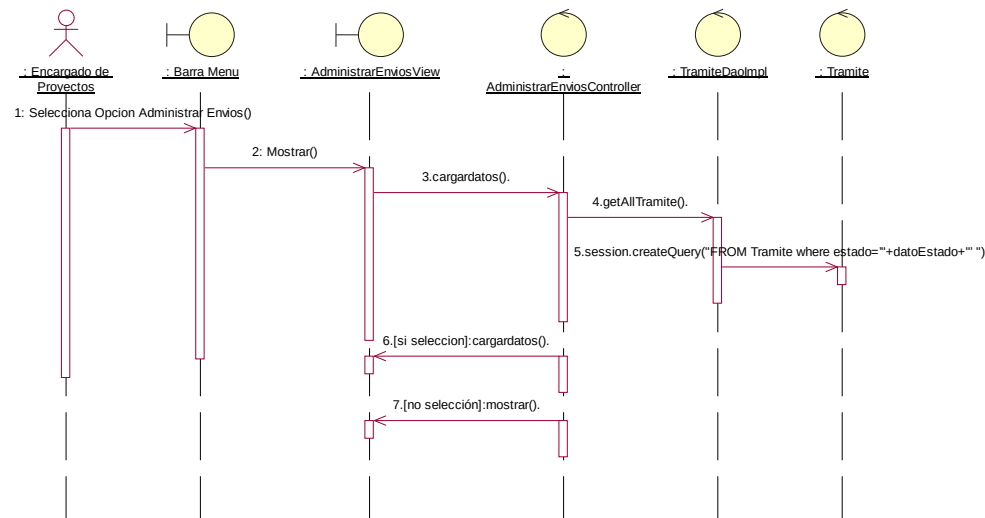


Figura 226: Diagrama de Secuencia: Administrar Envíos

III.2.2.2 .27 Diagrama de Secuencia Registrar Envíos

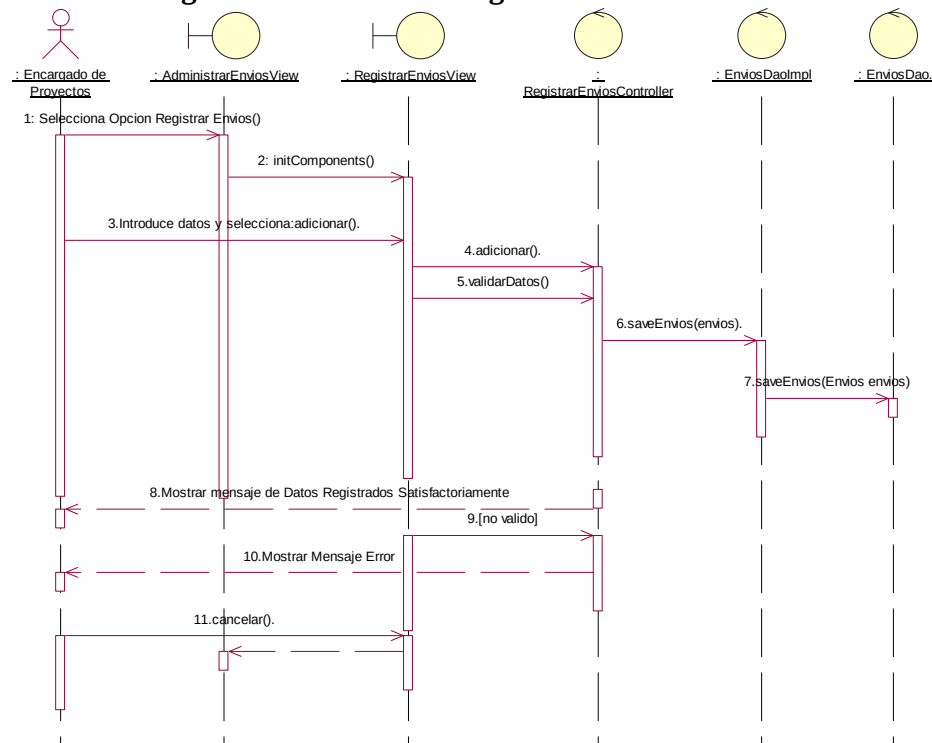


Figura 227: Diagrama de Secuencia: Registrar Envíos

III.2.2.2 .28 Diagrama de Secuencia Modificar Envíos

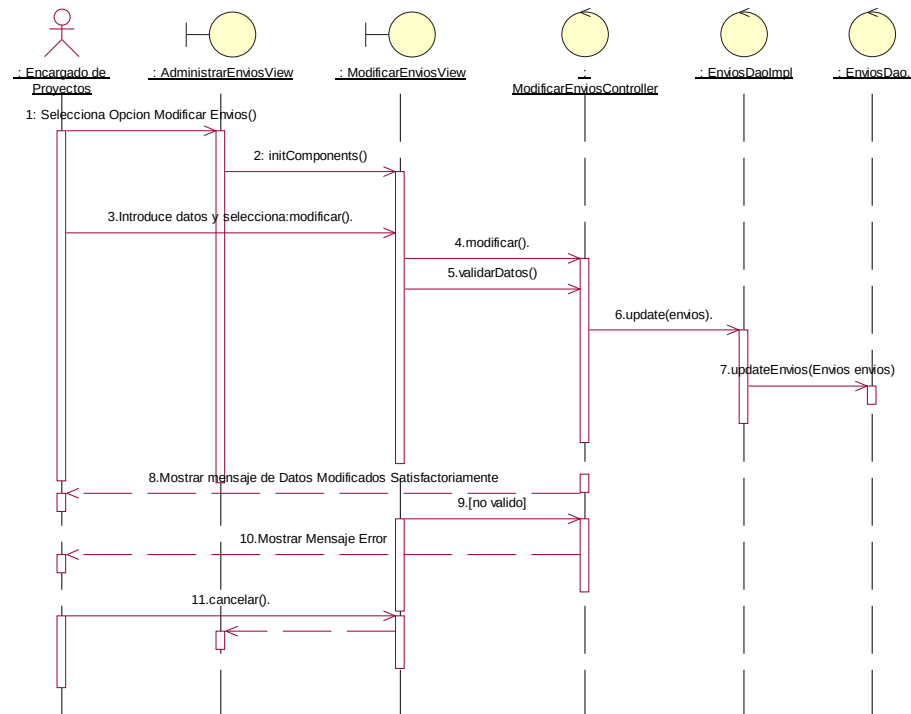


Figura 228: Diagrama de Secuencia: Modificar Envíos

III.2.2.2 .29 Diagrama de Secuencia Ver Envíos

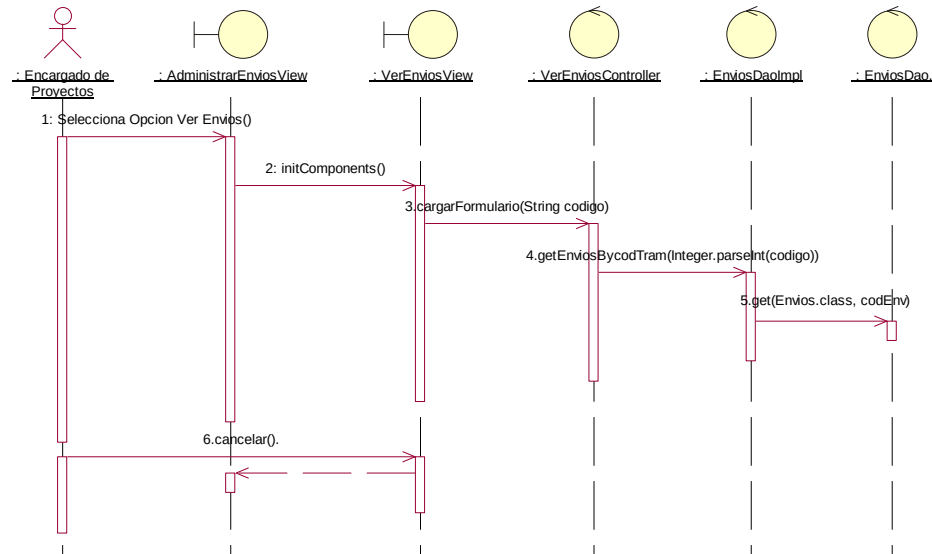


Figura 229: Diagrama de Secuencia: Ver Envíos

III.2.2.2 .30 Diagrama de Secuencia Administrar Recepción

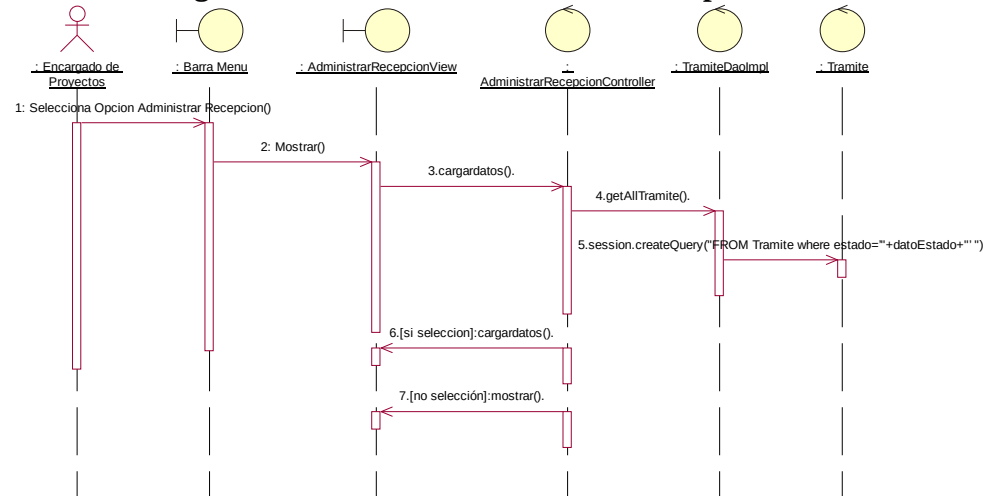


Figura 230: Diagrama de Secuencia: Administrar Recepción

III.2.2.2 .31 Diagrama de Secuencia Registrar Recepción

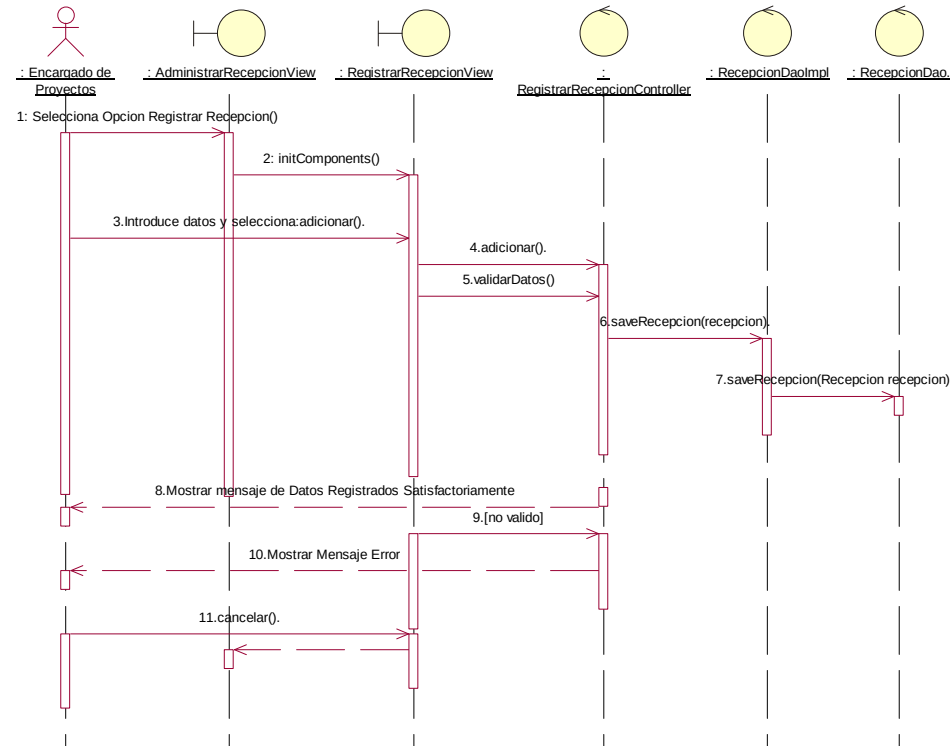


Figura 231: Diagrama de Secuencia: Registrar Recepción

III.2.2.2 .32 Diagrama de Secuencia Modificar Recepción

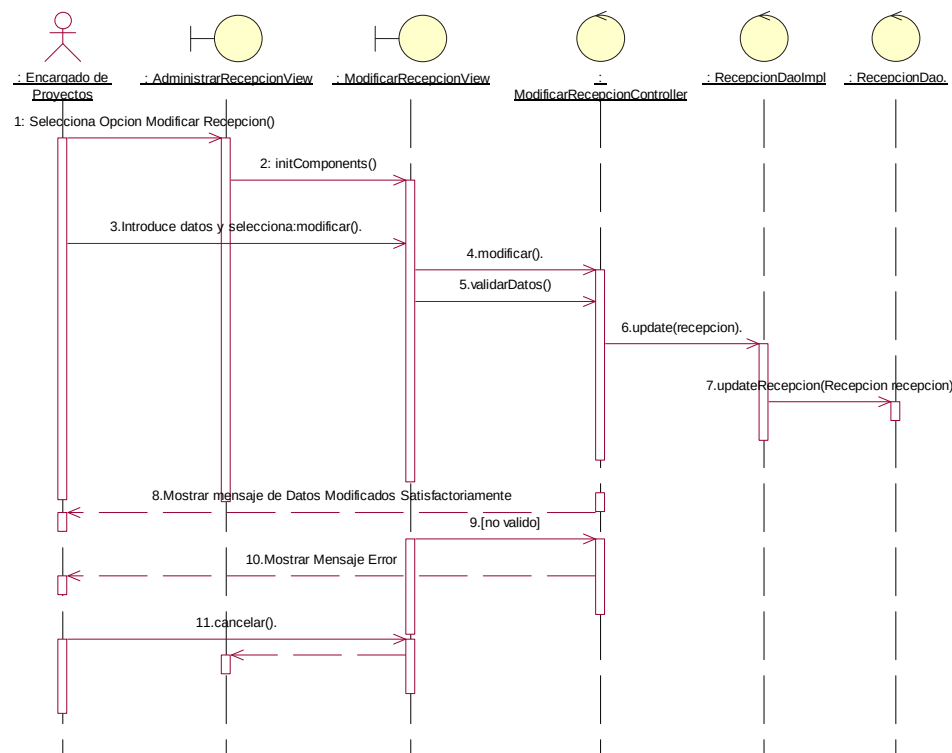


Figura 232: Diagrama de Secuencia: Modificar Recepción

III.2.2.2 .33 Diagrama de Secuencia Ver Recepción

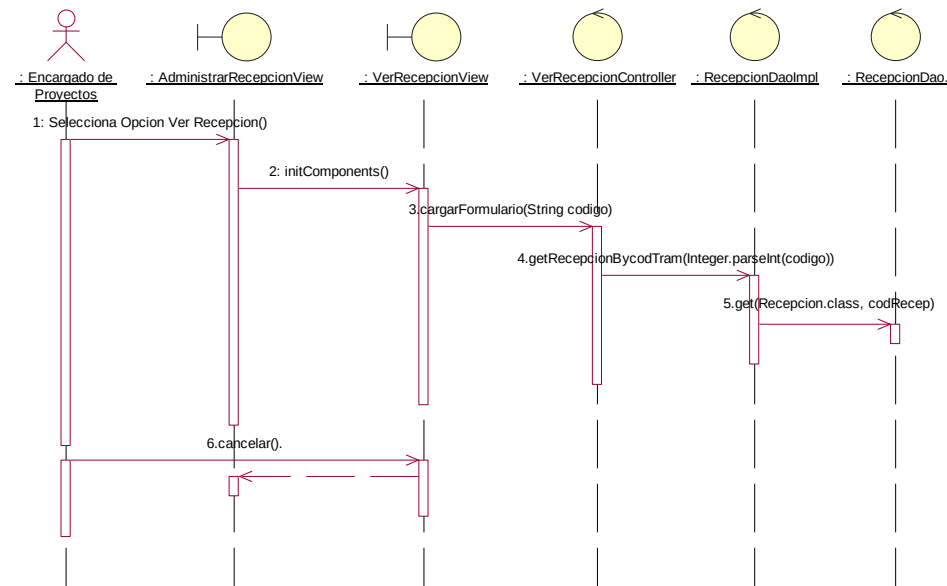


Figura 233: Diagrama de Secuencia: Ver Recepción

III.2.2.2 .34 Diagrama de Secuencia Asignar Numero Oficial

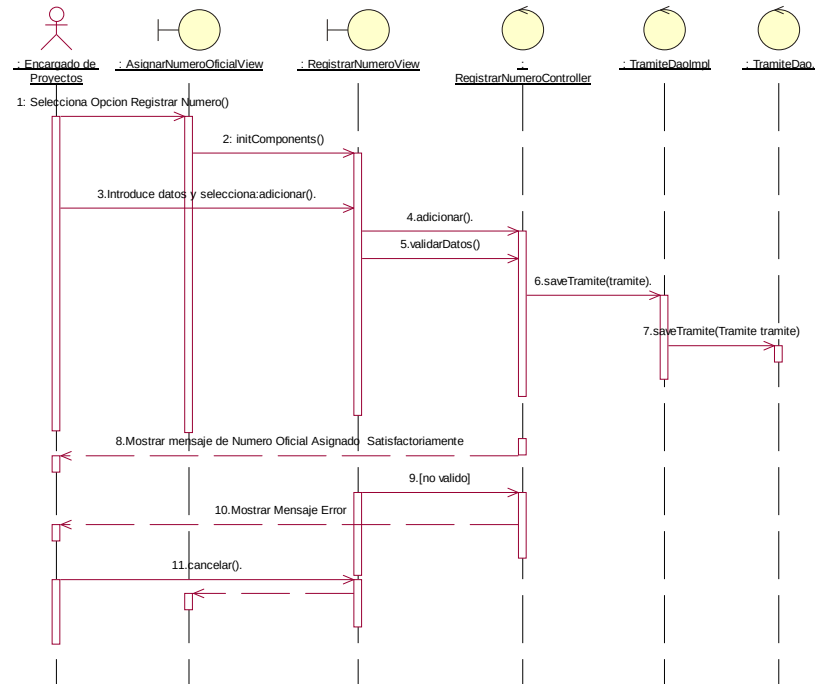


Figura 234: Diagrama de Secuencia: Asignar Numero Oficial

III.2.2.2 .35 Diagrama de Secuencia Asignar Personal Trámite

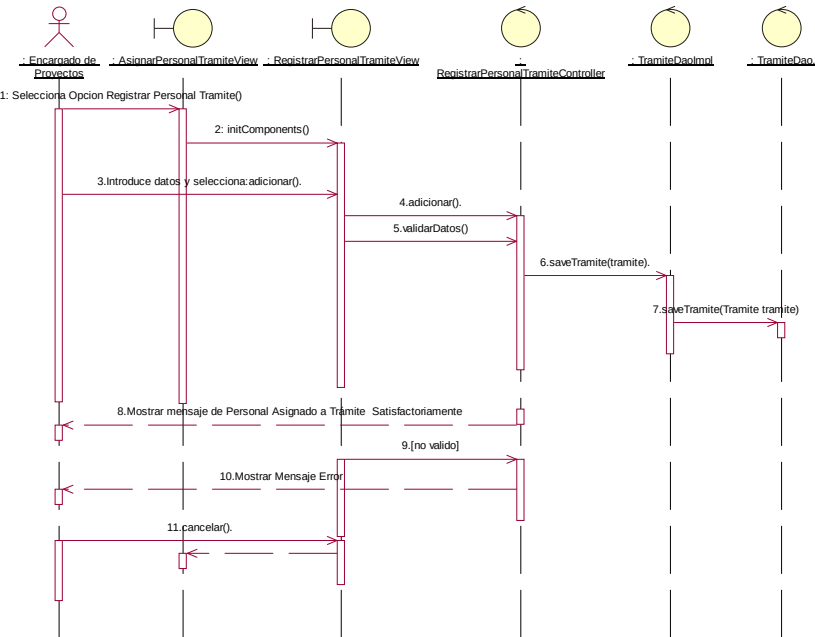


Figura 235: Diagrama de Secuencia: Asignar Personal Trámite

III.2.2.2 .36 Diagrama de Secuencia Listar Mensajes

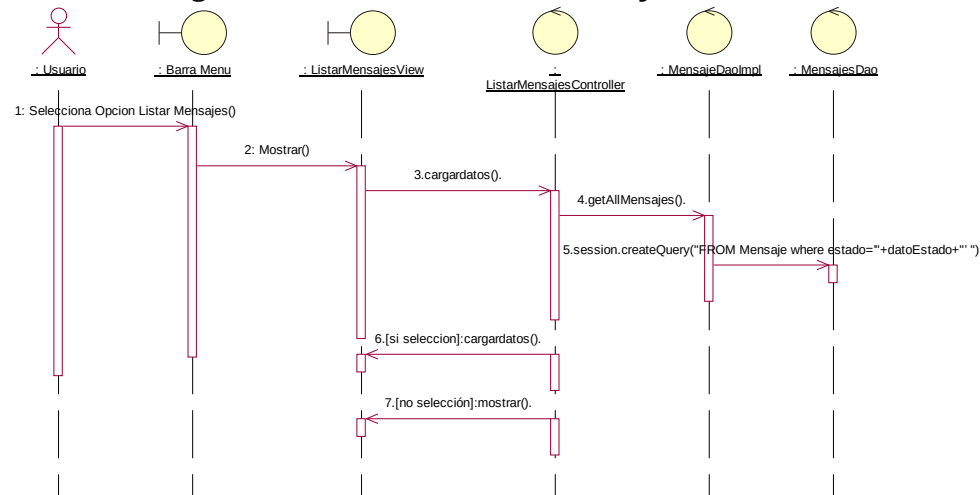


Figura 236: Diagrama de Secuencia: Listar Mensajes

III.2.2.2 .38 Diagrama de Secuencia Enviar Mensaje

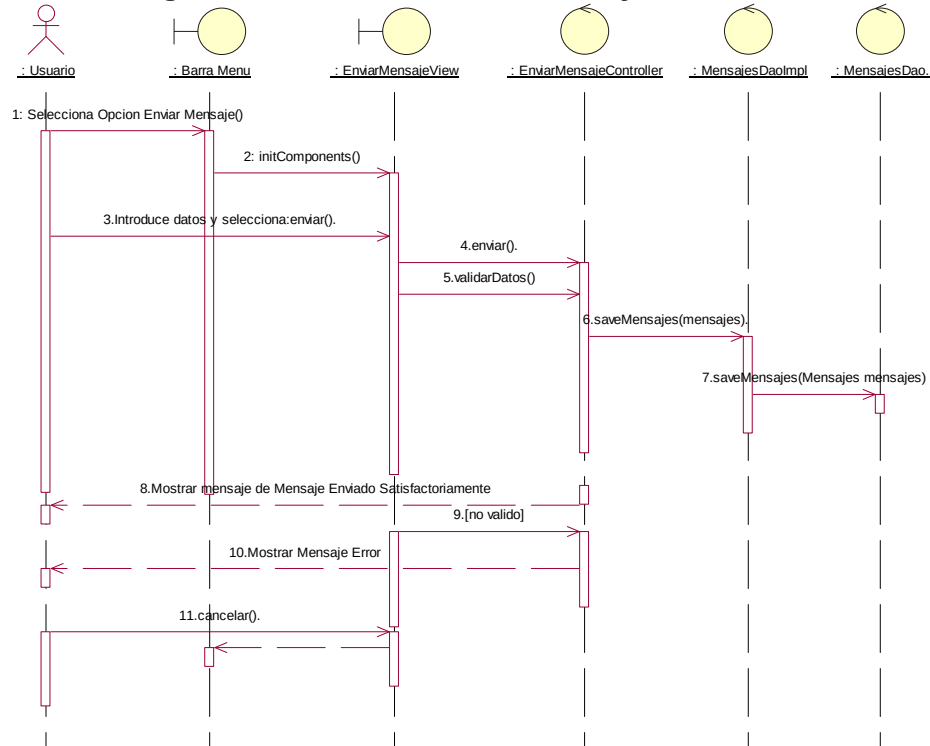


Figura 237: Diagrama de Secuencia: Enviar Mensaje

III.2.2.2. 39 Diagrama de Secuencia Eliminar Mensaje

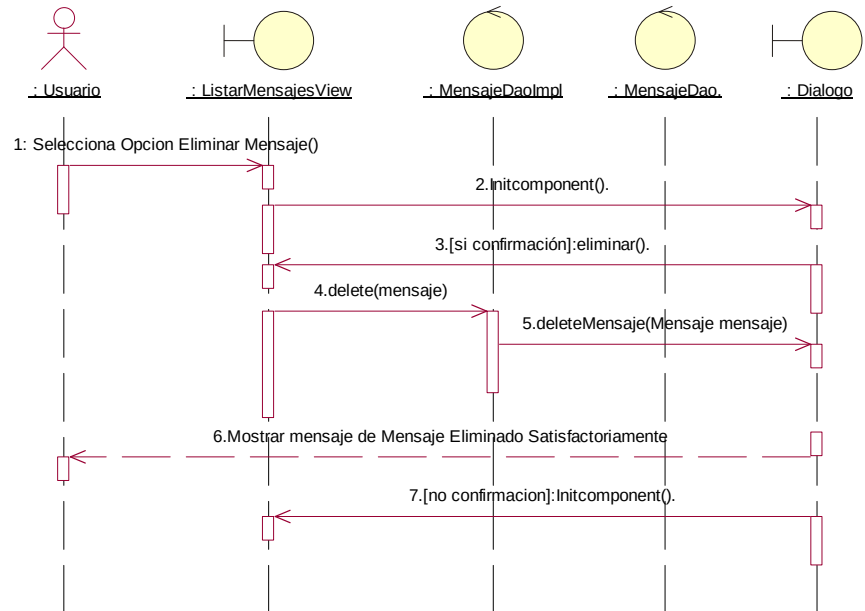


Figura 238: Diagrama de Secuencia: Eliminar Mensaje

III.2.2.2 .40 Diagrama de Secuencia Ver Mensaje

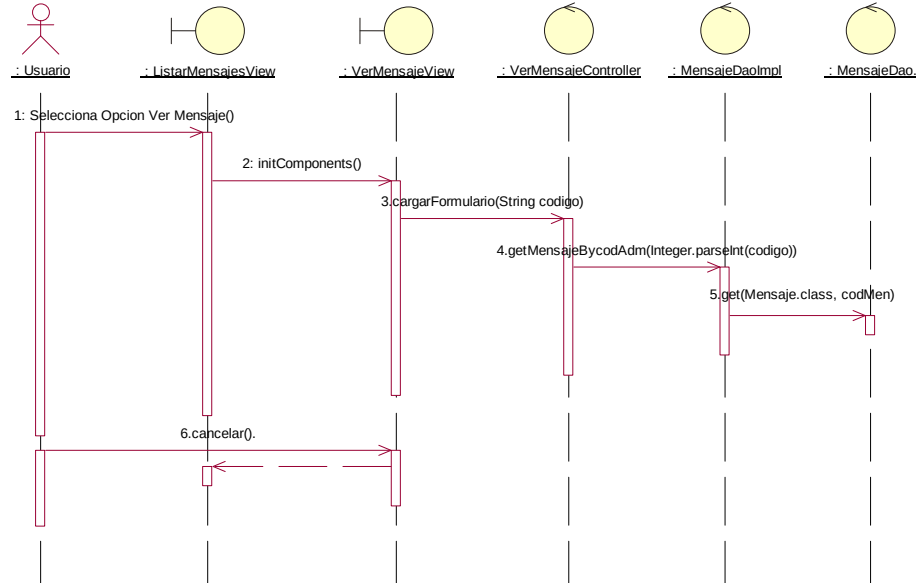


Figura 239: Diagrama de Secuencia: Ver Mensaje

III.2.2.2 .41 Diagrama de Secuencia Administrar Observaciones

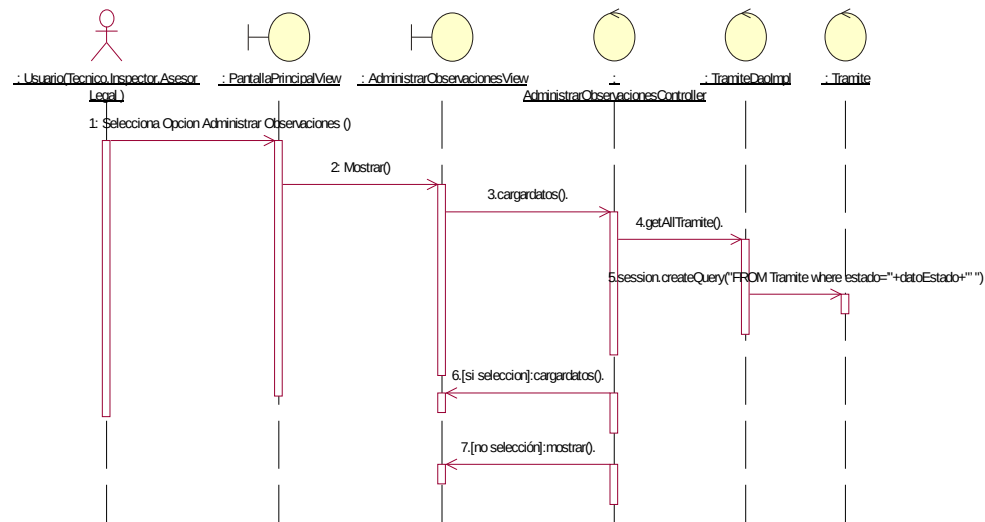


Figura 240: Diagrama de Secuencia: Administrar Observaciones

III.2.2.2 .42 Diagrama de Secuencia Adicionar Observaciones

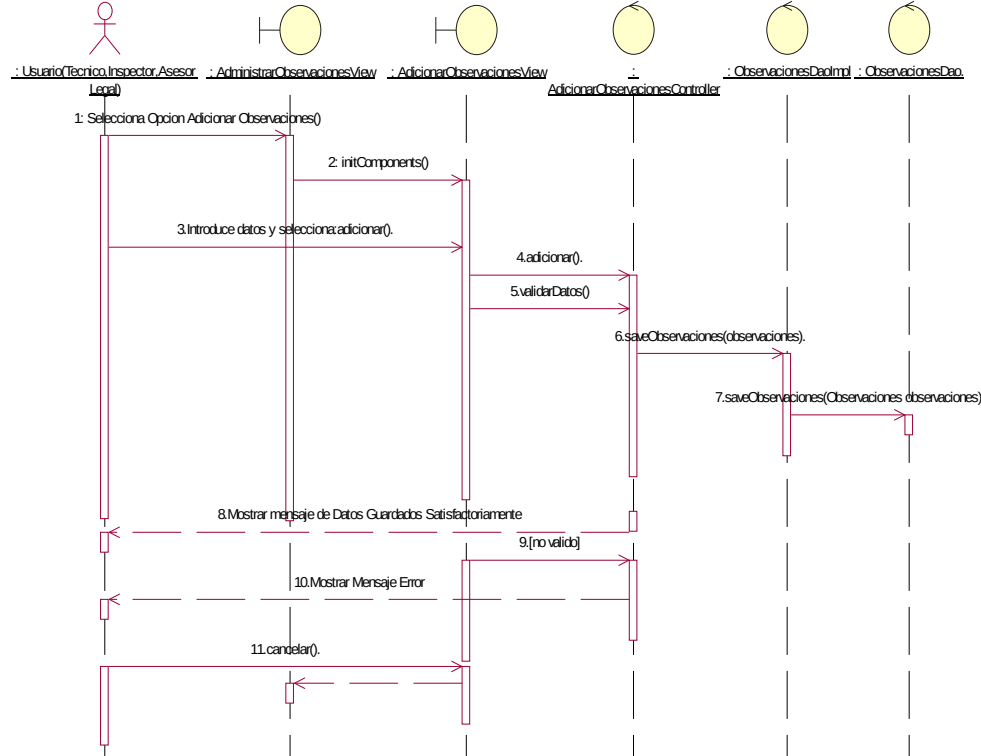


Figura 241: Diagrama de Secuencia: Adicionar Observaciones

III.2.2.2 .43 Diagrama de Secuencia Modificar Observaciones

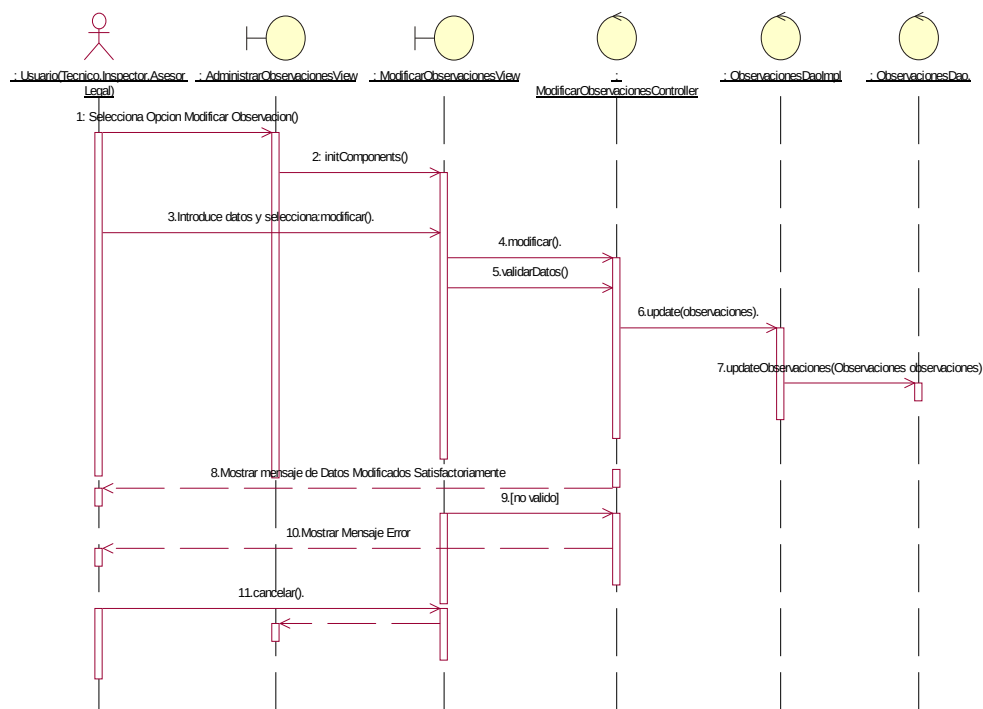


Figura 242: Diagrama de Secuencia: Modificar Observaciones

III.2.2.2 .45 Diagrama de Secuencia Ver Observaciones

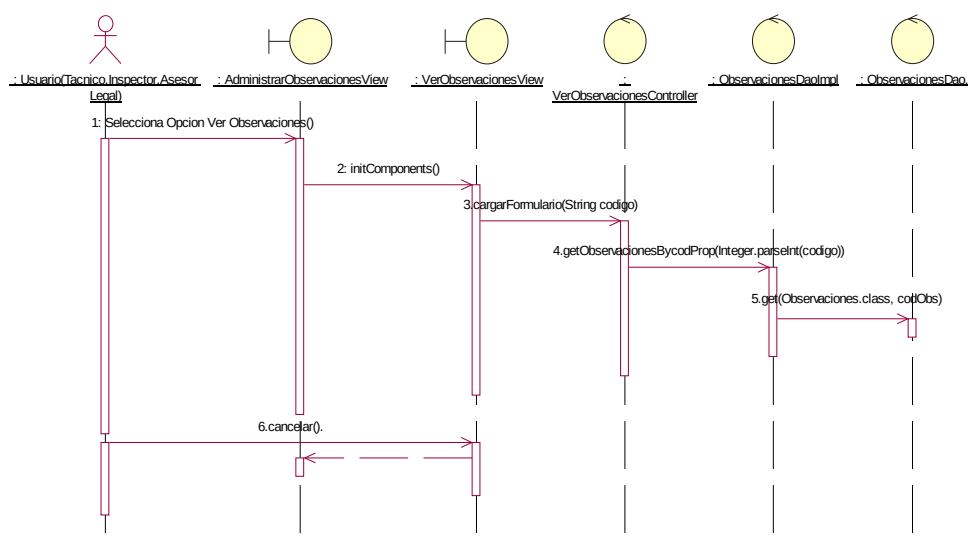


Figura 243: Diagrama de Secuencia: Ver Observaciones

III.2.2.2 .56 Diagrama de Secuencia Administrar Observaciones Jefe Proyectos

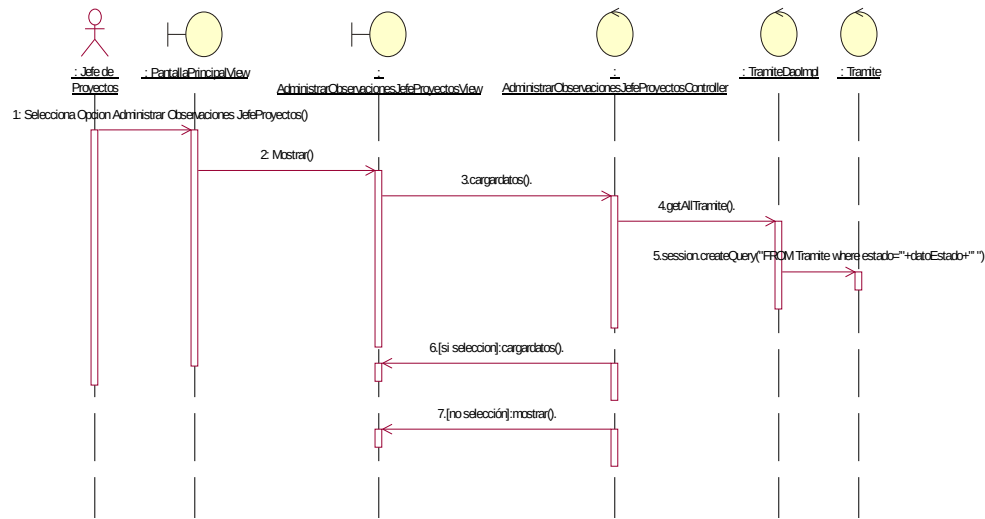


Figura 244: Diagrama de Secuencia: Administrar Observaciones Jefe Proyectos

III.2.2.2 .57 Diagrama de Secuencia Adicionar Observaciones Jefe Proyectos

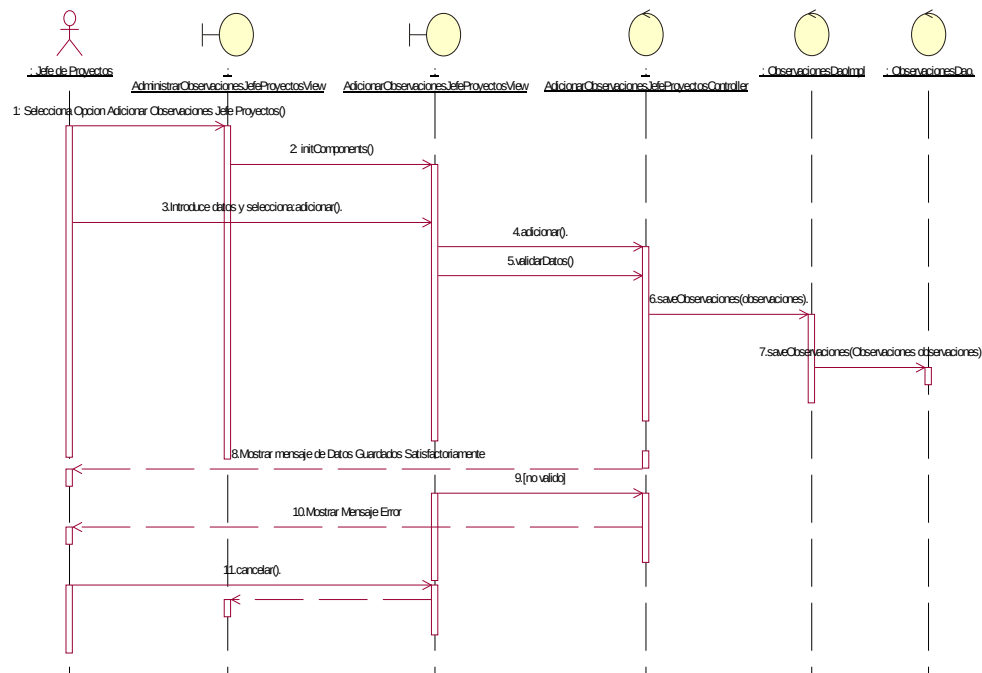


Figura 245: Diagrama de Secuencia: Adicionar Observaciones Jefe Proyectos

III.2.2.2 .58 Diagrama de Secuencia Modificar Observaciones Jefe Proyectos

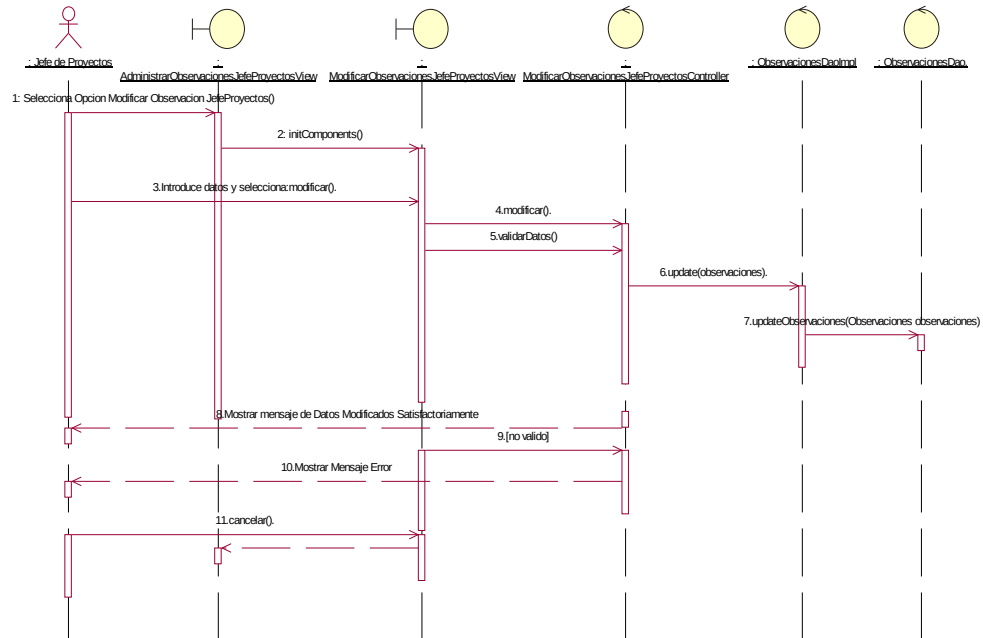


Figura 246: Diagrama de Secuencia: Modificar Observaciones Jefe Proyectos

III.2.2.2 .60 Diagrama de Secuencia Ver Observaciones Jefe Proyectos

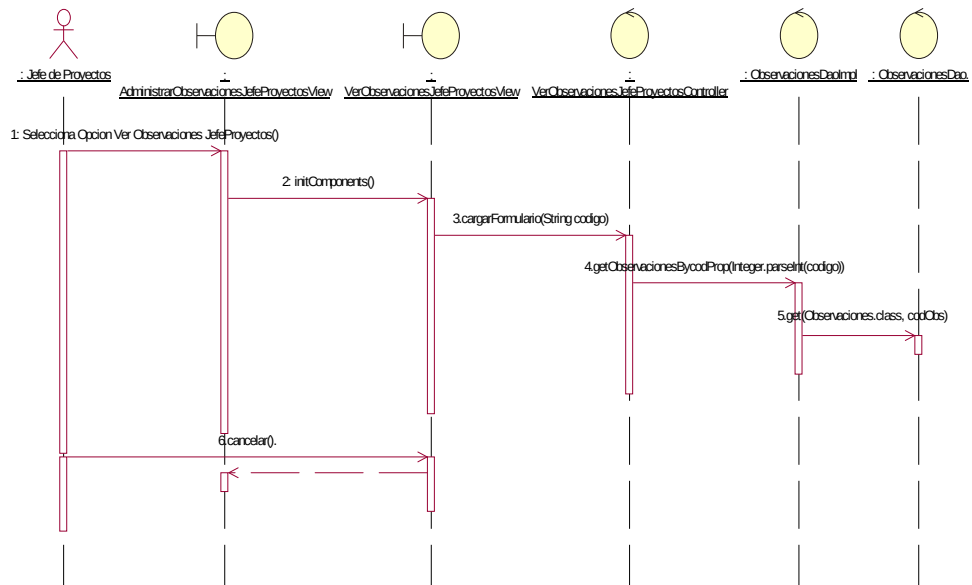


Figura 247: Diagrama de Secuencia: Ver Observaciones Jefe Proyectos

III.2.2.2 .61 Diagrama de Secuencia Dar Alta Trámite

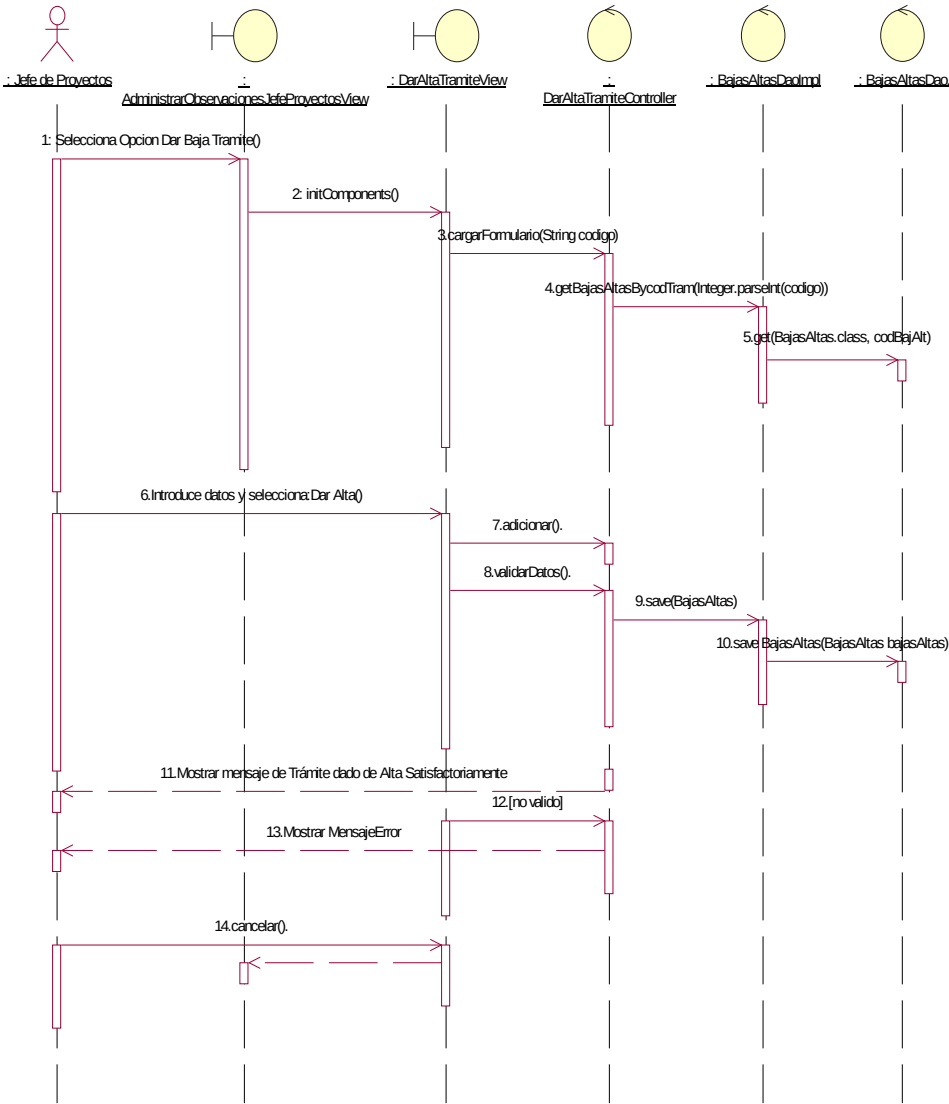


Figura 248: Diagrama de Secuencia: Dar Alta Trámite

III.2.2.2 .62 Diagrama de Secuencia Aprobar Tramite

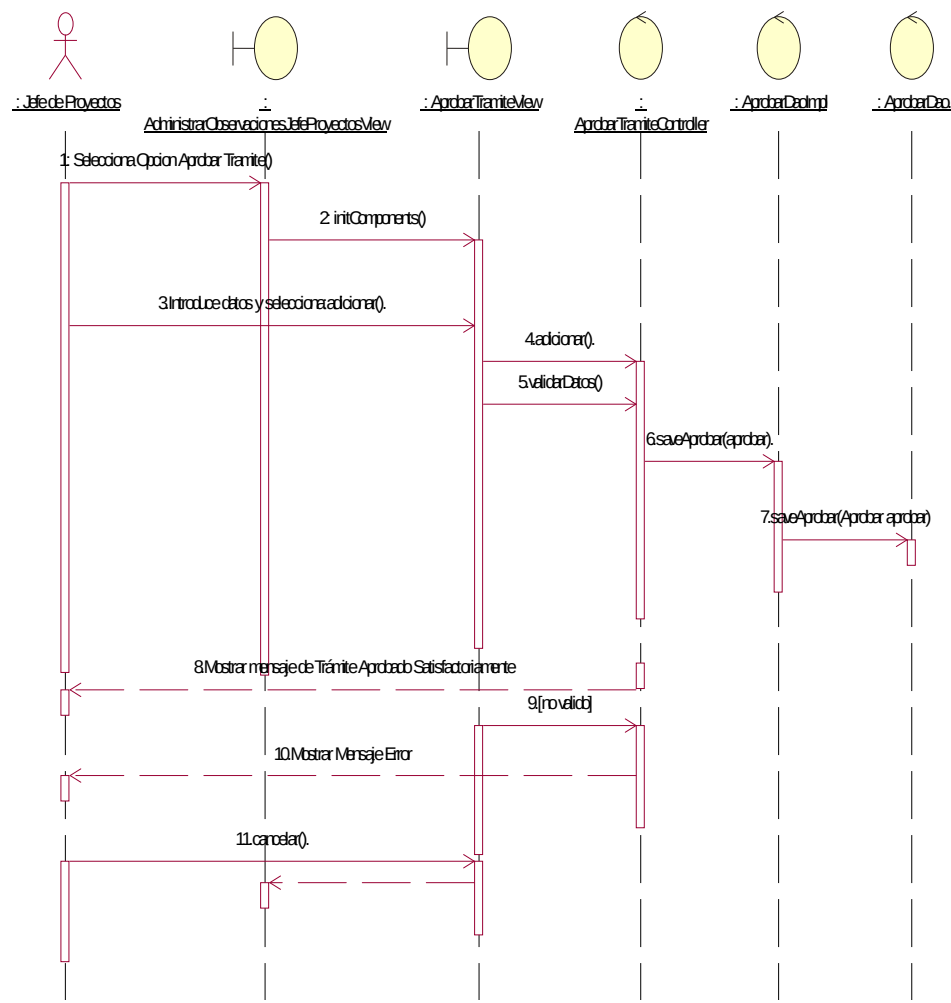


Figura 249: Diagrama de Secuencia: Aprobar Tramite

III.2.2.3 MODULO DE LOTES

III.2.2.3.1 Diagrama de Secuencia Administrar Propietarios

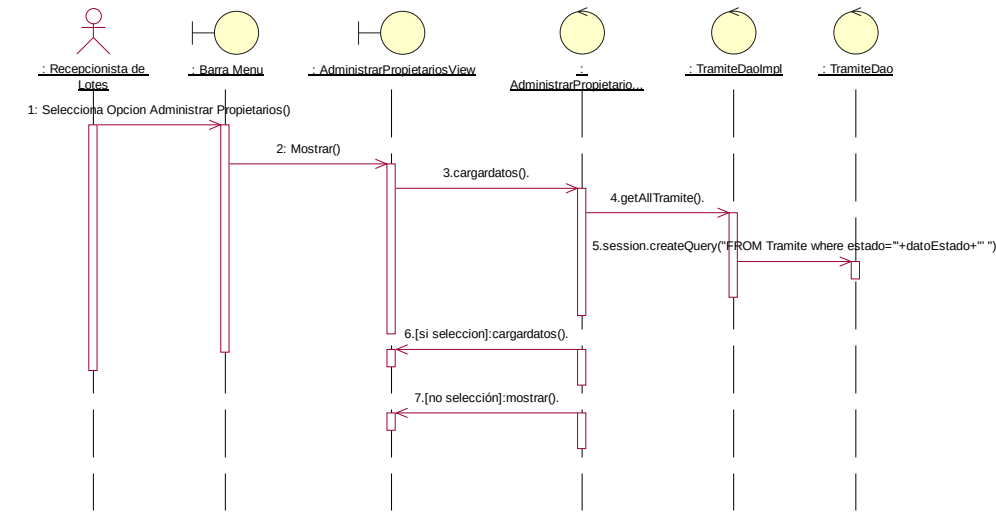


Figura 250: Diagrama de Secuencia: Administrar Propietarios

III.2.2.3.2 Diagrama de Secuencia Adicionar Propietarios

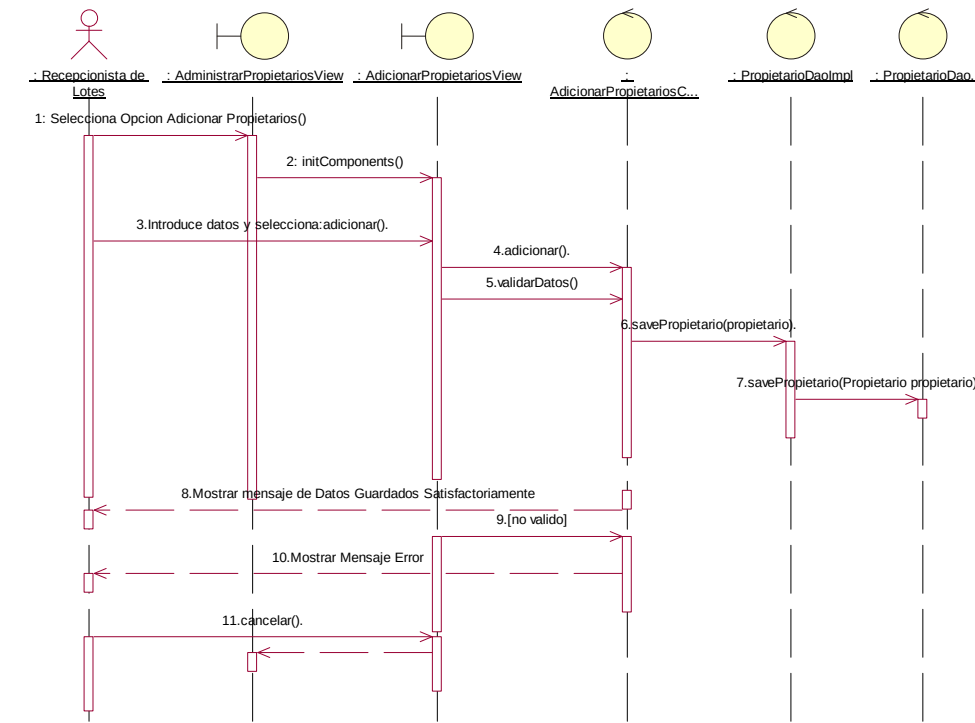


Figura 251: Diagrama de Secuencia: Adicionar Propietarios

III.2.2.3.3 Diagrama de Secuencia Adicionar Projectista

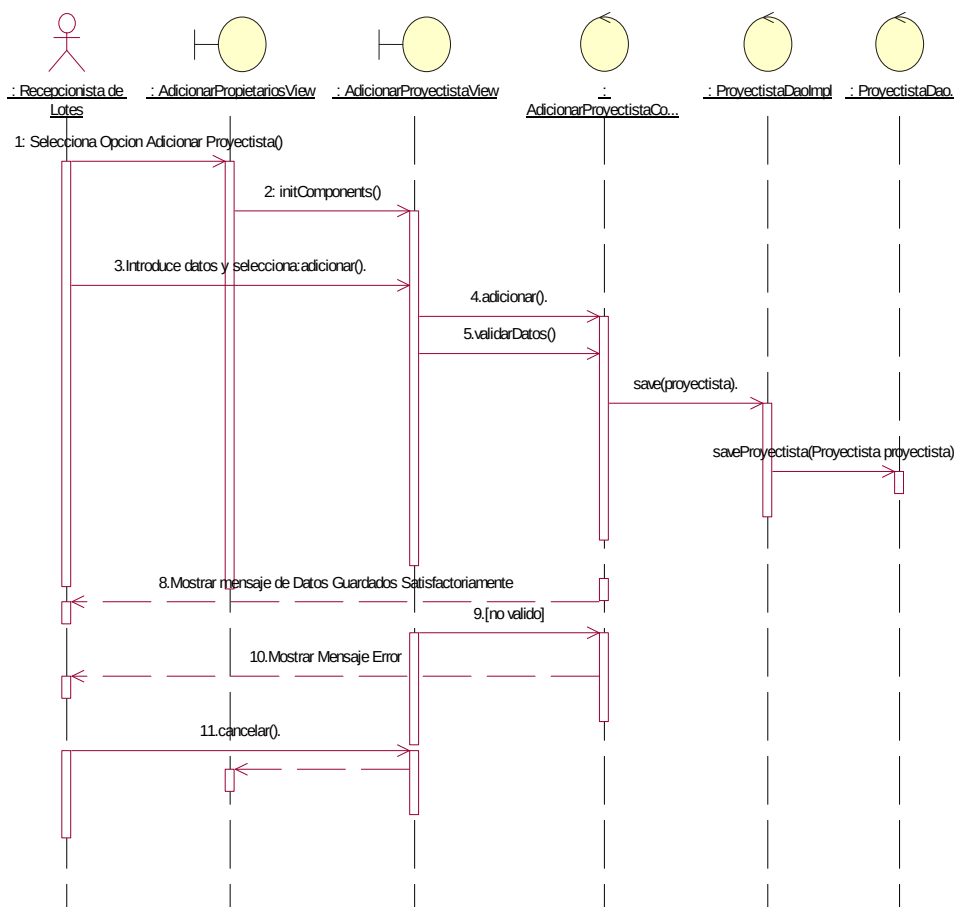


Figura 252: Diagrama de Secuencia: Adicionar Projectista

III.2.2.3.4 Diagrama de Secuencia Adicionar Apoderado

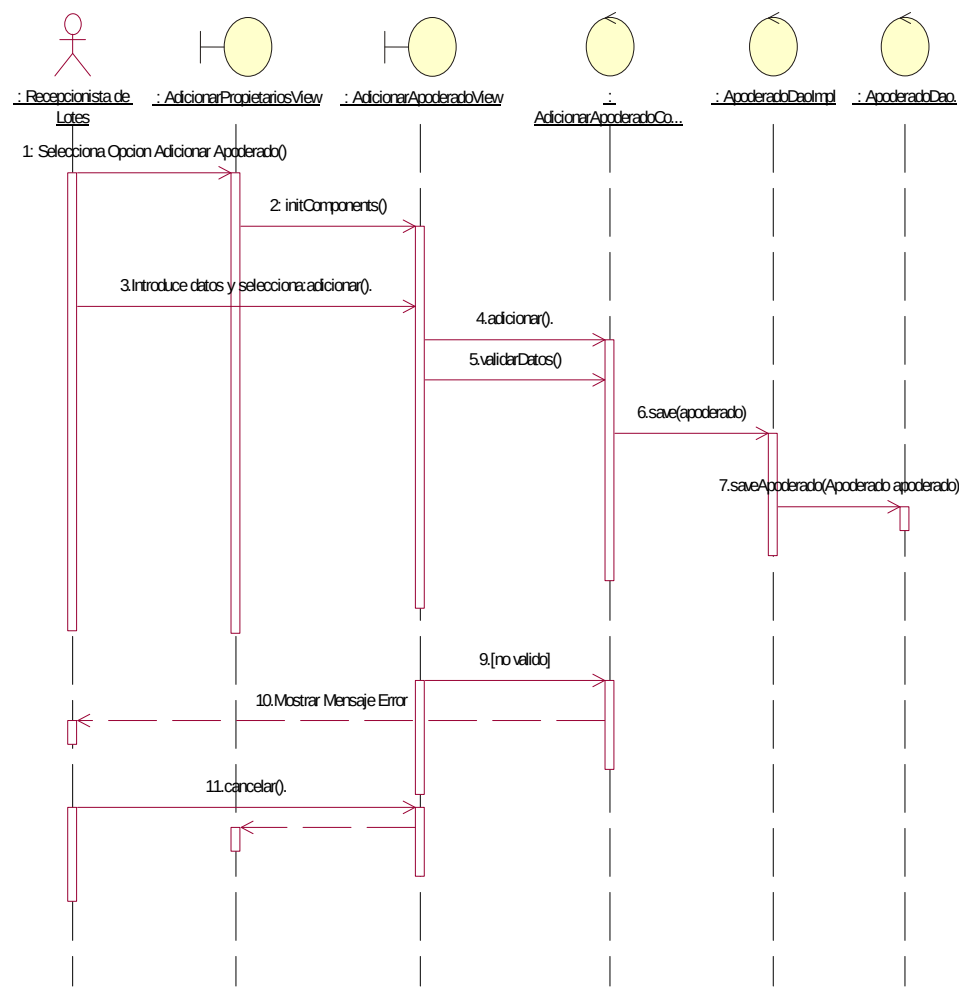


Figura 253: Diagrama de Secuencia: Adicionar Apoderado

III.2.2.3.5 Diagrama de Secuencia Modificar Propietarios

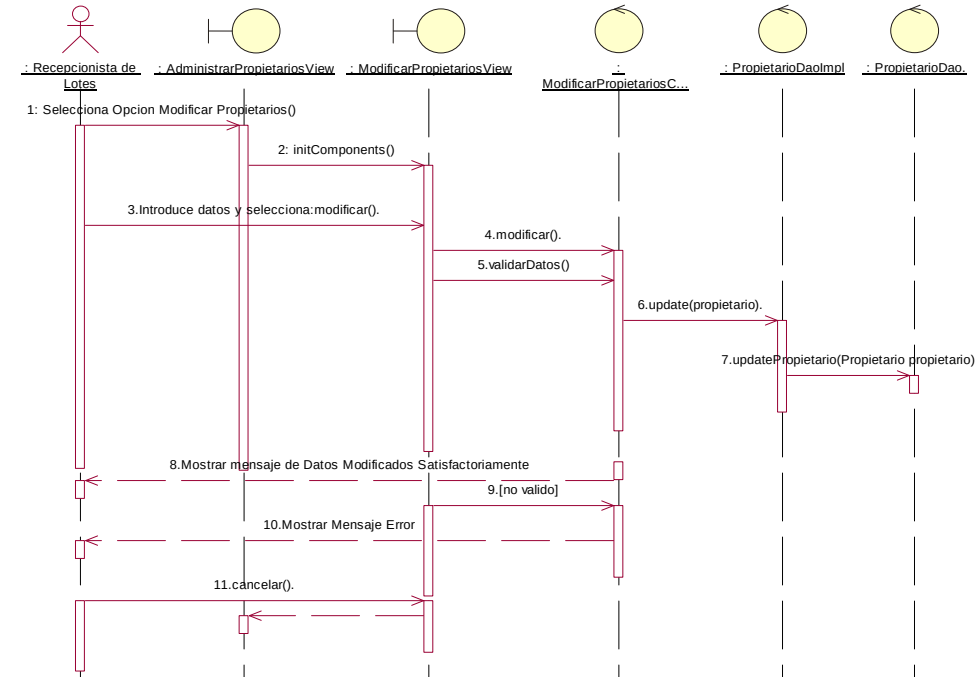


Figura 254: Diagrama de Secuencia: Modificar Propietarios

III.2.2.3.6 Diagrama de Secuencia Modificar Proyectista

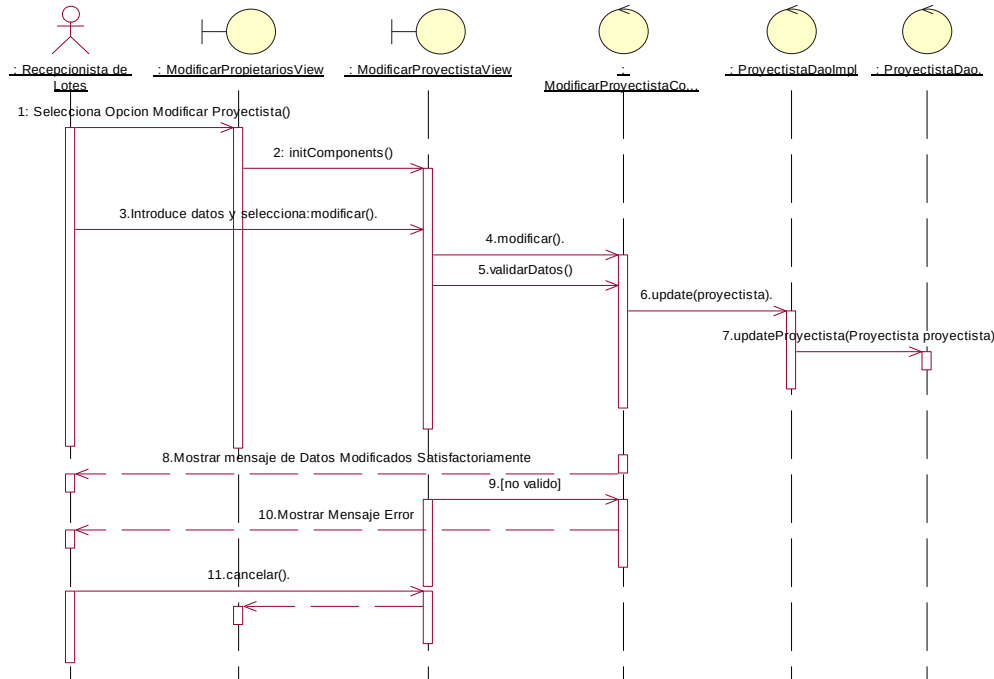


Figura 255: Diagrama de Secuencia: Modificar Proyectista

III.2.2.3.7 Diagrama de Secuencia Modificar Apoderado

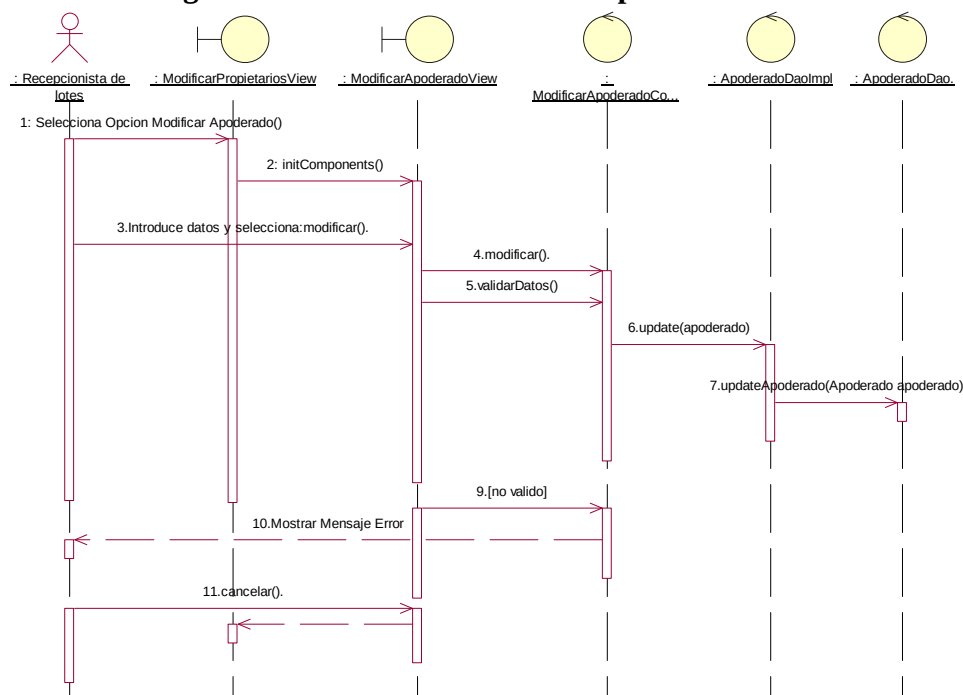


Figura 256: Diagrama de Secuencia: Modificar Apoderado

III.2.2.3.8 Diagrama de Secuencia Ver Propietarios

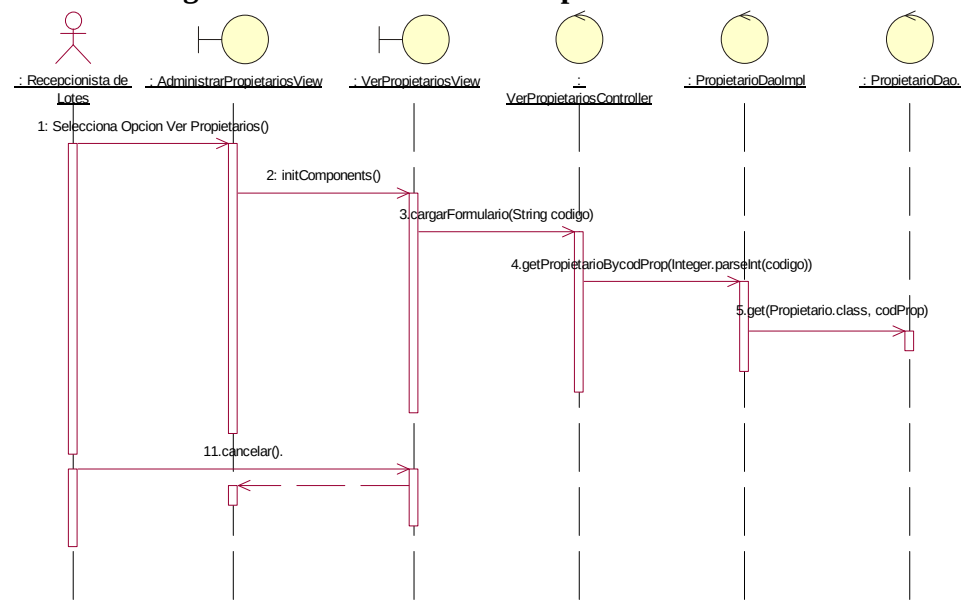


Figura 257: Diagrama de Secuencia: Ver Propietarios

III.2.2.3.9 Diagrama de Secuencia Administrar Observaciones Generales

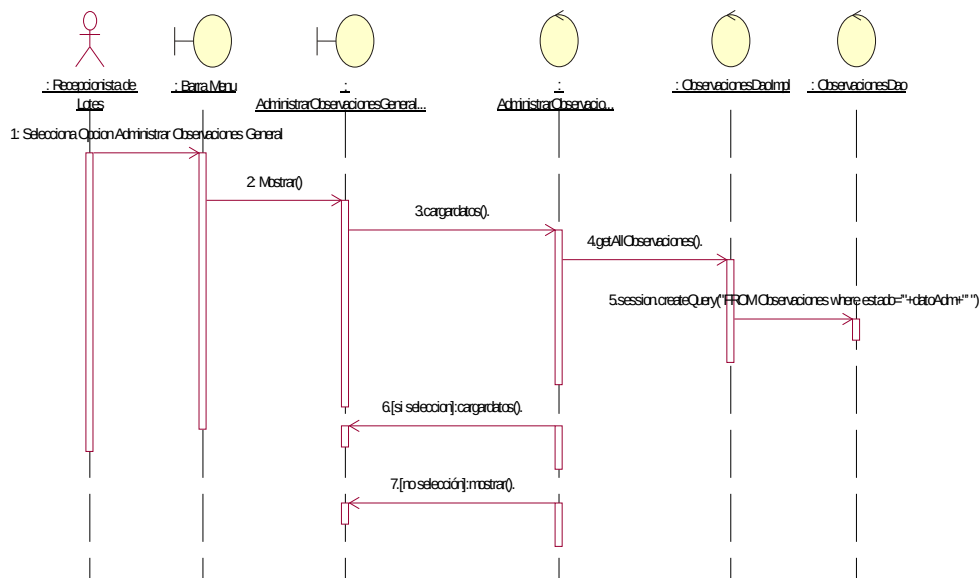


Figura 258: Diagrama de Secuencia: Administrar Observaciones Generales

III.2.2.3.12 Diagrama de Secuencia Ver Observaciones Generales

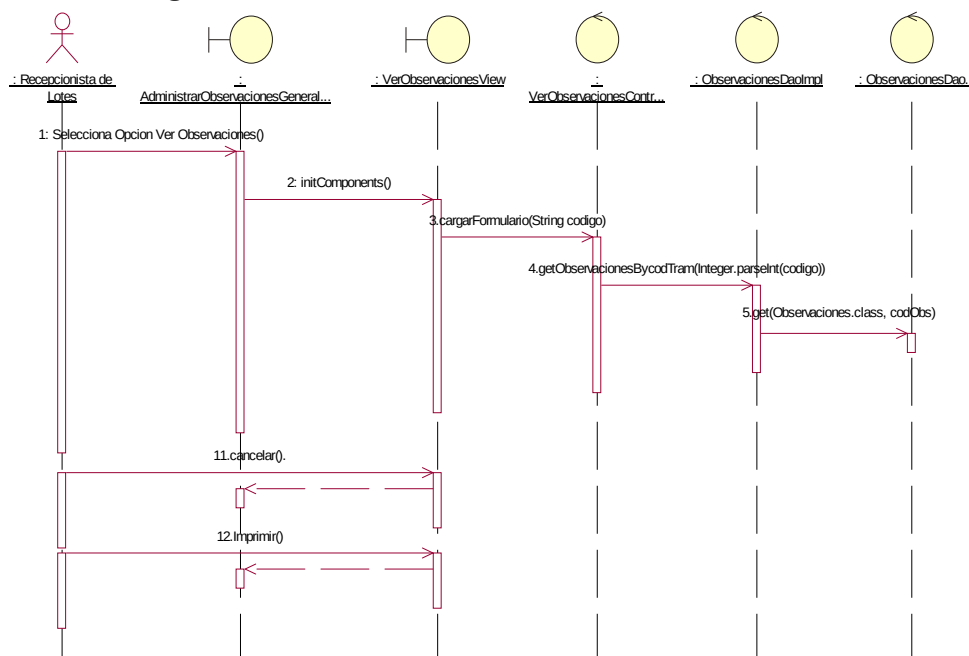


Figura 259: Diagrama de Secuencia: Ver Observaciones Generales

III.2.2.3.12 Diagrama de Secuencia Dar Baja Trámite

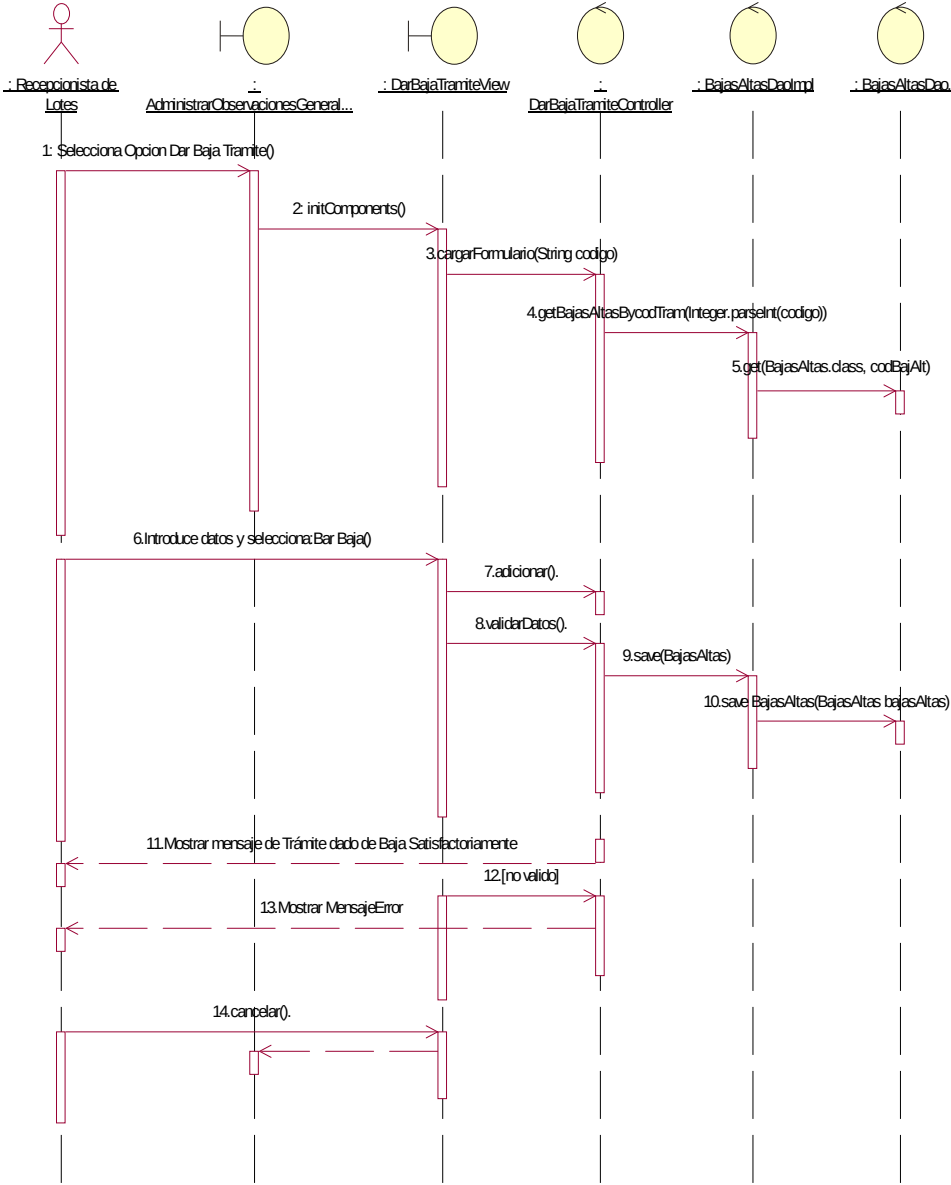


Figura 260: Diagrama de Secuencia: Dar Baja Trámite

III.2.2.3.12 Diagrama de Secuencia Reasignar Trámite

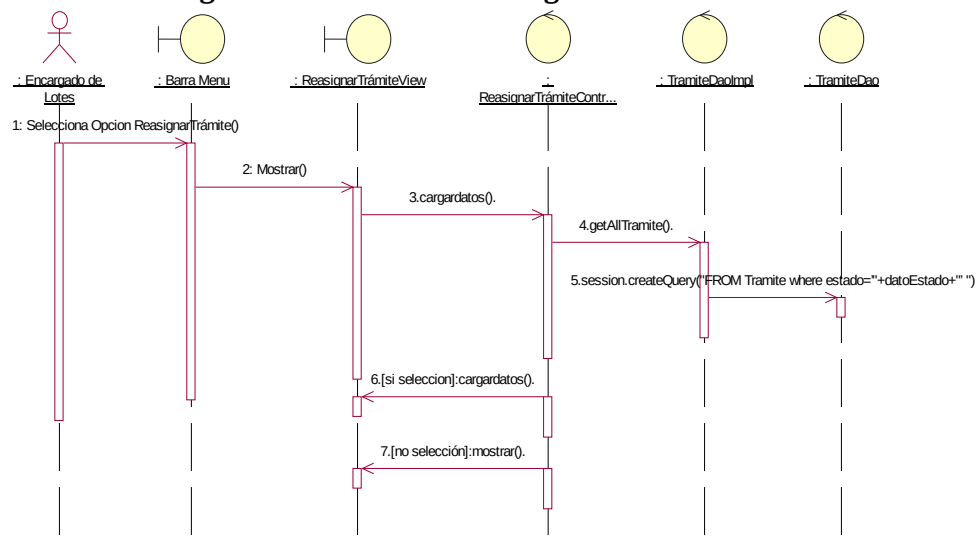


Figura 261: Diagrama de Secuencia: Reasignar Trámite

III.2.2.3.13 Diagrama de Secuencia Registrar Tramite

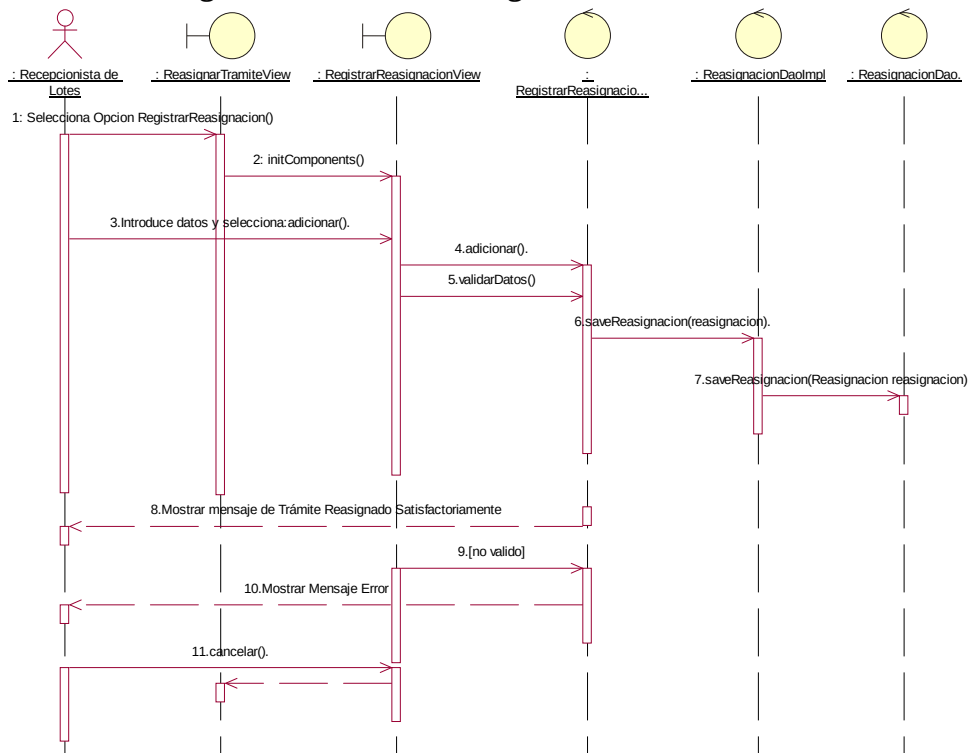


Figura 262: Diagrama de Secuencia: Registrar Tramite

III.2.2.3.14 Diagrama de Secuencia Ver Reasignación

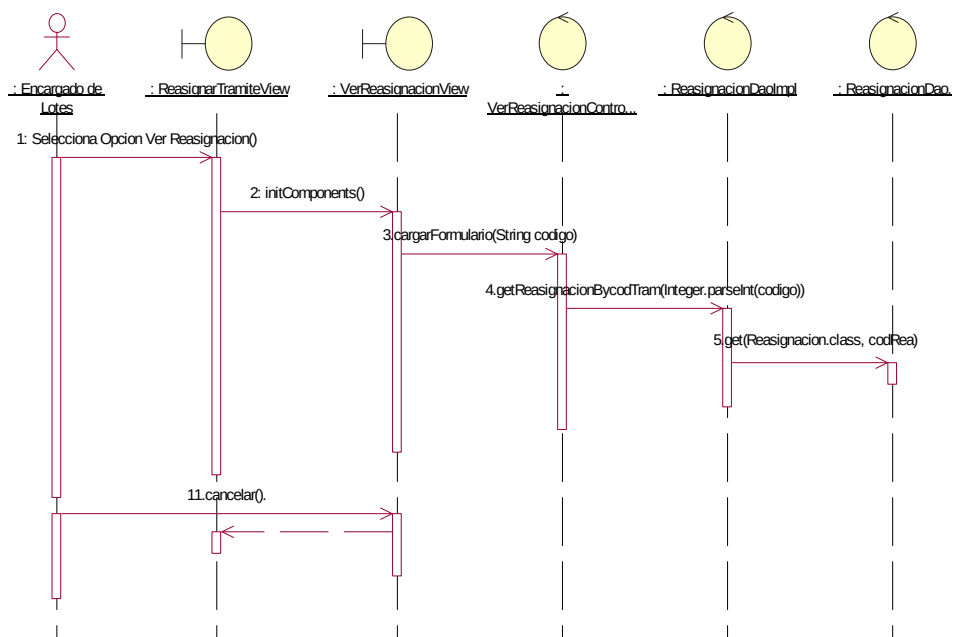


Figura 263: Diagrama de Secuencia: Ver Reasignación

III.2.2.3.15 Diagrama de Secuencia Ver Datos Trámite

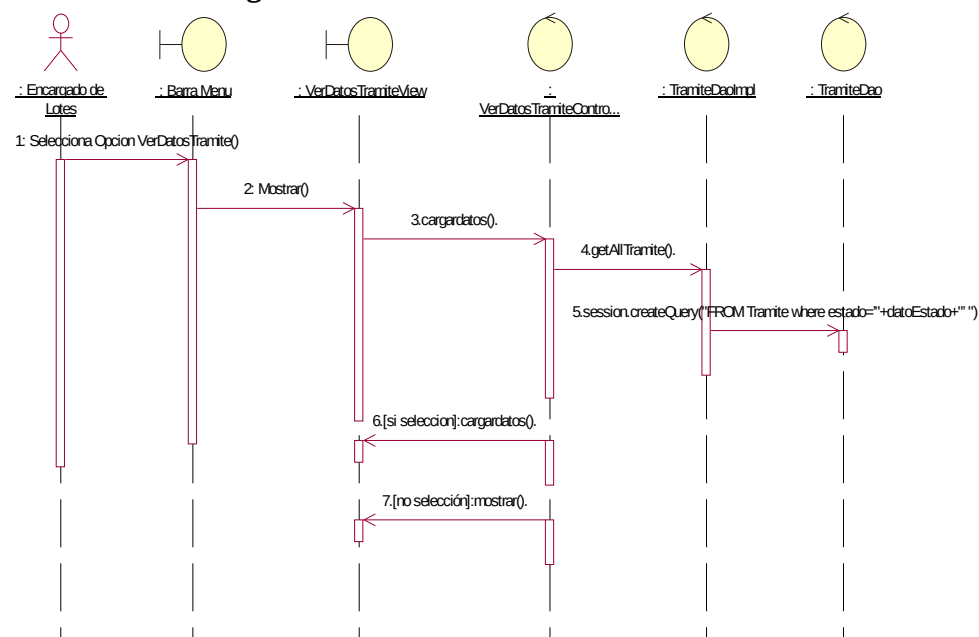


Figura 264: Diagrama de Secuencia: Ver Datos Trámite

III.2.2.3.16 Diagrama de Secuencia Ver Estado Tramite

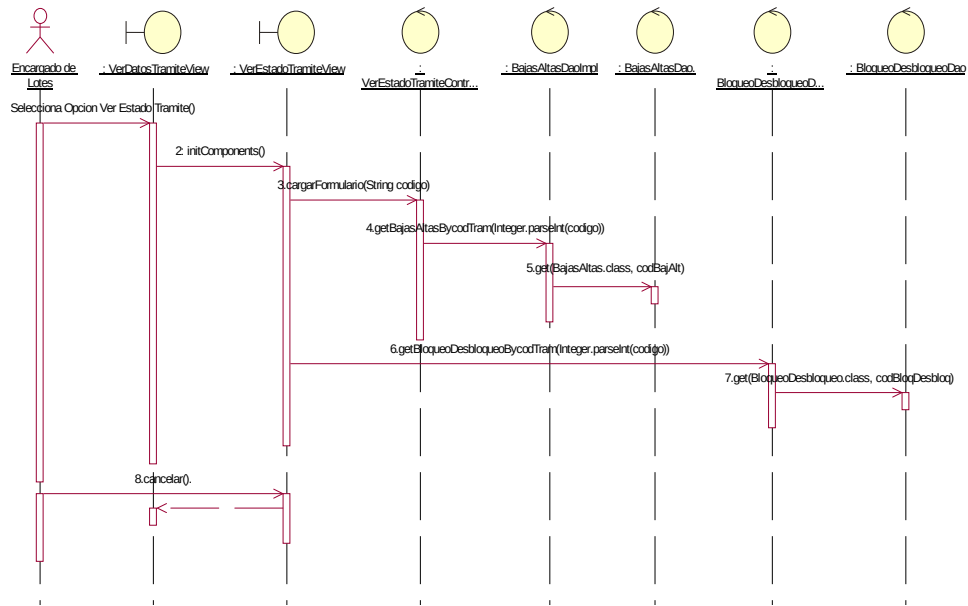


Figura 265: Diagrama de Secuencia: Ver Estado Tramite

III.2.2.3.17 Diagrama de Secuencia Ver Detalle Observación

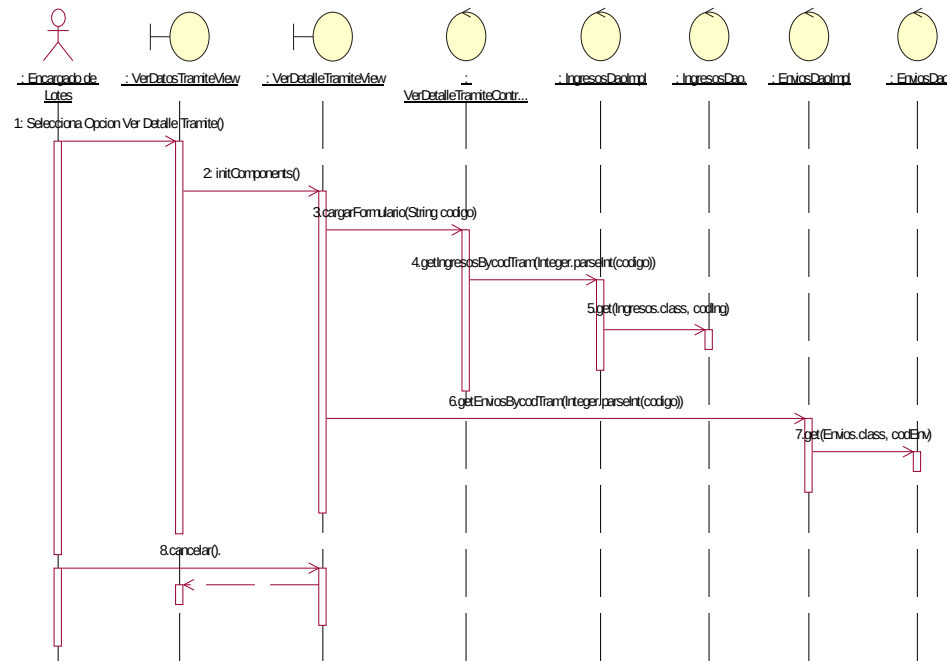


Figura 266: Diagrama de Secuencia: Ver Detalle Observación

III.2.2.3.18 Diagrama de Secuencia Administrar Ingresos

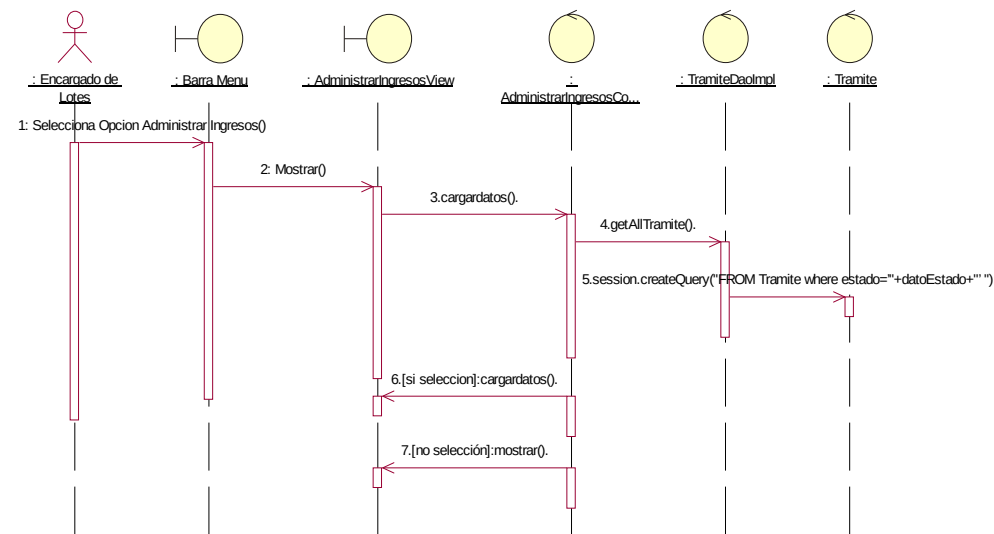


Figura 267: Diagrama de Secuencia: Administrar Ingresos

III.2.2.3.19 Diagrama de Secuencia Registrar Ingresos

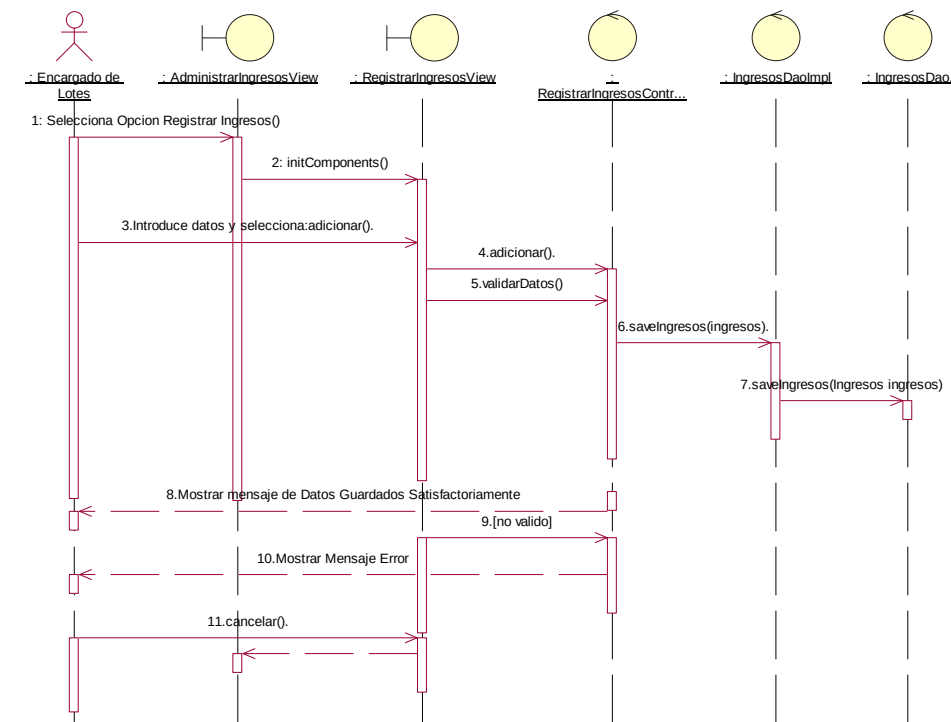


Figura 268: Diagrama de Secuencia: Registrar Ingresos

III.2.2.3.20 Diagrama de Secuencia Modificar Ingresos

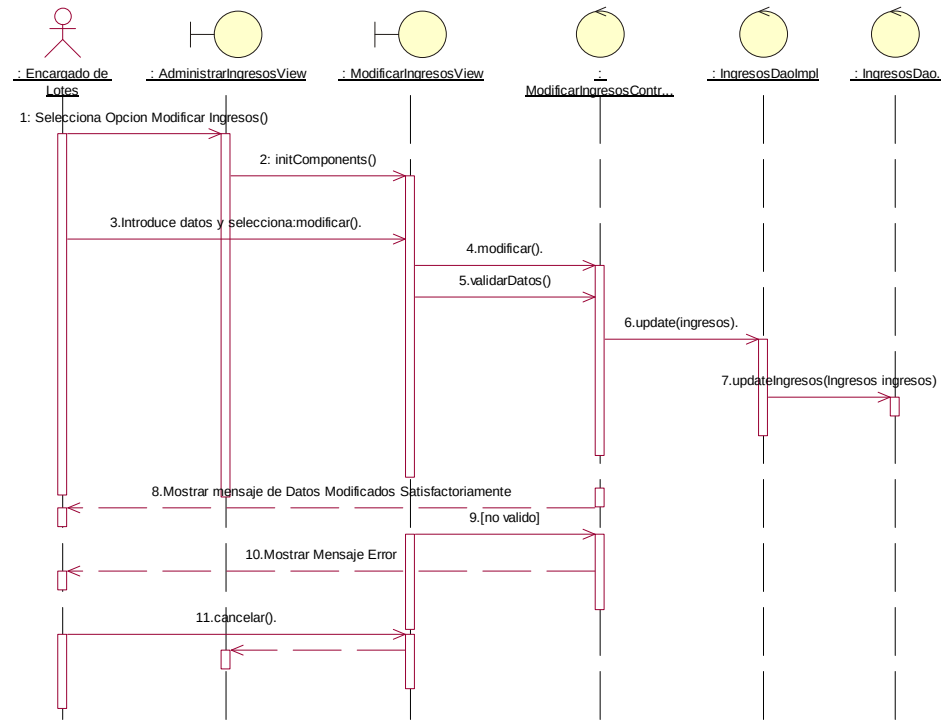


Figura 269: Diagrama de Secuencia: Modificar Ingresos

III.2.2.3.21 Diagrama de Secuencia Ver Ingresos

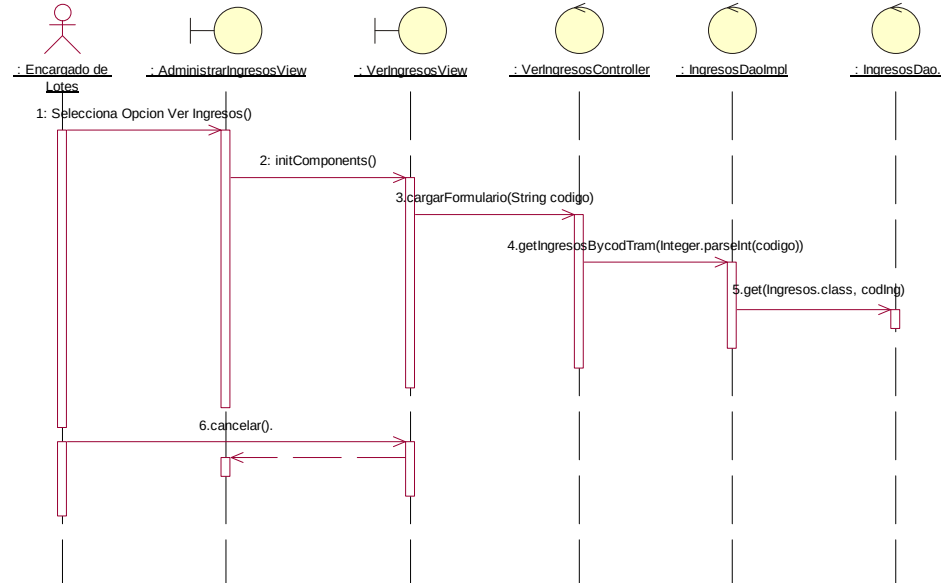


Figura 270 Diagrama de Secuencia: Ver Ingresos

III.2.2.3.22 Diagrama de Secuencia Administrar Salidas

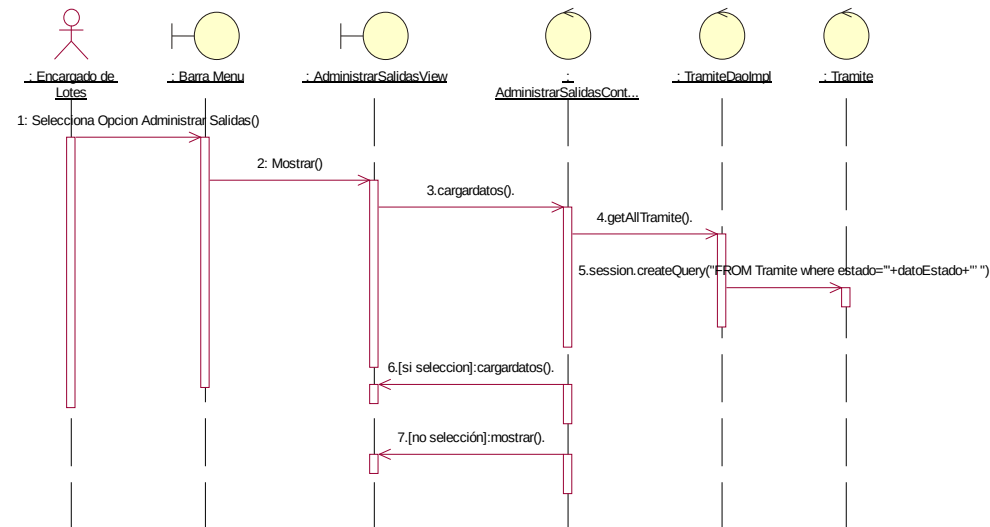


Figura 271: Diagrama de Secuencia: Administrar Salidas

III.2.2.3.23 Diagrama de Secuencia Registrar Salidas

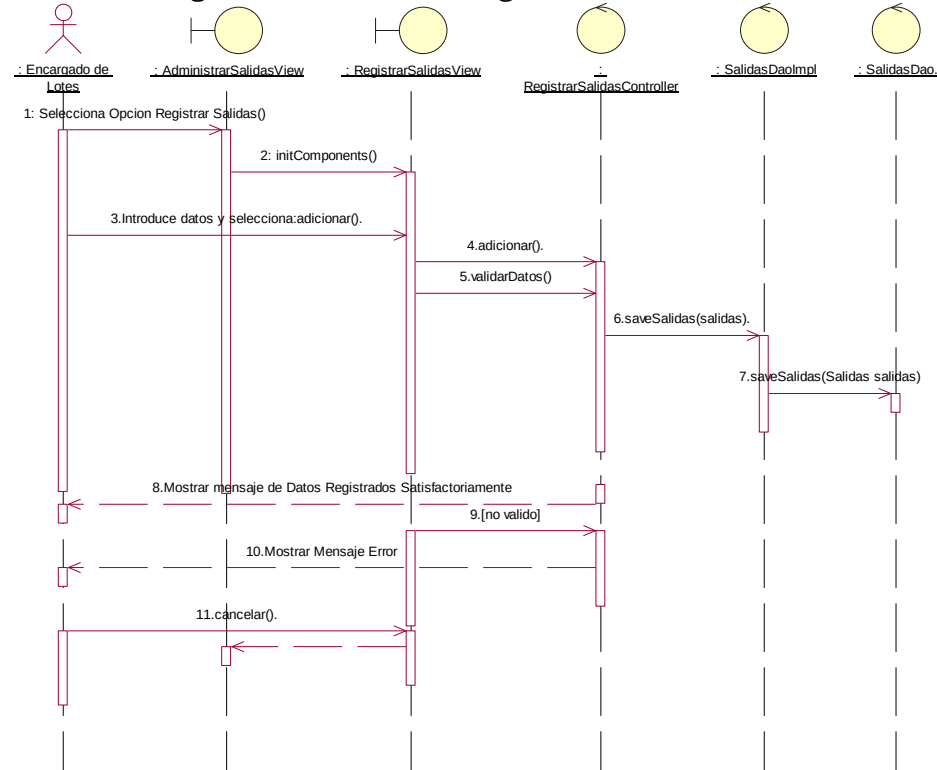


Figura 272: Diagrama de Secuencia: Registrar Salidas

III.2.2.3.24 Diagrama de Secuencia Modificar Salidas

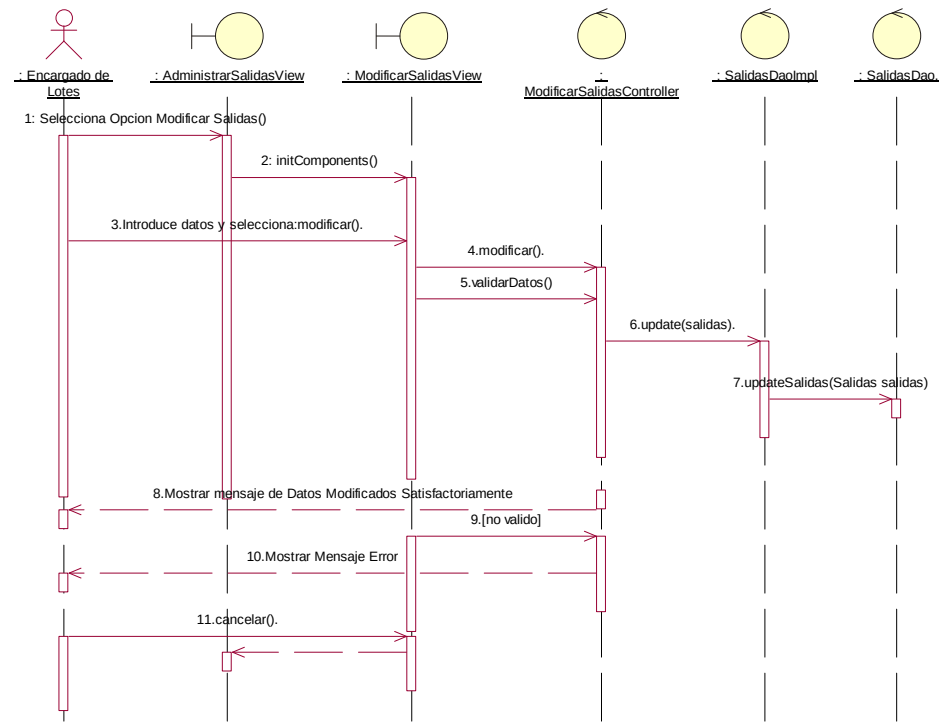


Figura 273: Diagrama de Secuencia: Modificar Salidas

III.2.2.3.25 Diagrama de Secuencia Ver Salidas

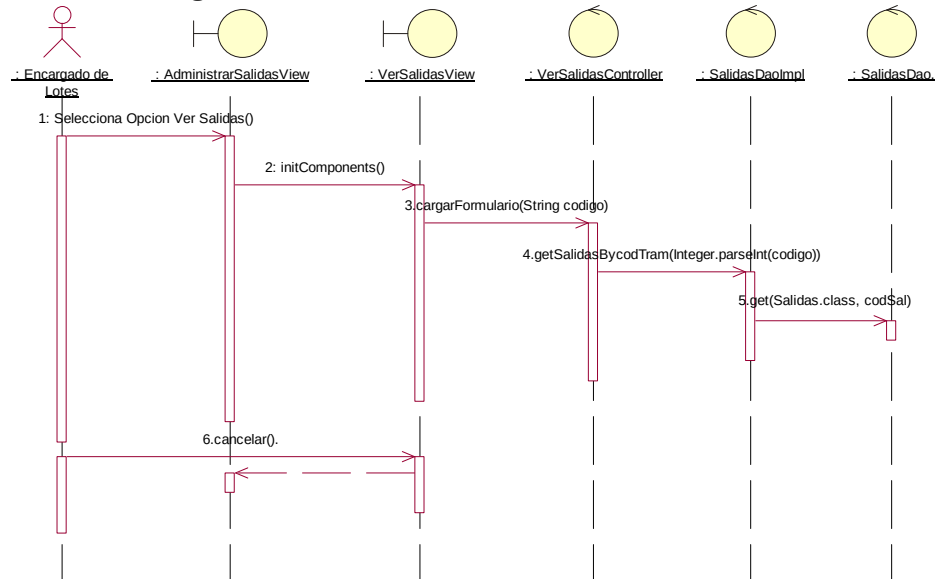


Figura 274: Diagrama de Secuencia: Ver Salidas

III.2.2.3.26 Diagrama de Secuencia Administrar Envíos

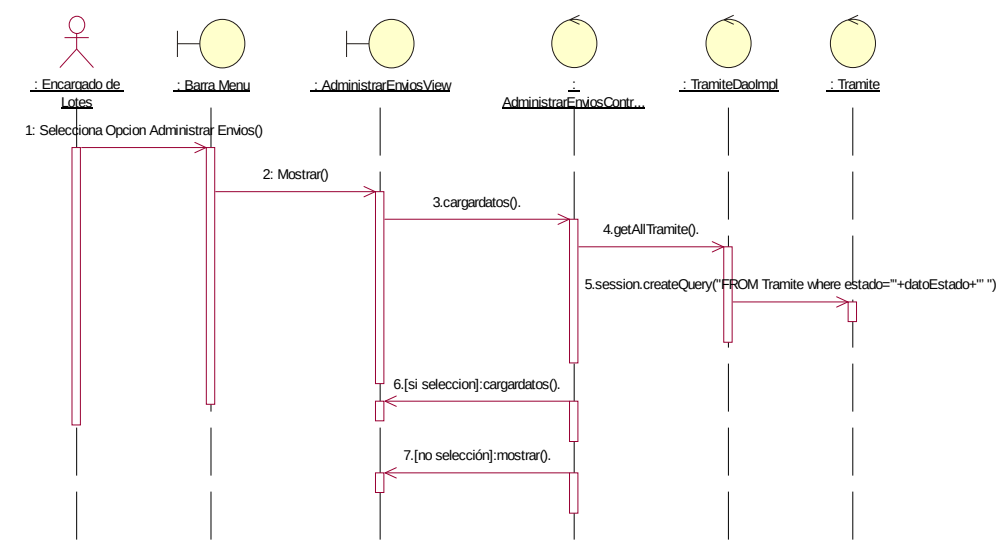


Figura 275: Diagrama de Secuencia: Administrar Envíos

III.2.2.3.27 Diagrama de Secuencia Registrar Envíos

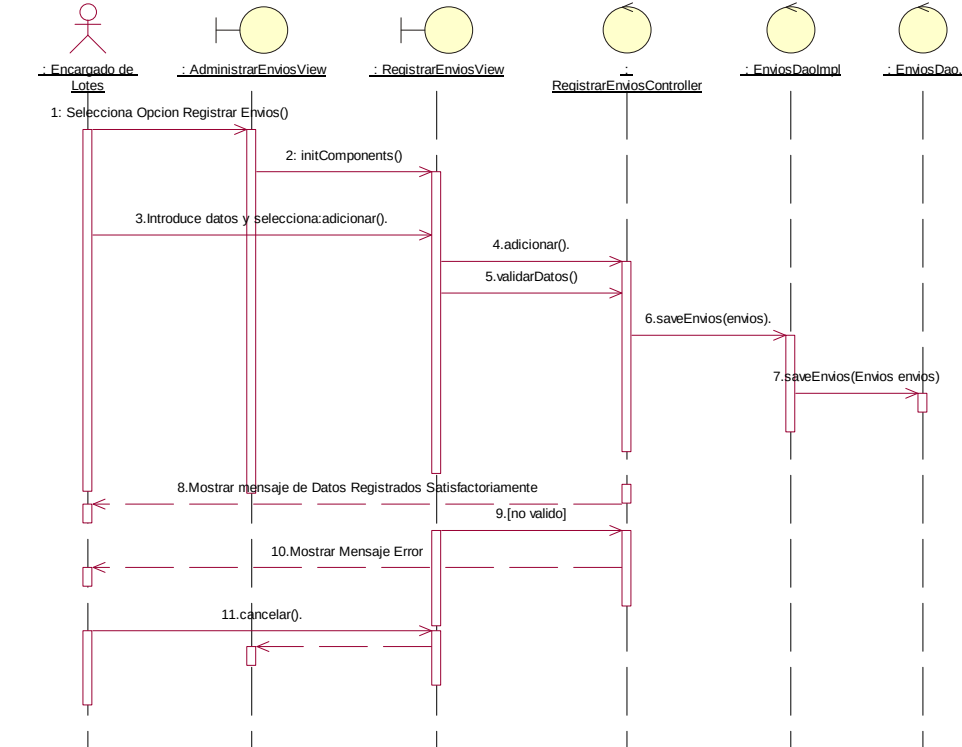


Figura 276: Diagrama de Secuencia: Registrar Envíos

III.2.2.3.28 Diagrama de Secuencia Modificar Envíos

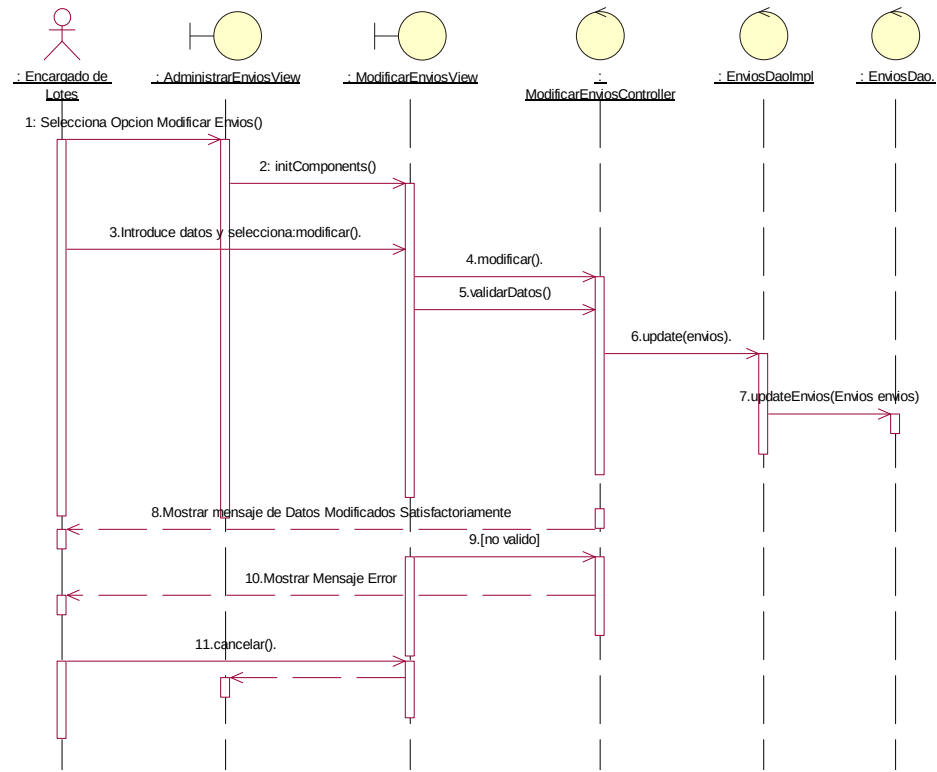


Figura 277: Diagrama de Secuencia: Modificar Envíos

III.2.2.3.29 Diagrama de Secuencia Ver Envíos

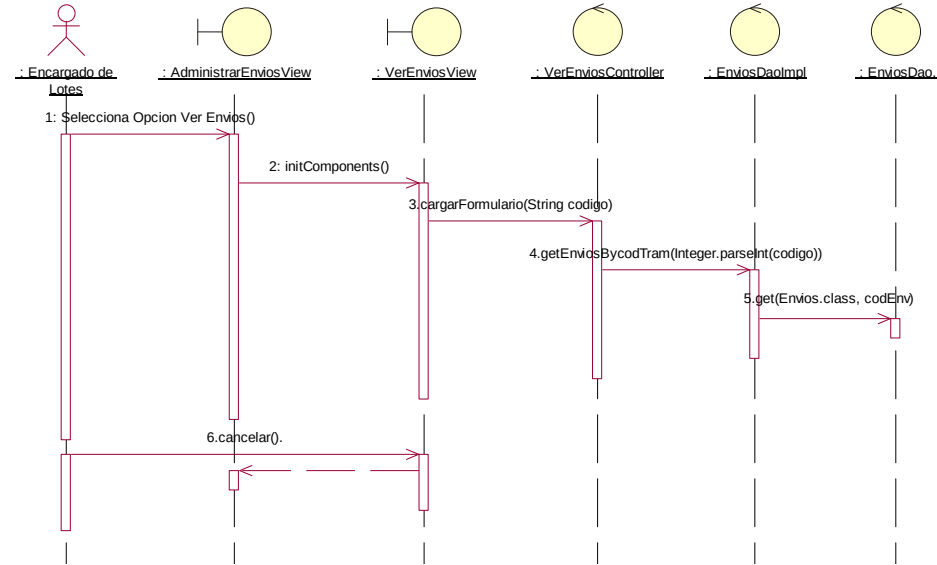


Figura 278: Diagrama de Secuencia: Ver Envíos

III.2.2.3.30 Diagrama de Secuencia Administrar Recepción

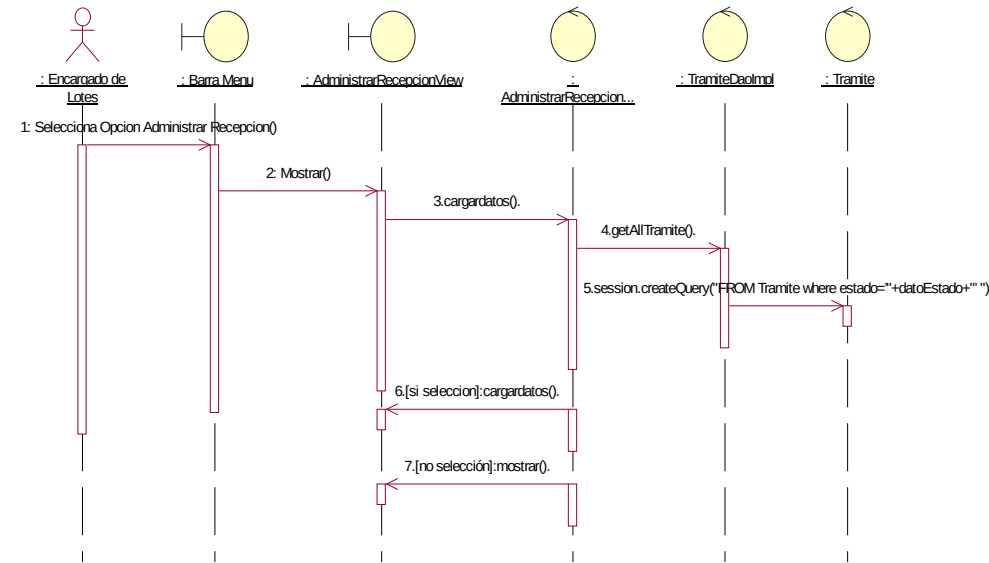


Figura 279: Diagrama de Secuencia: Administrar Recepción

III.2.2.3.31 Diagrama de Secuencia Registrar Recepción

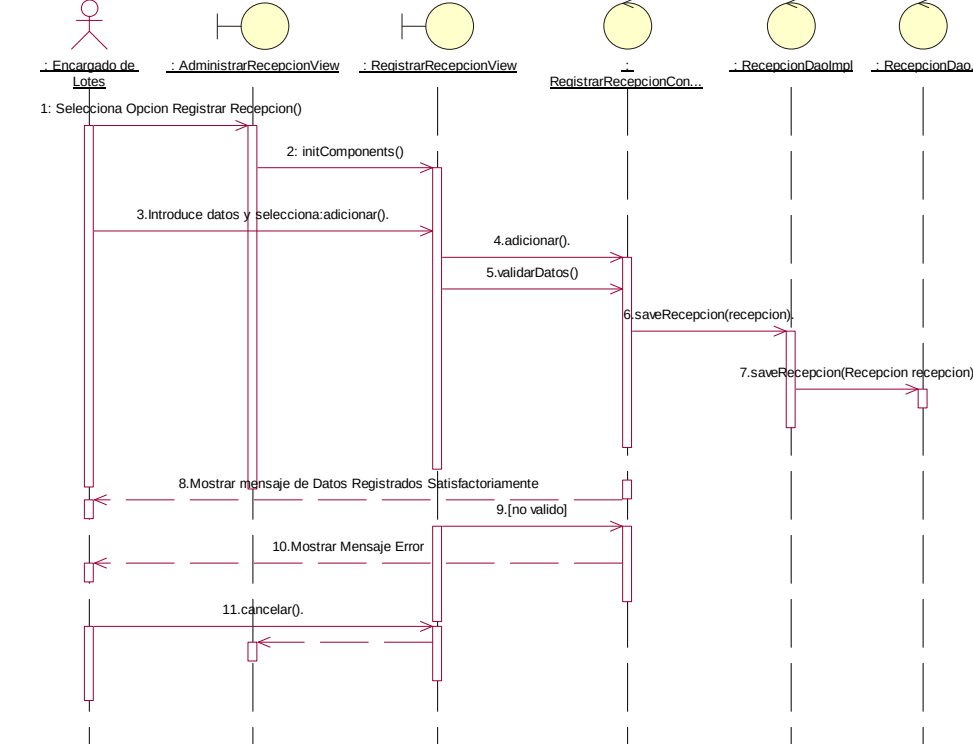


Figura 280: Diagrama de Secuencia: Registrar Recepción

III.2.2.3.32 Diagrama de Secuencia Modificar Recepción

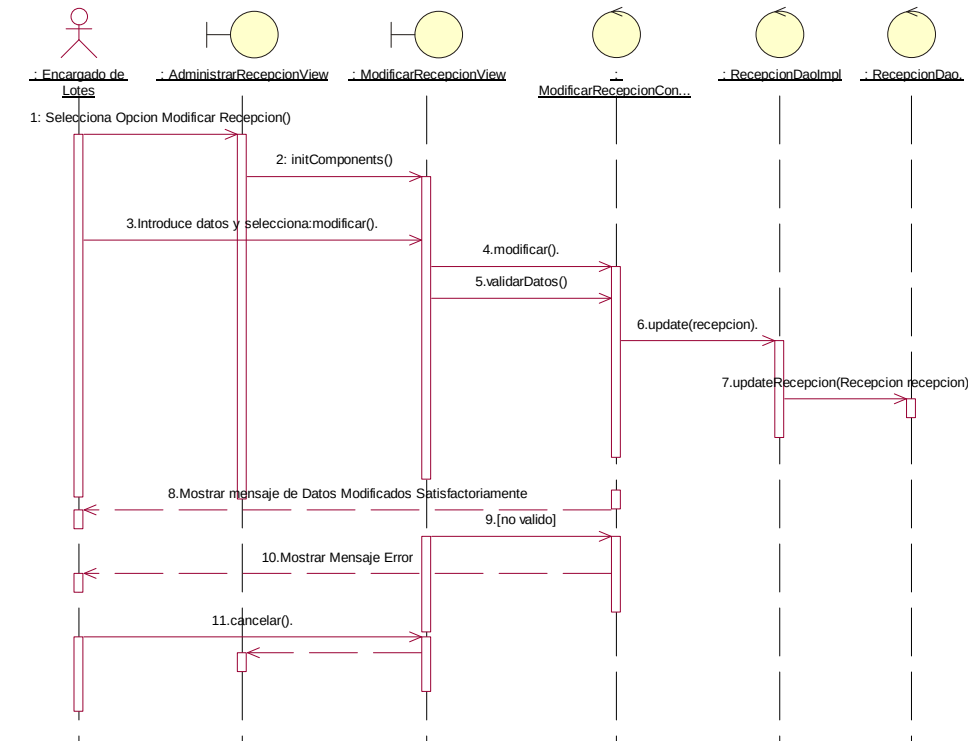


Figura 281: Diagrama de Secuencia: Modificar Recepción

III.2.2.3.33 Diagrama de Secuencia Ver Recepción

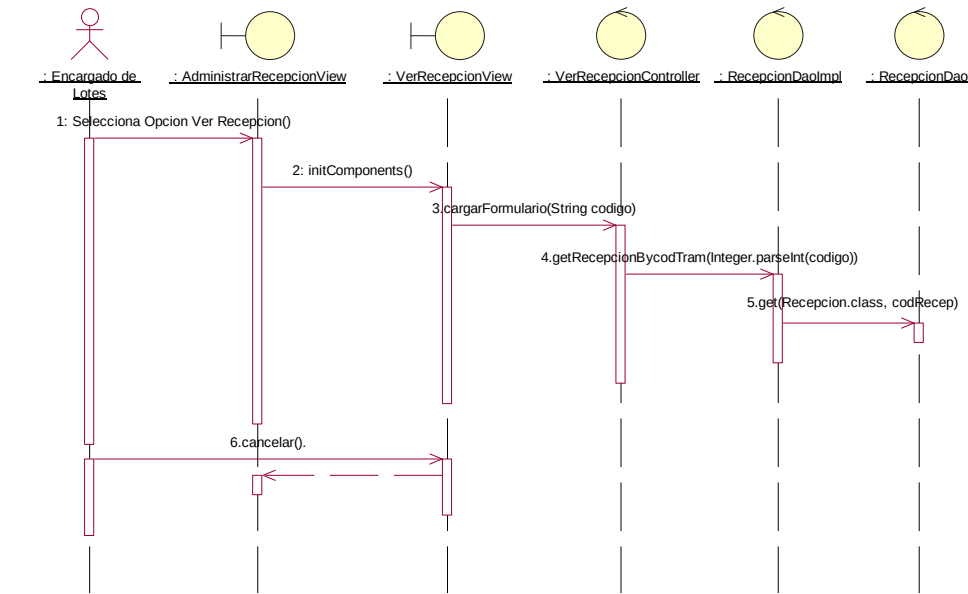


Figura 282: Diagrama de Secuencia: Ver Recepción

III.2.2.3.34 Diagrama de Secuencia Asignar Numero Oficial

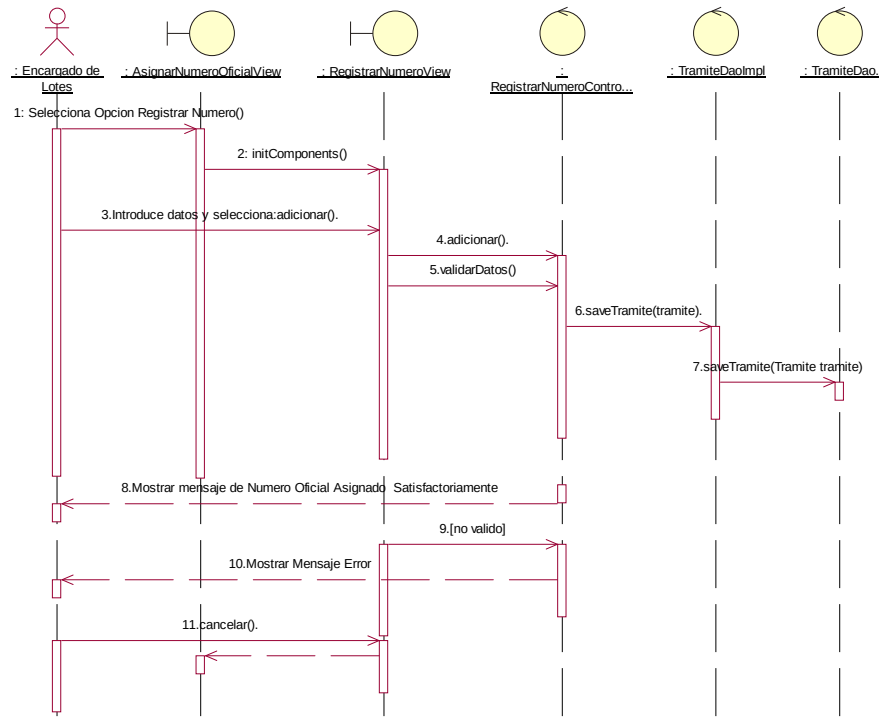


Figura 283: Diagrama de Secuencia: Asignar Numero Oficial

III.2.2.3.35 Diagrama de Secuencia Asignar Personal Trámite

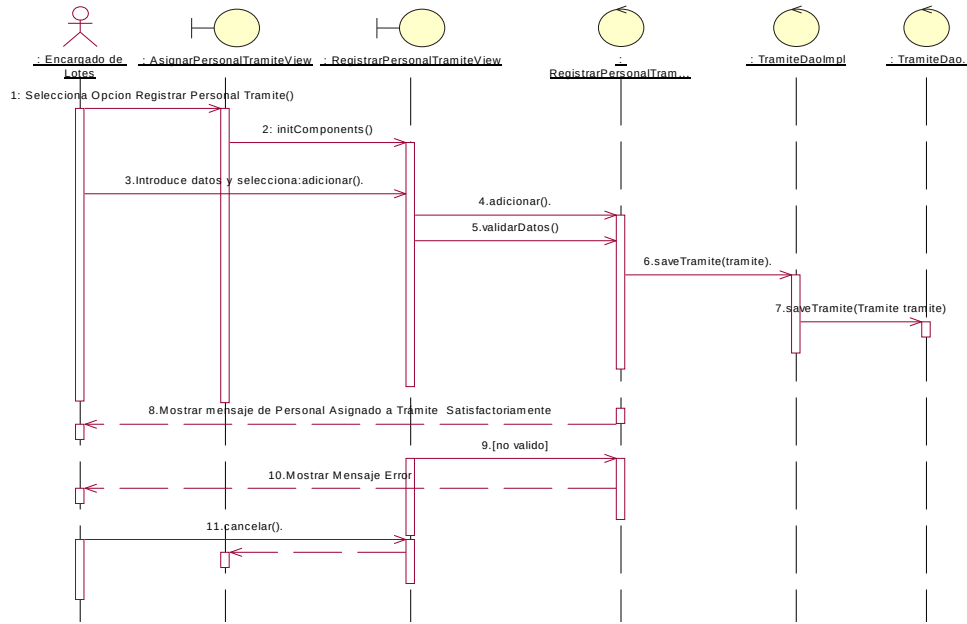


Figura 284: Diagrama de Secuencia: Asignar Personal Trámite

III.2.2.2 .36 Diagrama de Secuencia Listar Mensajes

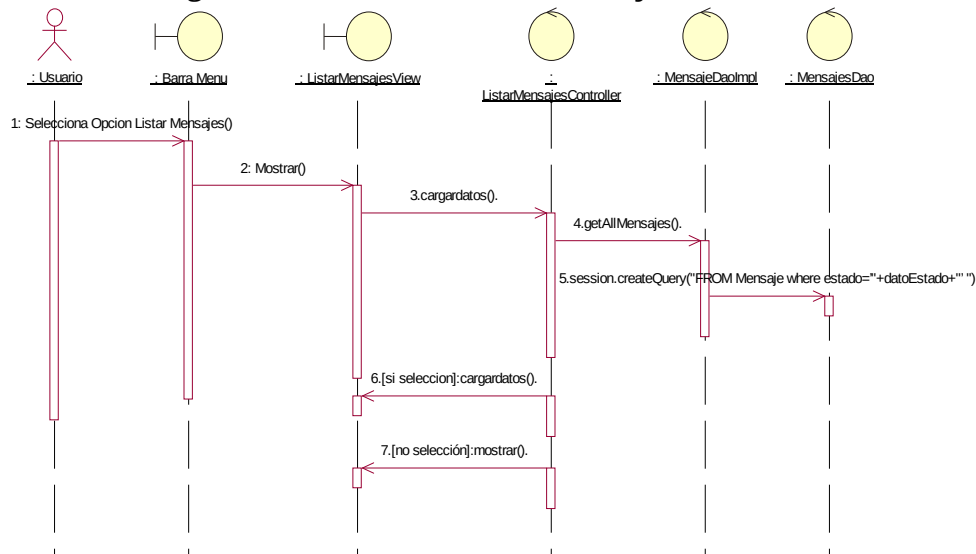


Figura 285: Diagrama de Secuencia: Listar Mensajes

III.2.2.2 .38 Diagrama de Secuencia Enviar Mensaje

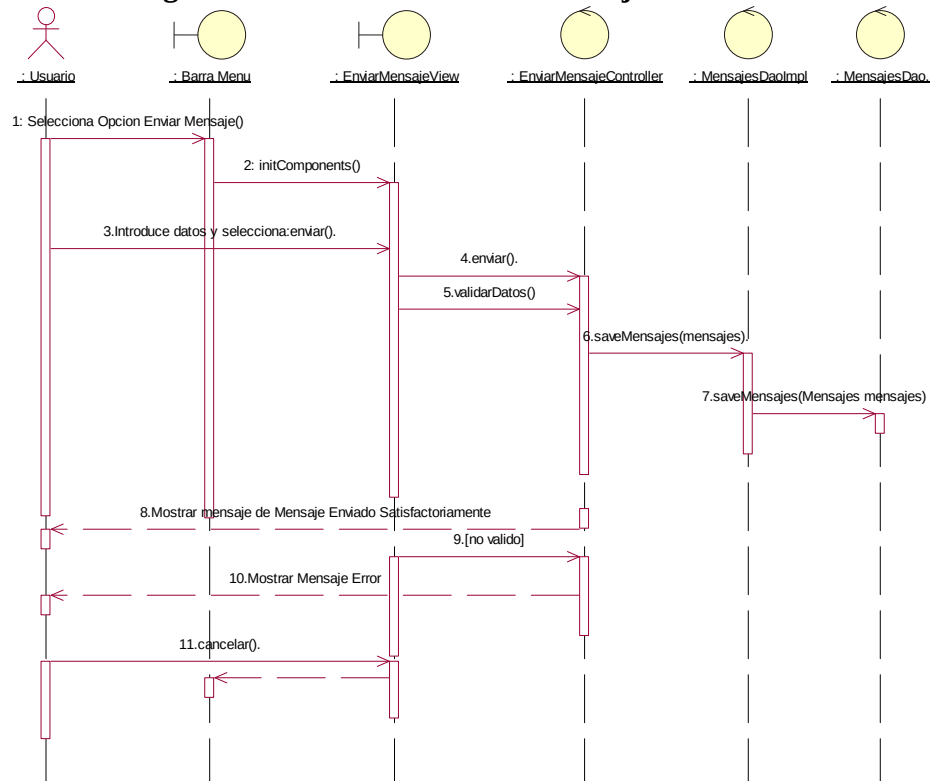


Figura 286: Diagrama de Secuencia: Enviar Mensaje

III.2.2.2. 39 Diagrama de Secuencia Eliminar Mensaje

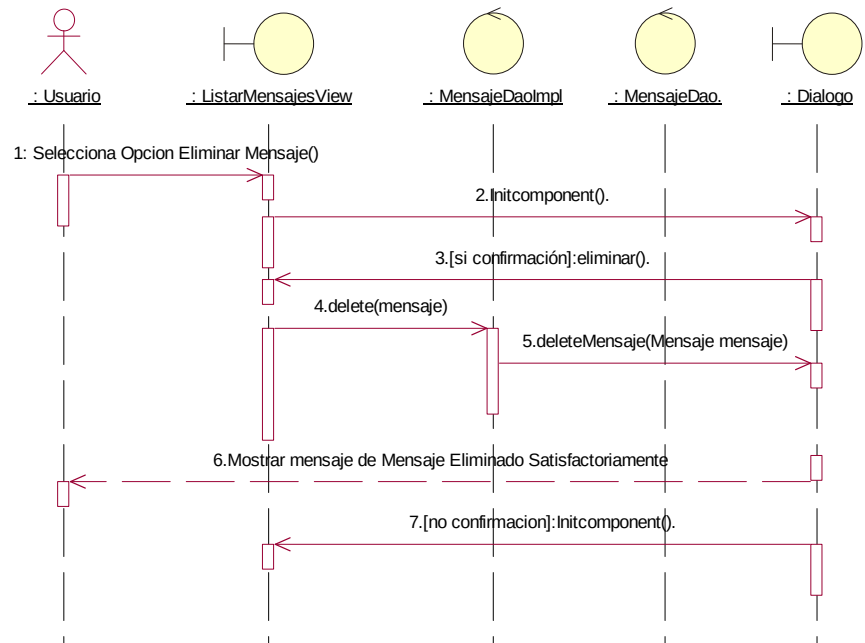


Figura 287: Diagrama de Secuencia: Eliminar Mensaje

III.2.2.2 .40 Diagrama de Secuencia Ver Mensaje

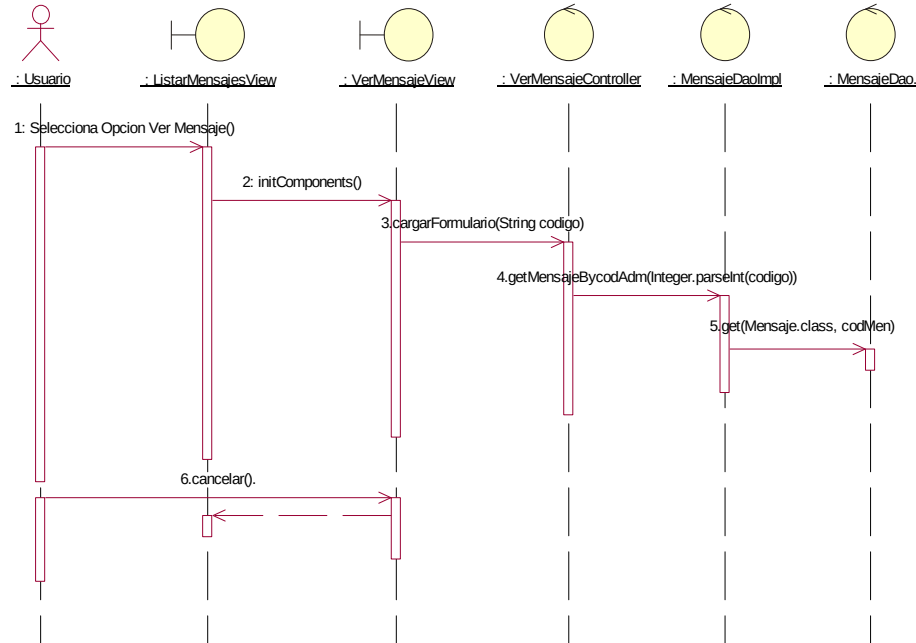


Figura 288: Diagrama de Secuencia: Ver Mensaje

III.2.2.3.40 Diagrama de Secuencia Administrar Observaciones

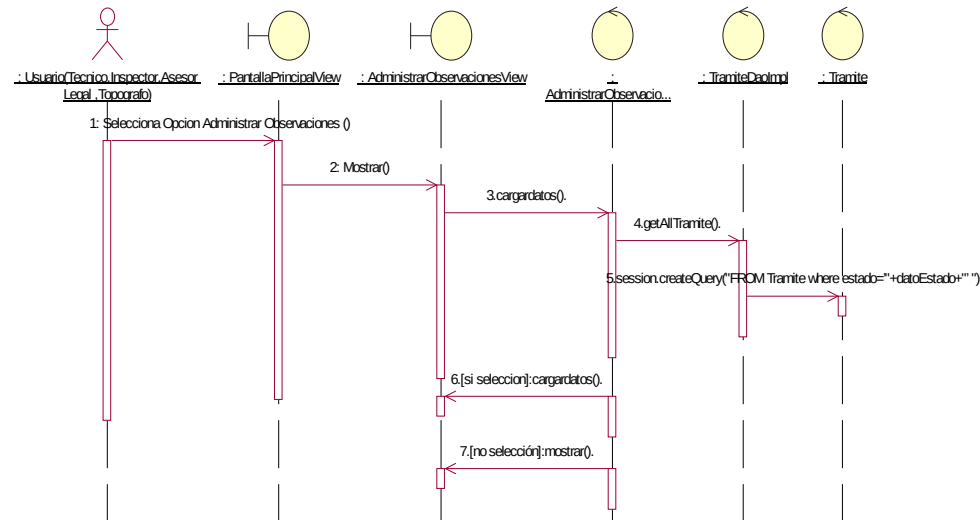


Figura 289: Diagrama de Secuencia: Adicionar Observaciones

III.2.2.3.41 Diagrama de Secuencia Adicionar Observaciones

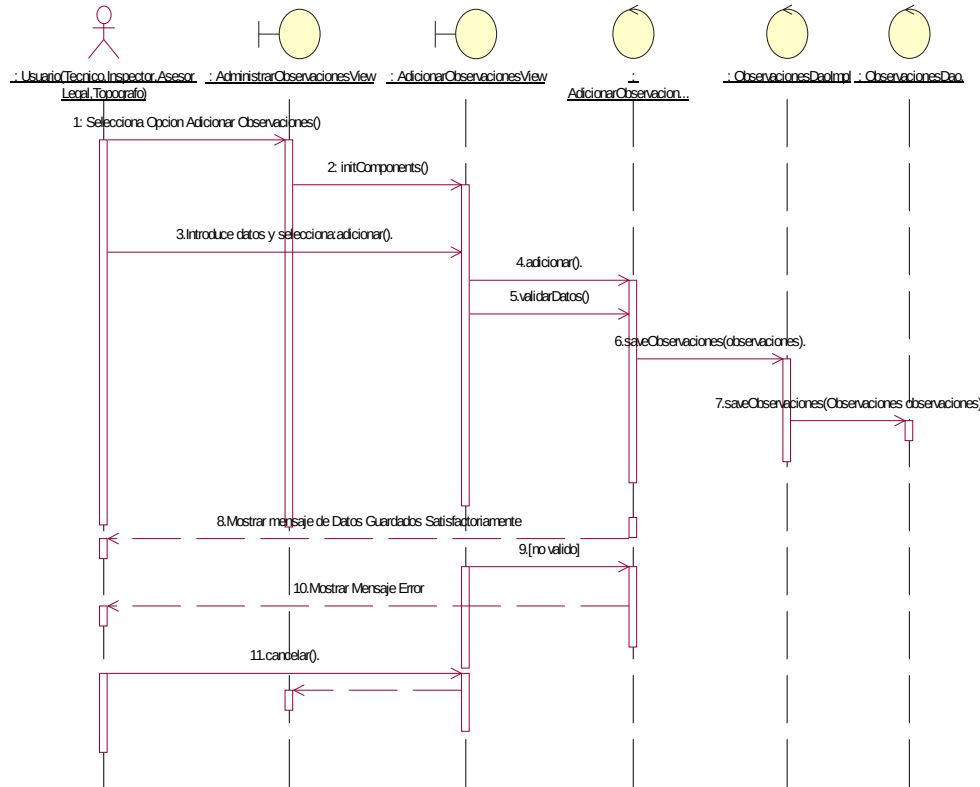


Figura 290: Diagrama de Secuencia: Adicionar Observaciones

III.2.2.3.43 Diagrama de Secuencia Modificar Observaciones

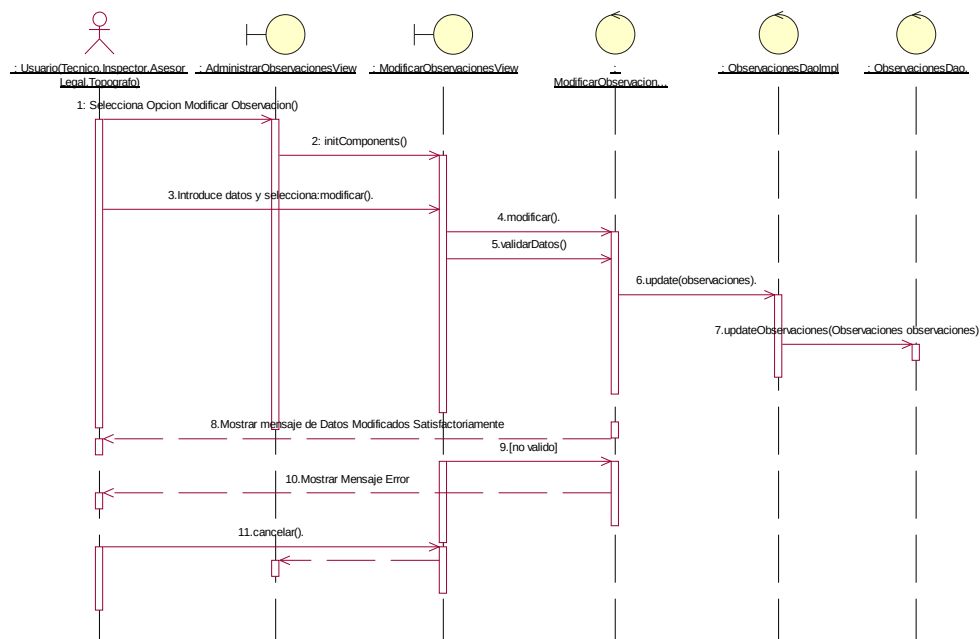


Figura 291: Diagrama de Secuencia: Modificar Observaciones

III.2.2.3.43 Diagrama de Secuencia Ver Observaciones

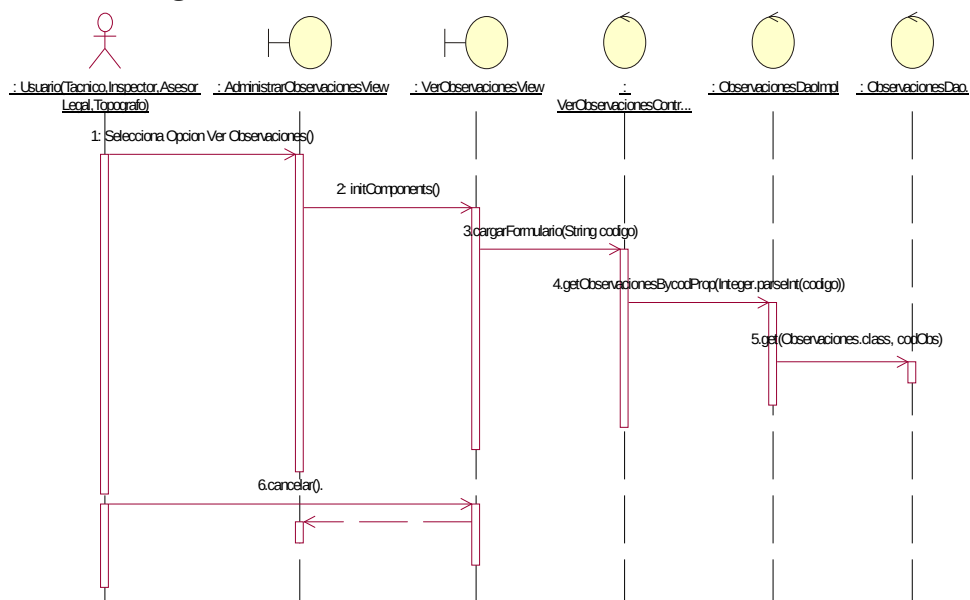


Figura 292: Diagrama de Secuencia: Ver Observaciones

III.2.2.3.59 Diagrama de Secuencia Administrar Observaciones Jefe Lotes

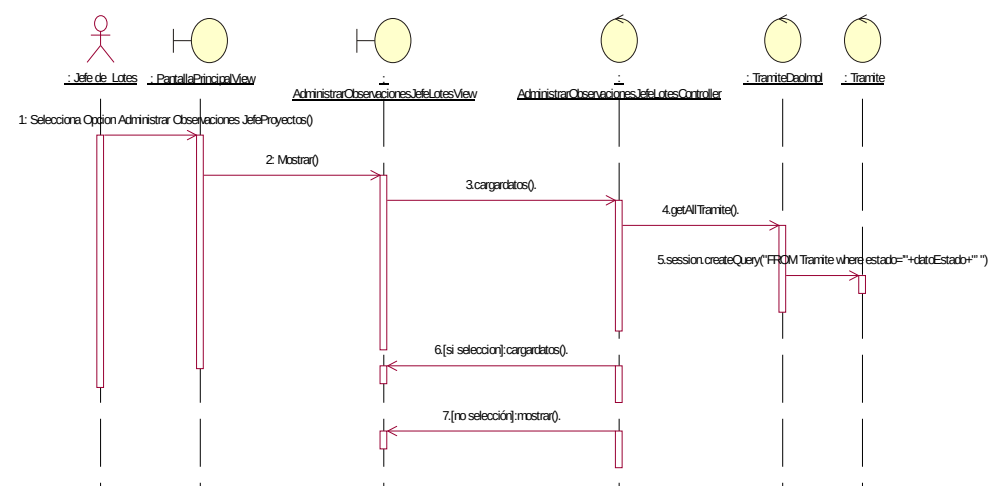


Figura 293: Diagrama de Secuencia: Administrar Observaciones Jefe Lotes

III.2.2.3.60 Diagrama de Secuencia Adicionar Observaciones Jefe Lotes

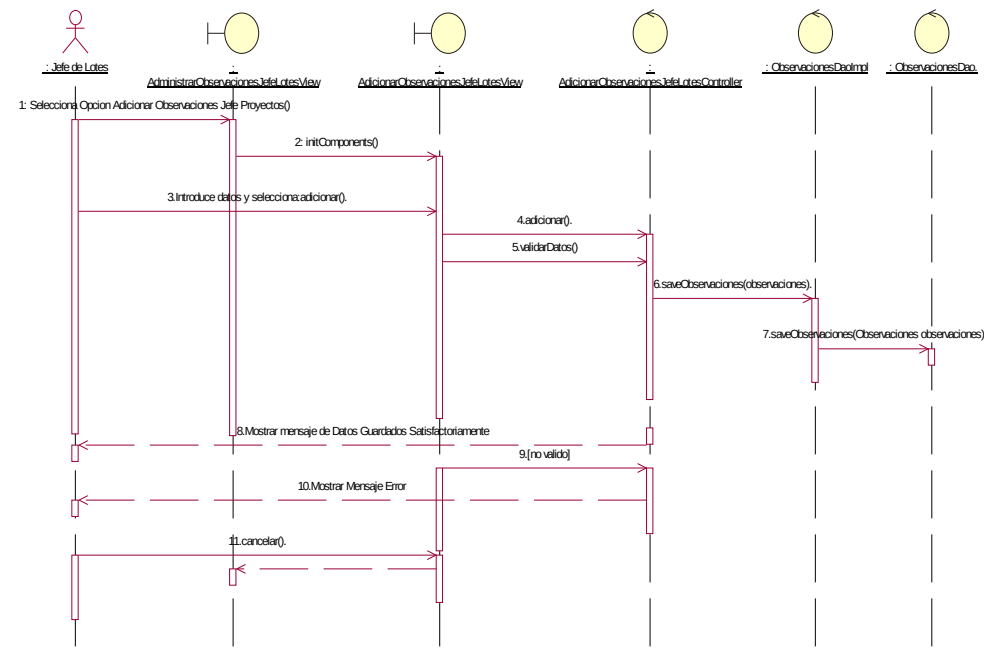


Figura 294: Diagrama de Secuencia: Adicionar Observaciones Jefe Lotes

III.2.2.3.61 Diagrama de Secuencia Modificar Observaciones Jefe Lotes

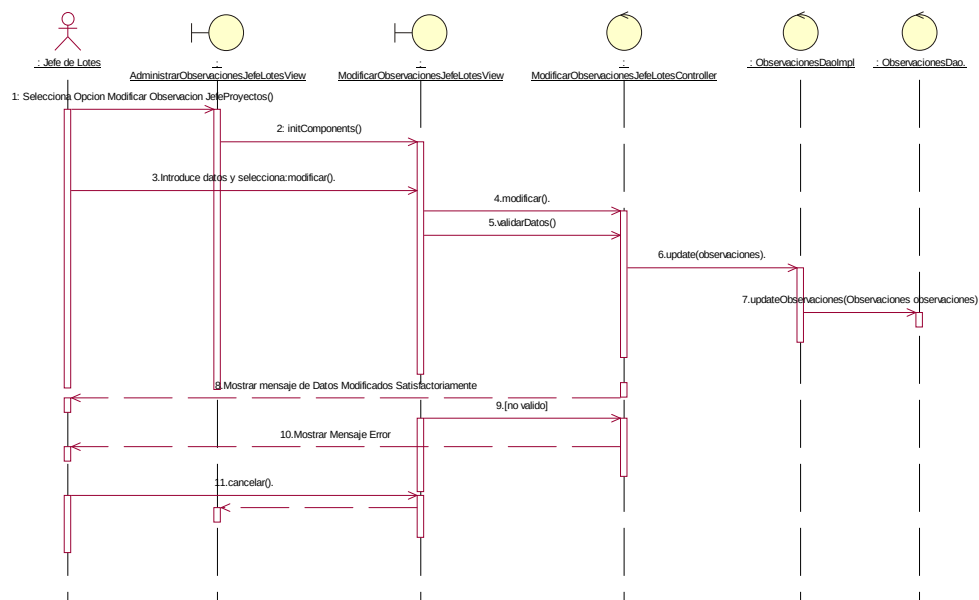


Figura 295: Diagrama de Secuencia: Modificar Observaciones Jefe Lotes

III.2.2.3.63 Diagrama de Secuencia Ver Observaciones Jefe Lotes

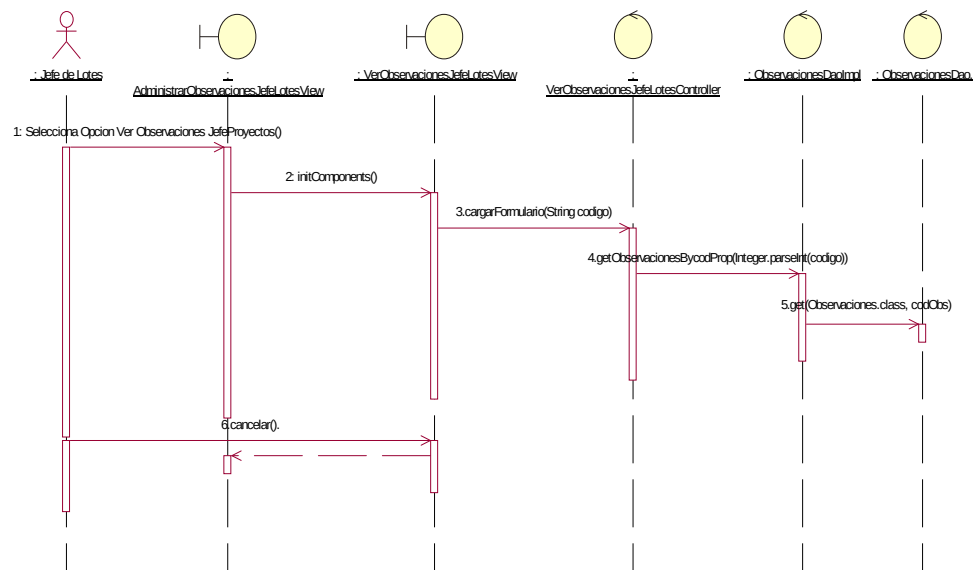


Figura 296: Diagrama de Secuencia: Ver Observaciones Jefe Lotes

III.2.2.3.64 Diagrama de Secuencia Dar Alta Trámite

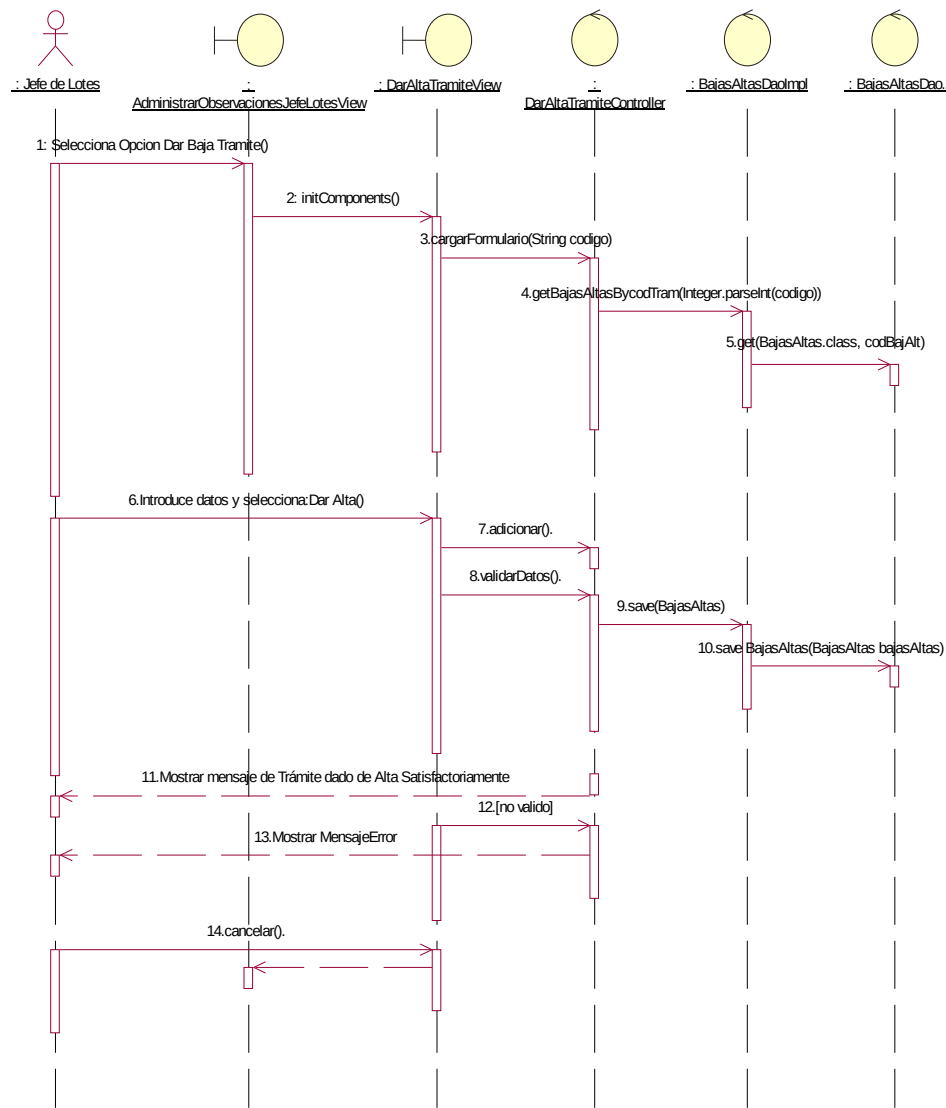


Figura 297: Diagrama de Secuencia: Dar Alta Trámite

III.2.2.3.65 Diagrama de Secuencia Aprobar Tramite

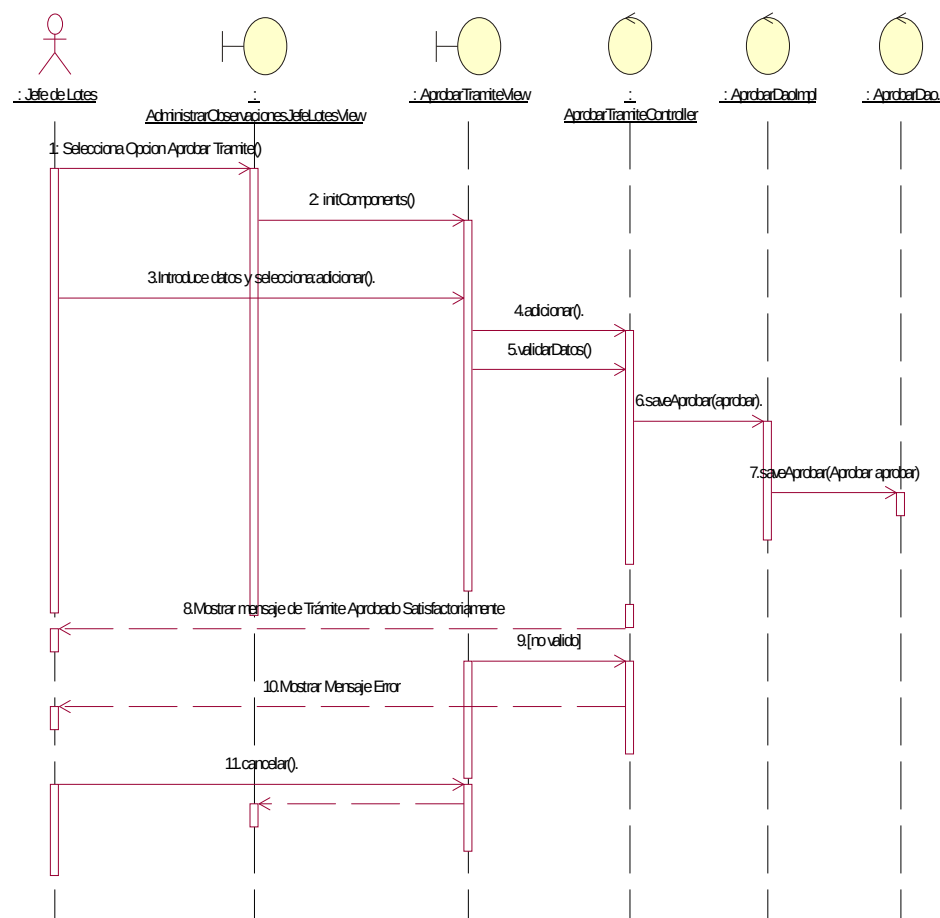


Figura 298: Diagrama de Secuencia: Aprobar Tramite

III.2.2.4 MODULO ADMINISTRACION

III.2.2.4.1 Diagrama de Secuencia Administrar Personal

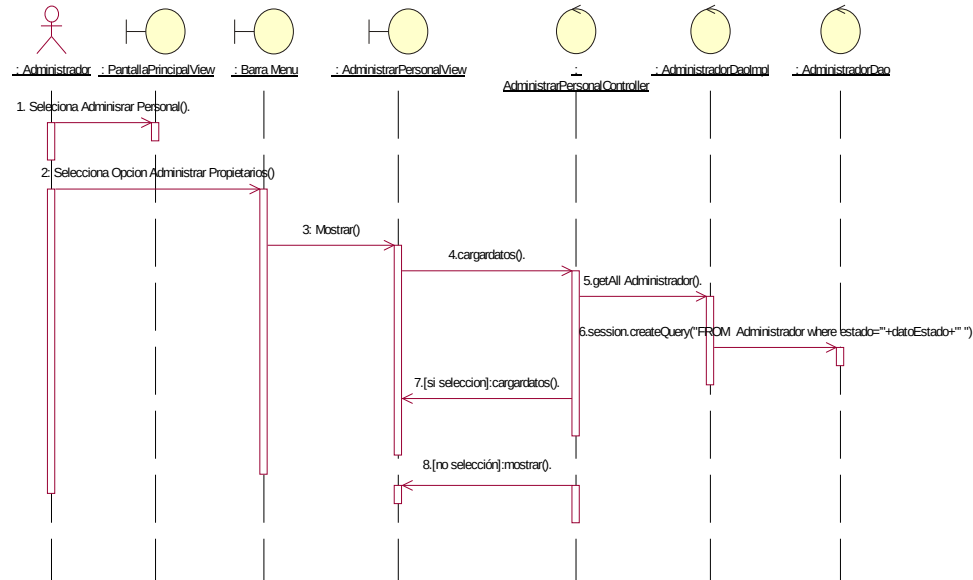


Figura 299: Diagrama de Secuencia: Administrar Personal

III.2.2.4.2 Diagrama de Secuencia Adicionar Personal

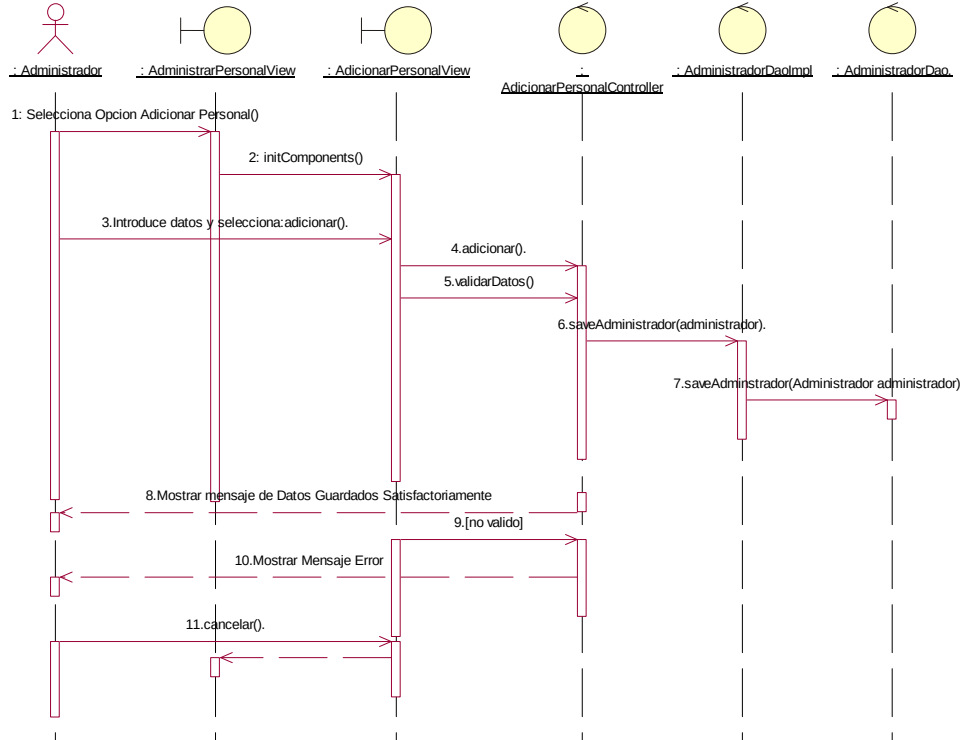


Figura 300: Diagrama de Secuencia: Adiconar Personal

III.2.2.4.3 Diagrama de Secuencia Modificar Personal

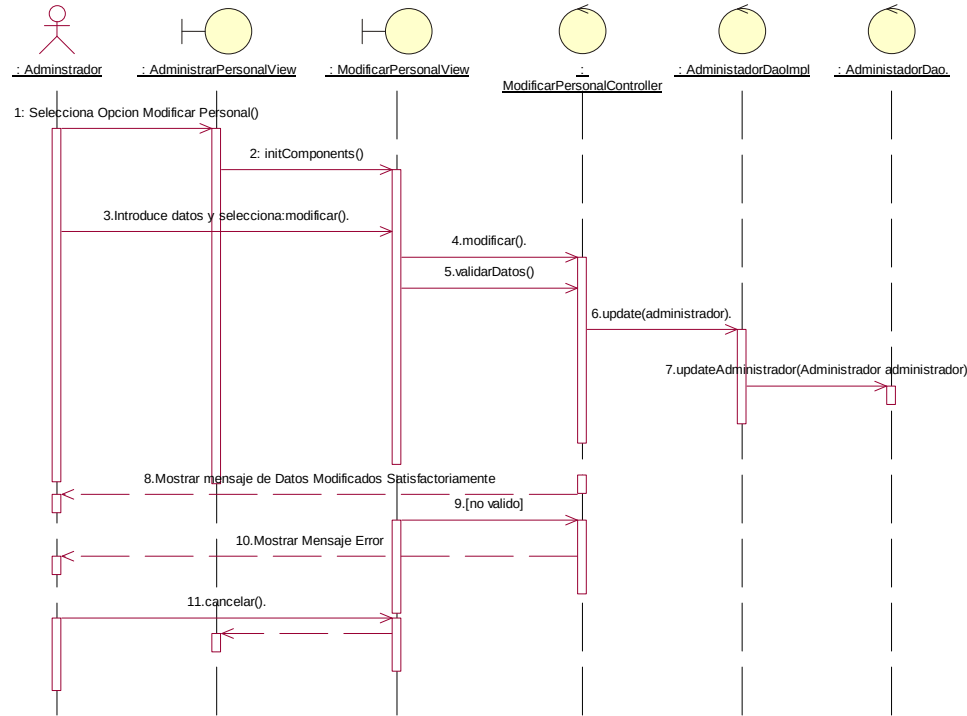


Figura 301: Diagrama de Secuencia: Modificar Personal

III.2.2.4.5 Diagrama de Secuencia Ver Personal

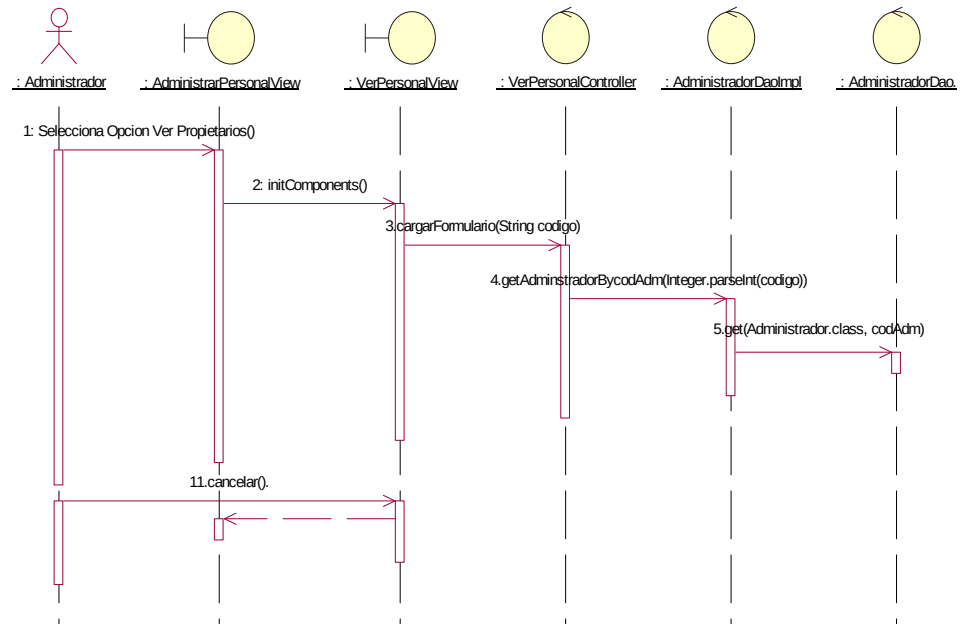


Figura 302: Diagrama de Secuencia: Ver Personal

III.2.2.4.6 Diagrama de Secuencia Administrar Roles

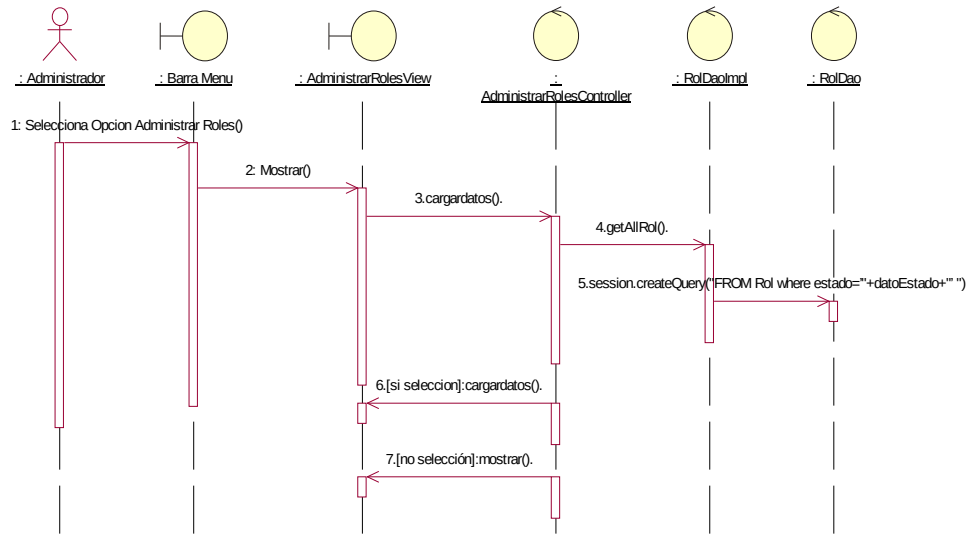


Figura 303: Diagrama de Secuencia: Administrar Roles

III.2.2.4.7 Diagrama de Secuencia Adicionar Roles

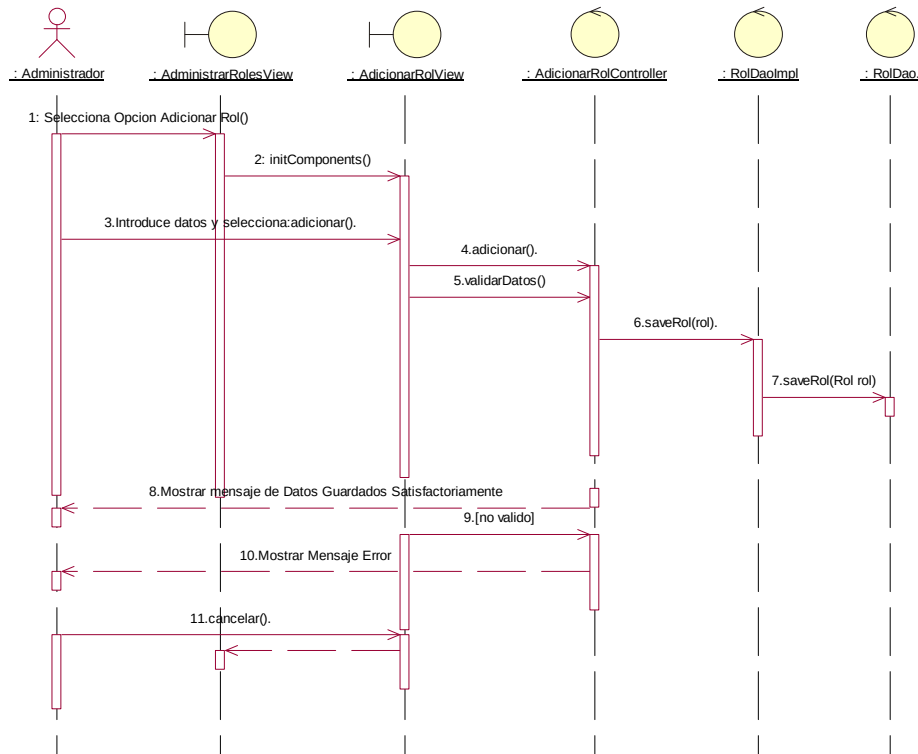


Figura 304: Diagrama de Secuencia: Adiconar Roles

III.2.2.4.8 Diagrama de Secuencia Modificar Roles

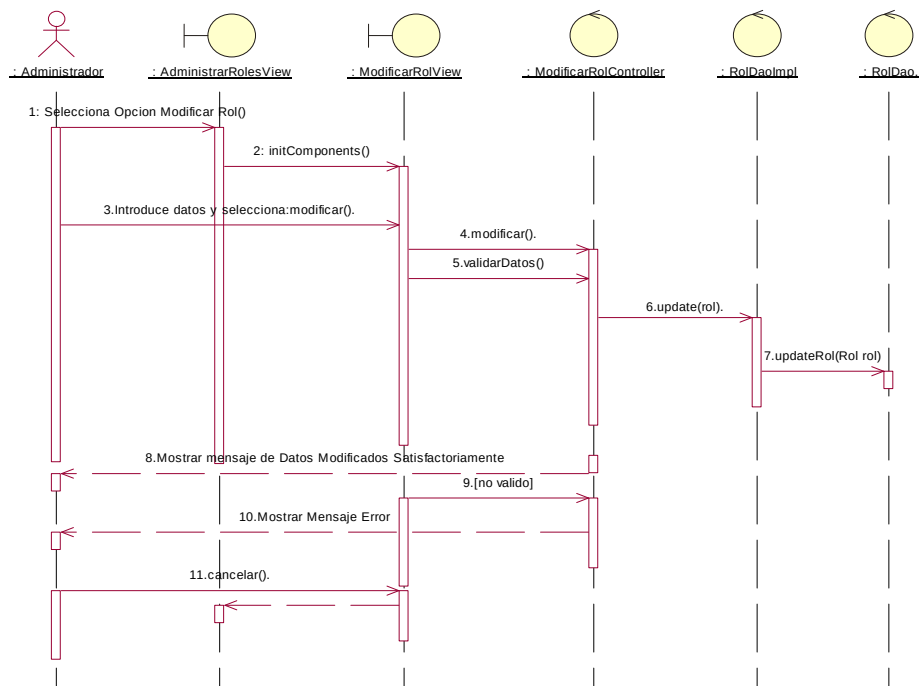


Figura 305: Diagrama de Secuencia: Modificar Roles

III.2.2.4.10 Diagrama de Secuencia Eliminar Roles

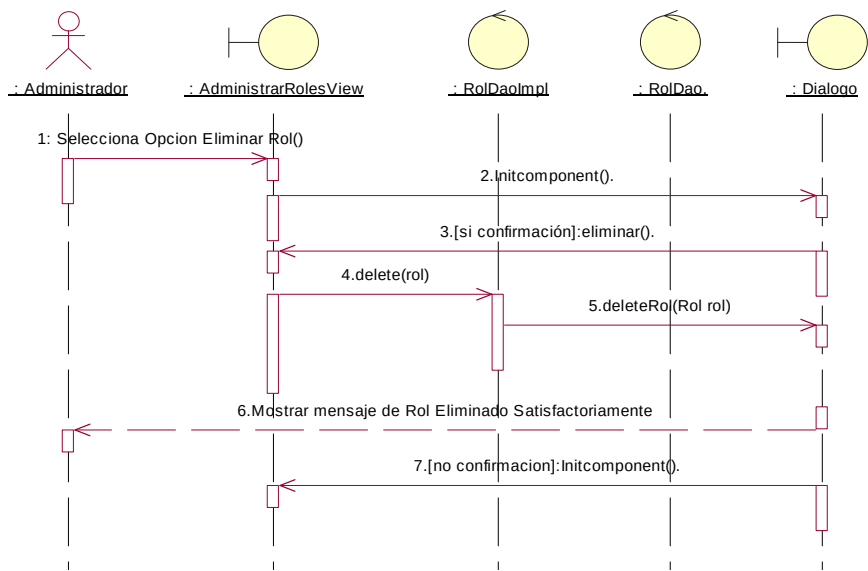


Figura 306: Diagrama de Secuencia: Eliminar Roles

III.2.2.4.1 1 Diagrama de Secuencia Cambiar Contraseña

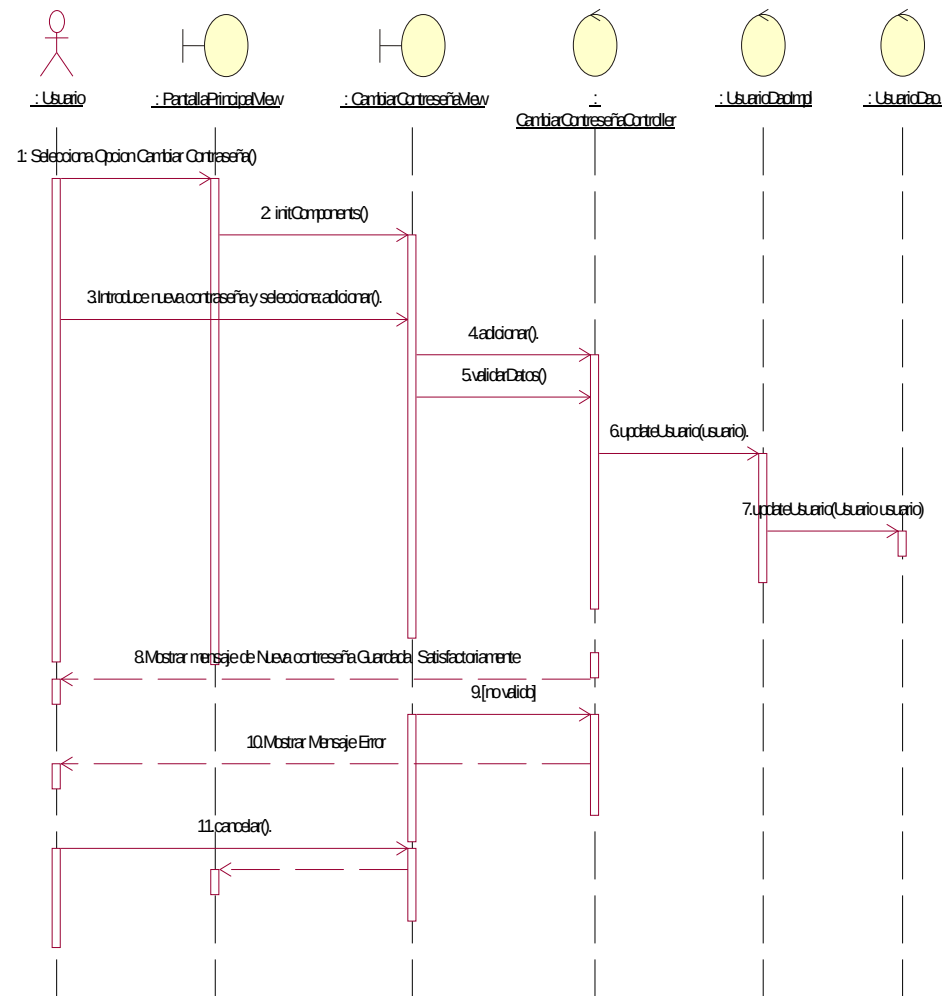


Figura 307: Diagrama de Secuencia: Cambiar Contraseña

III.2.2.5 MODULO DE REPORTES

III.2.2.5 .1 Diagrama de Secuencia Generar Reportes de Tramite

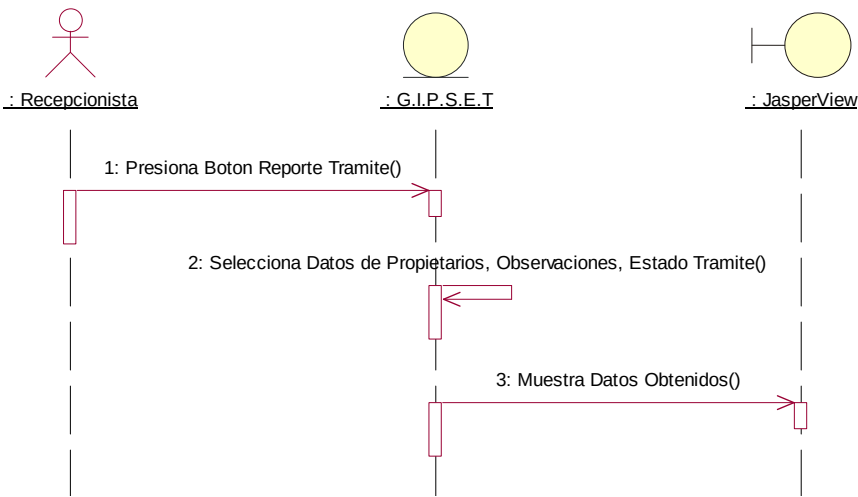


Figura 308: Diagrama de Secuencia: Generar Reportes de Tramite

III.2.2.5 .3 Diagrama de Secuencia Generar Reporte Técnico

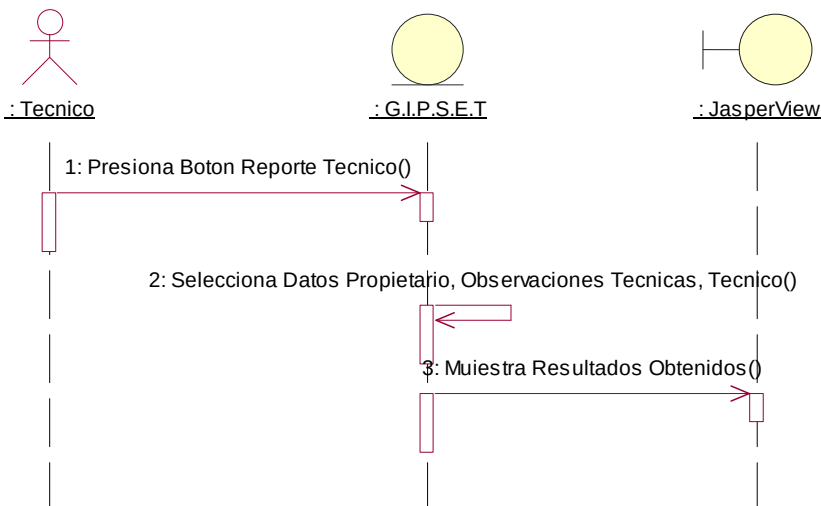


Figura 309: Diagrama de Secuencia: Generar Reporte Técnico

III.2.2.5 .4 Diagrama de Secuencia Generar Reporte de Inspección

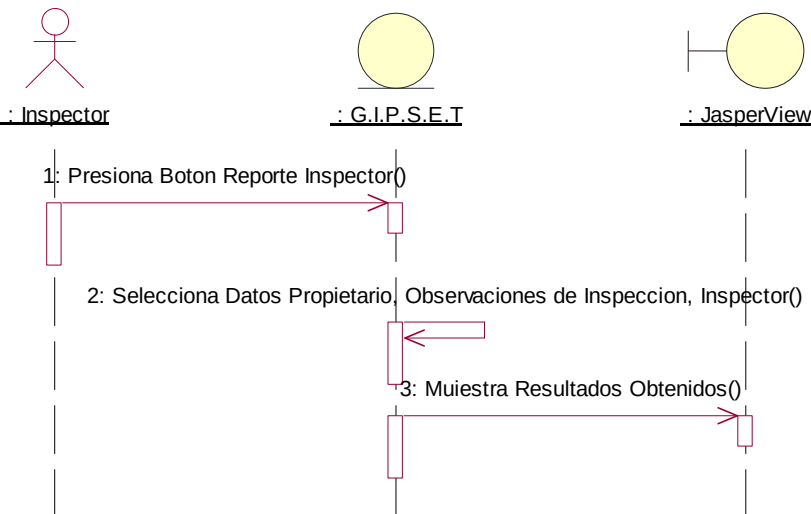


Figura 310: Diagrama de Secuencia: Generar Reporte de Inspección

III.2.2.5 .5 Diagrama de Secuencia Generar Reporte Legal

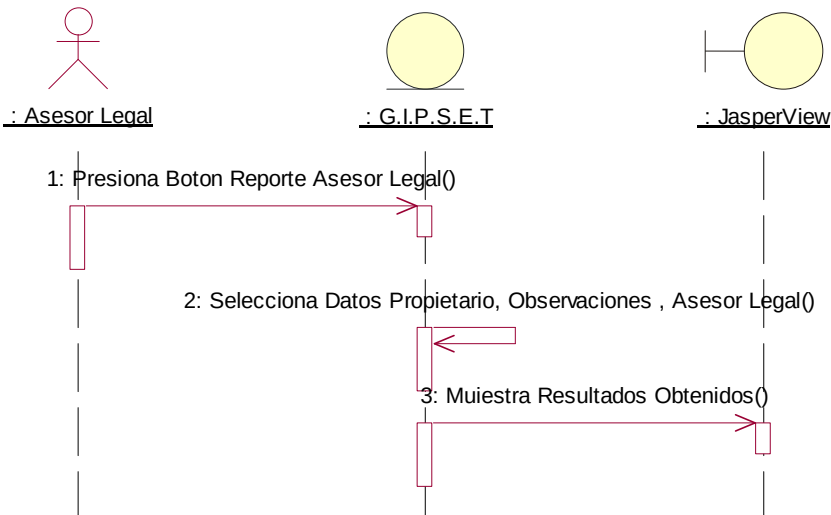


Figura 311: Diagrama de Secuencia: Generar Reporte Legal

III.2.2.5 .6 Diagrama de Secuencia Generar Reporte Topográfico

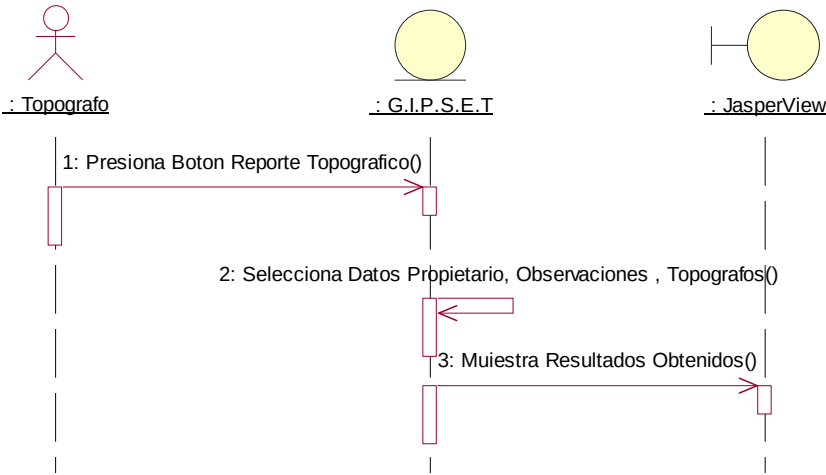


Figura 312: Diagrama de Secuencia: Generar Reporte Topográfico

III.2.3 DIAGRAMAS DE COLABORACION

Interactuar con Recursos del Sistema:

III.2.3.1. Diagrama de Colaboración Ingresar Sistema

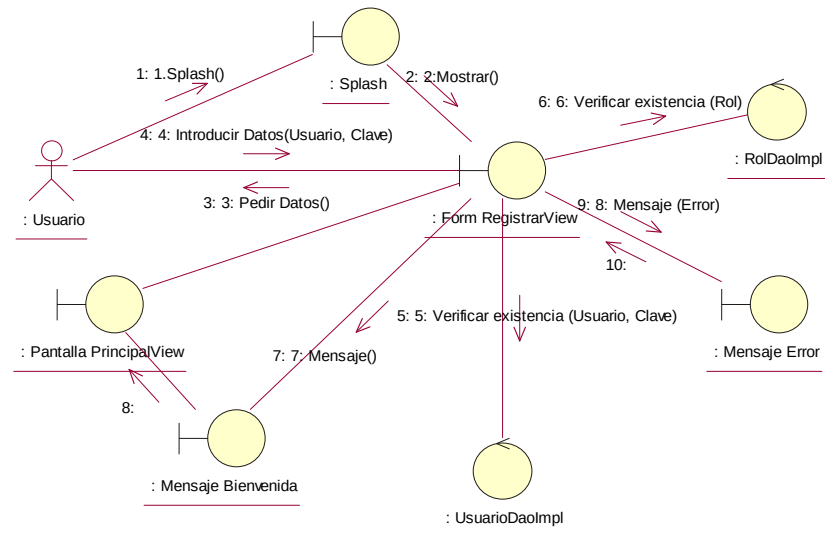


Figura 313: Diagrama de Colaboración: Ingresar Sistema

III.2.3.2. Diagrama de Colaboración Resguardar Base de Datos

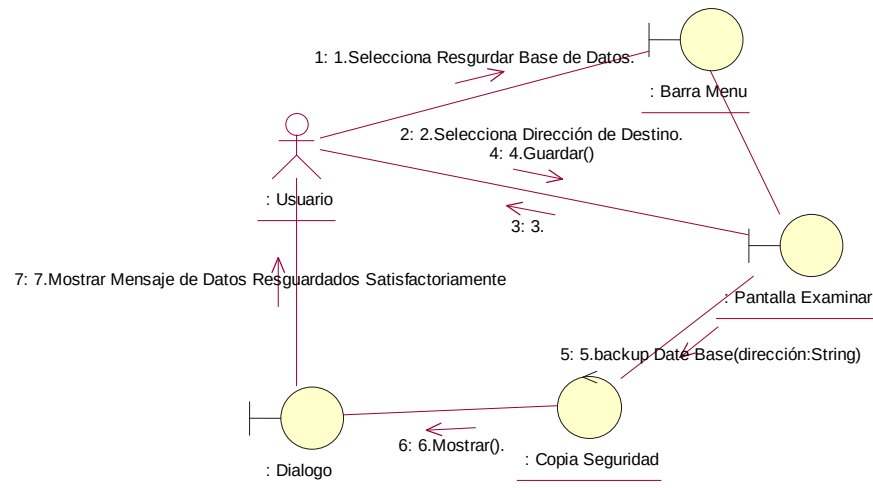


Figura 314: Diagrama de Colaboración: Resguardar Base de Datos

III.2.3.3. Diagrama de Colaboración Restaurar Base de Datos

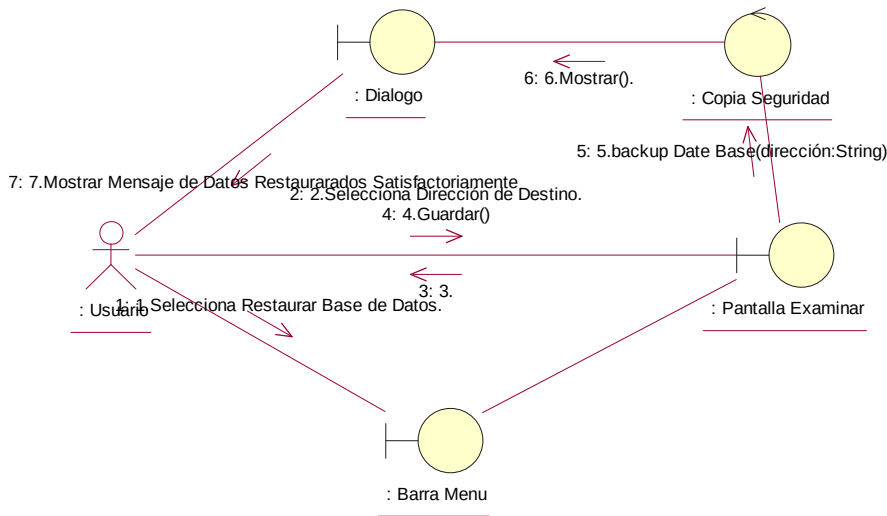


Figura 315: Diagrama de Colaboración: Restaurar Base de Datos

III.2.3.4. Diagrama de Colaboración Cerrar Conexión

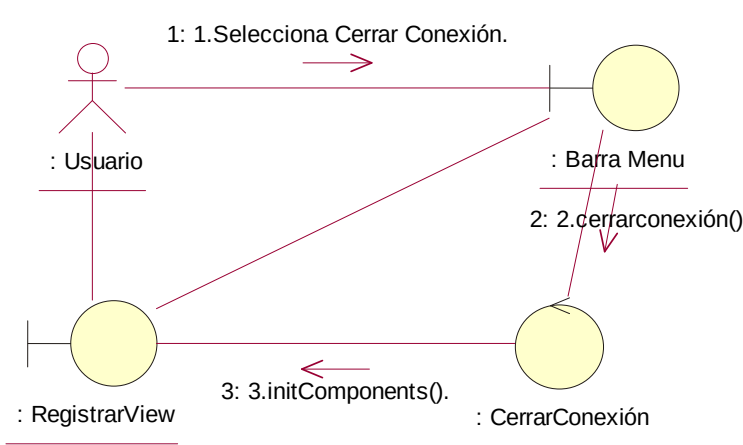


Figura 316: Diagrama de Colaboración: Cerrar Conexión

III.2.3.5. Diagrama de Colaboración Salir Sistema

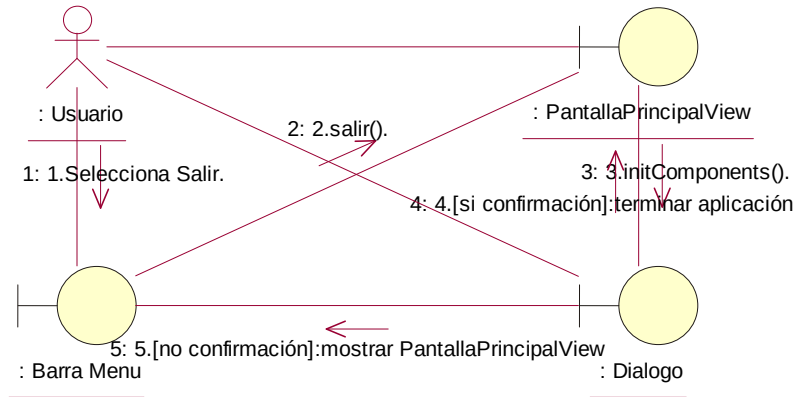


Figura 317: Diagrama de Colaboración: Salir Sistema

III.2.3.6. Diagrama de Colaboración Obtener Ayuda

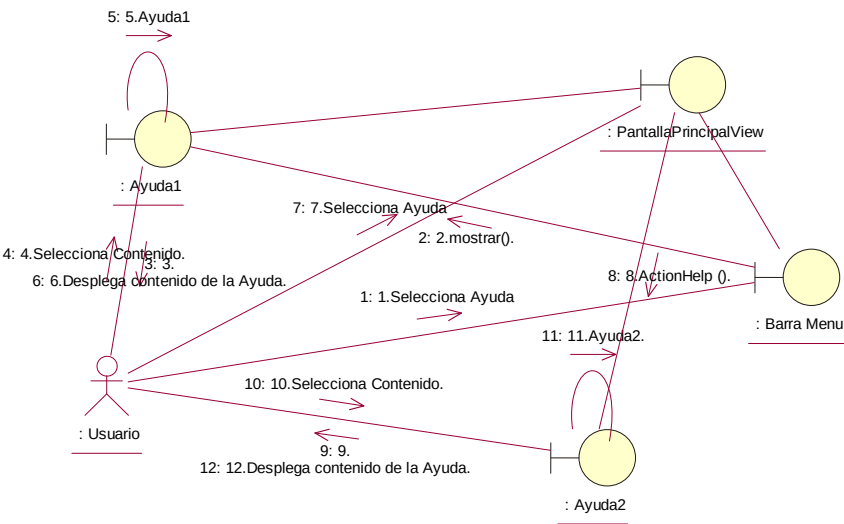


Figura 318: Diagrama de Colaboración: Obtener Ayuda

III.2.3.2 MODULO DE PROYECTOS

III.2.3.2 .1 Diagrama de Colaboración Administrar Propietarios

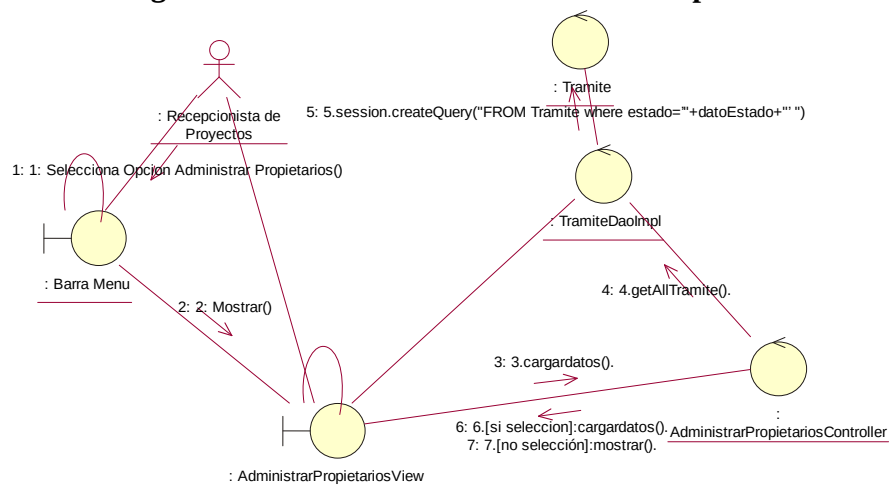


Figura 319: Diagrama de Colaboración: Administrar Propietarios

III.2.3.2 .2 Diagrama de Colaboración Adicionar Propietarios

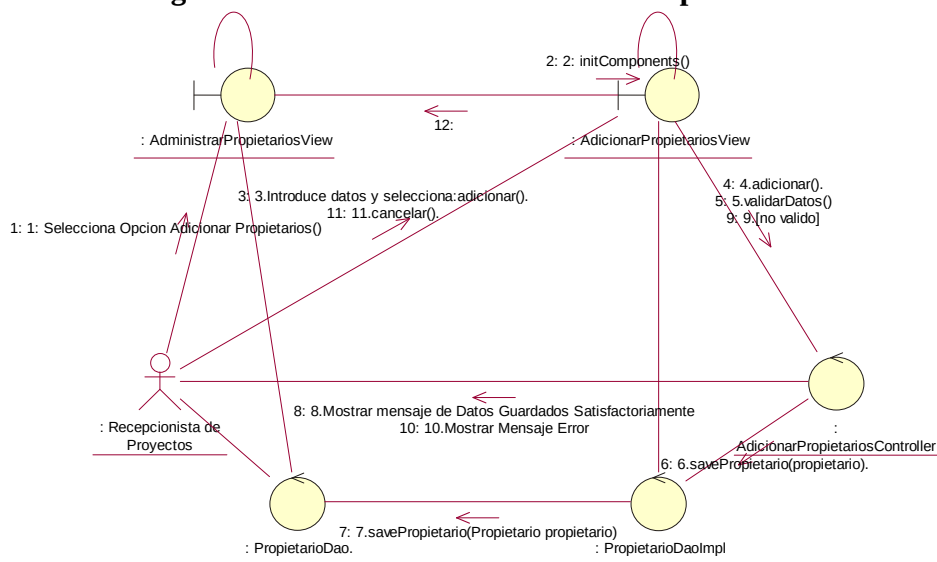


Figura 320: Diagrama de Colaboración: Adicionar Propietarios

III.2.3.2.3 Diagrama de Colaboración Adicionar Proyectista

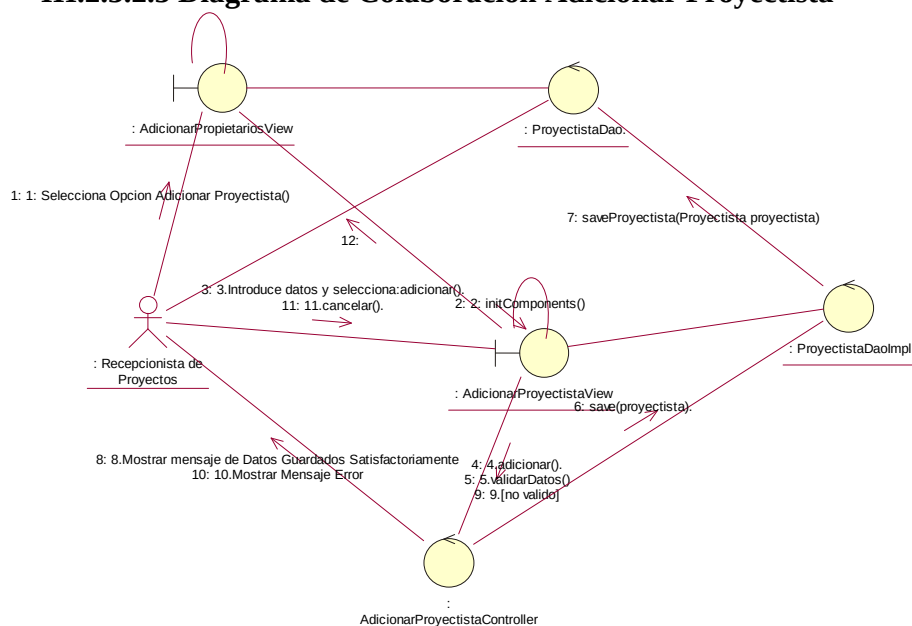


Figura 321: Diagrama de Colaboración: Adicionar Proyectista

III.2.3.2.4 Diagrama de Colaboración Adicionar Apoderado

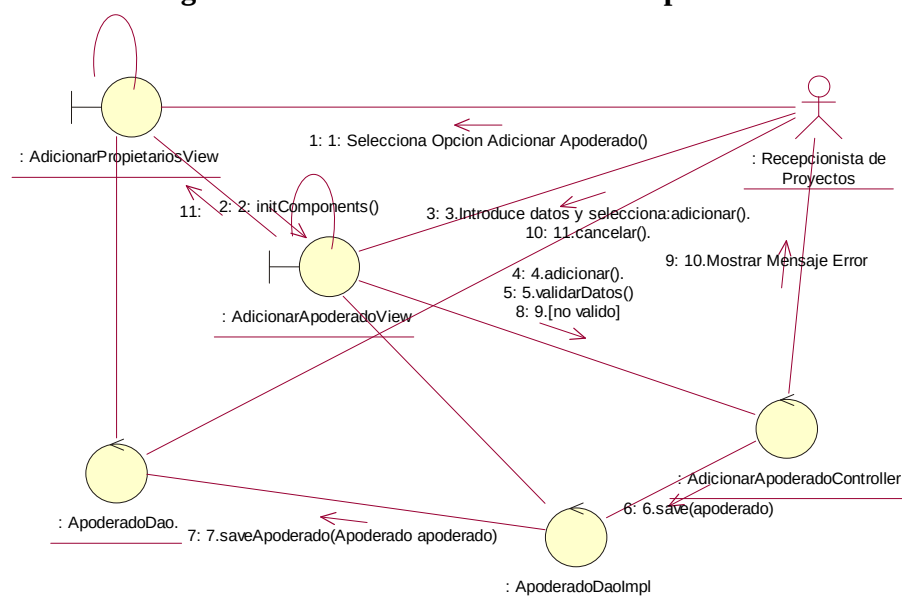


Figura 322: Diagrama de Colaboración: Adicionar Apoderado

III.2.3.2.5 Diagrama de Colaboración Modificar Propietarios

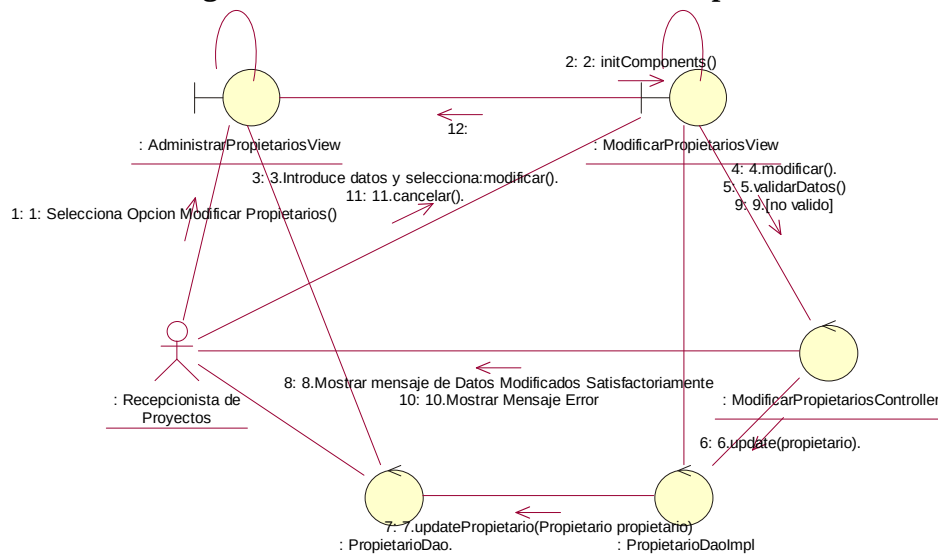


Figura 323: Diagrama de Colaboración: Modificar Propietarios

III.2.3.2.6 Diagrama de Colaboración Modificar Projectista

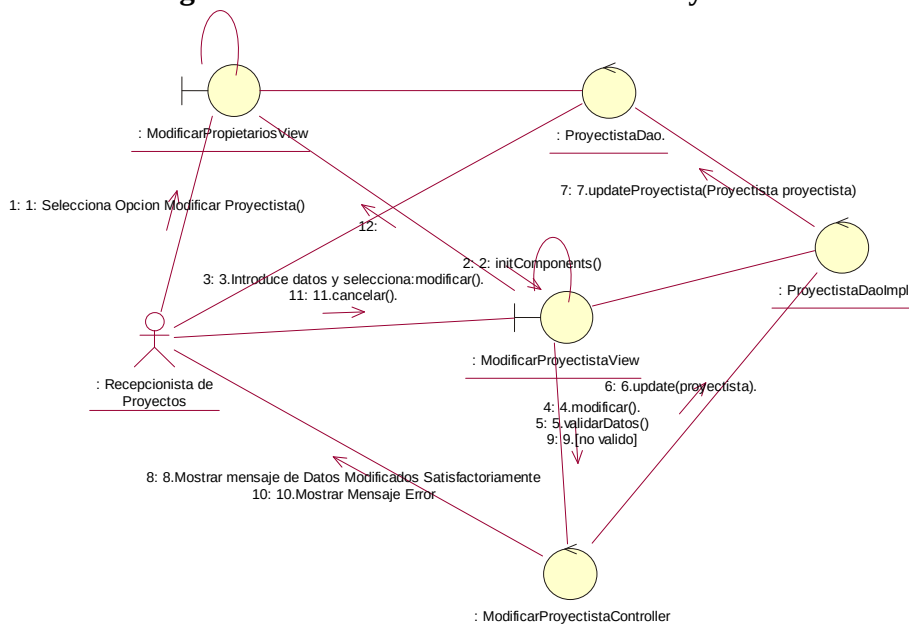


Figura 324: Diagrama de Colaboración: Modificar Projectista

III.2.3.2.7 Diagrama de Colaboración Modificar Apoderado

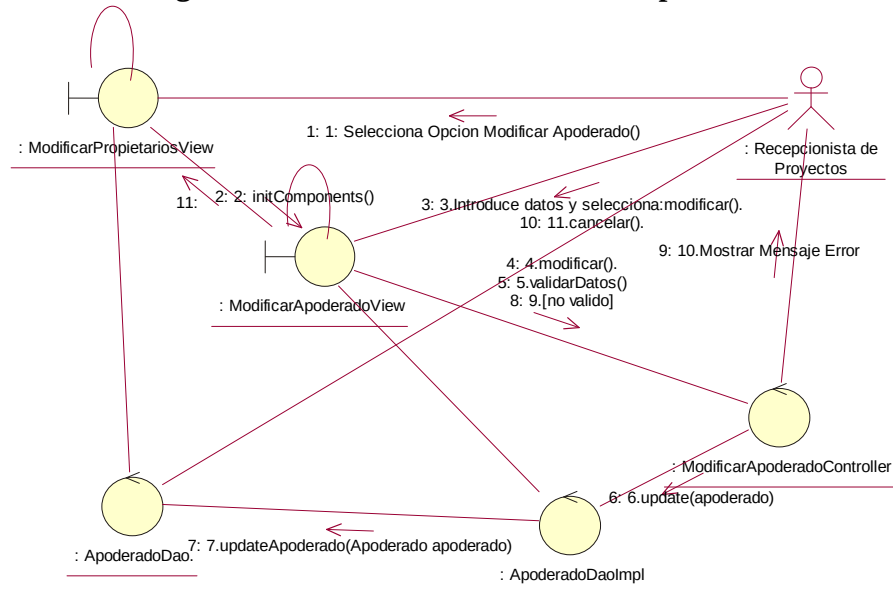


Figura 325: Diagrama de Colaboración: Modificar Apoderado

III.2.3.2.8 Diagrama de Colaboración Ver Propietarios

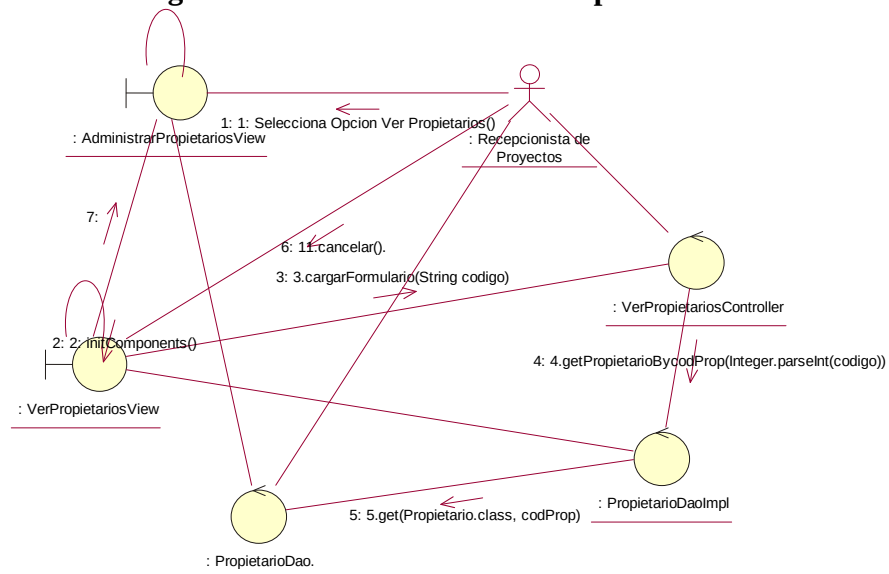


Figura 326: Diagrama de Colaboración: Ver Propietarios

III.2.3.2.9 Diagrama de Colaboración Administrar Observaciones Generales

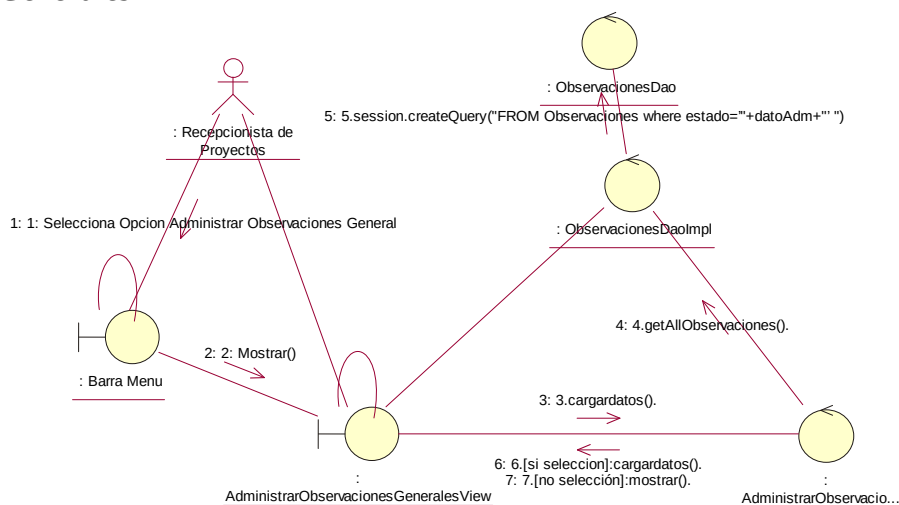


Figura 327: Diagrama de Colaboración: Administrar Observaciones Generales

III.2.3.2.10 Diagrama de Colaboración Ver Observaciones Generales

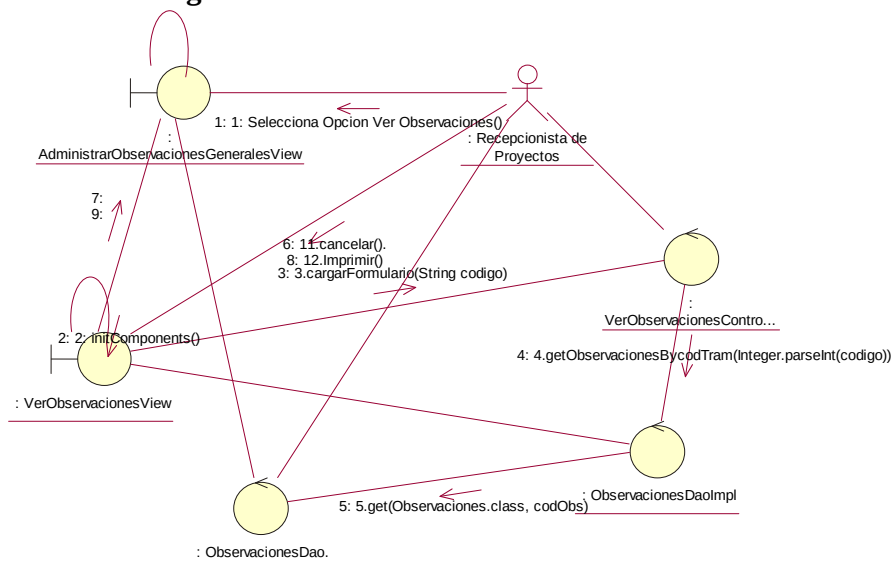


Figura 228: Diagrama de Colaboración: Ver Observaciones Generales

III.2.3.2.11 Diagrama de Colaboración Dar Baja Trámite

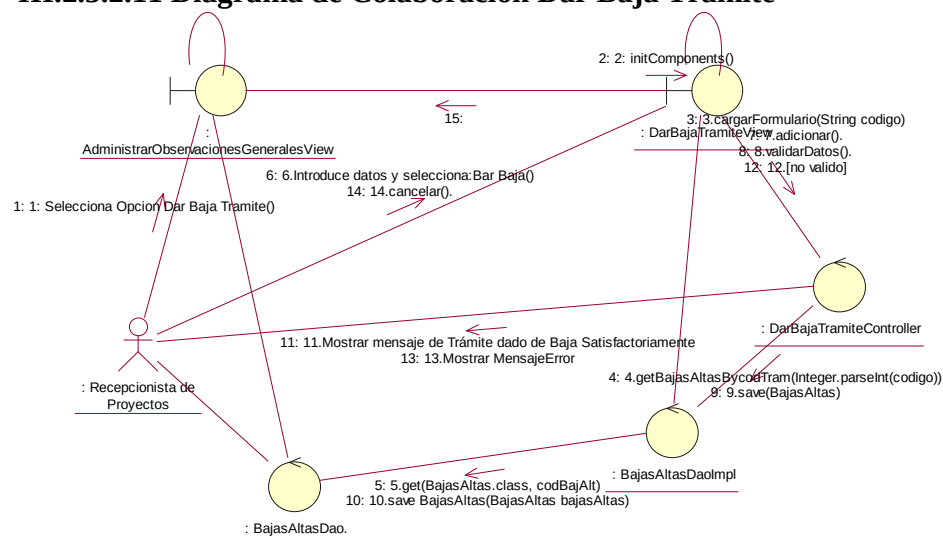


Figura 329: Diagrama de Colaboración: Dar Baja Trámite

III.2.3.2 .12 Diagrama de Colaboración Reasignar Tramite

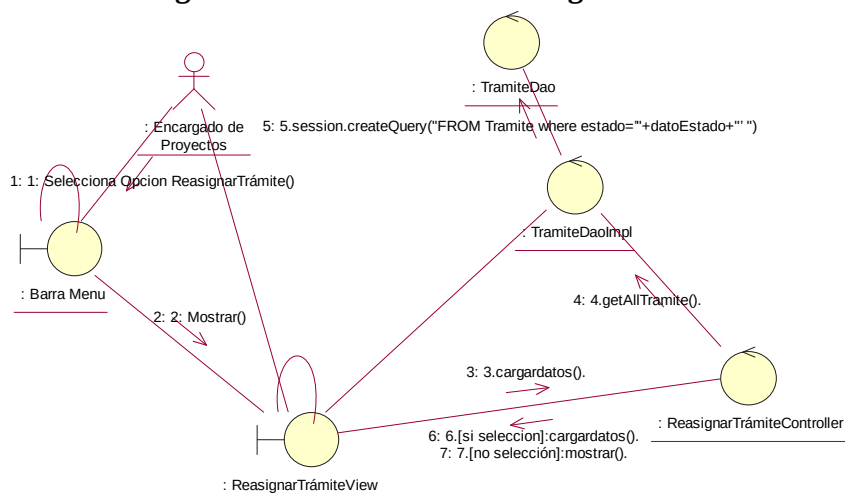


Figura 330: Diagrama de Colaboración: Reasignar Trámite

III.2.3.2 .13 Diagrama de Colaboración Registrar Reasignación

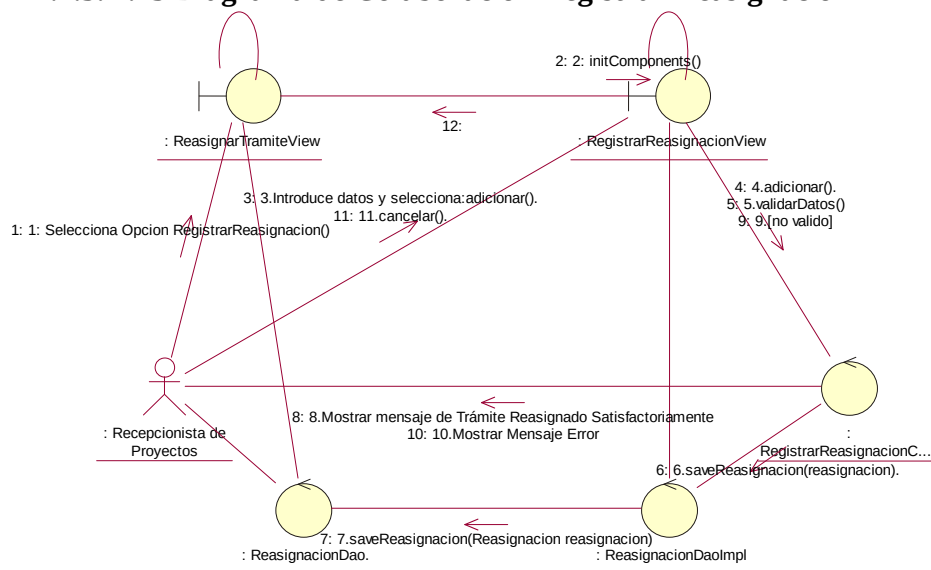


Figura 331: Diagrama de Colaboración: Registrar Tramite

III.2.3.2 .14 Diagrama de Colaboración Ver Reasignación

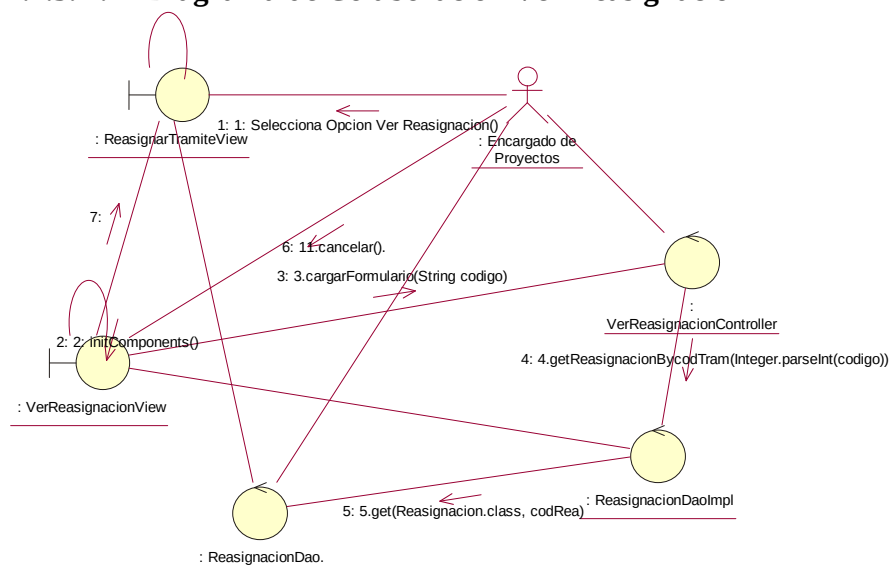


Figura 332: Diagrama de Colaboración: Ver Reasignación

III.2.3.2 .15 Diagrama de Colaboración Ver Datos Trámite

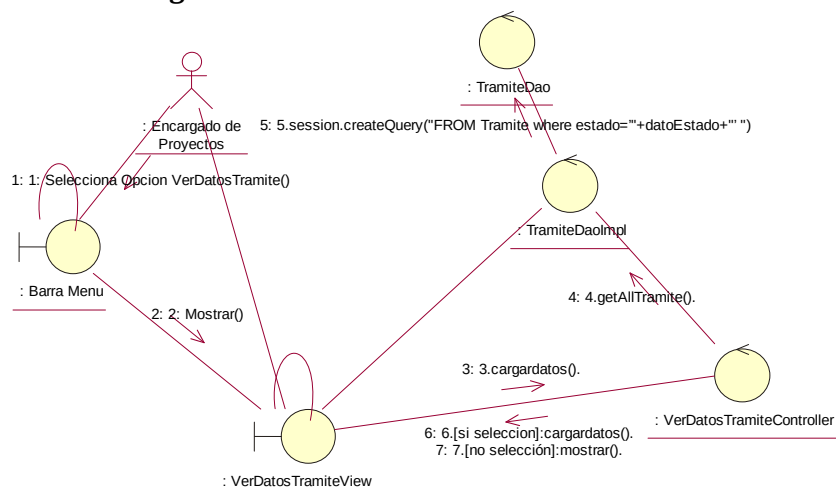


Figura 333: Diagrama de Colaboración: Ver Datos Trámite

III.2.3.2 .16 Diagrama de Colaboración Ver Estado Trámite

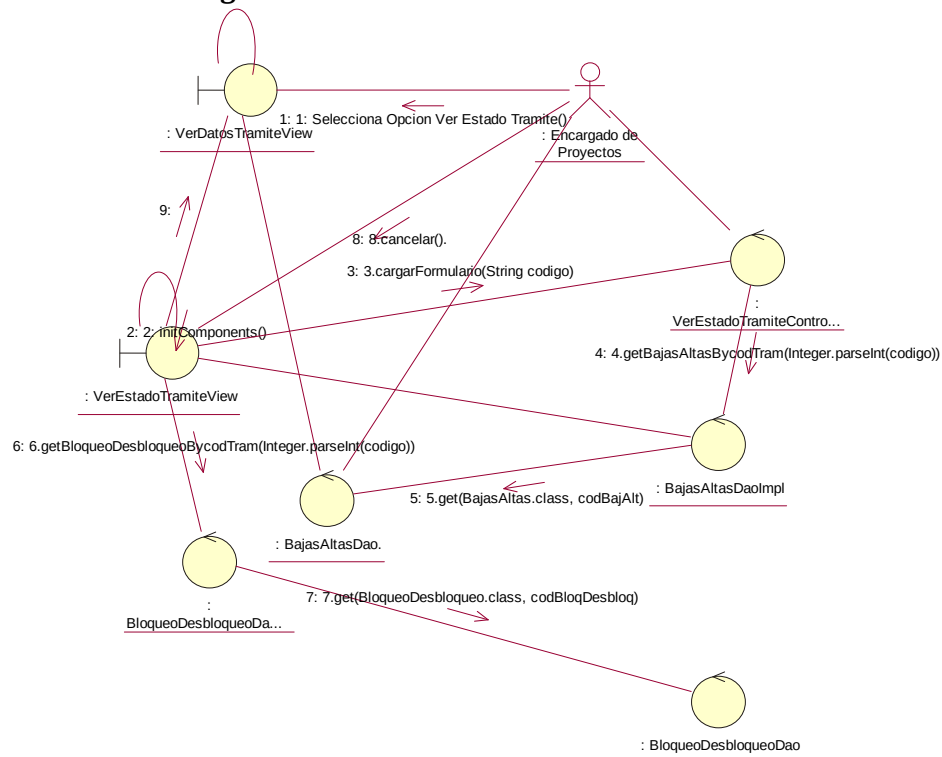


Figura 334: Diagrama de Colaboración: Ver Estado Tramite

III.2.3.2 .17 Diagrama de Colaboración Ver Detalle Trámite

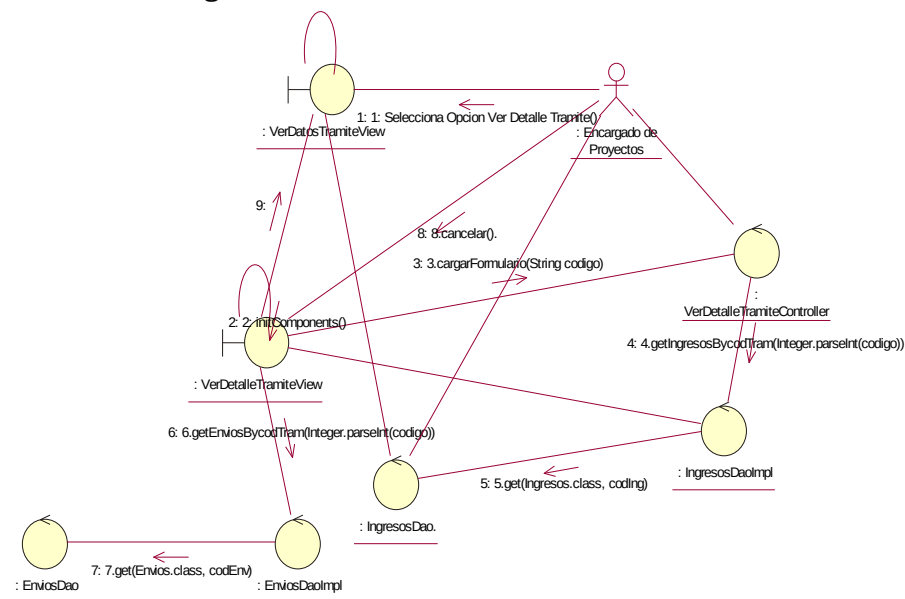


Figura 335: Diagrama de Colaboración: Ver Detalle Trámite

III.2.3.2 .18 Diagrama de Colaboración Administrar Ingresos

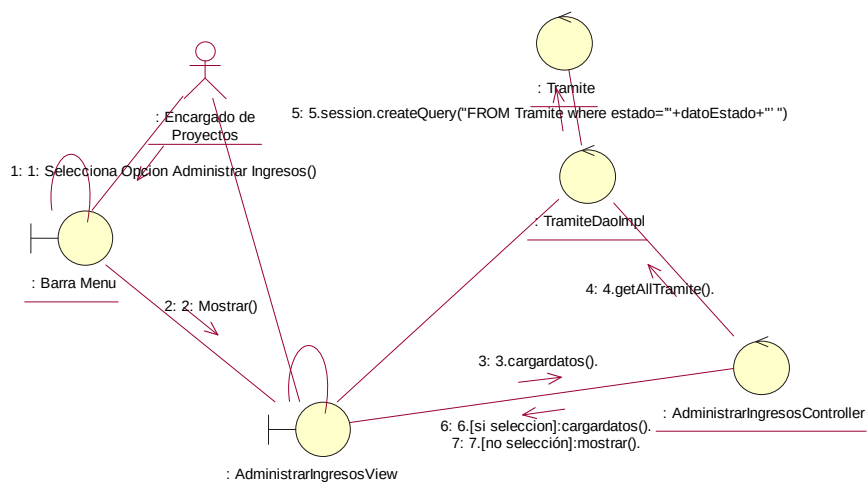


Figura 336: Diagrama de Colaboración: Administrar Ingresos

III.2.3.2 .19 Diagrama de Colaboración Registrar Ingresos

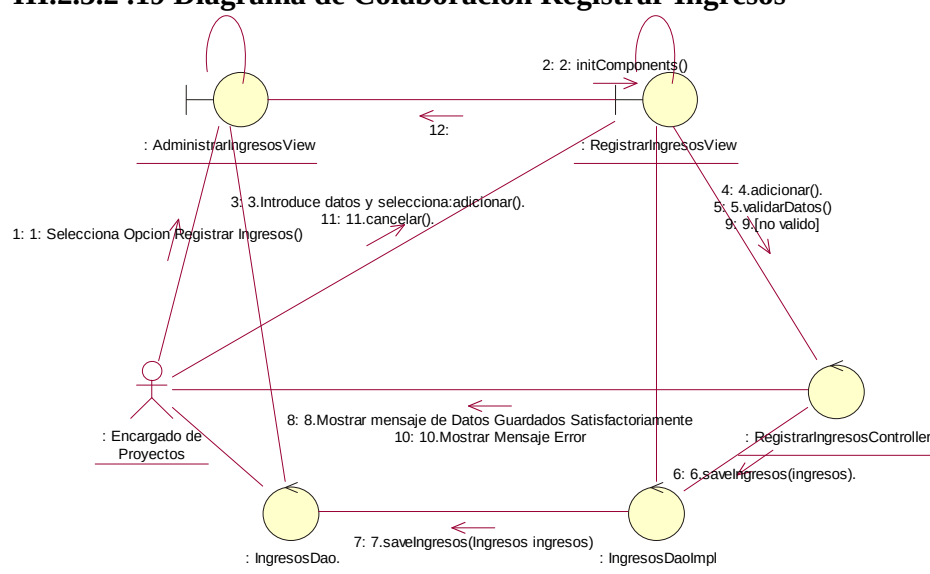


Figura 337: Diagrama de Colaboración: Registrar Ingresos

III.2.3.2 .20 Diagrama de Colaboración Modificar Ingresos

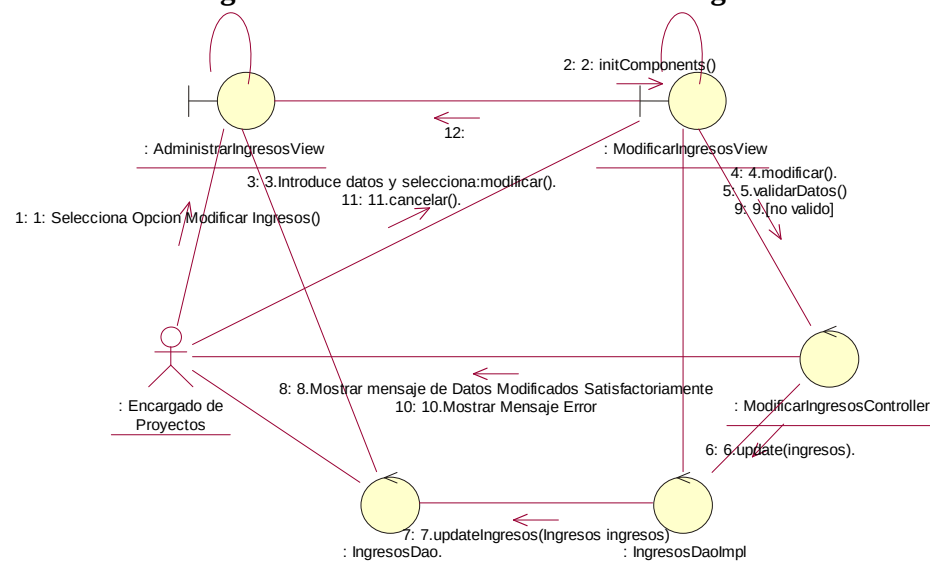


Figura 338: Diagrama de Colaboración: Modificar Ingresos

III.2.3.2 .21 Diagrama de Colaboración Ver Ingresos

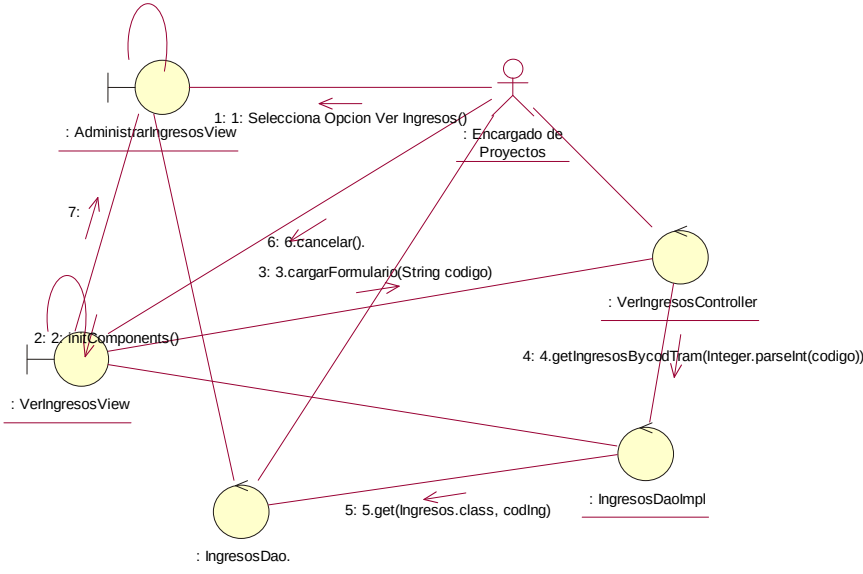


Figura 339: Diagrama de Colaboración: Ver Ingresos

III.2.3.2 .22 Diagrama de Colaboración Administrar Salidas

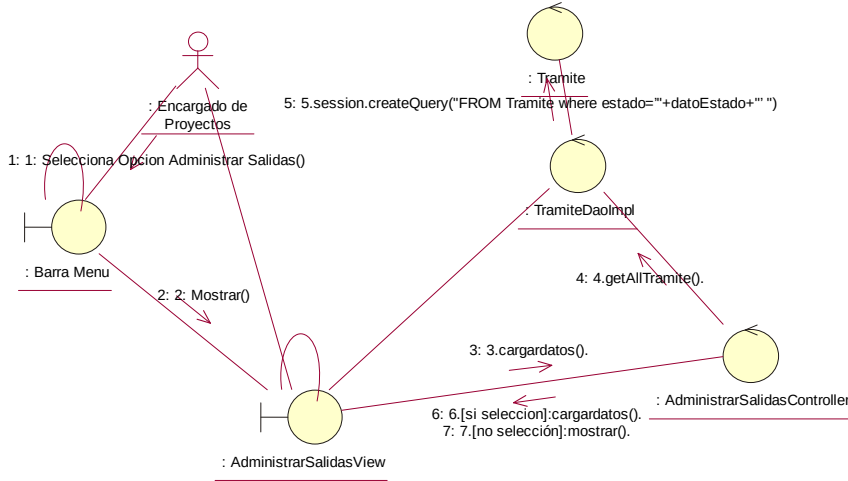


Figura 340: Diagrama de Colaboración: Administrar Salidas

III.2.3.2 .23 Diagrama de Colaboración Registrar Salidas

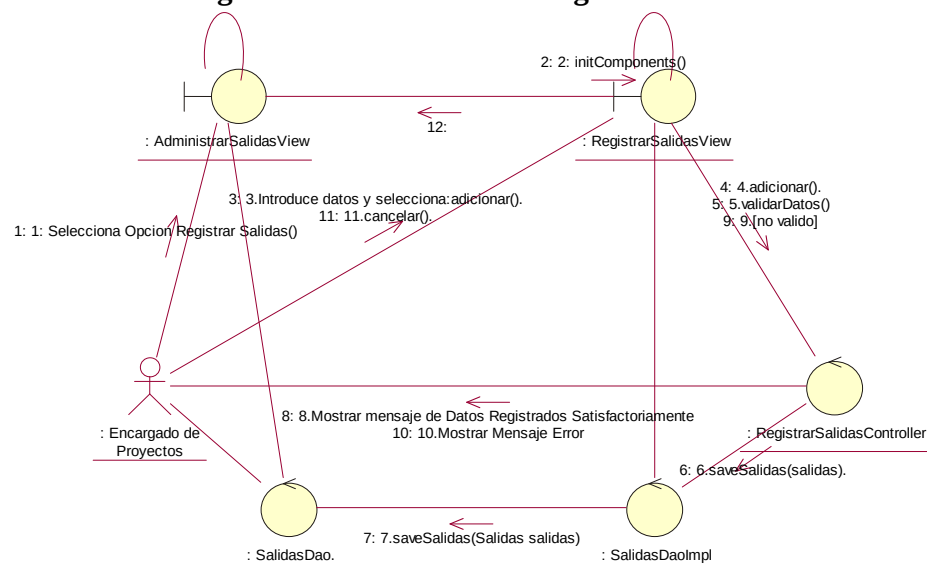


Figura 341: Diagrama de Colaboración: Registrar Salidas

III.2.3.2 .24 Diagrama de Colaboración Modificar Salidas

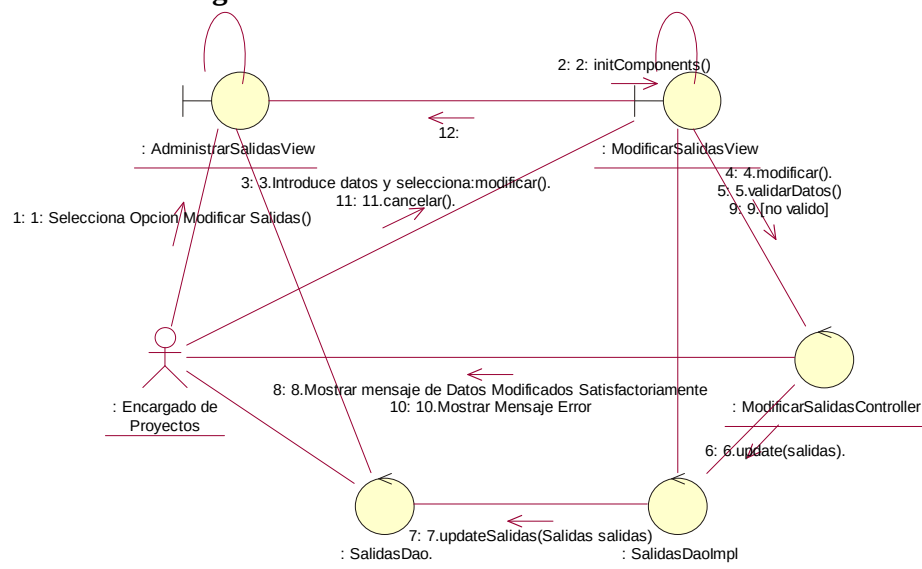


Figura 342: Diagrama de Colaboración: Modificar Salidas

III.2.3.2 .25 Diagrama de Colaboración Ver Salidas

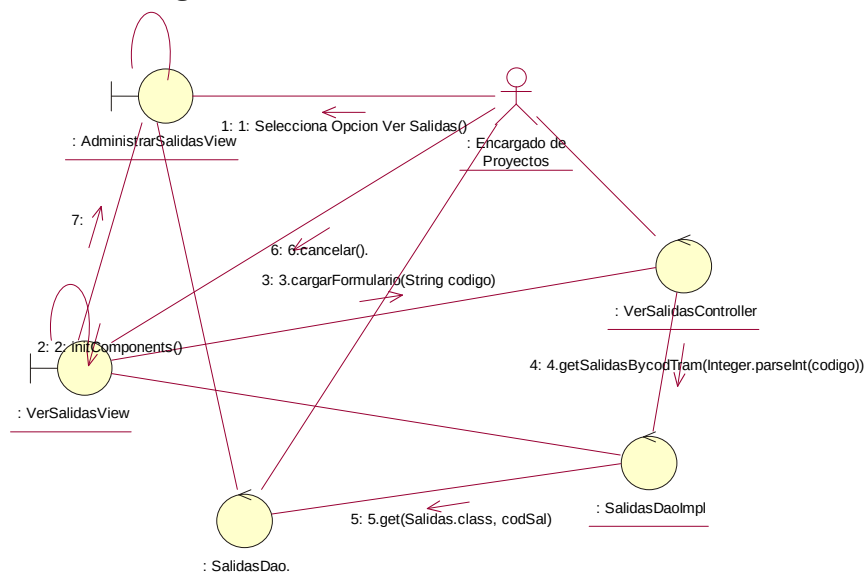


Figura 343: Diagrama de Colaboración: Ver Salidas

III.2.3.2 .26 Diagrama de Colaboración Administrar Envíos

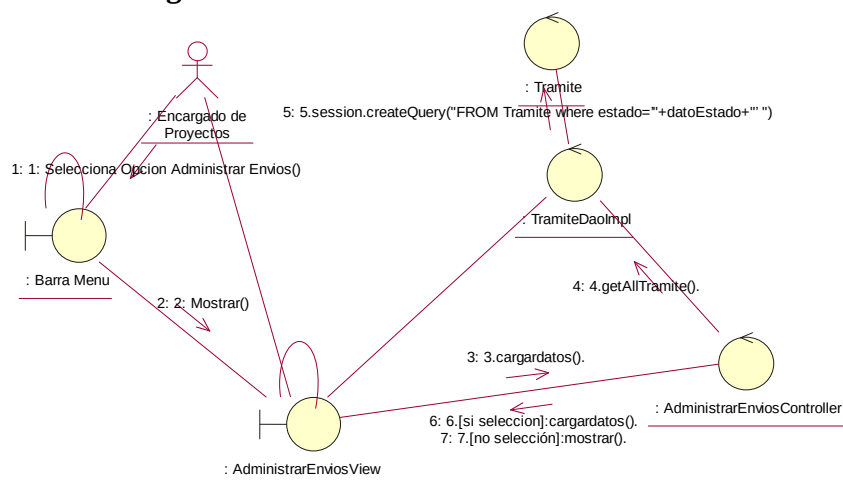


Figura 344: Diagrama de Colaboración: Administrar Envíos

III.2.3.2 .27 Diagrama de Colaboración Registrar Envíos

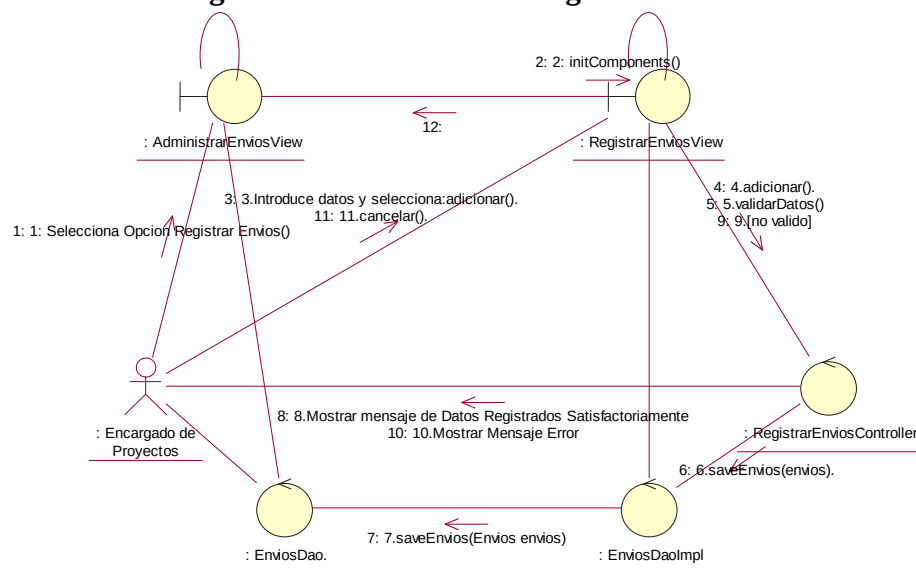


Figura 345: Diagrama de Colaboración: Registrar Envíos

III.2.3.2 .28 Diagrama de Colaboración Modificar Envíos

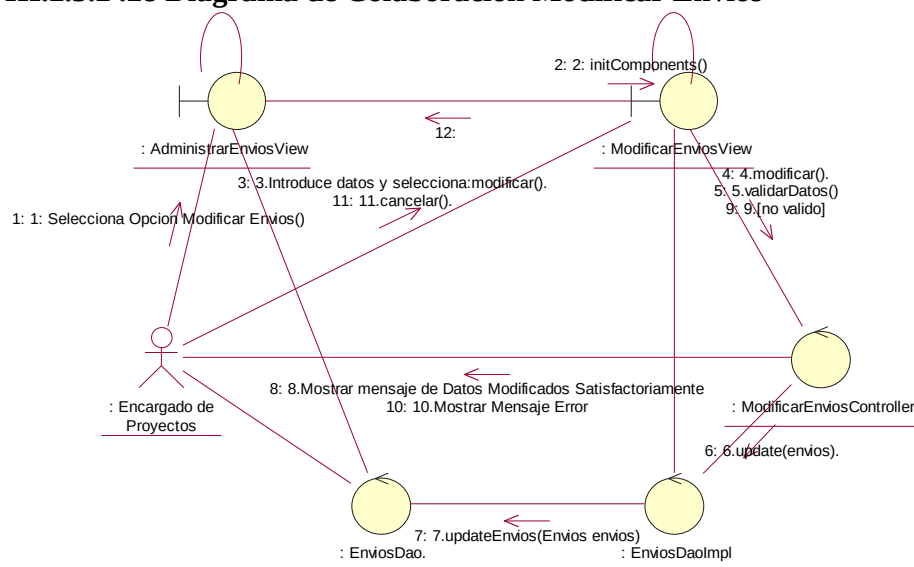


Figura 346: Diagrama de Colaboración: Modificar Envíos

III.2.3.2 .29 Diagrama de Colaboración Ver Envíos

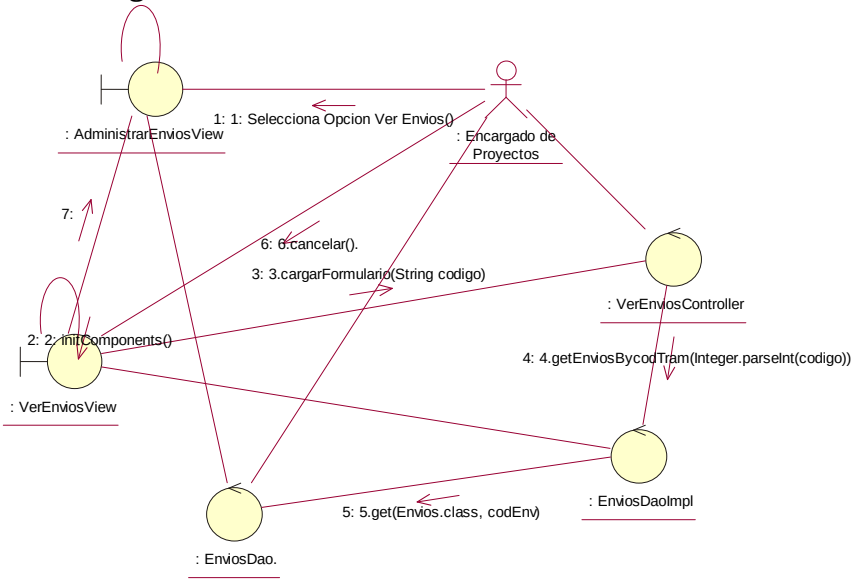


Figura 347: Diagrama de Colaboración: Ver Envíos

III.2.3.2 .30 Diagrama de Colaboración Administrar Recepción

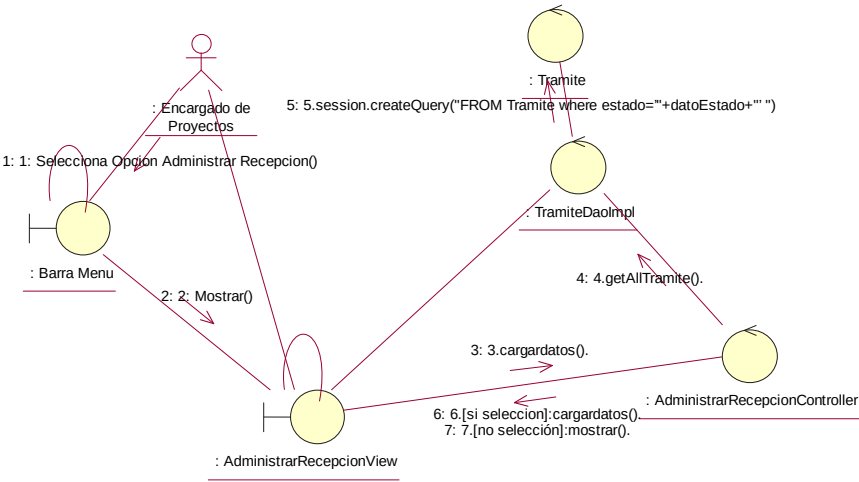


Figura 348: Diagrama de Colaboración: Administrar Recepción

III.2.3.2 .31 Diagrama de Colaboración Registrar Recepción

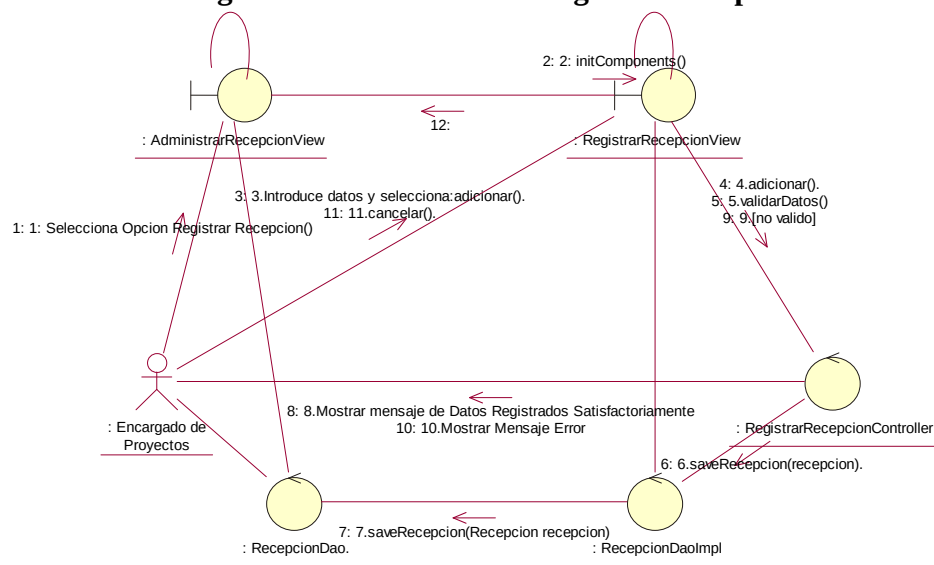


Figura 349: Diagrama de Colaboración: Registrar Recepción

III.2.3.2 .32 Diagrama de Colaboración Modificar Recepción

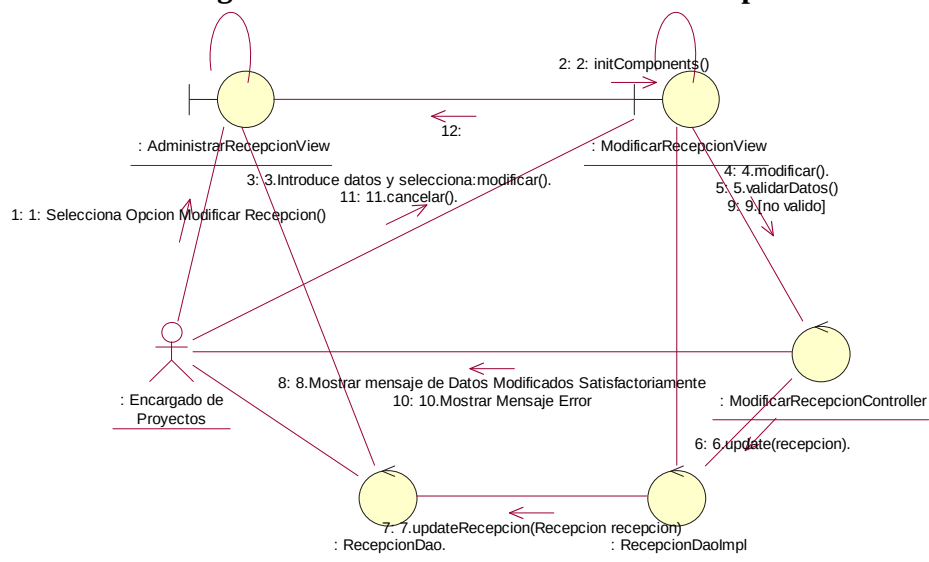


Figura 350: Diagrama de Colaboración: Modificar Recepción

III.2.3.2 .33 Diagrama de Colaboración Ver Recepción

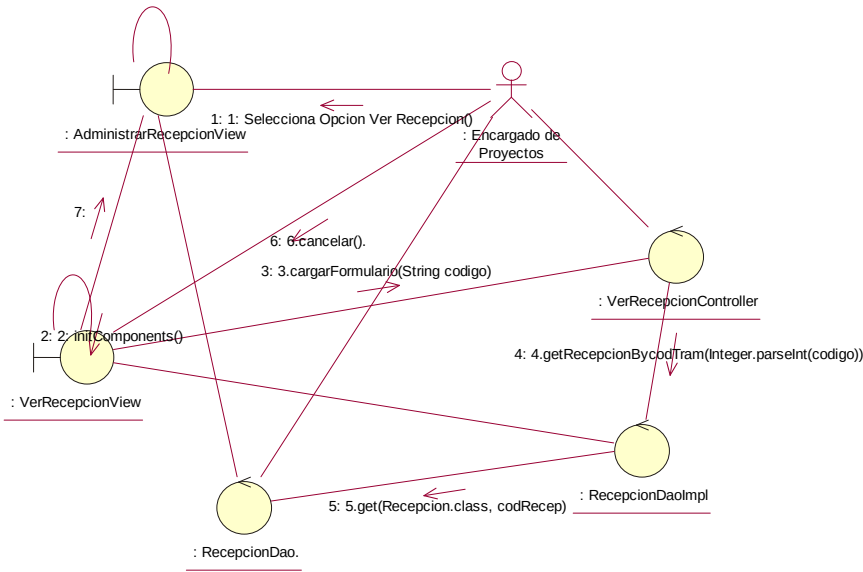


Figura 351: Diagrama de Colaboración: Ver Recepción

III.2.3.2 .34 Diagrama de Colaboración Asignar Numero Oficial

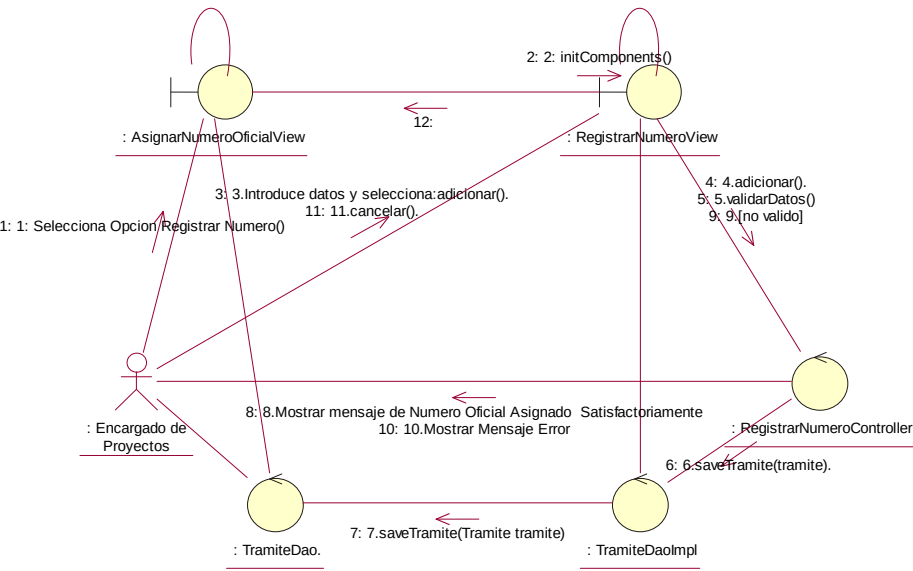


Figura 352: Diagrama de Colaboración: Asignar Numero Oficial

III.2.3.2 .35 Diagrama de Colaboración Asignar Personal Tramite

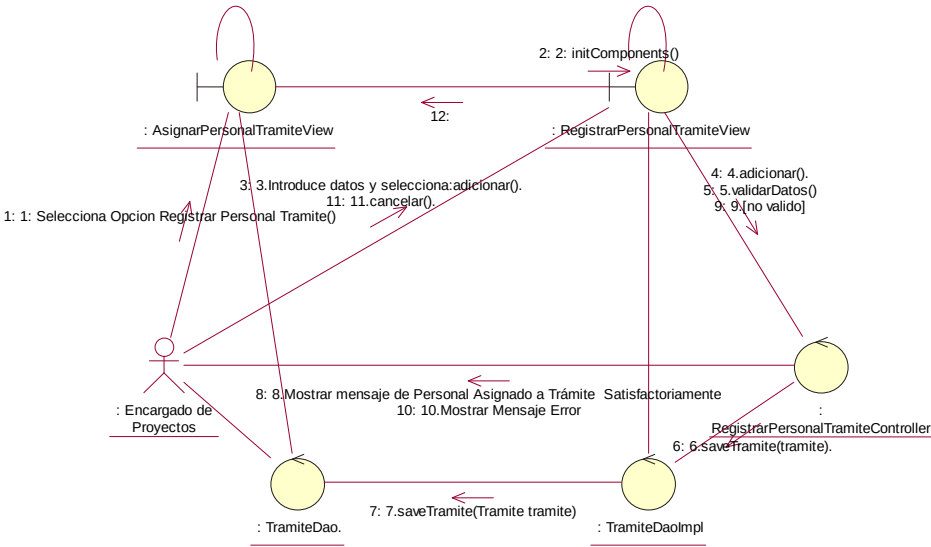


Figura 353: Diagrama de Colaboración: Asignar Personal Trámite

III.2.3.2 .36 Diagrama de Colaboración Listar Mensajes

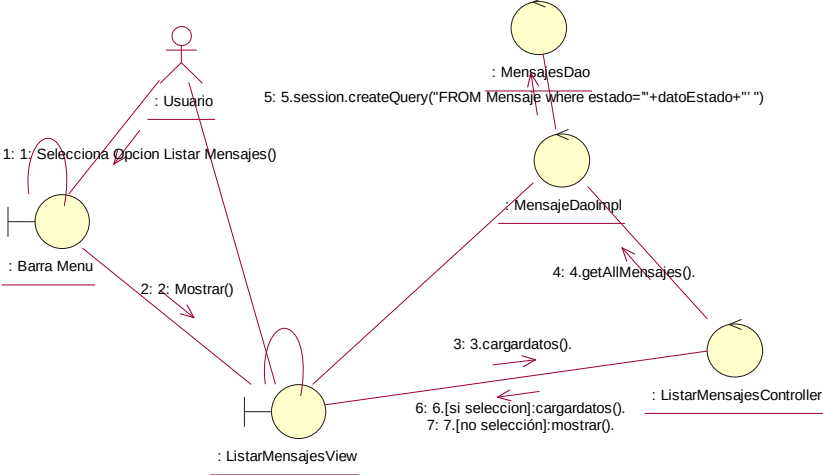


Figura 354: Diagrama de Colaboración: Listar Mensajes

III.2.3.2 .37 Diagrama de Colaboración Enviar Mensaje

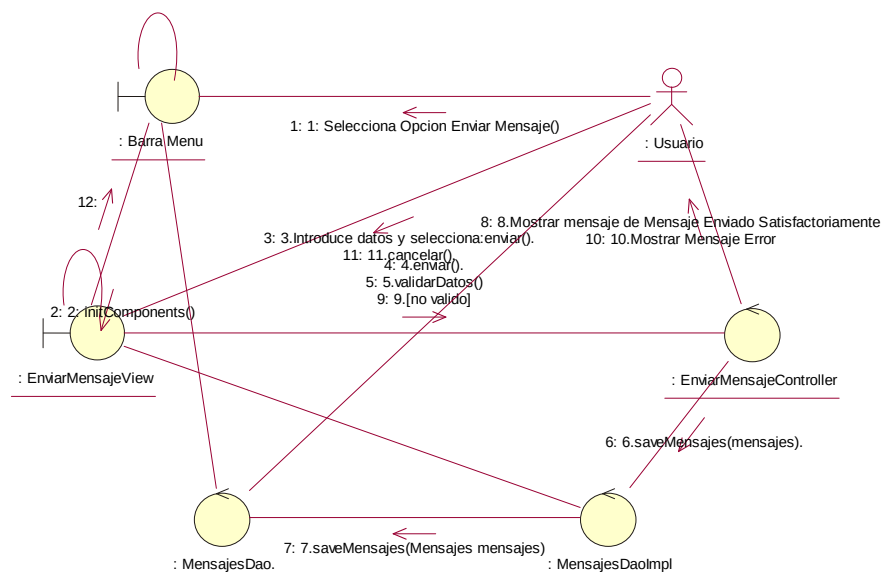


Figura 355: Diagrama de Colaboración: Enviar Mensaje

III.2.3.2 .38 Diagrama de Colaboración Eliminar Mensaje

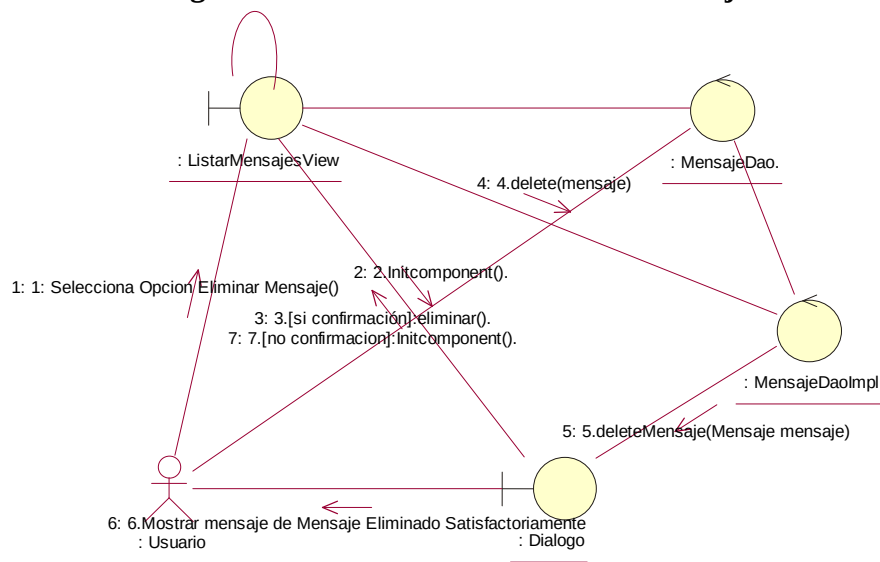


Figura 356: Diagrama de Colaboración: Eliminar Mensaje

III.2.3.2 .39 Diagrama de Colaboración Ver Mensaje

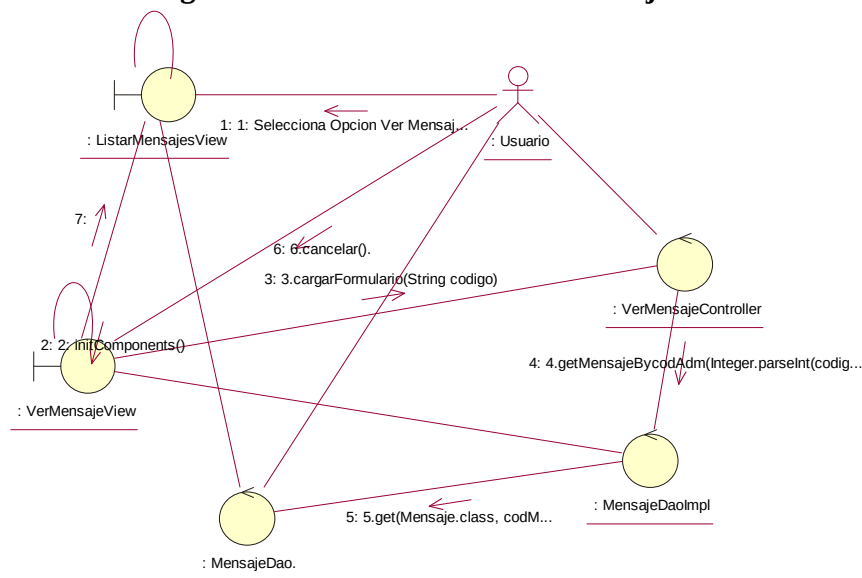


Figura 357: Diagrama de Colaboración: Ver Mensaje

III.2.3.2 .40 Diagrama de Colaboración Administrar Observaciones

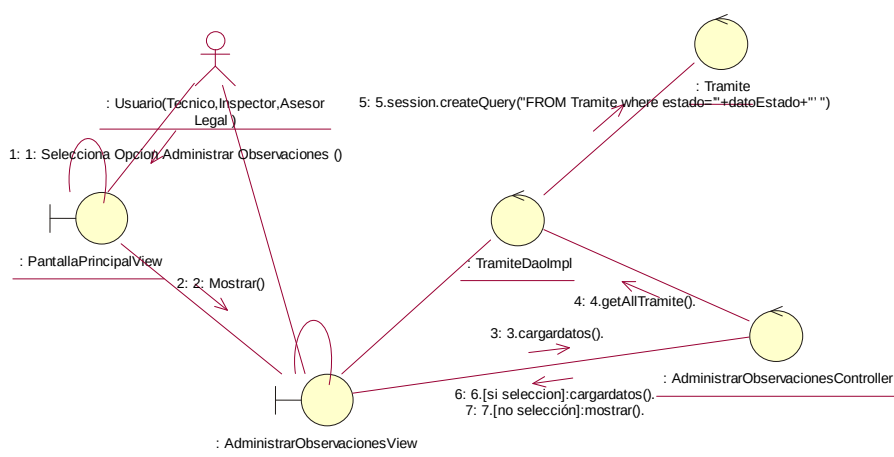


Figura 358: Diagrama de Colaboración: Administrar Observaciones

III.2.3.2 .41 Diagrama de Colaboración Adicionar Observaciones

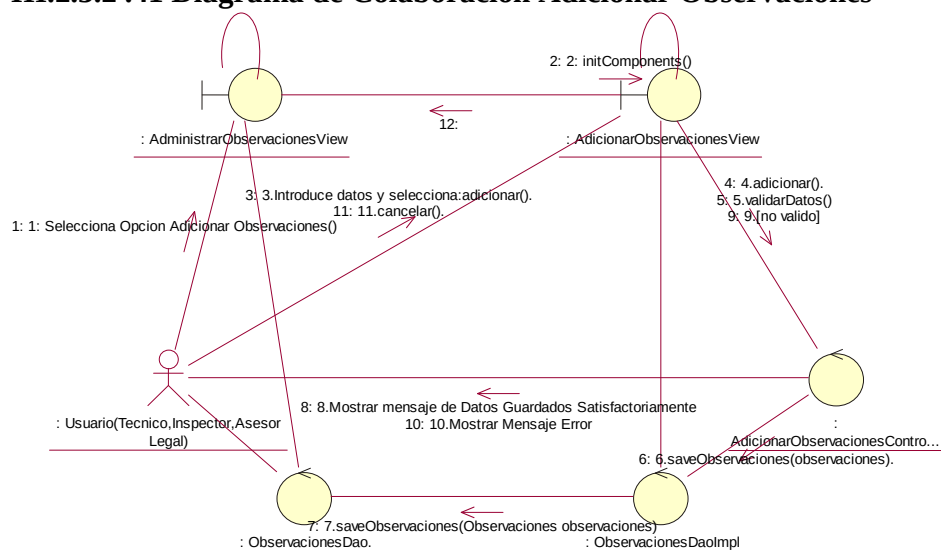


Figura 359: Diagrama de Colaboración: Adicionar Observaciones

III.2.3.2 .42 Diagrama de Colaboración Modificar Observaciones

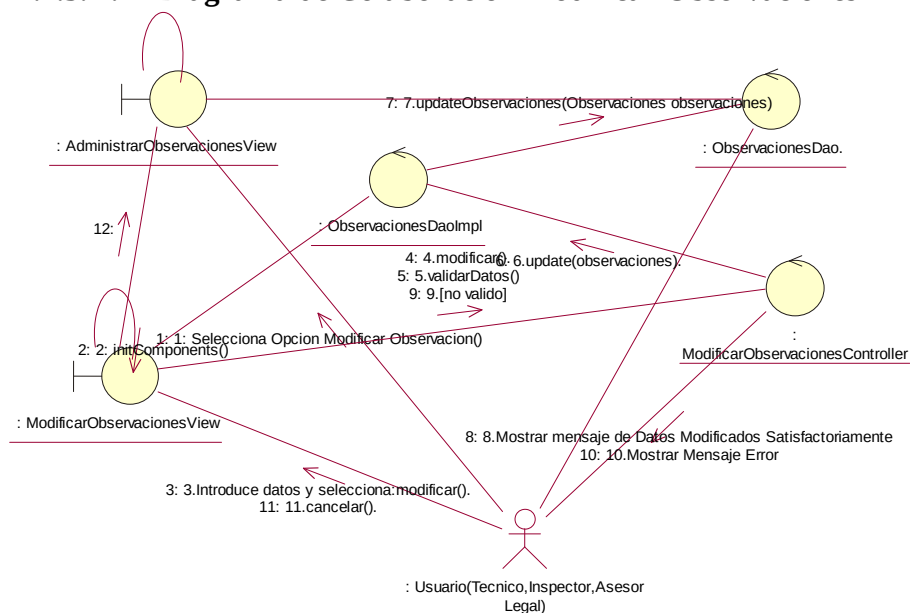


Figura 360: Diagrama de Colaboración: Modificar Observaciones

III.2.3.2 .43 Diagrama de Colaboración Ver Observaciones

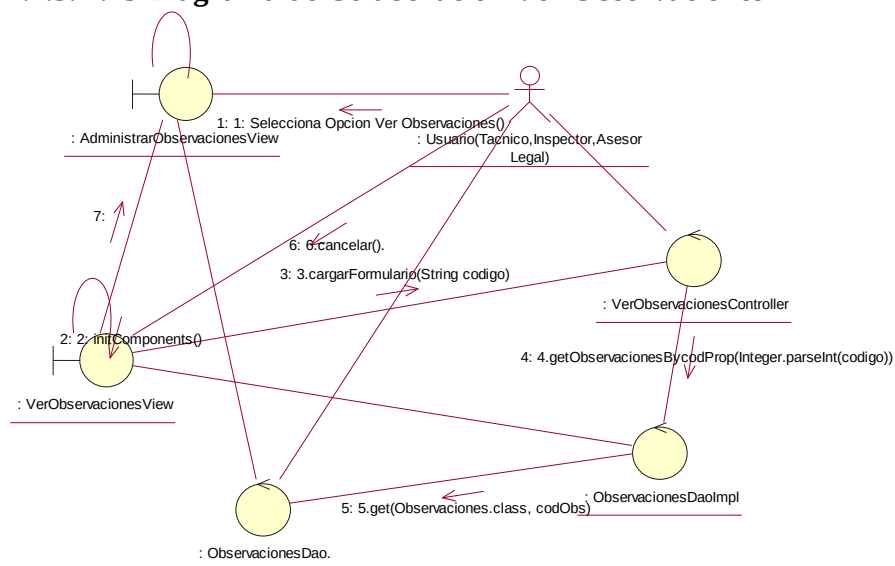


Figura 361: Diagrama de Colaboración: Ver Observaciones

III.2.3.2 .44 Diagrama de Colaboración Administrar Observaciones Jefe Proyectos

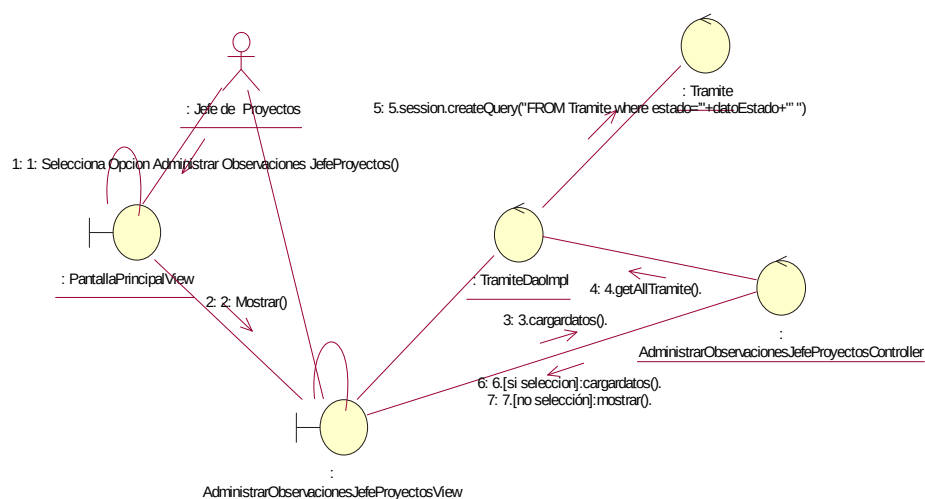


Figura 362: Diagrama de Colaboración: Adm. Observaciones Jefe Lotes

III.2.3.2 .45 Diagrama de Colaboración Adicionar Observaciones Jefe Proyectos

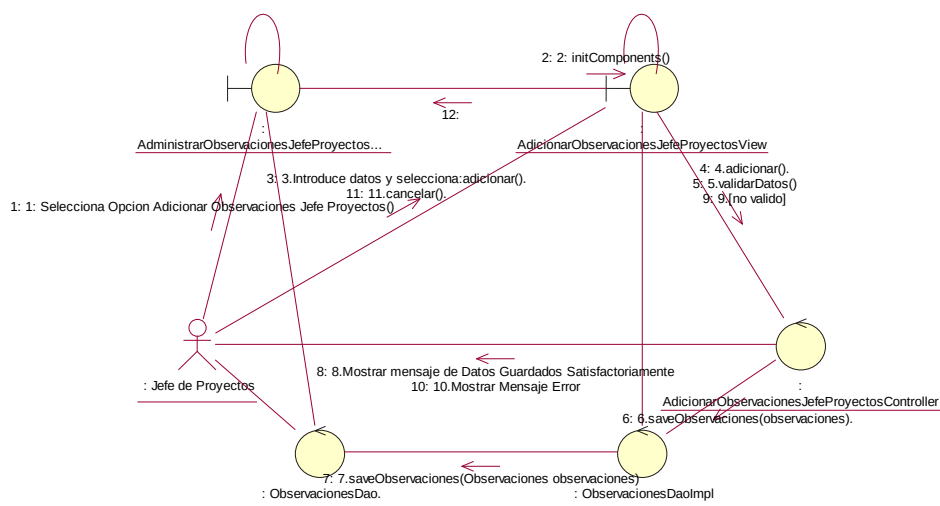


Figura 363: Diagrama de Colaboración: Adicionar Obser. Jefe Lotes

III.2.3.2 .46 Diagrama de Colaboración Modificar Observaciones Jefe Proyectos

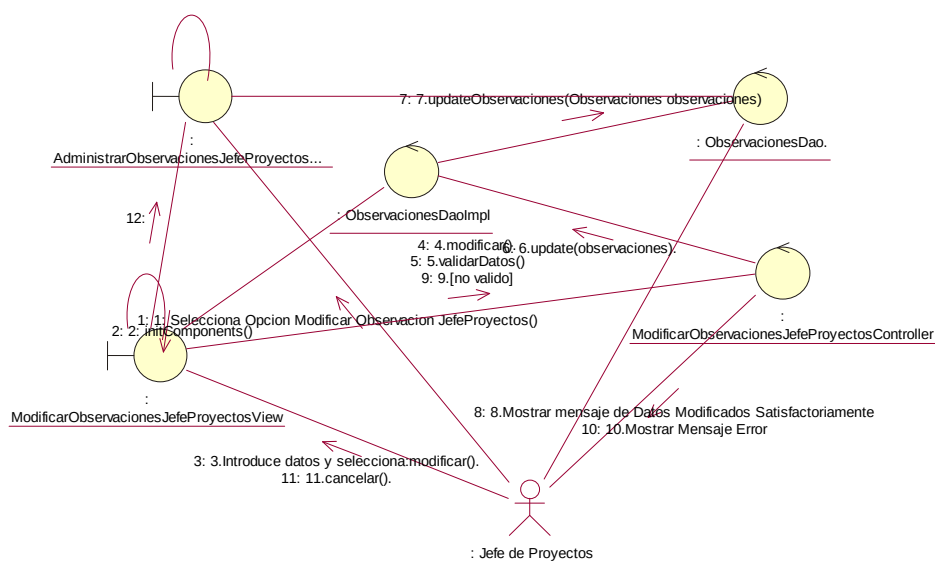


Figura 364: Diagrama de Colaboración: Modificar Obser. Jefe Lotes

III.2.3.2 .47 Diagrama de Colaboración Ver Observaciones Jefe Proyectos

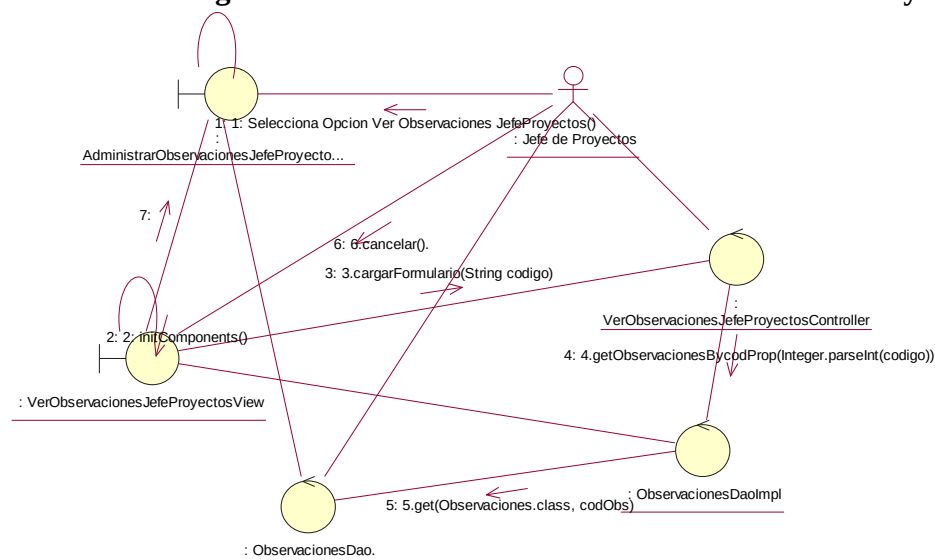


Figura 365: Diagrama de Colaboración: Ver Observaciones Jefe Lotes

III.2.3.2 .48 Diagrama de Colaboración Dar Alta

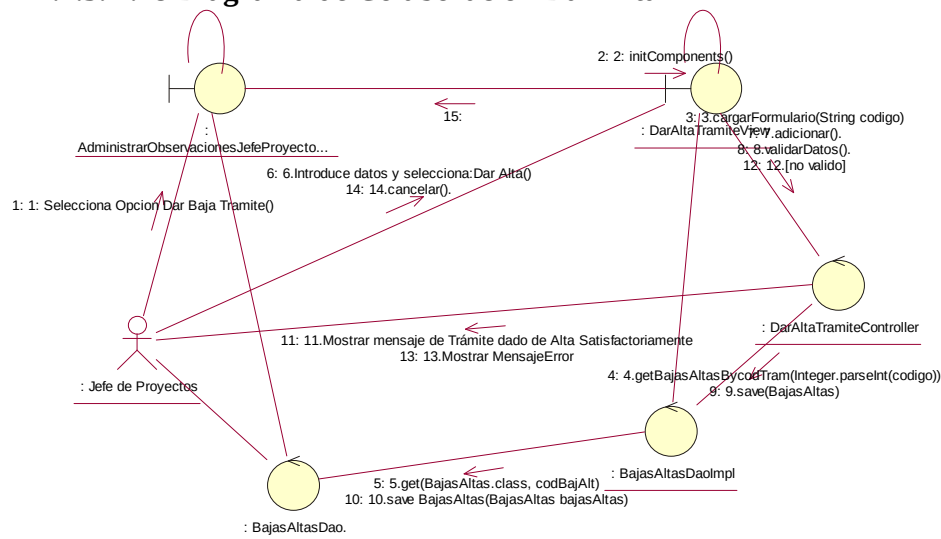


Figura 366: Diagrama de Colaboración: Dar Alta Trámite

III.2.3.2 .49 Diagrama de Colaboración Aprobar Tramite

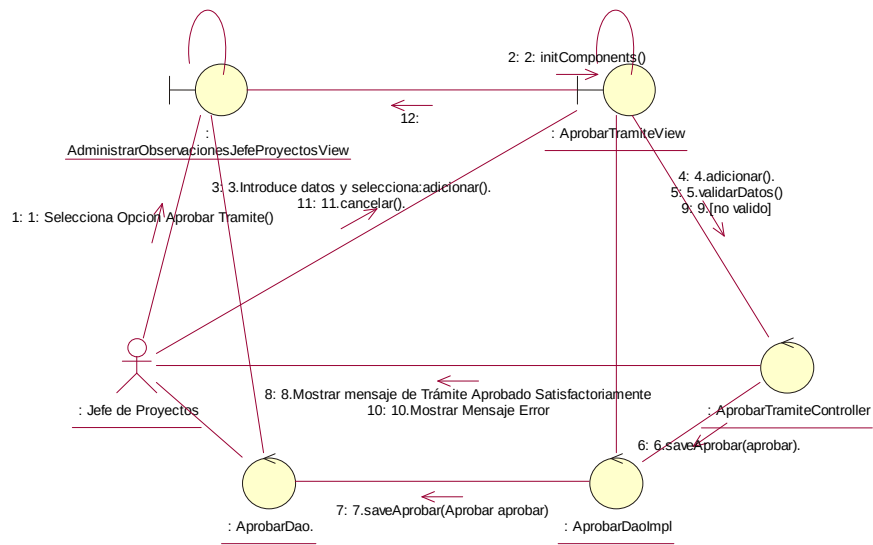


Figura 367: Diagrama de Colaboración: Aprobar Trámite

III.2.3.3 MODULO DE LOTES

III.2.3.3.1 Diagrama de Colaboración Administrar Propietarios

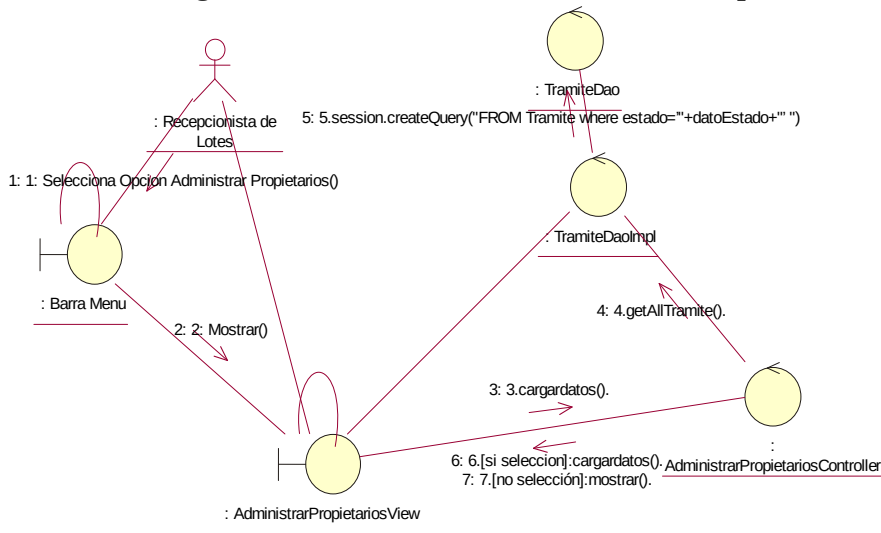


Figura 368: Diagrama de Colaboración: Administrar Propietarios

III.2.3.3.2 Diagrama de Colaboración Adicionar Propietarios

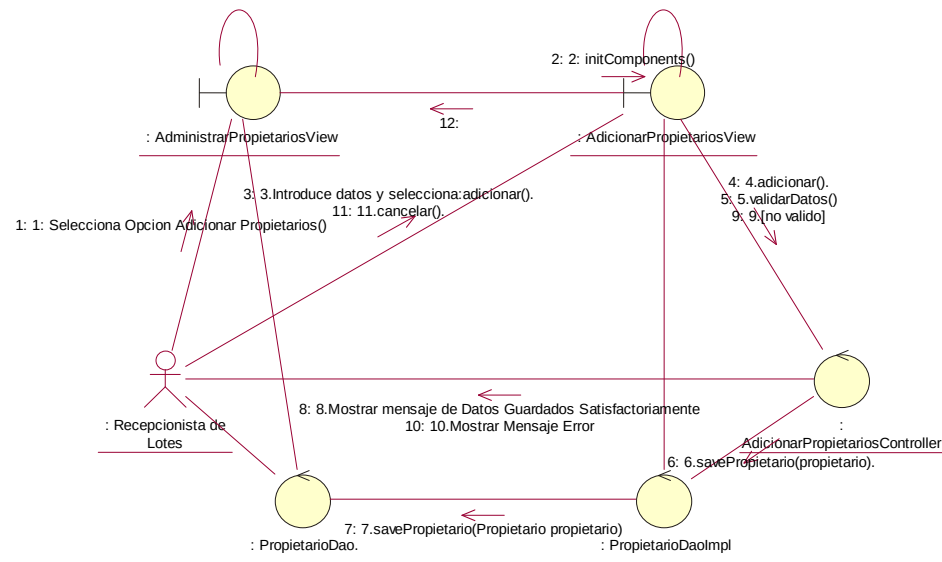


Figura 369: Diagrama de Colaboración: Adicionar Propietarios

III.2.3.3.3 Diagrama de Colaboración Adicionar Projectista

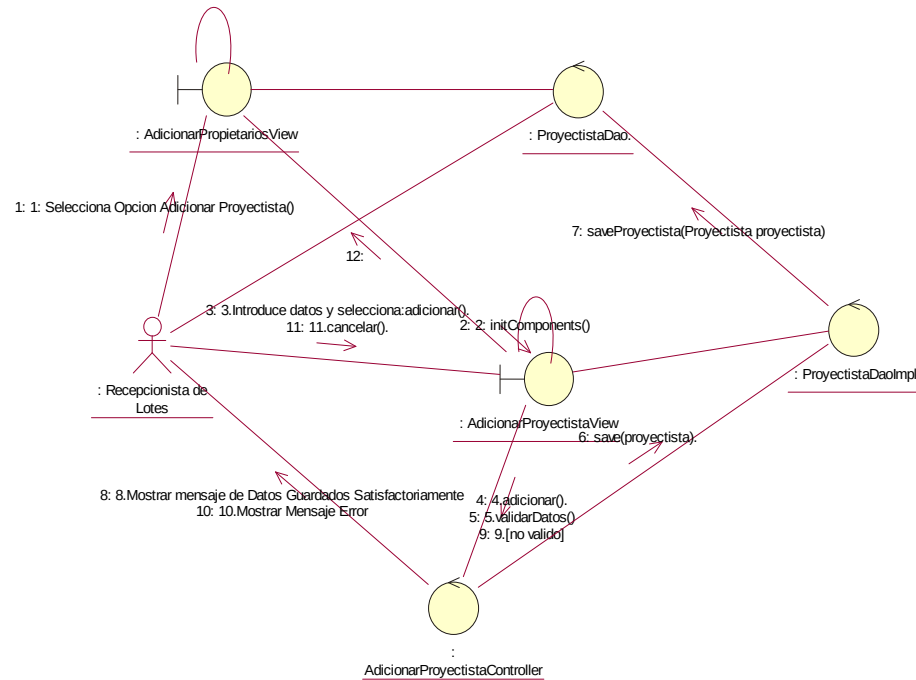


Figura 370: Diagrama de Colaboración: Adicionar Projectista

III.2.3.3.4 Diagrama de Colaboración Adicionar Apoderado

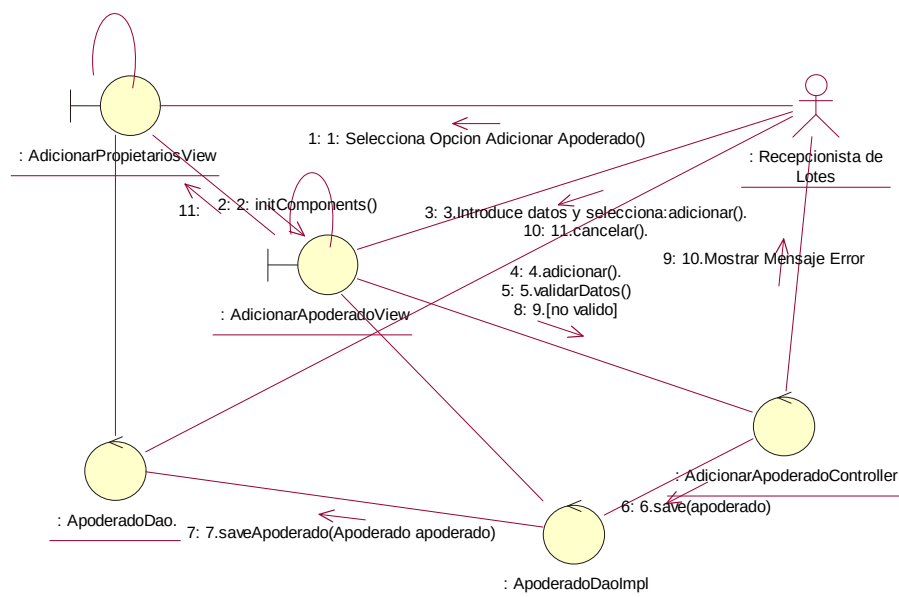


Figura 371: Diagrama de Colaboración: Adicionar Apoderado

III.2.3.3.5 Diagrama de Colaboración Modificar Propietarios

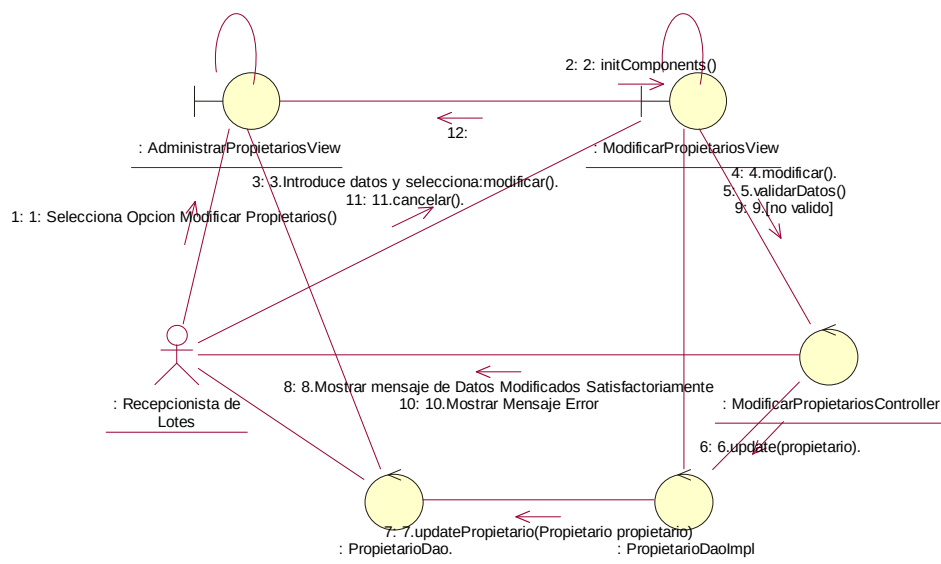


Figura 372: Diagrama de Colaboración: Modificar Propietarios

III.2.3.3.6 Diagrama de Colaboración Modificar Projectista

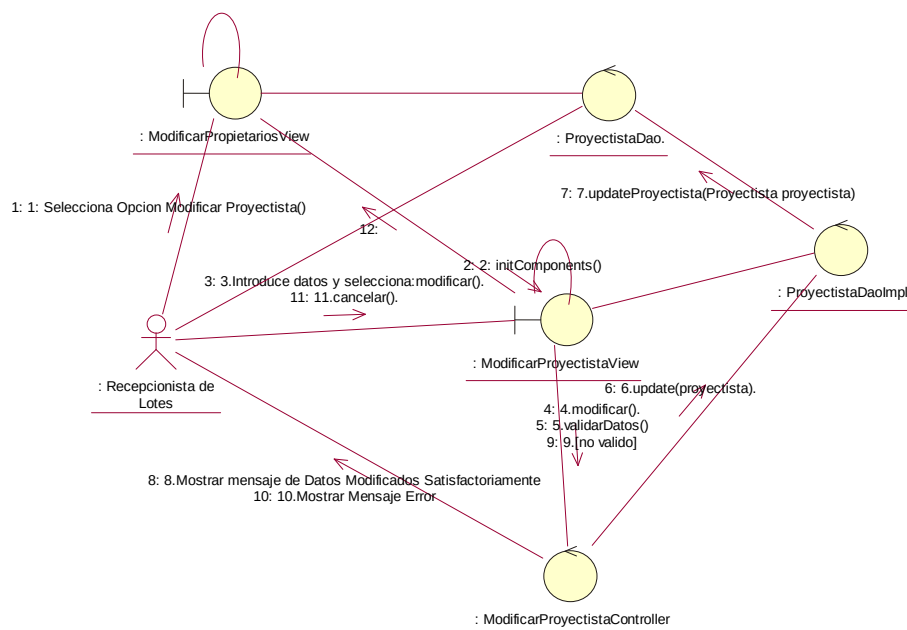


Figura 373: Diagrama de Colaboración: Modificar Projectista

III.2.3.3.7 Diagrama de Colaboración Modificar Apoderado

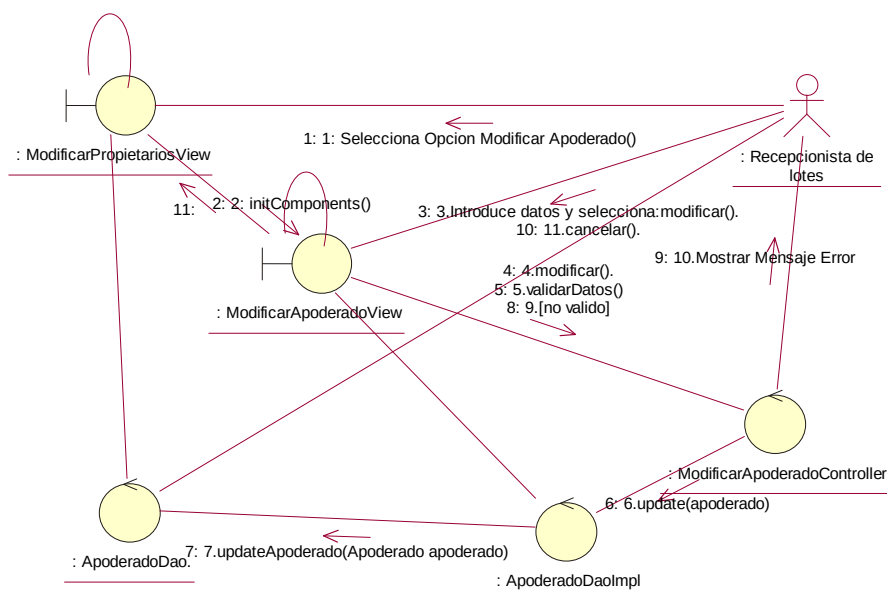


Figura 374: Diagrama de Colaboración: Modificar Apoderado

III.2.3.3.8 Diagrama de Colaboración Ver Propietarios

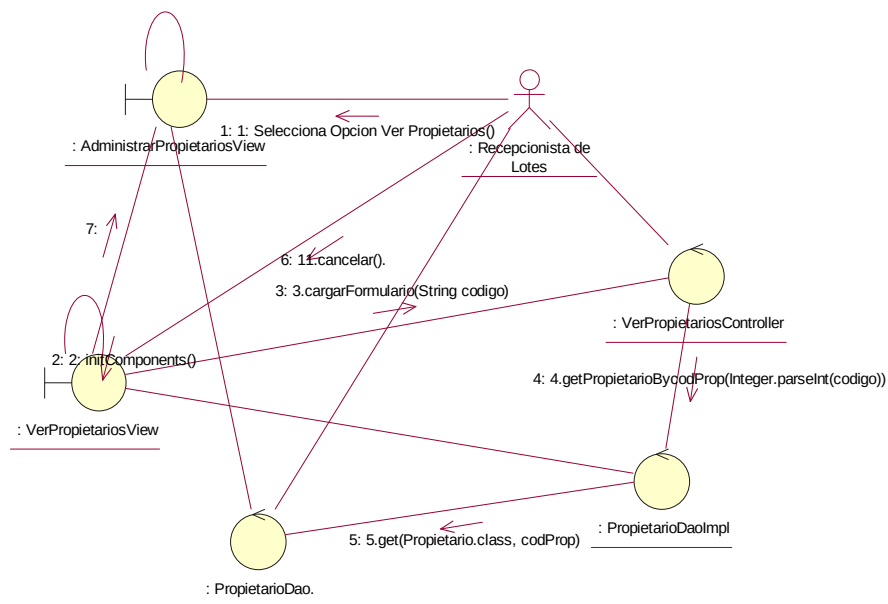


Figura 375: Diagrama de Colaboración: Ver Propietarios

III.2.3.3.9 Diagrama de Colaboración Administrar Observaciones Generales

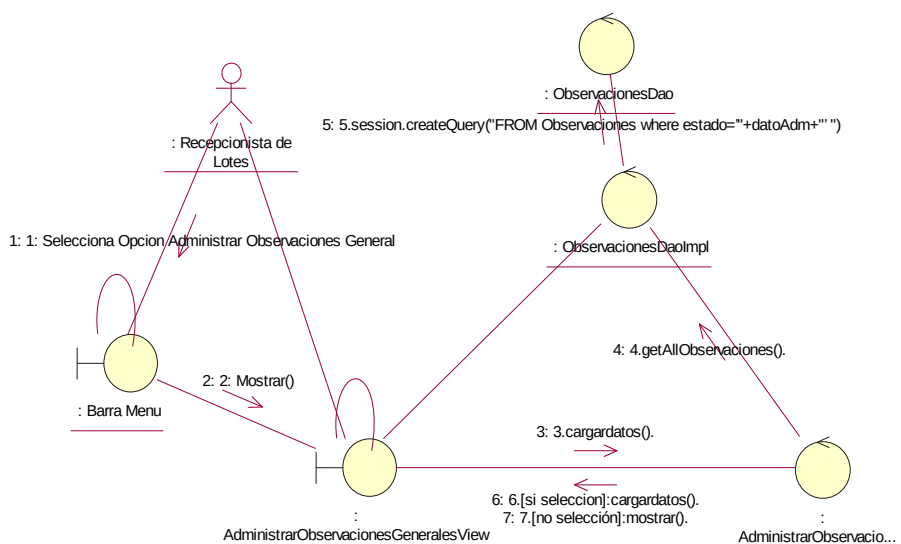


Figura 376: Diagrama de Colaboración: Administrar Observaciones Generales

III.2.3.3.10 Diagrama de Colaboración Ver Observaciones Generales

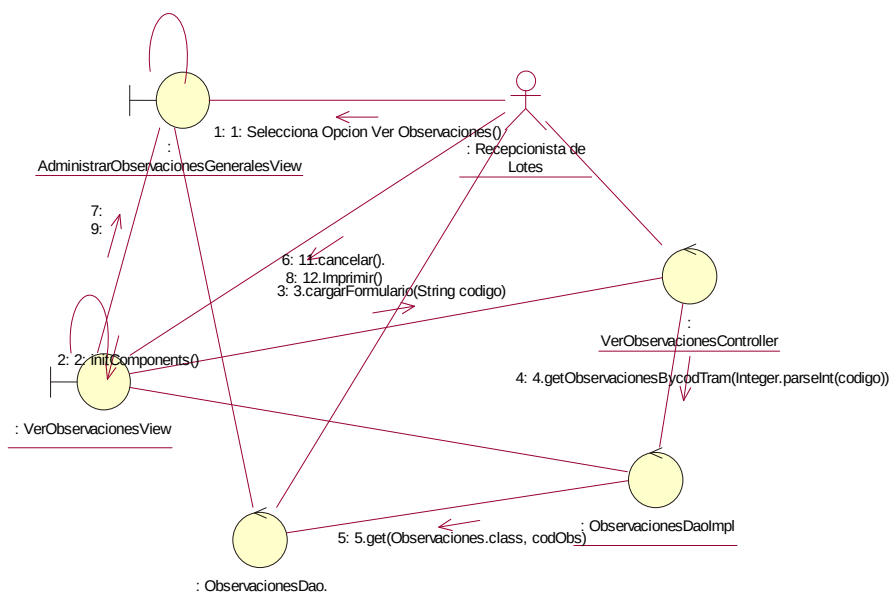


Figura 377: Diagrama de Colaboración: Ver Observaciones Generales

III.2.3.3.11 Diagrama de Secuencia Dar Baja Trámite

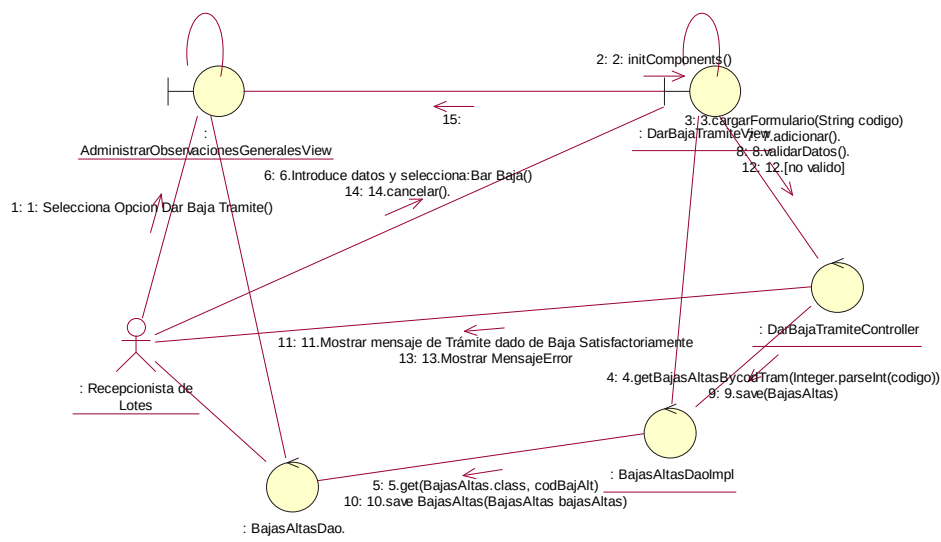


Figura 378: Diagrama de Colaboración: Dar Baja Trámite

III.2.3.3.12 Diagrama de Colaboración Reasignar Trámite

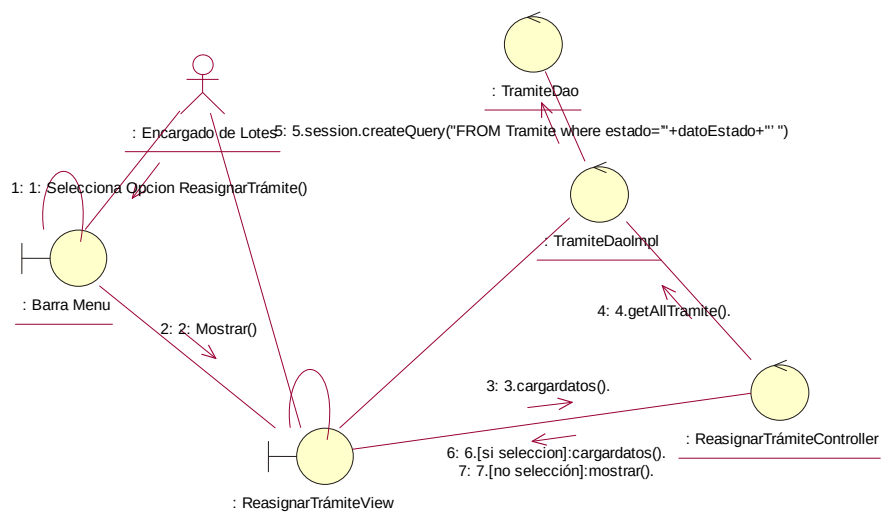


Figura 379: Diagrama de Colaboración: Reasignar Trámite

III.2.3.3.13 Diagrama de Colaboración Registrar Tramite

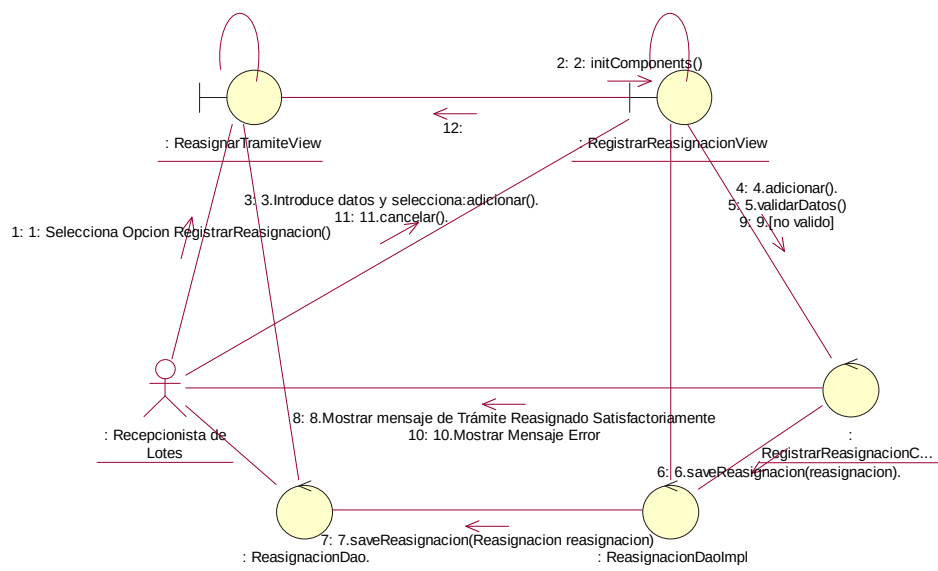


Figura 380: Diagrama de Colaboración: Registrar Tramite

III.2.3.3.14 Diagrama de Colaboración Ver Reasignación

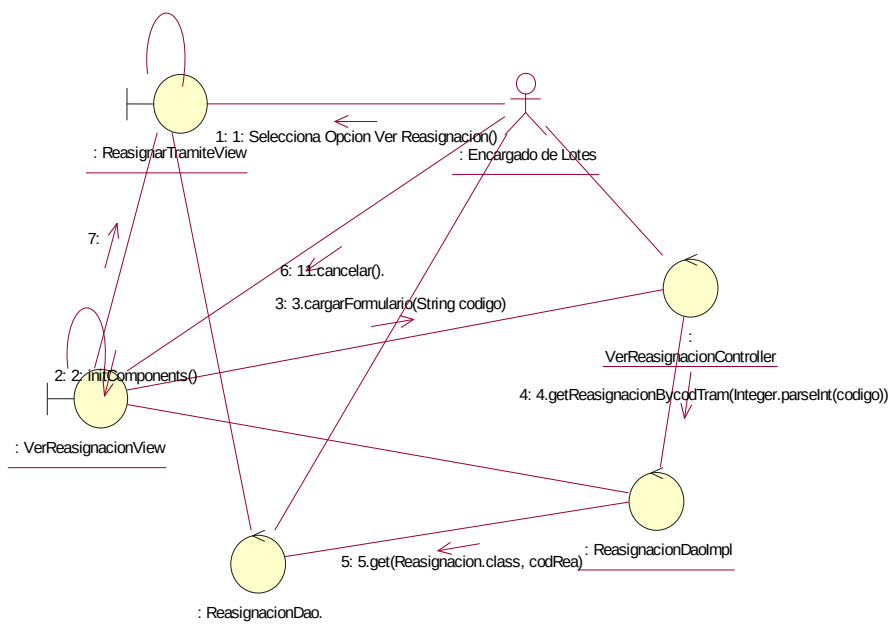


Figura 381: Diagrama de Colaboración: Ver Reasignación

III.2.3.3.15 Diagrama de Colaboración Ver Datos Trámite

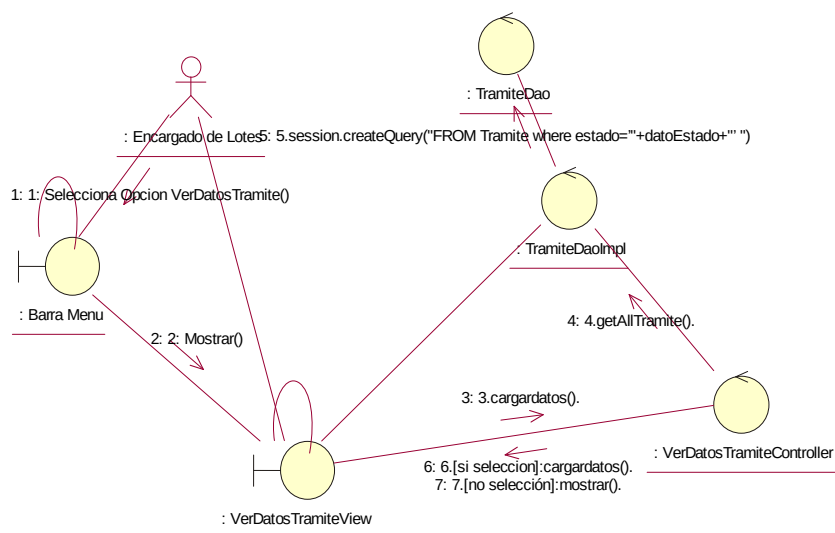


Figura 382: Diagrama de Colaboración: Ver Datos Trámite

III.2.3.3.16 Diagrama de Colaboración Ver Estado Tramite

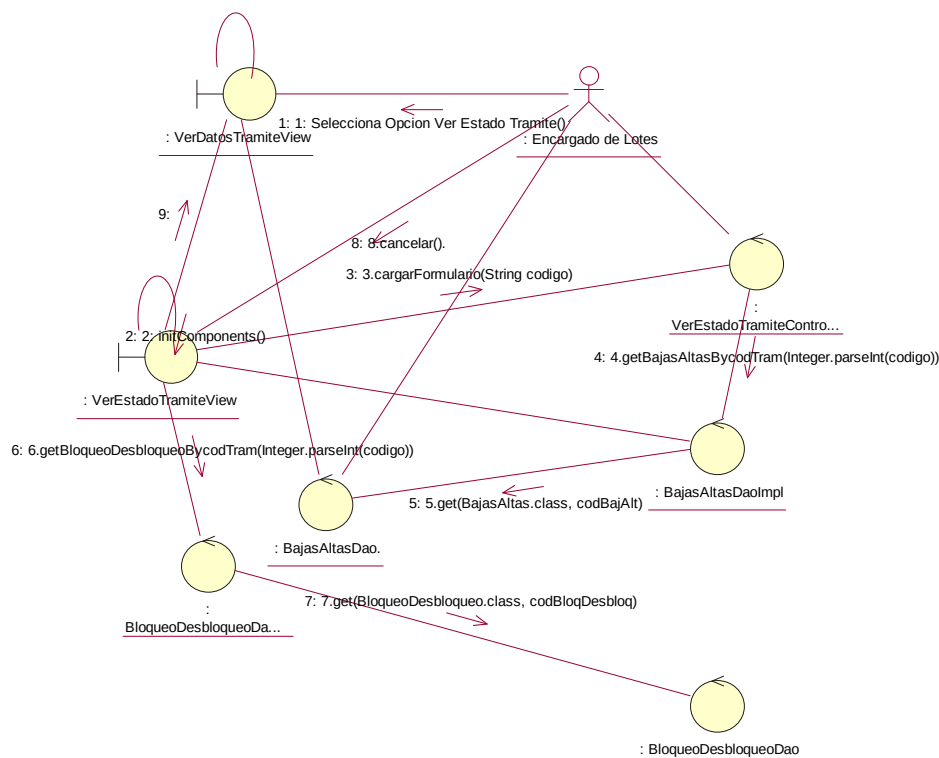


Figura 383: Diagrama de Colaboración: Ver Estado Tramite

III.2.3.3.17 Diagrama de Colaboración Ver Detalle Observación

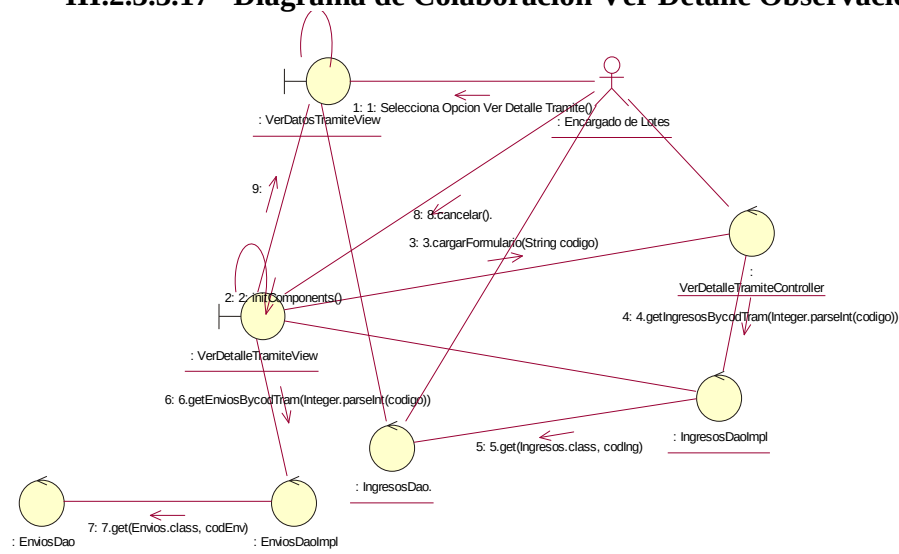


Figura 384: Diagrama de Colaboración: Ver Detalle Observación

III.2.3.3.18 Diagrama de Colaboración Administrar Ingresos

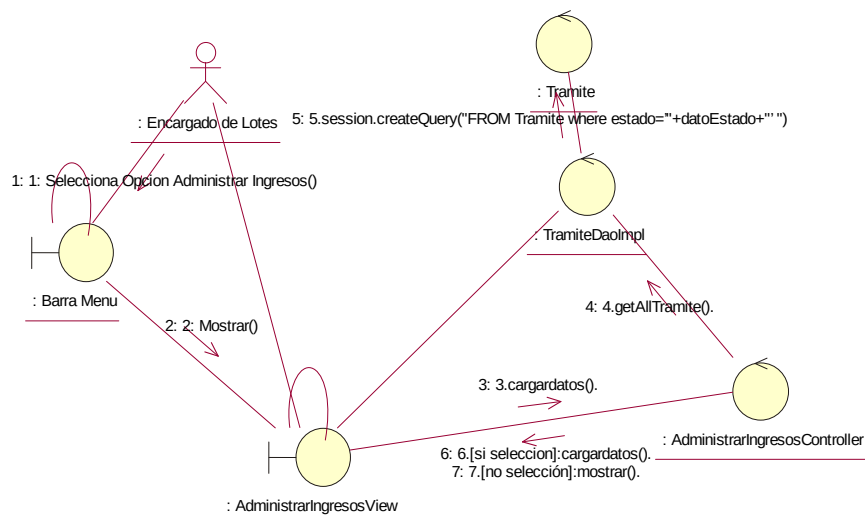


Figura 385: Diagrama de Colaboración: Administrar Ingresos

III.2.3.3.19 Diagrama de Colaboración Registrar Ingresos

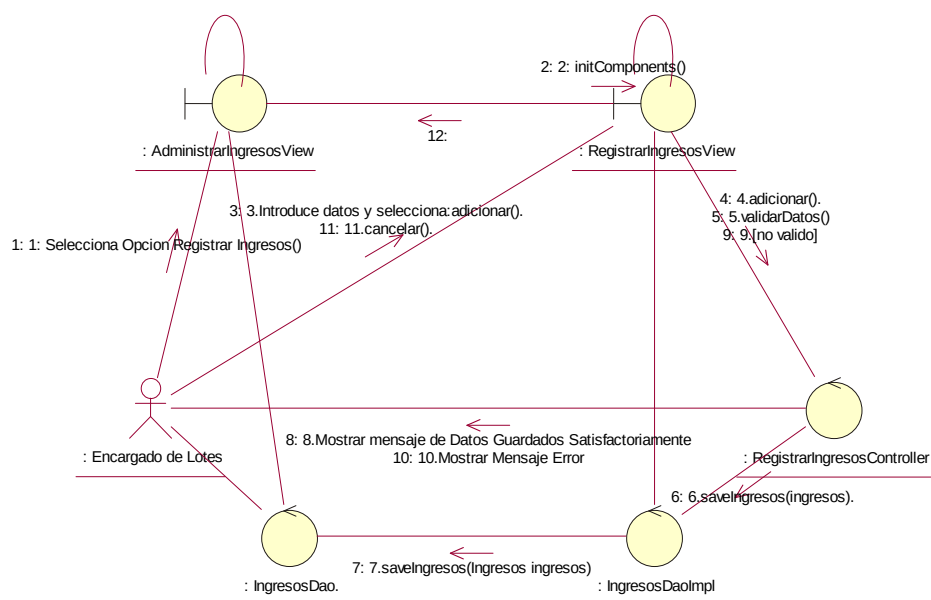


Figura 386: Diagrama de Colaboración: Registrar Ingresos

III.2.3.3.20 Diagrama de Colaboración Modificar Ingresos

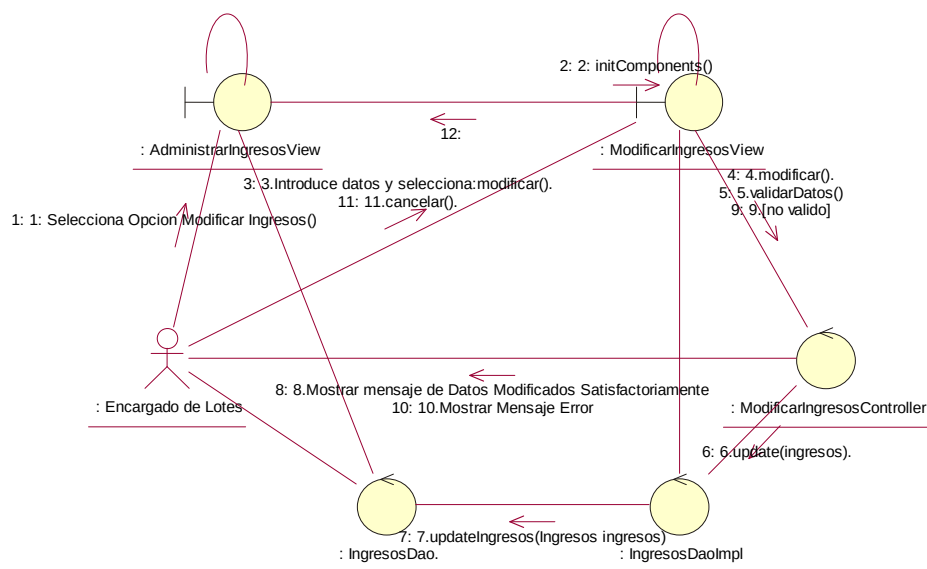


Figura 387: Diagrama de Colaboración: Modificar Ingresos

III.2.3.3.21 Diagrama de Colaboración Ver Ingresos

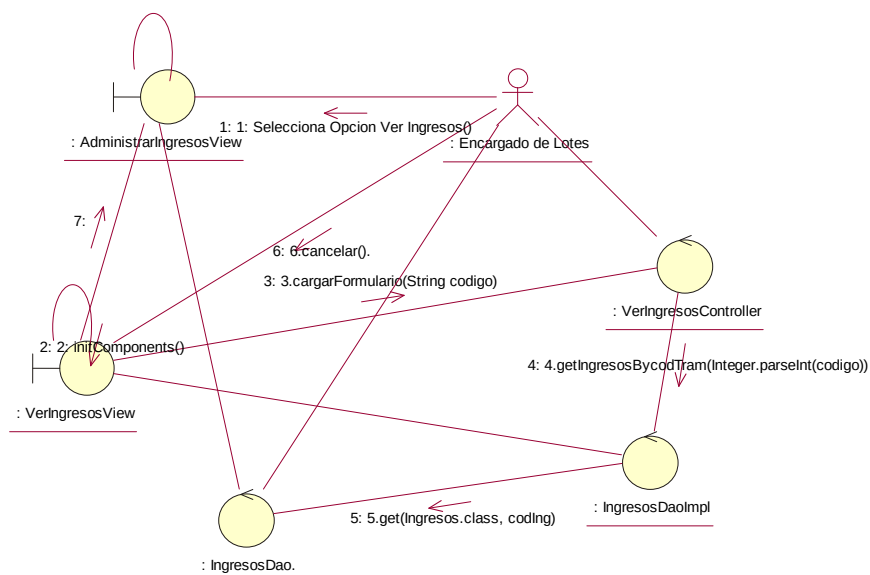


Figura 388: Diagrama de Colaboración: Ver Ingresos

III.2.3.3.22 Diagrama de Colaboración Administrar Salidas

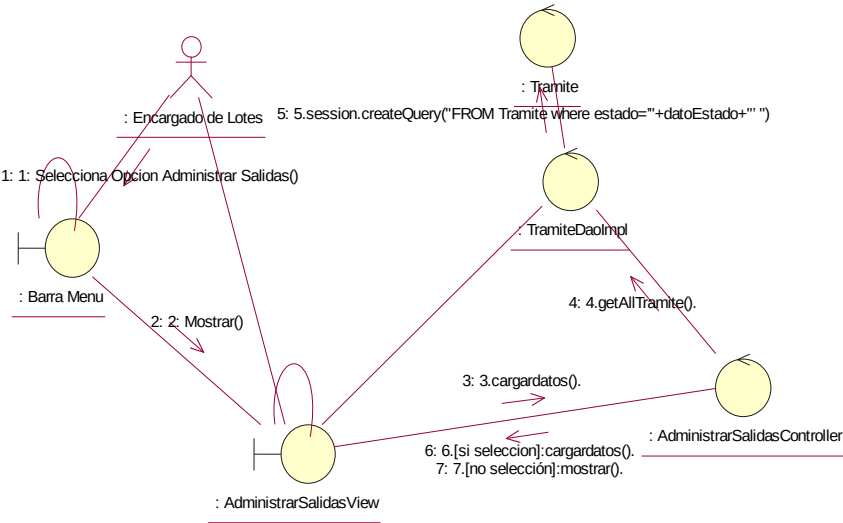


Figura 389: Diagrama de Colaboración: Administrar Salidas

III.2.3.3.23 Diagrama de Colaboración Registrar Salidas

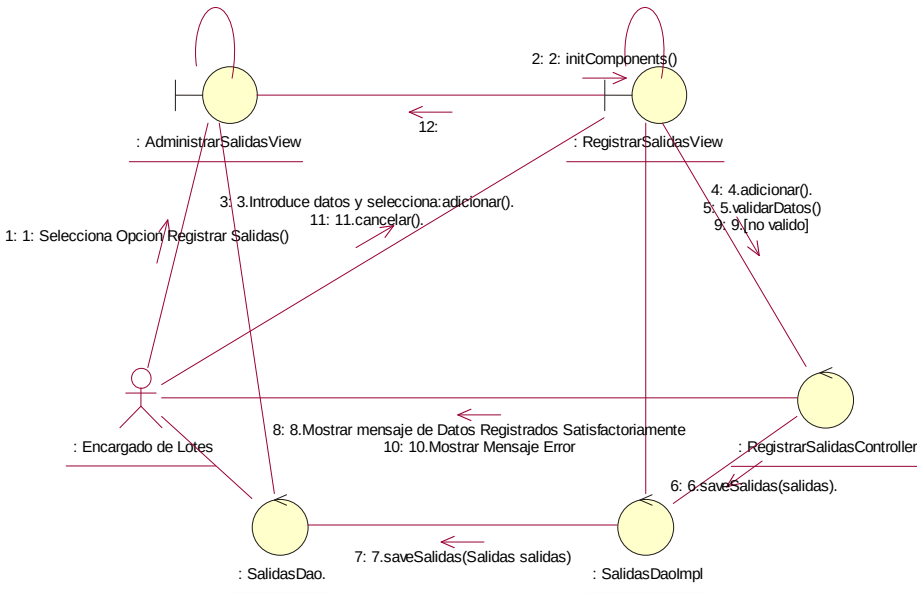


Figura 390: Diagrama de Colaboración: Registrar Salidas

III.2.3.3.24 Diagrama de Colaboración Modificar Salidas

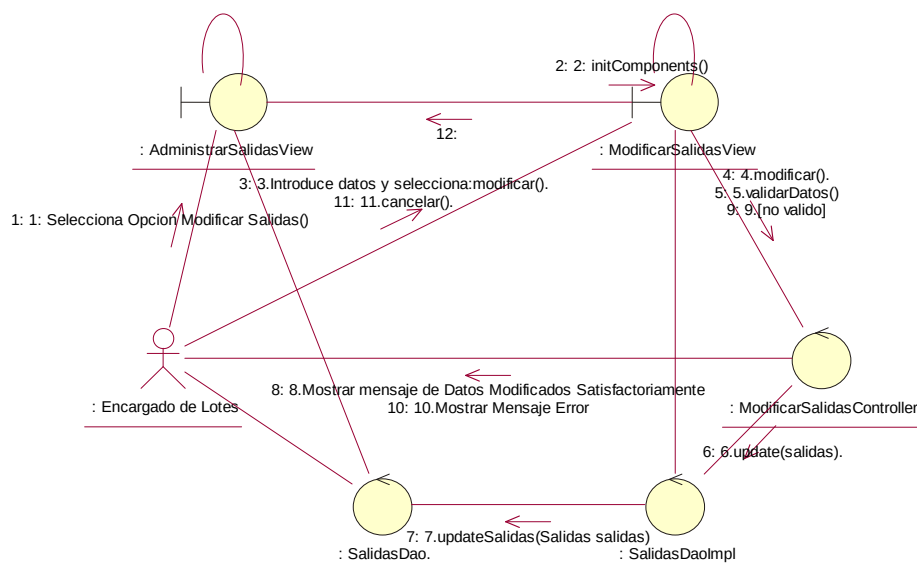


Figura 391: Diagrama de Colaboración: Modificar Salidas

III.2.3.3.25 Diagrama de Colaboración Ver Salidas

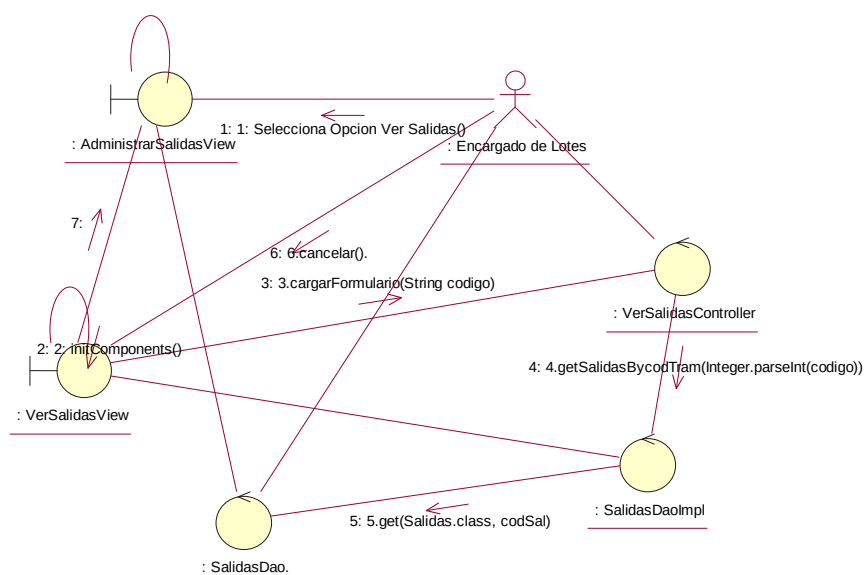


Figura 392: Diagrama de Colaboración: Ver Salidas

III.2.3.3.26 Diagrama de Colaboración Administrar Envíos

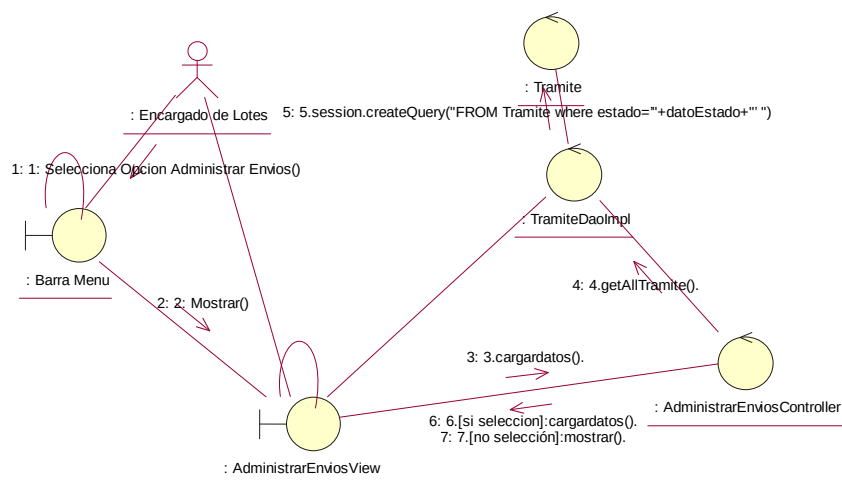


Figura 393: Diagrama de Colaboración: Administrar Envíos

III.2.3.3.27 Diagrama de Colaboración Registrar Envíos

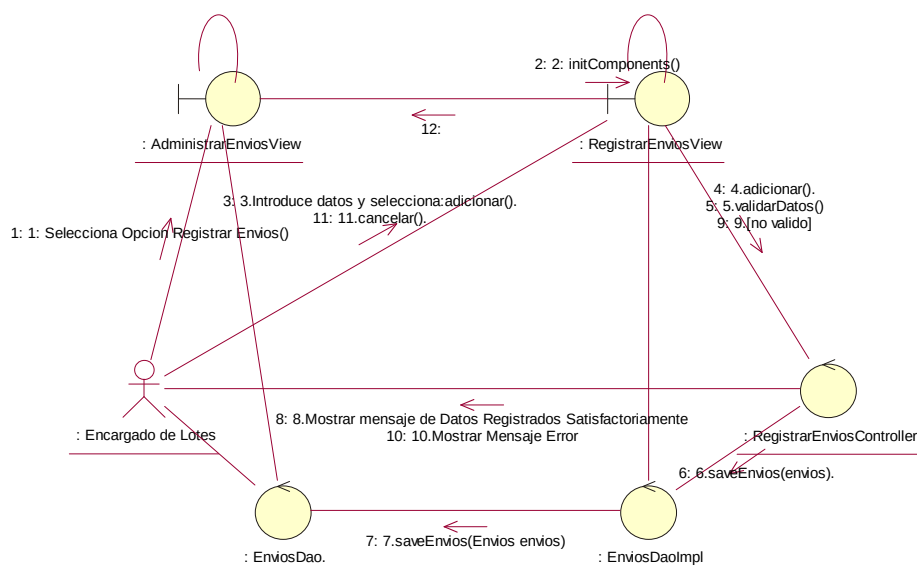


Figura 394: Diagrama de Colaboración: Registrar Envíos

III.2.3.3.28 Diagrama de Colaboración Modificar Envíos

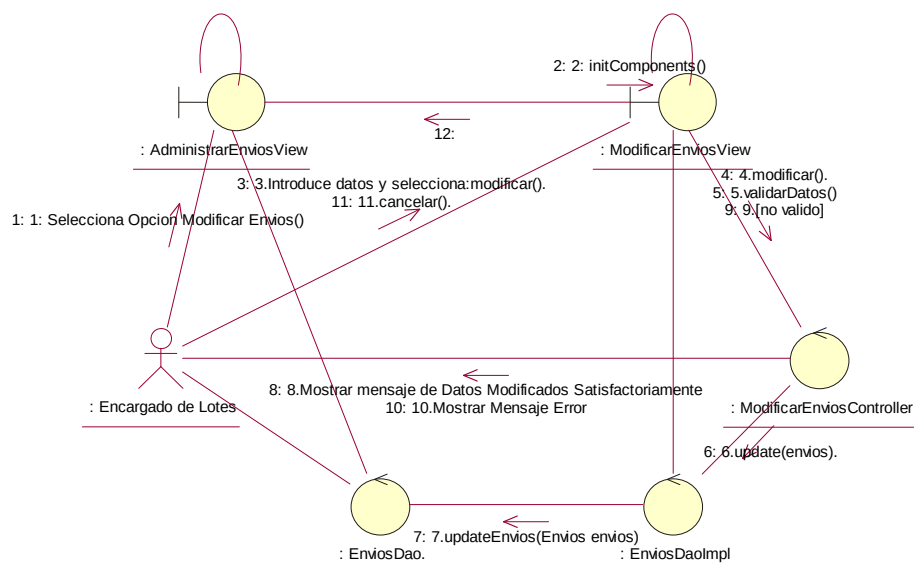


Figura 395: Diagrama de Colaboración: Modificar Envíos

III.2.3.3.29 Diagrama de Colaboración Ver Envíos

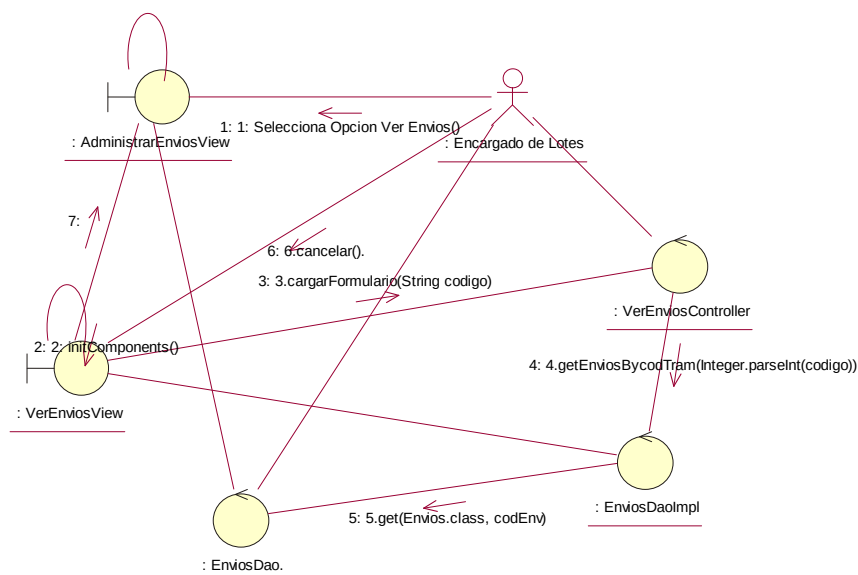


Figura 396: Diagrama de Colaboración: Ver Envíos

III.2.3.3.29 Diagrama de Colaboración Administrar Recepción

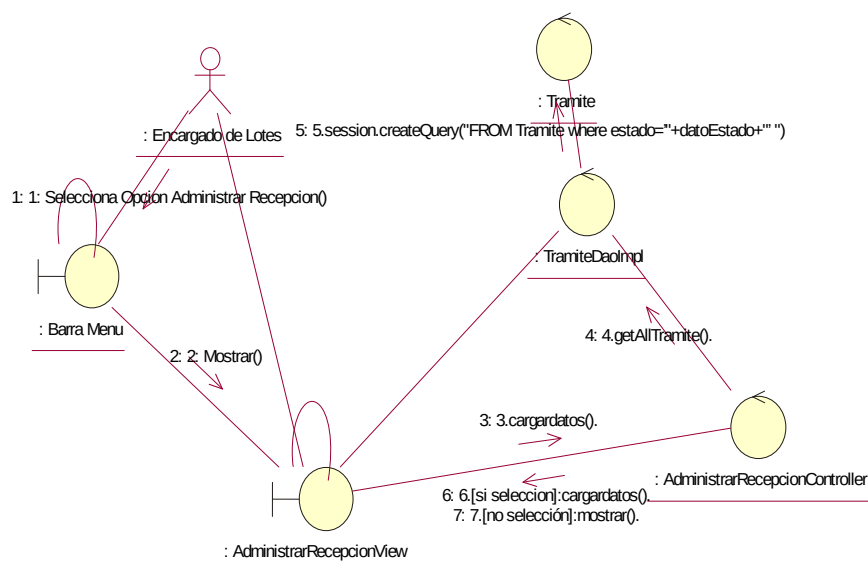


Figura 397: Diagrama de Colaboración: Administrar Recepción

III.2.3.3.30 Diagrama de Colaboración Registrar Recepción

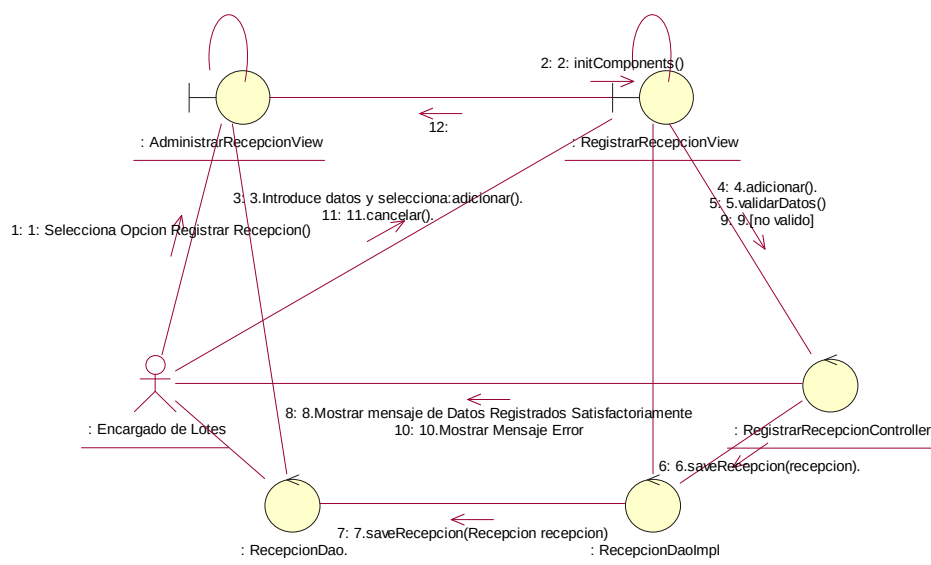


Figura 398: Diagrama de Colaboración: Registrar Recepción

III.2.3.3.31 Diagrama de Colaboración Modificar Recepción

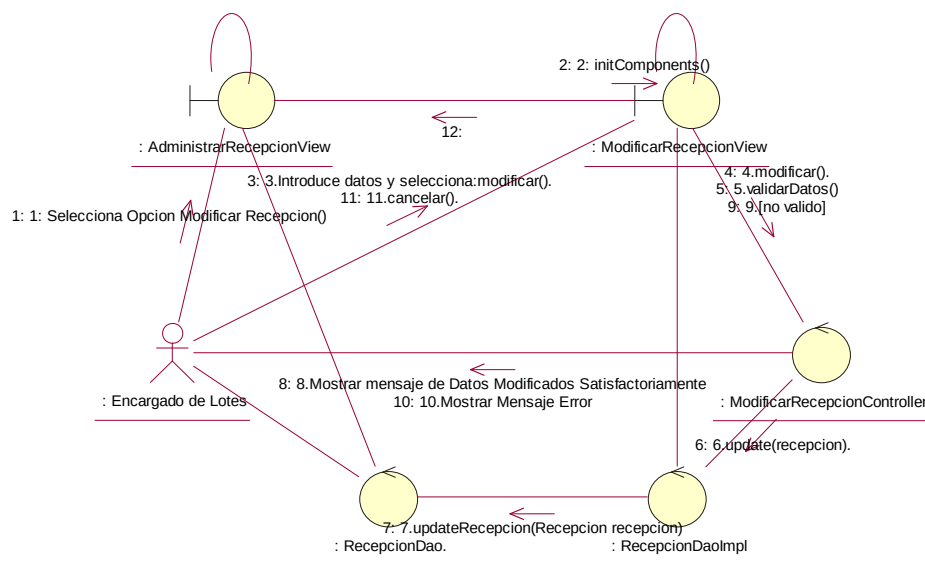


Figura 399: Diagrama de Colaboración: Modificar Recepción

III.2.3.3.32 Diagrama de Colaboración Ver Recepción

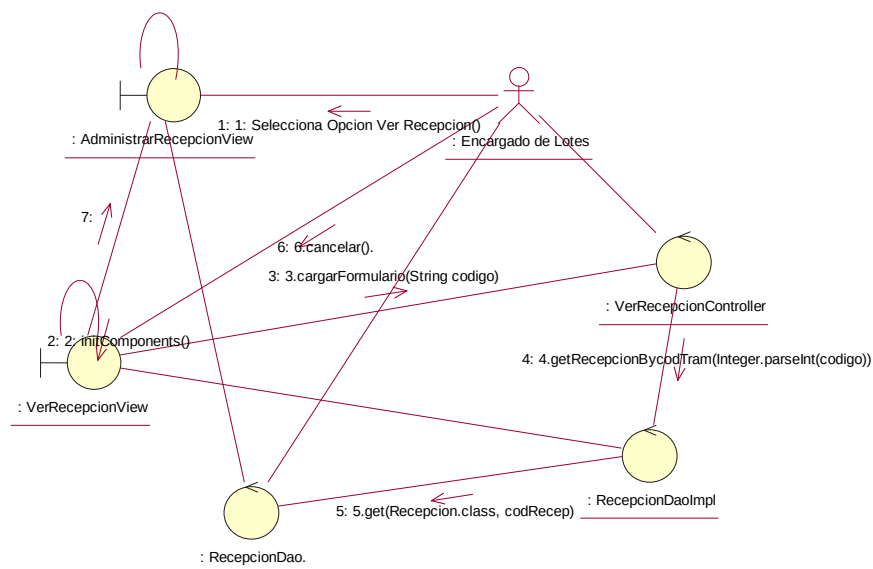


Figura 400: Diagrama de Colaboración: Ver Recepción

III.2.3.3.33 Diagrama de Colaboración Asignar Numero Oficial

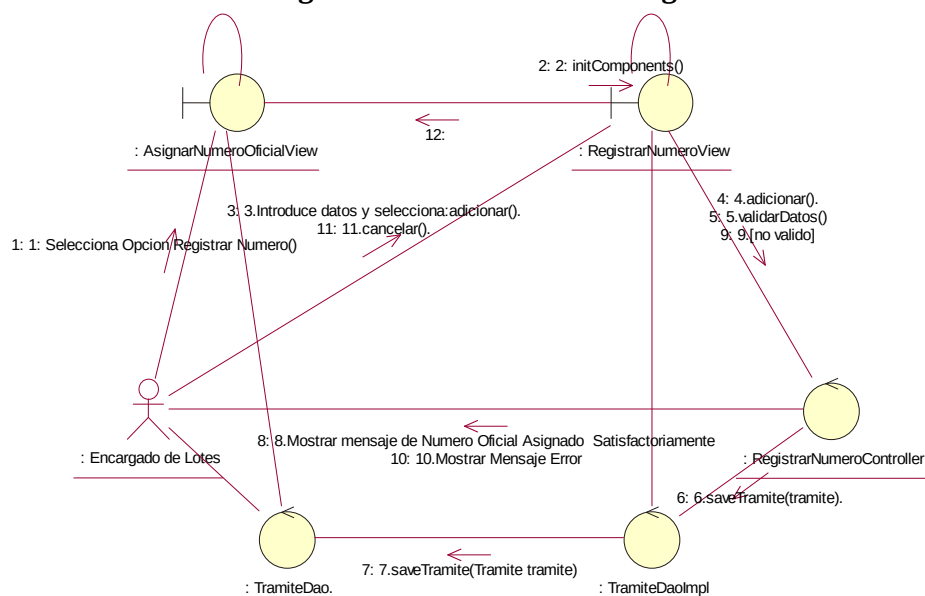


Figura 401: Diagrama de Colaboración: Asignar Numero Oficial

III.2.3.3.34 Diagrama de Colaboración Asignar Personal Trámite

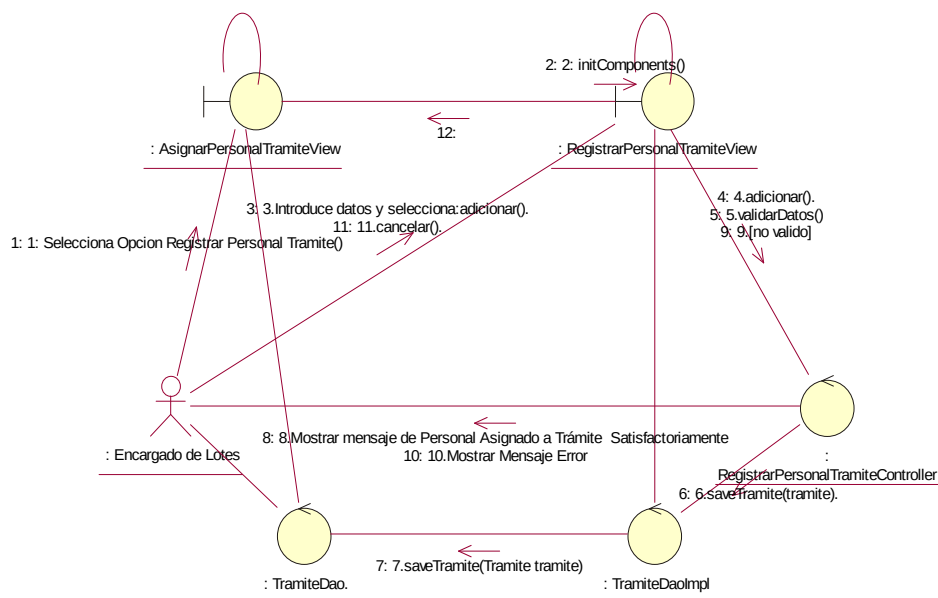


Figura 402: Diagrama de Colaboración: Asignar Personal Trámite

III.2.3.3.35 Diagrama de Colaboración Listar Mensajes

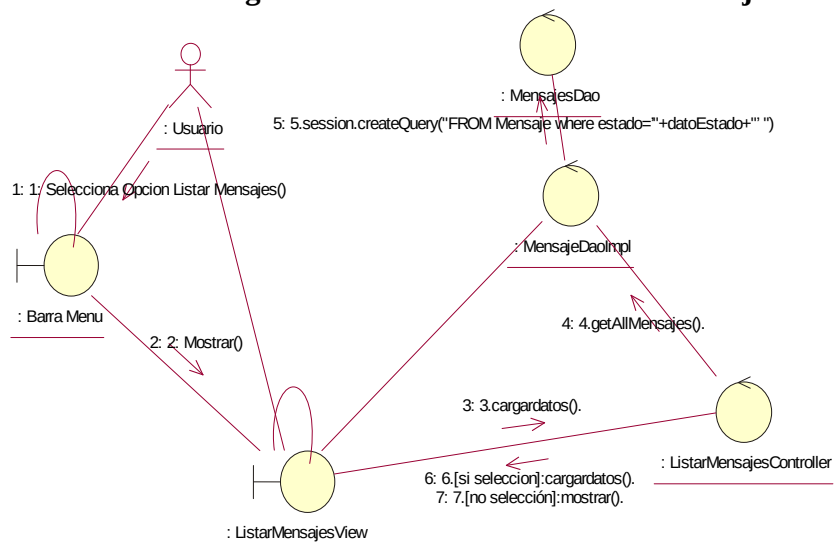


Figura 403: Diagrama de Colaboración: Listar Mensajes

III.2.3.3.36 Diagrama de Colaboración Enviar Mensaje

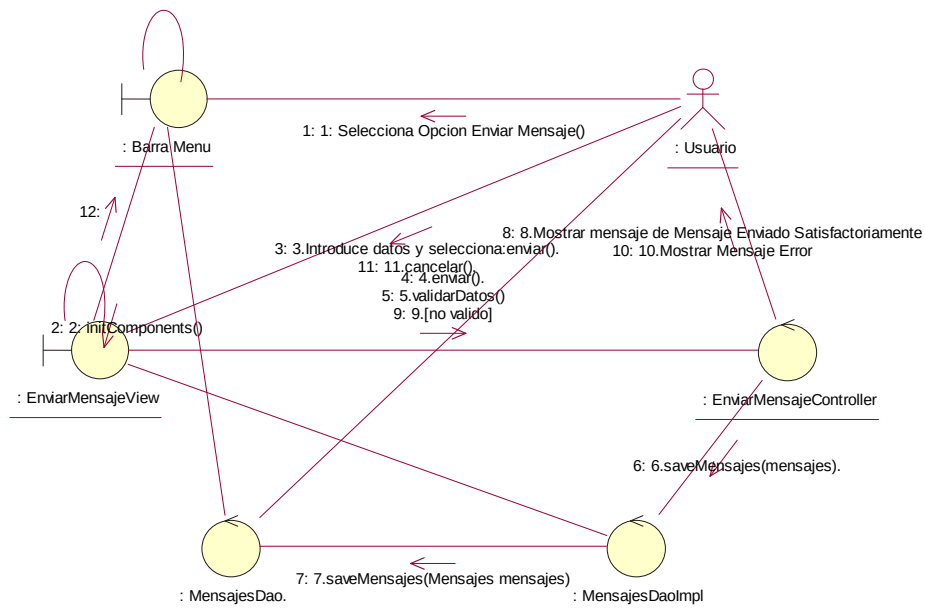


Figura 404: Diagrama de Colaboración: Enviar Mensaje

III.2.3.3.37 Diagrama de Colaboración Eliminar Mensaje

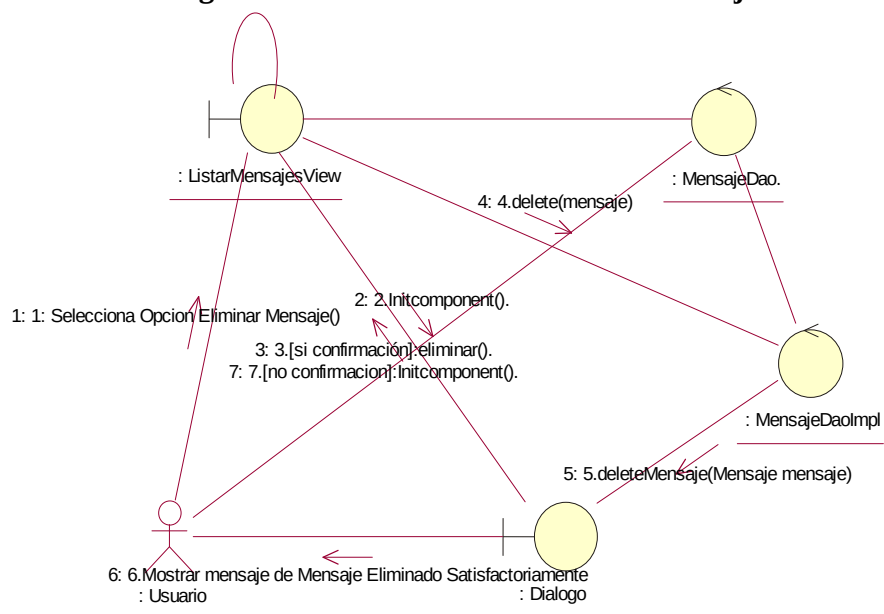


Figura 405: Diagrama de Colaboración: Eliminar Mensaje

III.2.3.3.38 Diagrama de Colaboración Ver Mensaje

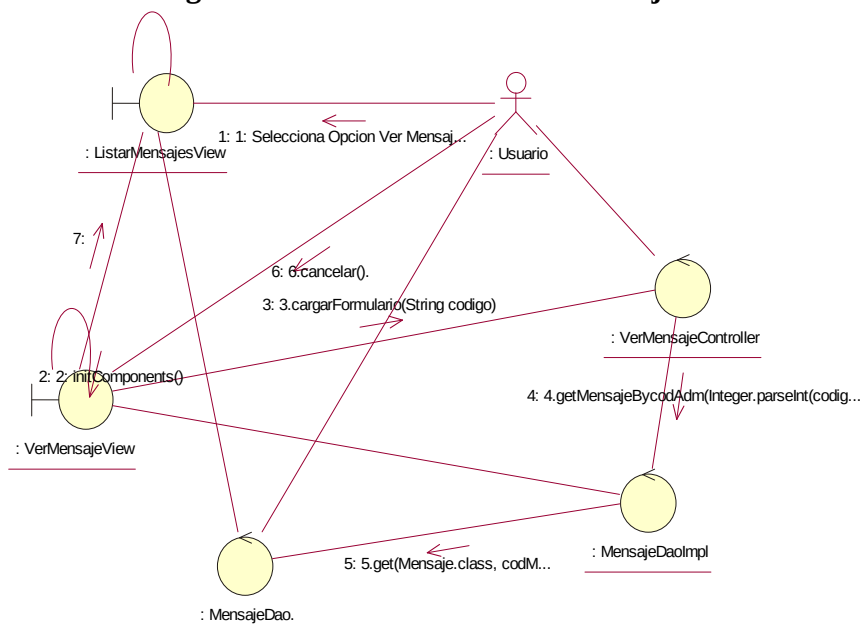


Figura 406: Diagrama de Colaboración: Ver Mensaje

III.2.3.3.39 Diagrama de Colaboración Administrar Observaciones

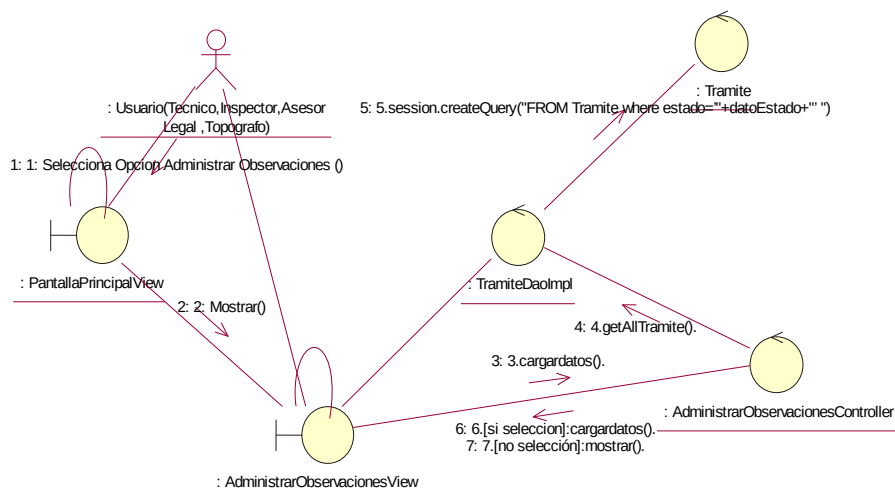


Figura 407: Diagrama de Colaboración: Administrar Observaciones

III.2.3.3.40 Diagrama de Colaboración Adicionar Observaciones

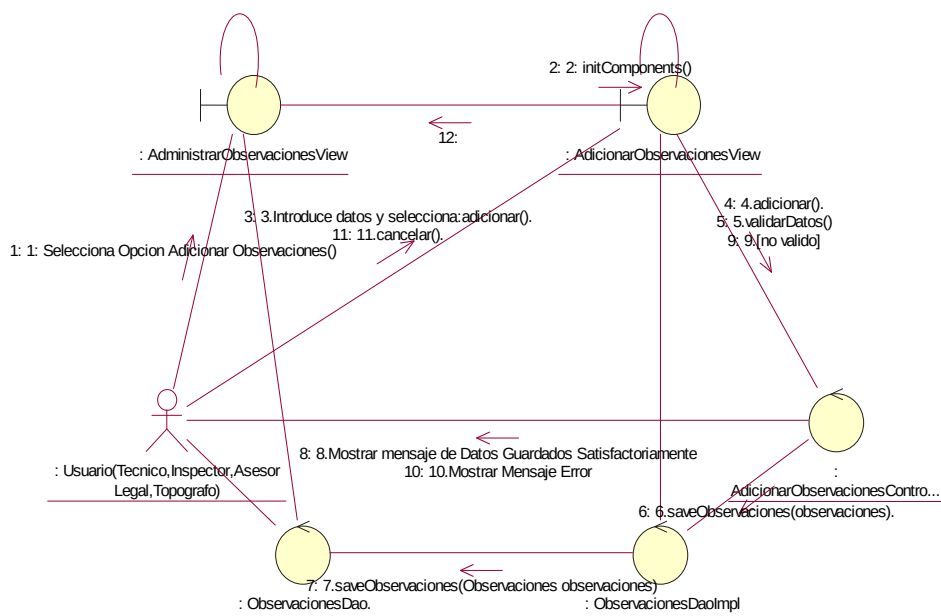


Figura 408: Diagrama de Colaboración: Adicionar Observaciones

III.2.3.3.40 Diagrama de Colaboración Modificar Observaciones

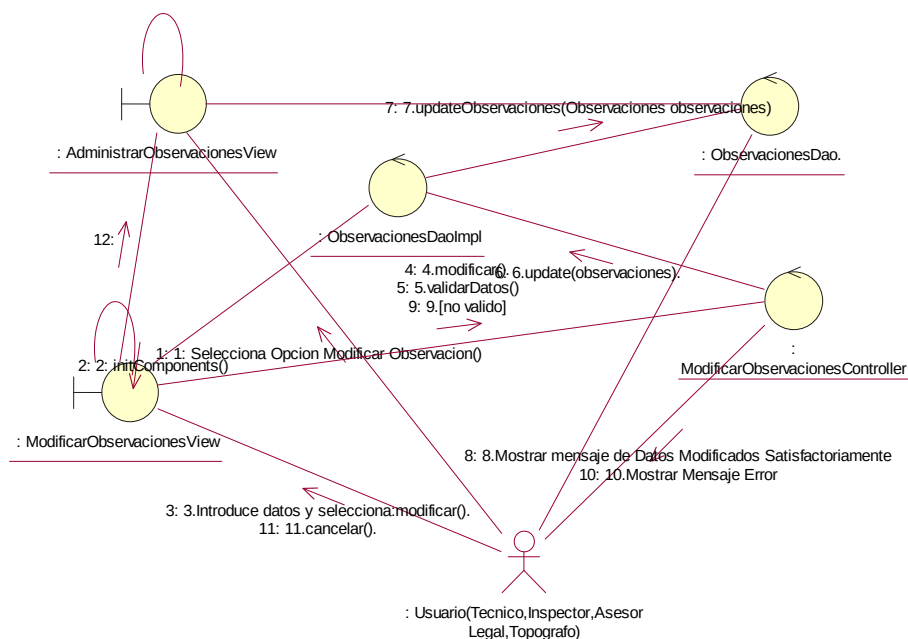


Figura 409: Diagrama de Colaboración: Modificar Observaciones

III.2.3.3.41 Diagrama de Colaboración Ver Observaciones

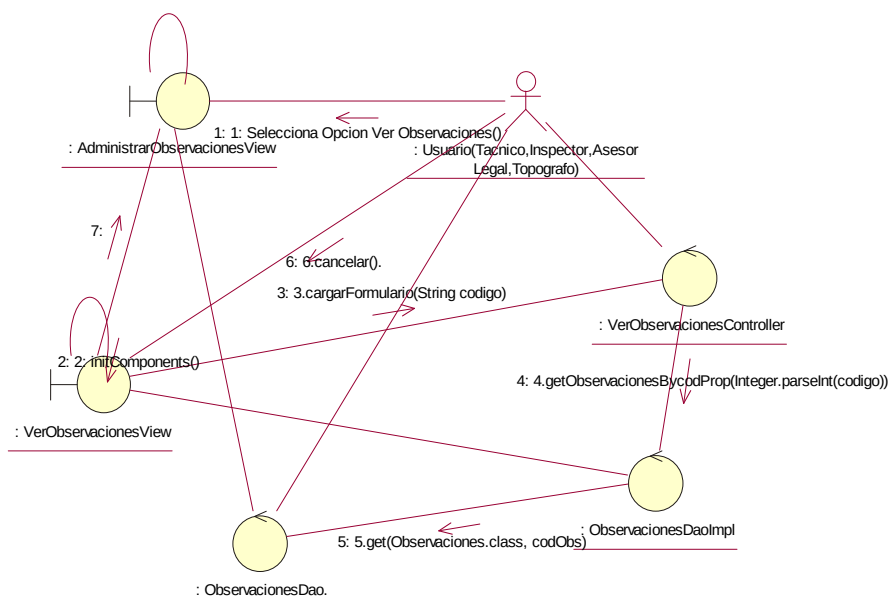


Figura 410: Diagrama de Colaboración: Ver Observaciones

III.2.3.3.42 Diagrama de Colaboración Administrar Observaciones Jefe Lotes

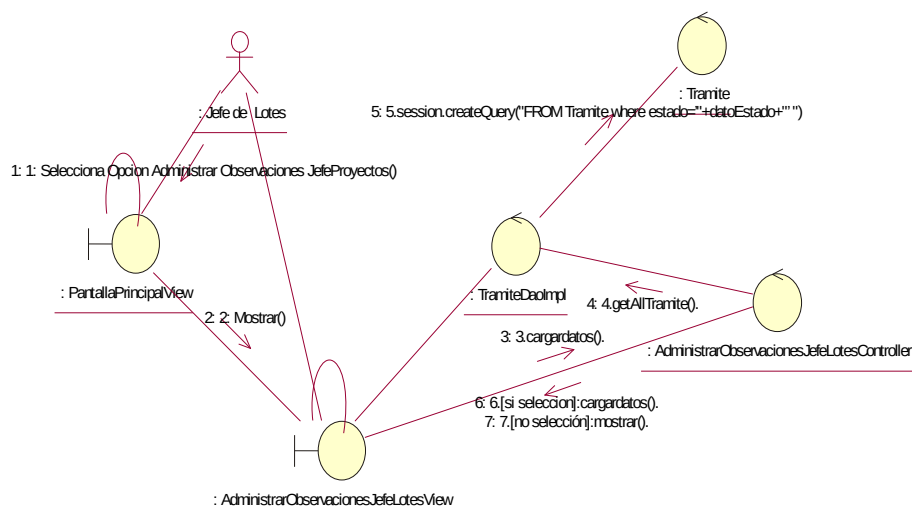


Figura 411: Diagrama de Colaboración: Adm. Observaciones Jefe Lotes

III.2.3.3.43 Diagrama de Colaboración Adicionar Observaciones Jefe Lotes

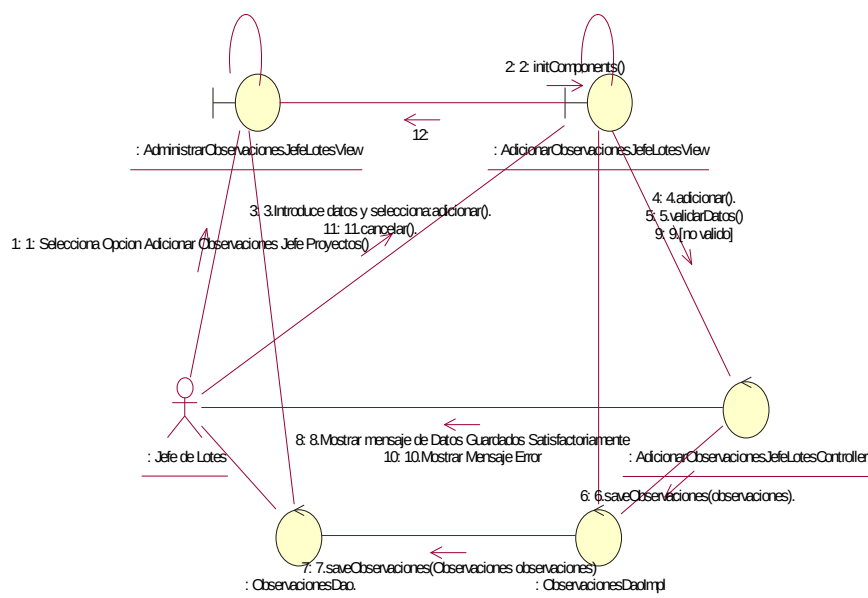


Figura 412: Diagrama de Colaboración: Adicionar Obser. Jefe Lotes

III.2.3.3.44 Diagrama de Colaboración Modificar Observaciones Jefe Lotes

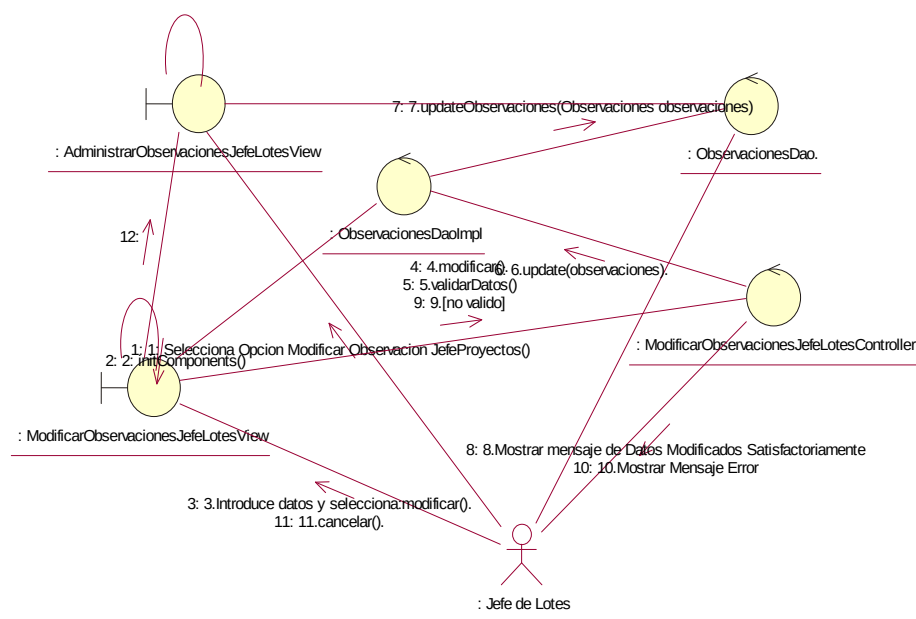


Figura 413: Diagrama de Colaboración: Modificar Obser. Jefe Lotes

III.2.3.3.45 Diagrama de Colaboración Ver Observaciones Jefe Lotes

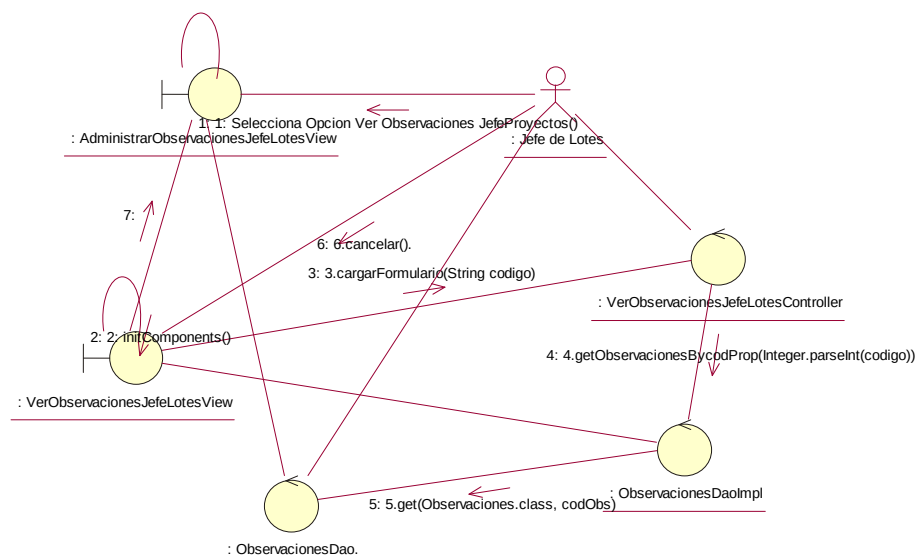


Figura 414: Diagrama de Colaboración: Ver Obser. Jefe Lotes

III.2.3.3.46 Diagrama de Colaboración Dar Alta Trámite

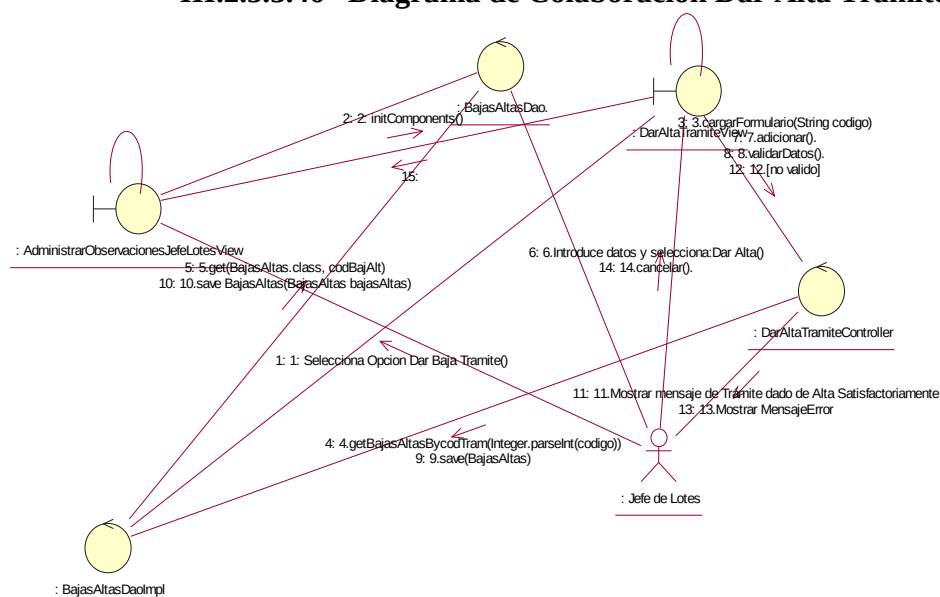


Figura 415: Diagrama de Colaboración: Dar Alta Trámite

III.2.3.3.47 Diagrama de Colaboración Aprobar Tramite

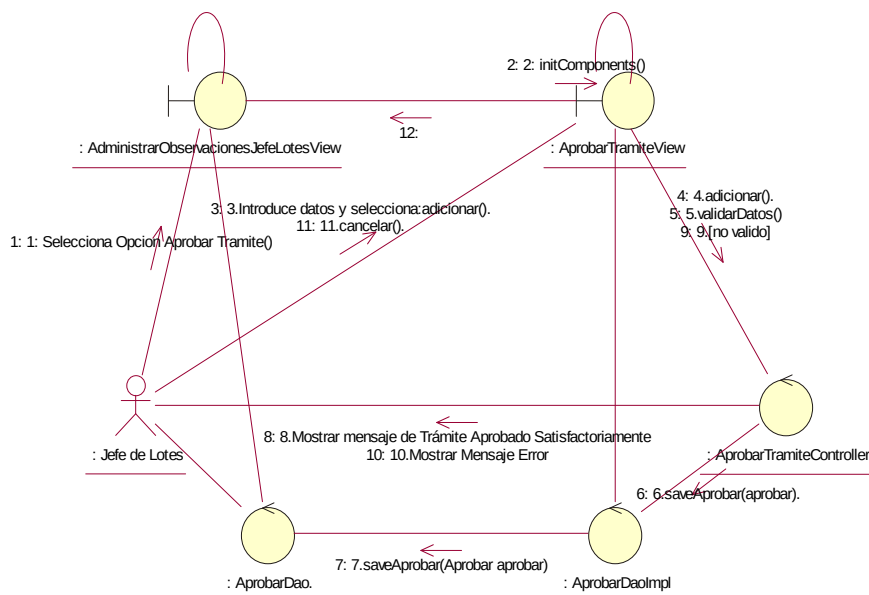


Figura 416: Diagrama de Colaboración: Aprobar Trámite

III.2.3.4 MODULO ADMINISTRACION

III.2.3.4.1 Diagrama de Colaboración Administrar Personal

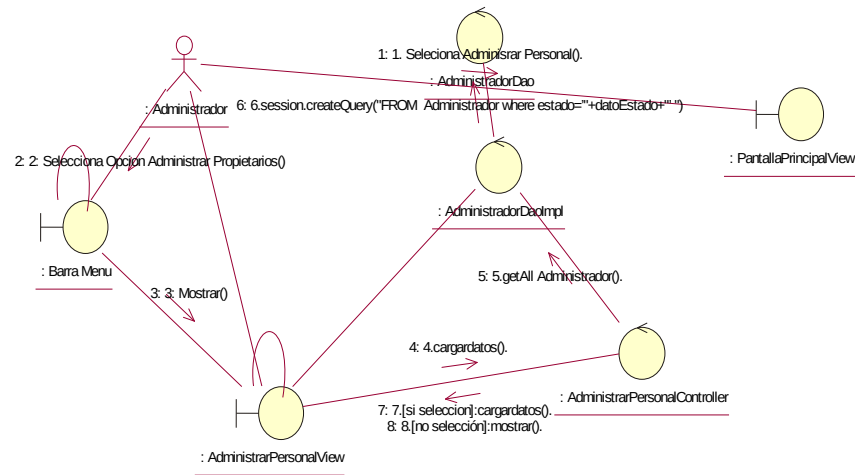


Figura 417: Diagrama de Colaboración: Administrar Personal

III.2.3.4.2 Diagrama de Colaboración Adicionar Personal

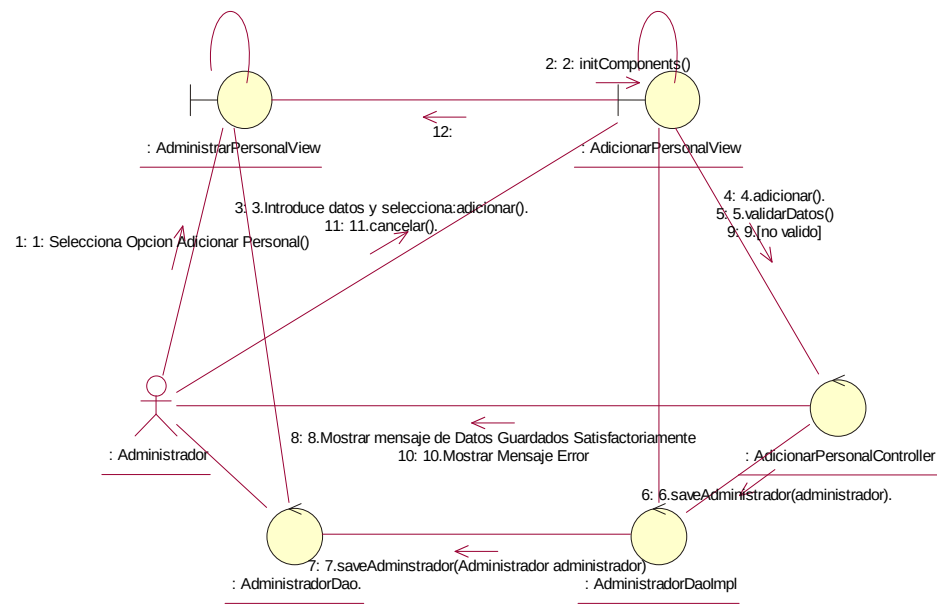


Figura 418: Diagrama de Colaboración: Adicionar Personal

III.2.3.4.3 Diagrama de Colaboración Modificar Personal

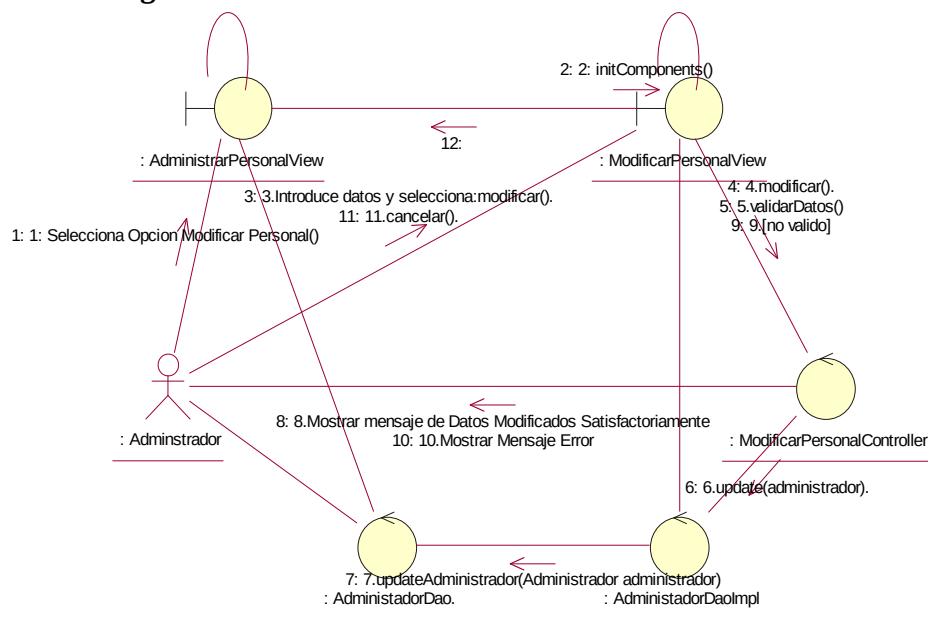


Figura 418: Diagrama de Colaboración: Modificar Personal

III.2.3.4.4 Diagrama de Colaboración Ver Personal

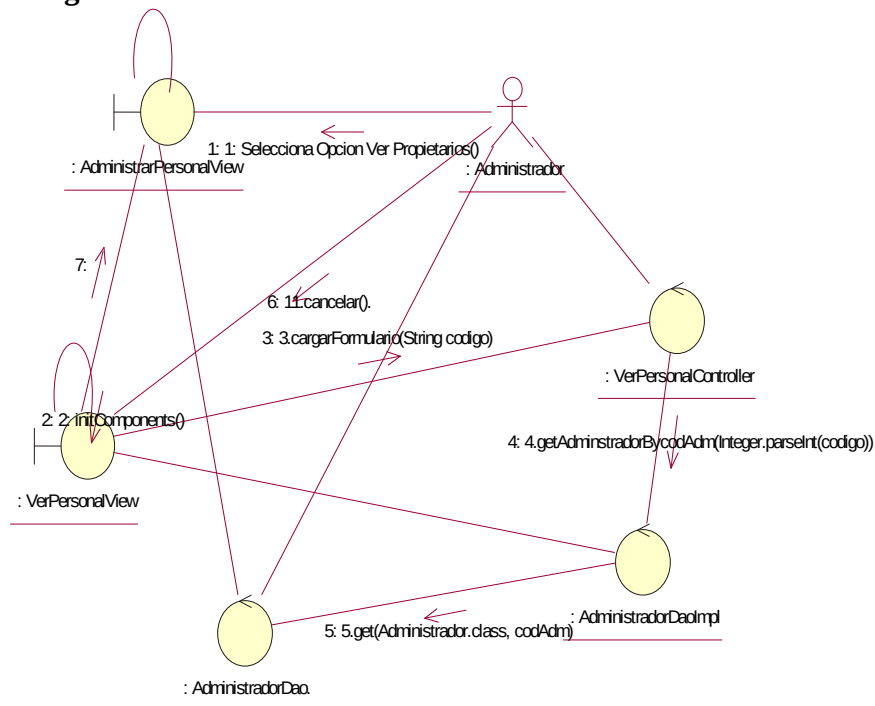


Figura 419: Diagrama de Colaboración: Ver Personal

III.2.3.4.5 Diagrama de Colaboración Administrar Roles

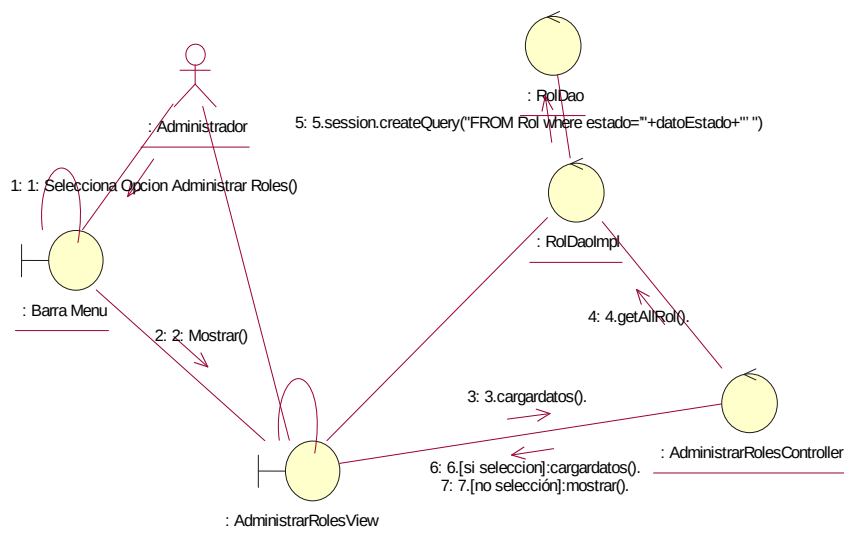


Figura 420: Diagrama de Colaboración: Administrar Roles

III.2.3.4.6 Diagrama de Colaboración Adicionar Roles

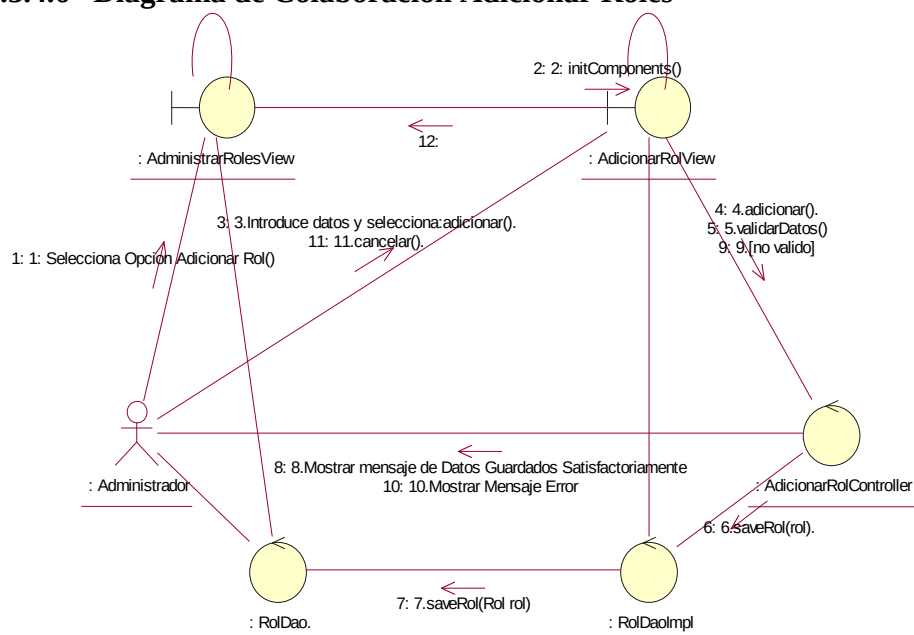


Figura 420: Diagrama de Colaboración: Adicionar Roles

III.2.3.4.7 Diagrama de Colaboración Modificar Roles

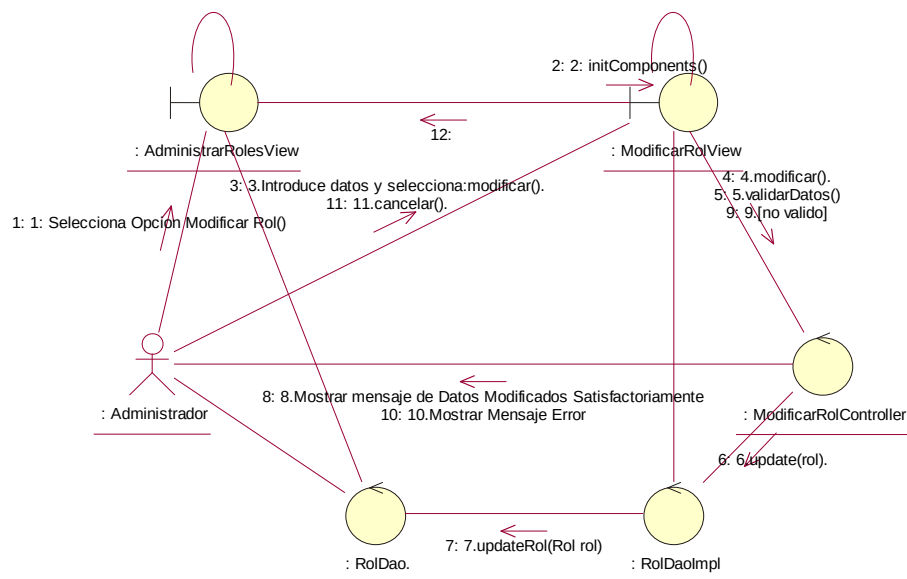


Figura 421: Diagrama de Colaboración: Modificar Roles

III.2.3.4.8 Diagrama de Colaboración Eliminar Roles

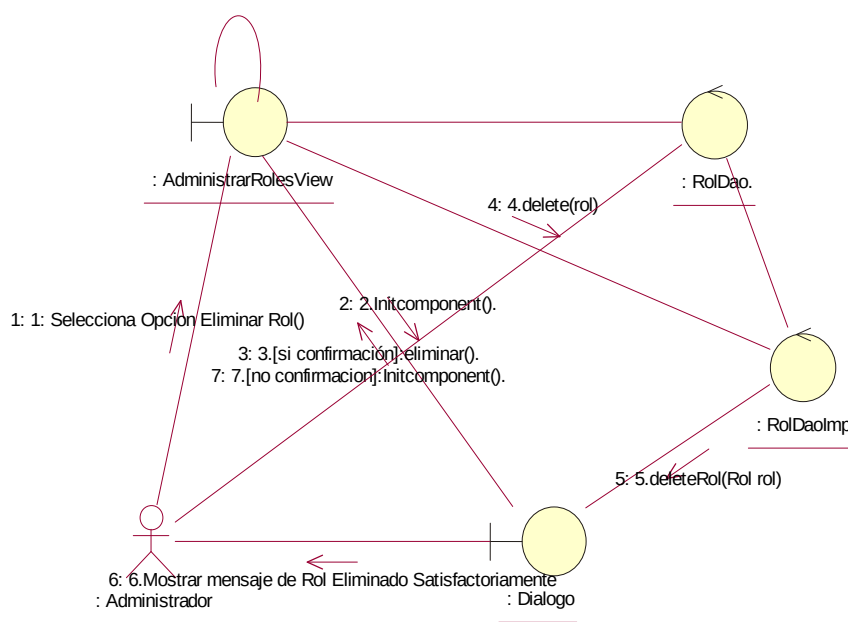


Figura422: Diagrama de Colaboración: Eliminar Roles

III.2.3.4.9 Diagrama de Colaboración Cambiar Contraseña

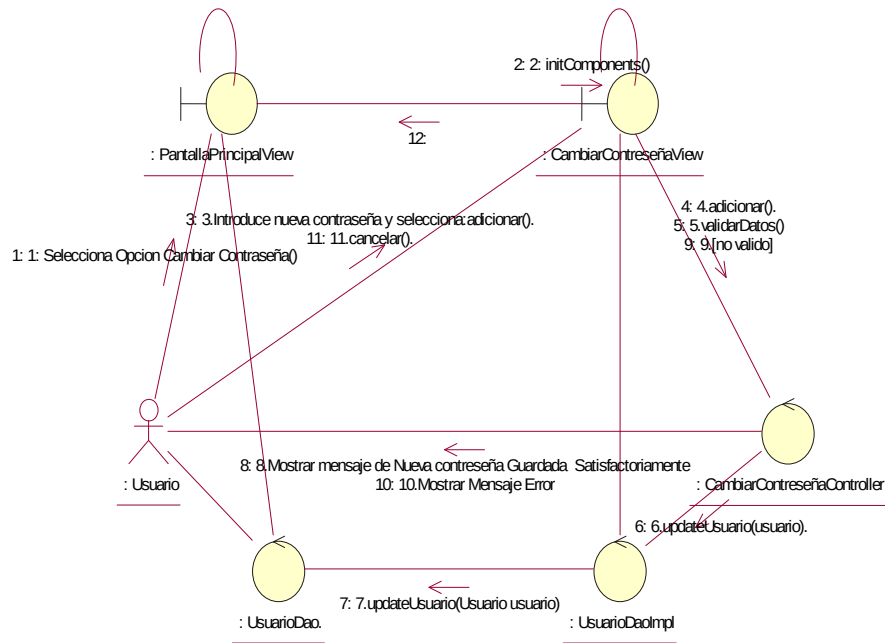


Figura 423: Diagrama de Colaboración: Cambiar Contraseña

III.2.3.5 MODULO DE REPORTES

III.2.3.5.1 Diagrama de Secuencia Generar Reportes de Tramite

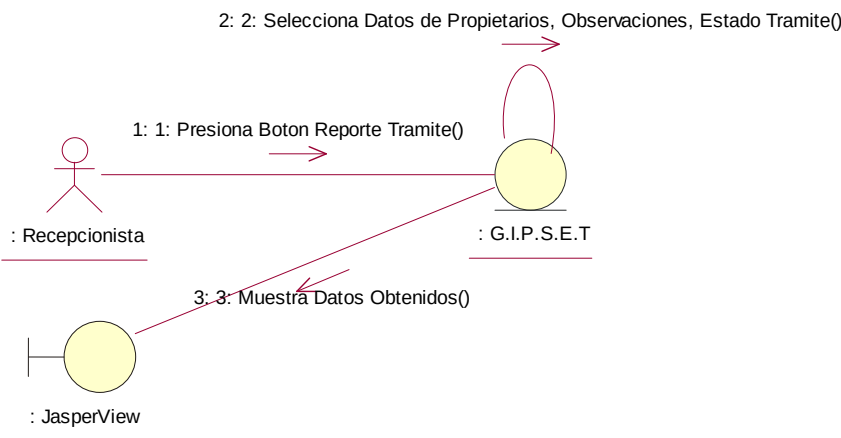


Figura 424: Diagrama de Secuencia: Generar Reportes de Tramite

III.2.3.5 .2 Diagrama de Secuencia Generar Reporte Técnico

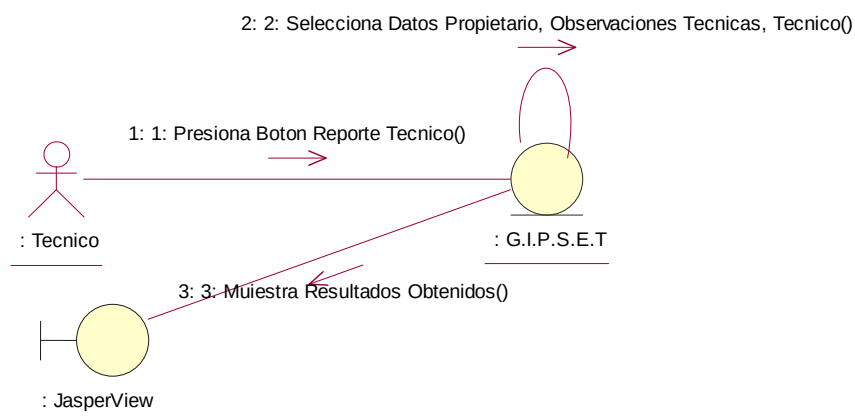


Figura 425: Diagrama de Secuencia: Generar Reporte Técnico

III.2.3.5 .3 Diagrama de Secuencia Generar Reporte de Inspección

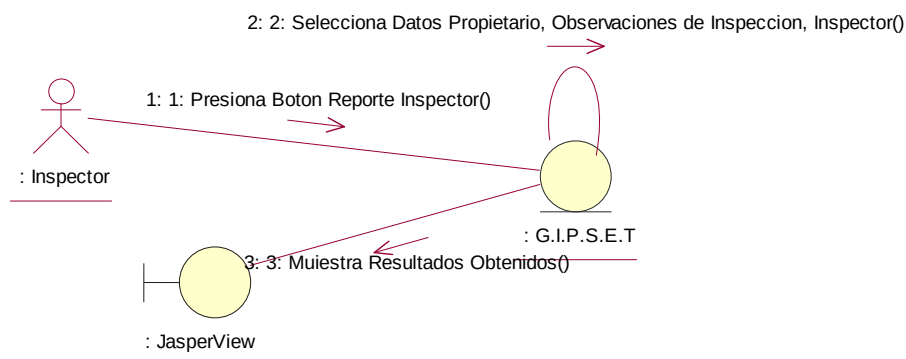


Figura 426: Diagrama de Secuencia: Generar Reporte de Inspección

III.2.3.5 .4 Diagrama de Secuencia Generar Reporte Legal

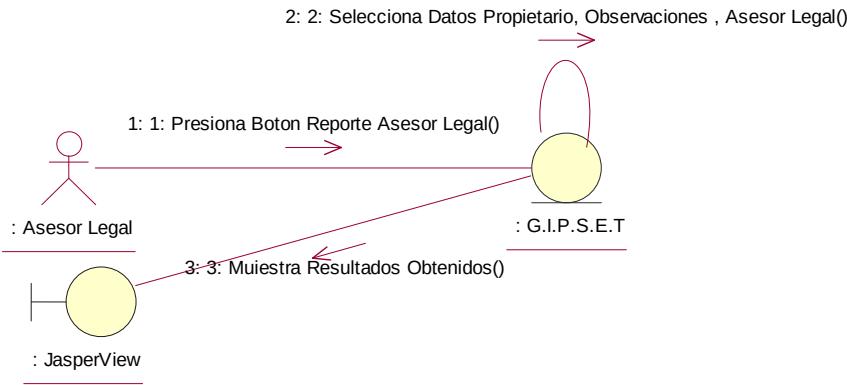


Figura 427: Diagrama de Secuencia: Generar Reporte Legal

III.2.2.5 .6 Diagrama de Secuencia Generar Reporte Topográfico

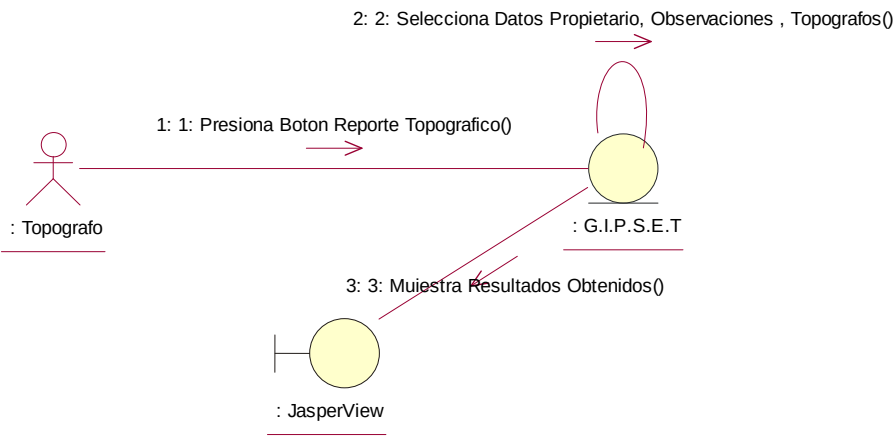


Figura 428: Diagrama de Secuencia: Generar Reporte Topográfico

II.1.12. PROTOTIPOS DE INTERFAZ DE USUARIO

II.1.12.1. Introducción

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

II.1.12.1.1. Propósito

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

II.1.12.1.2. Alcance

- Mostrar los Prototipos de Pantallas, solamente el diseño que adoptarán todas.

II.1.12.2. Prototipos de Pantallas

Pantalla 1 (Ingresar al Sistema)

Esta pantalla se muestra al iniciar el sistema.

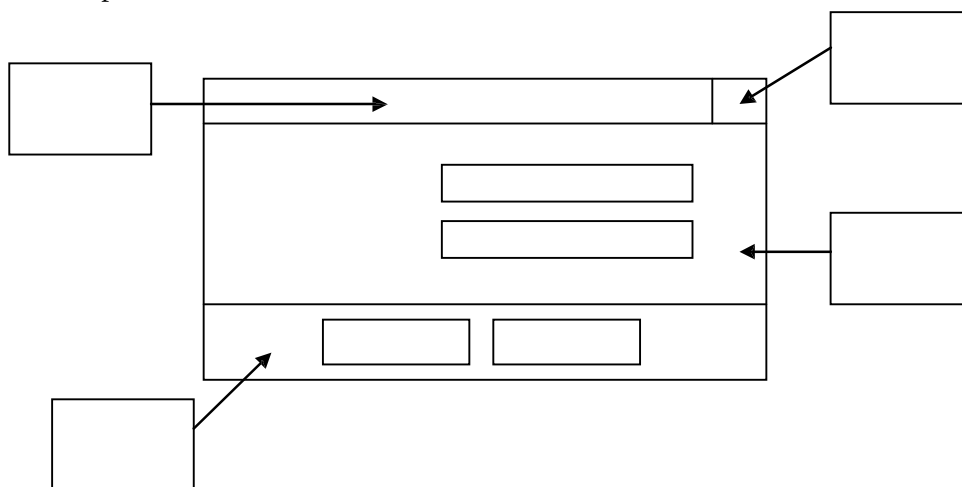


Figura 429: Pantalla 1: Ingresar al Sistema.

Pantalla 2 (Pantalla Principal Proyectos/Lotes)

Esta pantalla contiene el menú del sistema, el cual contiene los enlaces a los distintos sectores del mismo que nos facilitaran el trabajo de administrar.

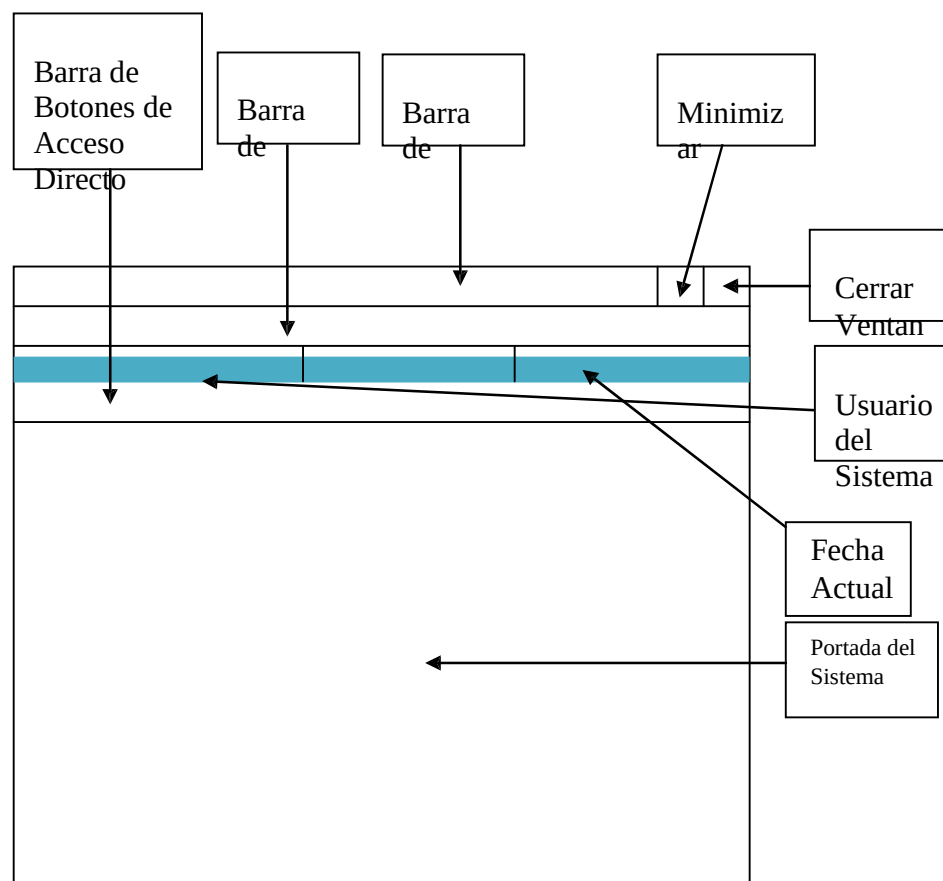


Figura 430: Pantalla 2: Pantalla Principal Proyectos/Lotes.

Pantalla 3: (Administrar / Propietarios)

Esta pantalla se obtiene al hacer clic en la barra de menú de la pantalla 2 (Pantalla Principal), donde ésta nueva pantalla se carga dentro de la pantalla 2 como una nueva pestaña. Aquí el usuario puede ver en la izquierda una tabla con la lista de propietarios, o lo que se quiera administrar o reasignar. En la parte derecha se muestra el detalle de cada fila de la tabla mencionada.

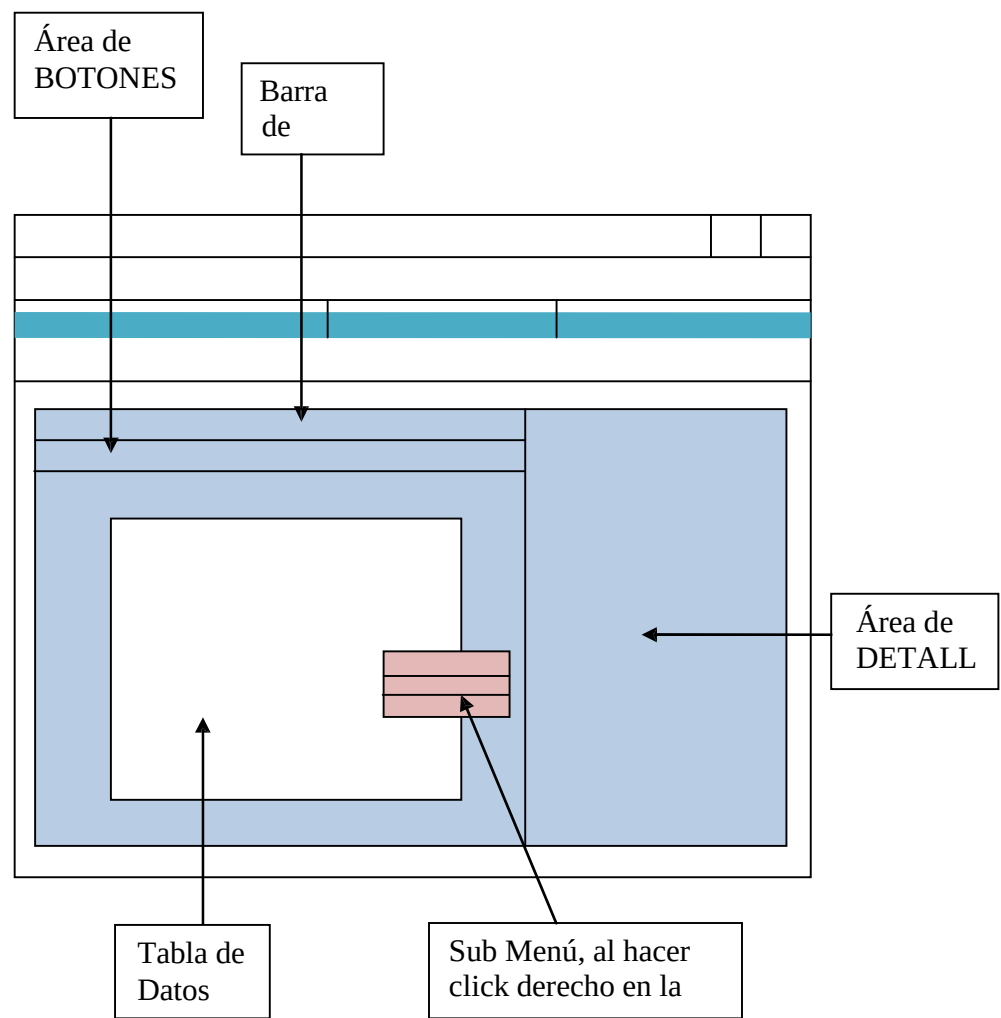


Figura 431: Pantalla 3: Administrar / Propietarios

Pantalla 4: (Pantalla Principal)

Esta pantalla contiene el menú del sistema, el cual contiene los enlaces a los distintos Módulos del Sistema el mismo que nos facilitaran el trabajo de administrar Proyectos, Personal, Cambiar Contraseña, Ayuda sobre el Sistema, Observaciones, Mensajes.

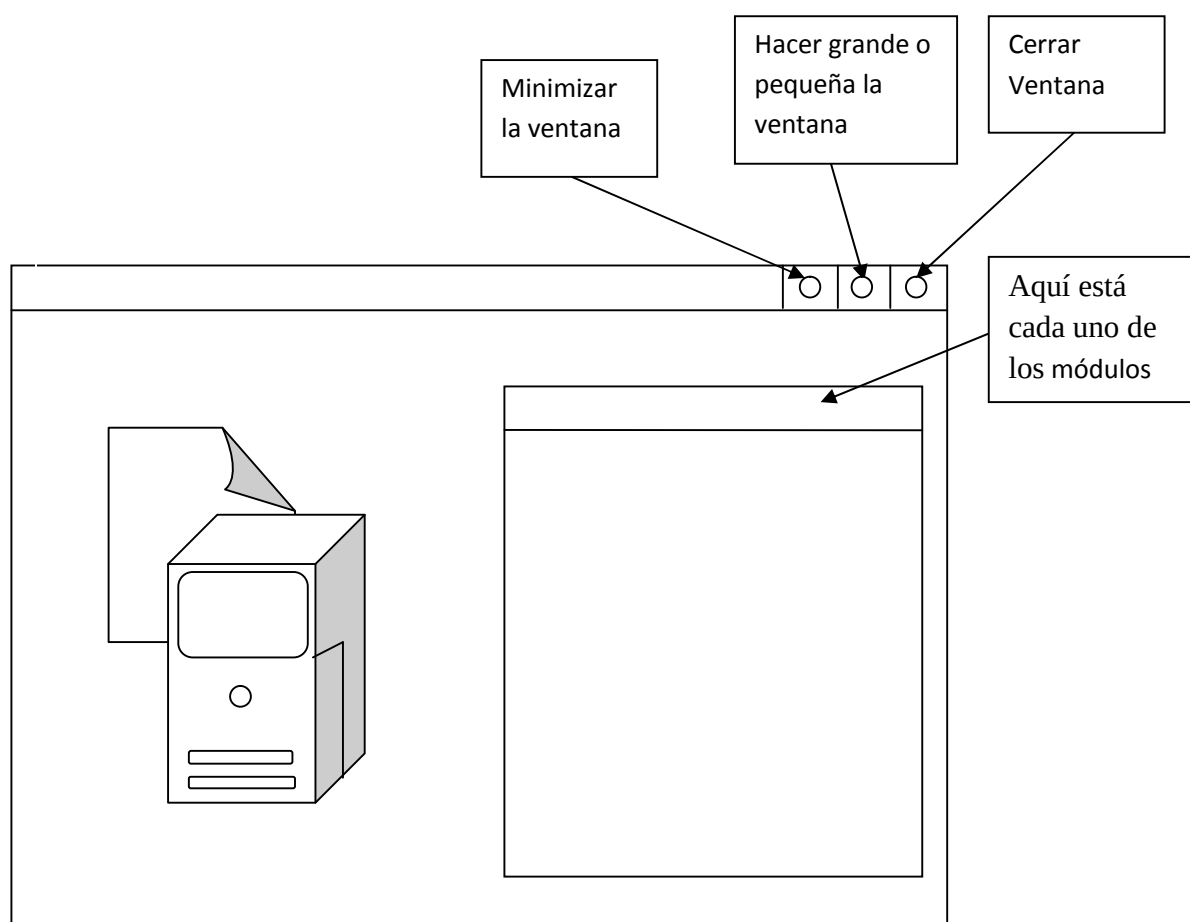


Figura 432: Pantalla 4: Pantalla Principal

Pantalla 5: (Pantalla Adminstrar Personal)

Esta pantalla se obtiene al hacer clic en la barra de menú de la donde ésta nueva pantalla se carga como una nueva pestaña. Aquí el usuario puede ver datos del personal, adicionar un nuevo personal, modificar y ver sus datos.

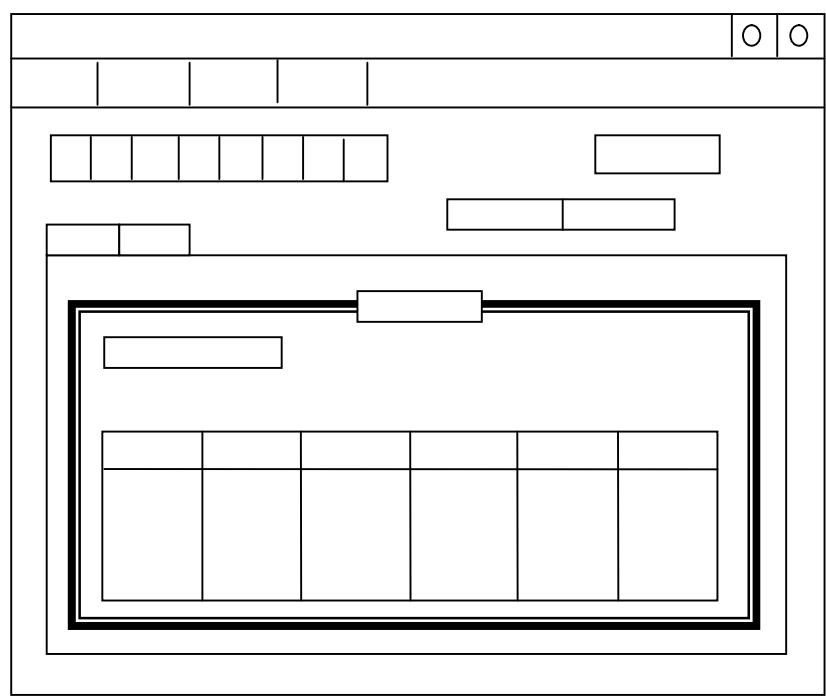


Figura 433: Pantalla 5: Pantalla Adminstrar Personal

Pantalla 6: (Pantalla Adminstrar Observaciones Generales)

Esta pantalla se obtiene al hacer clic en la barra de menú de la donde ésta nueva pantalla se carga como una nueva pestaña. Aquí el usuario puede ver datos del propietario y su trámite. En la parte derecha se muestra el detalle de cada fila de la tabla mencionada.

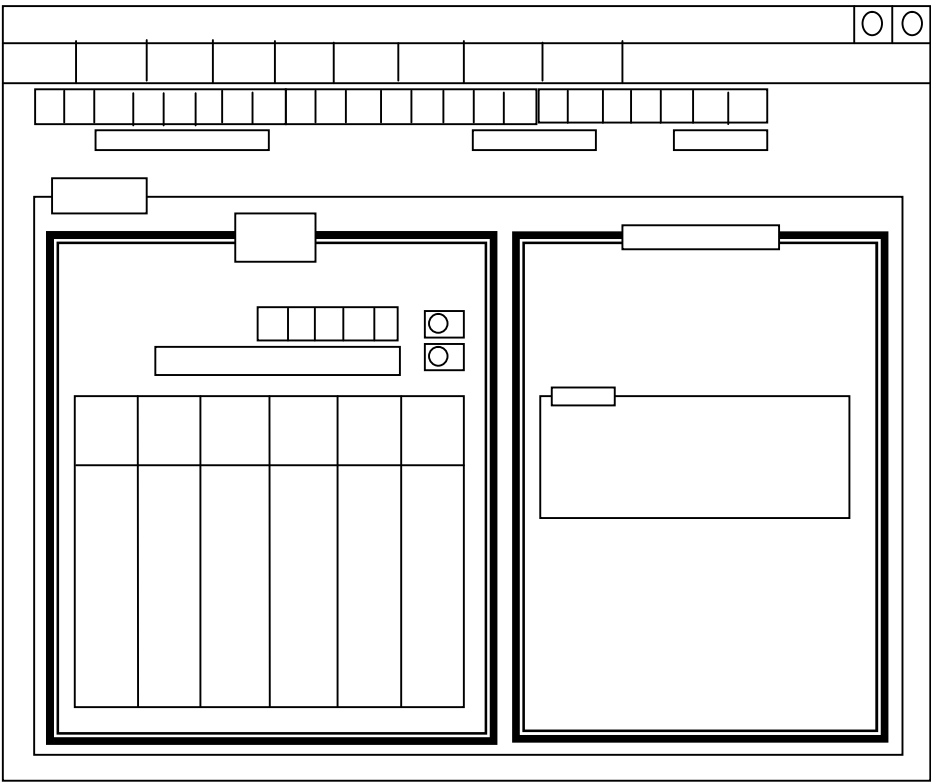


Figura 434: Pantalla 6: Pantalla Adminstrar Observaciones Generales

Pantalla 7: (Pantalla Registrar Ingresos)

Esta pantalla se obtiene al hacer clic en el de menú de la Tabla administrar ingresos donde ésta aparece en una nueva ventana. Aquí el usuario puede introducir el detalle del ingreso además que previamente se cargar algunos datos importantes.

Figura 435: Pantalla 7: Pantalla Registrar Ingresos

Pantalla 8: (Pantalla Reasignar Trámite)

Esta pantalla se obtiene al hacer clic en la barra de menú de la donde ésta nueva pantalla se carga como una nueva pestaña. Aquí el usuario puede ver datos del propietario y su trámite. En la parte derecha se muestra el detalle de cada fila de la tabla mencionada.

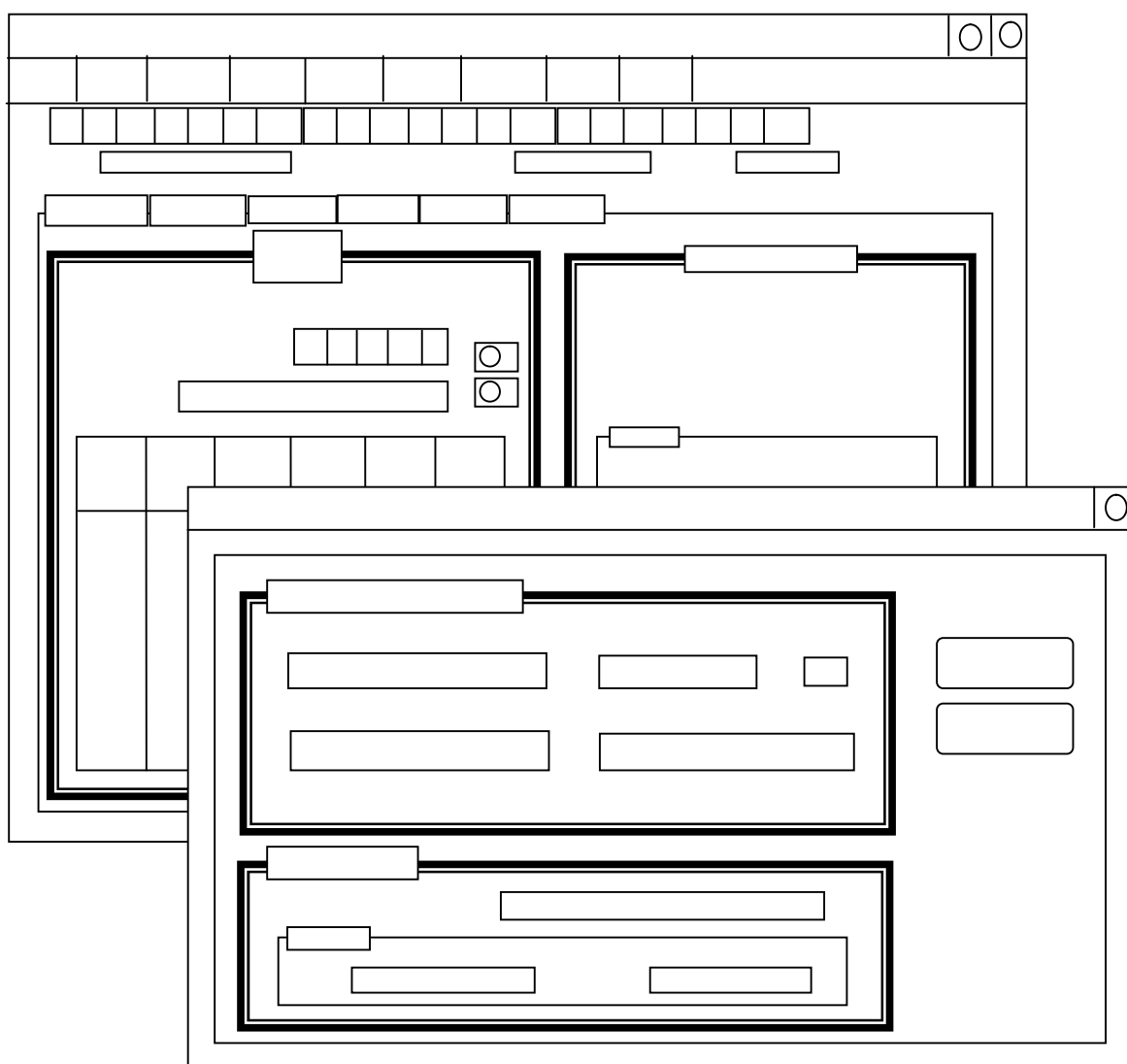


Figura 436: Pantalla 8: Pantalla Reasignar Trámite

CAPITULO IV:

BASE DE DATOS

CAPITULO IV: BASE DE DATOS

IV.1. MODELADO DE DIAGRAMA DE CLASES

IV.1.1. Introducción

El Modelado de Diagrama de Clases es uno más de los diagramas requeridos en la fase de Análisis/Diseño de la metodología que se está implementando.

IV.1.2. Propósito

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

IV.1.3. Alcance

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

IV.2 Creación de la Base de Datos.

```
create table persona(  
cod_per serial not null,  
ci integer not null,  
nombre varchar(50) not null,  
ap varchar(25),  
am varchar(25),  
primary key (cod_per)  
);
```

```
create table proyectista(  
cod_proy serial not null,  
cod_per integer not null,  
foreign key (cod_per) references persona (cod_per),  
primary key (cod_proy)  
);
```

```
create table apoderado(  
cod_apod serial not null,  
cod_per integer not null,  
foreign key (cod_per) references persona (cod_per),  
primary key (cod_apod)  
);
```



```
create table propietario(  
  cod_prop serial not null,  
  cod_per integer not null,  
  cod_proy integer not null,  
  cod_apod integer not null,  
  foreign key (cod_per) references persona (cod_per),  
  foreign key (cod_proy) references proyectista (cod_proy),  
  foreign key (cod_apod) references apoderado (cod_apod),  
  primary key (cod_prop)  
);
```

```
create table rol(  
  cod_rol serial not null,  
  rol varchar(30),  
  primary key(cod_rol)  
);
```

```
create table usuario(  
  cod_us serial not null,  
  usuario varchar(12) not null unique,  
  clave varchar(12) not null unique,  
  cod_adm integer not null,  
  cod_rol integer not null,  
  foreign key (cod_adm) references administrador (cod_adm),
```

```
foreign key (cod_rol) references rol (cod_rol),  
primary key (cod_us)  
);
```

```
create table zona(  
cod_zona serial not null,  
zona varchar(1000),  
primary key (cod_zona)  
);
```

```
create table direccion(  
cod_direc serial not null,  
zona varchar(50) not null,  
calle varchar(50)not null,  
num_casa integer not null,  
cod_per integer not null,  
cod_zona integer not null,  
foreign key (cod_per) references persona (cod_per),  
foreign key (cod_zona) references zona (cod_zona),  
primary key (cod_direc )  
);
```

```
create table telefono(  
cod_telf serial not null,
```

```
numero varchar(30) default 'no tiene',  
cod_per integer not null,  
foreign key (cod_per) references persona (cod_per),  
primary key (cod_telf)  
);
```

```
create table tramite(  
cod_tram serial not null,  
fecha date,  
hora time,  
gestion integer not null,  
num_previo integer not null,  
num_oficial integer not null,  
proceso varchar (15) default 'normal',  
estado varchar (15) default 'alta',  
cod_prop integer not null,  
cod_tip_tram integer not null,  
foreign key (cod_prop) references propietario (cod_prop),  
foreign key (cod_tip_tram) references tipo_tramite (cod_tip_tram),  
primary key (cod_tram)  
);
```

```
create table detalle_adtram(  
cod_adtram serial not null,
```

```
cod_tram integer not null,  
cod_adm integer not null,  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_adm) references administrador (cod_adm),  
primary key (cod_adtram)  
);
```

```
create table num_archivo(  
cod_arch serial not null,  
num_arch integer not null,  
fecha_arch date,  
hora_arch time,  
cod_adm integer not null,  
cod_tram integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
primary key (cod_arch)  
);
```

```
create table mensaje(  
cod_men serial not null,  
fecha_men date,  
hora_men time,  
mensaje varchar(2000),
```

```
cod_adm_env integer not null,  
cod_adm_des integer not null,  
foreign key (cod_adm_env) references administrador (cod_adm),  
foreign key (cod_adm_des) references administrador (cod_adm),  
primary key (cod_men)  
);
```

```
create table imagenes(  
cod_img serial not null,  
imagen varchar(800),  
cod_tram integer not null,  
foreign key (cod_tram) references tramite (cod_tram),  
primary key (cod_img )  
);
```

```
create table observaciones(  
cod_obs serial not null,  
fecha_obs date,  
hora_obs time,  
observacion varchar(2500),  
cod_adm integer not null,  
cod_tram integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),
```

```
primary key (cod_obs)
```

```
);
```

```
create table dar_ok(
```

```
cod_ok serial not null,
```

```
fecha_ok date,
```

```
hora_ok time,
```

```
ok varchar(25) default 'pendiente',
```

```
cod_obs integer not null,
```

```
cod_tram integer not null,
```

```
foreign key (cod_obs) references observaciones (cod_obs),
```

```
foreign key (cod_tram) references tramite (cod_tram),
```

```
primary key (cod_ok)
```

```
);
```

```
create table datos_tramite(
```

```
cod_dat_tram serial not null,
```

```
zona varchar(50) not null,
```

```
calle varchar(50),
```

```
numero varchar(30) default 'sin numero',
```

```
descripcion varchar(2000),
```

```
cod_tram integer not null,
```

```
cod_zona integer not null,
```

```
foreign key (cod_tram) references tramite (cod_tram),
```

```
foreign key (cod_zona) references zona (cod_zona),  
primary key (cod_dat_tram)  
);
```

```
create table requisitos(  
cod_req serial not null,  
requisitos varchar(2000),  
cod_tip_tram integer not null,  
foreign key (cod_tip_tram) references tipo_tramite (cod_tip_tram),  
primary key (cod_req)  
);
```

```
create table detalle_requisitos(  
cod_det serial not null,  
cumple integer default '1',  
cod_tram integer not null,  
cod_req integer not null,  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_req) references requisitos (cod_req),  
primary key (cod_det)  
);
```

```
create table motivo(  
cod_mot serial not null,
```

```
motivo varchar(500) not null,  
descripcion varchar(3000) not null,  
primary key (cod_mot)  
);
```

```
create table bloqueo_desbloqueo(  
cod_bloq_des serial not null,  
fecha_b_d date,  
hora_b_d time,  
tipo varchar(30) default 'desbloqueado',  
cod_mot integer not null,  
cod_adm integer not null,  
cod_tram integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_mot) references motivo (cod_mot),  
primary key (cod_bloq_des)  
);
```

```
create table aprobar(  
cod_aprob serial not null,  
fecha_aprob date,  
hora_aprob time,  
cod_adm integer not null,
```



```
cod_tram integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
primary key (cod_aprob)  
);
```

```
create table bajas_altas(  
cod_baj_alt serial not null,  
fecha_b_a date,  
hora_b_a time,  
tipo varchar(30),  
cod_adm integer not null,  
cod_tram integer not null,  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_adm) references administrador (cod_adm),  
primary key (cod_baj_alt)  
);
```

```
create table salidas(  
cod_sal serial not null,  
fecha_sal date,  
hora_sal time,  
observacion varchar(2500),  
cod_adm integer not null,
```

```
cod_tram integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
primary key (cod_sal)  
);
```

```
create table ingresos(  
cod_ing serial not null,  
num_ing integer,  
fecha_ing date,  
hora_ing time,  
observacion varchar(2500),  
cod_adm integer not null,  
cod_tram integer not null,  
cod_sal integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_sal) references salidas (cod_sal),  
primary key (cod_ing)  
);
```

```
create table personal_baja(  
cod_per_baj serial not null,  
fecha_p_b date,
```

```
hora_p_b time,  
motivo varchar(600),  
detalle varchar(2500),  
estado varchar (10) default 'inactivo',  
cod_adm integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
primary key (cod_per_baj)  
);
```

```
create table destino(  
cod_des serial not null,  
destino varchar(1000),  
detalle varchar(2000),  
primary key (cod_des)  
);
```

```
create table envios(  
cod_env serial not null,  
fecha_env date,  
hora_env time,  
observacion varchar(2500),  
cod_adm integer not null,  
cod_tram integer not null,  
cod_des integer not null,
```

```
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_des) references destino (cod_des),  
primary key (cod_env)  
);
```

```
create table recepcion(  
cod_recep serial not null,  
fecha_recep date,  
hora_recep time,  
observacion varchar(2500),  
cod_adm integer not null,  
cod_tram integer not null,  
cod_env integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
foreign key (cod_env) references envios (cod_env),  
primary key (cod_recep)  
);
```

```
create table reasignacion(  
cod_rea serial not null,  
fecha_rea date,  
hora_rea time,
```

```
detalle_rea varchar(2500),  
cod_adm integer not null,  
cod_tram integer not null,  
foreign key (cod_adm) references administrador (cod_adm),  
foreign key (cod_tram) references tramite (cod_tram),  
primary key (cod_rea)  
);
```

```
create table asignacion(  
cod_asig serial not null,  
fecha_asig date,  
hora_asig time,  
observacion varchar(2500),  
cod_adm_tec integer not null,  
cod_adm_asig integer not null,  
estado varchar (10) default 'activo',  
foreign key (cod_adm_tec) references administrador (cod_adm),  
foreign key (cod_adm_asig) references administrador (cod_adm),  
primary key (cod_asig)  
);
```

```
create table tipo_tramite(  
cod_tip_tram serial not null,
```

```
tramite varchar(600) not null,  
cod_area integer not null,  
foreign key (cod_area) references area (cod_area),  
primary key (cod_tip_tram)  
);
```

```
create table area(  
cod_area serial not null,  
area varchar(200) not null,  
descripcion varchar(1000) not null,  
primary key (cod_area)  
);
```

```
create table administrador(  
cod_adm serial not null,  
fecha_nac date,  
fecha_ing date,  
email varchar (30) not null,  
estado varchar (20) default 'activo',  
cod_per integer not null,  
cod_area integer not null,  
foreign key (cod_area) references area (cod_area),  
foreign key (cod_per) references persona (cod_per),  
primary key (cod_adm)  
);
```

IV.3 Descripción de las clases.

Nombre	usuario		
Tipo	Entidad		
Descripción	Representa la información relacionada con los usuarios registrados en el sistema que hacen uso de los servicios.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_usu	serial	Privado	
login	varchar	Privado	
clave	Varchar	Privado	

Tabla N° 56: Descripción → Usuario

Nombre	rol		
Tipo	Entidad		
Descripción	Representa la información relacionada con el rol de los usuarios registrados en el sistema que hacen uso de los servicios.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_rol	serial	Privado	
rol	varchar	Privado	

Tabla N° 57: Descripción → Rol

Nombre	persona		
Tipo	Entidad		
Descripción	Representa la información relacionada con los personas que hacen uso del sistema que hacen uso de los servicios.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_person	serial	Privado	
ci	integer	Privado	
nombre	Varchar	Privado	
ap	Varchar	Privado	
am	Varchar	Privado	

Tabla N° 58: Descripción → Persona

Nombre	administrador		
Tipo	Entidad		
Descripción	Representa la información relacionada con los personas que van a administrar el sistema que hacen uso de los servicios.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_adm	serial	Privado	
Fecha_nac	date	Privado	
email	Varchar	Privado	

estado	Varchar	Privado	
--------	---------	---------	--

Tabla N° 59: Descripción → Administrador

Nombre	Propietario		
Tipo	Entidad		
Descripción	Representa la información relacionada con los propietarios que realizan sus trámites en el lugar.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_prop	serial	Privado	
proyectista	Varchar	Privado	
apoderado	Varchar	Privado	

Tabla N° 60: Descripción → Propietario

Nombre	Proyectista		
Tipo	Entidad		
Descripción	Representa la información relacionada con los propietarios que realizan sus trámites en el lugar.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_proyec	serial	Privado	
cod_per	integer	Privado	

Tabla N° 61: Descripción → Proyectista

Nombre	teléfono		
Tipo	Entidad		
Descripción	Representa la información relacionada con los teléfono registrados en el sistema, que poseen los usuarios.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_tel	serial	Privado	
numero	varchar	Privado	

Tabla N° 62: Descripción → Telefono

Nombre	dirección		
Tipo	Entidad		
Descripción	Representa la información relacionada a las direcciones registradas en el sistema, de los que utilizan el sistema.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_direc	serial	Privado	
calle	varchar	Privado	
zona	varchar	Privado	
num_casa	varchar	Privado	

Tabla N° 63: Descripción → Dirección

Nombre	num_archivo		
Tipo	Entidad		
Descripción	Representa la información relacionada con los números de archivo que se le da a los tramites registrados en el sistema, por el administrador.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_arch	serial	Privado	
num_arch	varchar	Privado	
fecha_arch	date	Privado	
hora_arch	time	Privado	

Tabla N° 64: Descripción → Num_Archivo

Nombre	envios		
Tipo	Entidad		
Descripción	Representa la información relacionada con los envios que se hacen aun trámite y son registrados en el sistema, por el encargado del área.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_env	serial	Privado	
fecha_env	date	Privado	
hora_env	time	Privado	
observacion	varchar	Privado	

Tabla N° 65: Descripción → Envios

Nombre	ingresos		
Tipo	Entidad		
Descripción	Representa la información relacionada a ingresos que se dan después que los trámites subsanan sus observaciones y son registrados en el sistema, por el encargado del área.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_ing	serial	Privado	
fecha_ing	date	Privado	
hora_ing	time	Privado	
observacion	varchar	Privado	

Tabla N° 66: Descripción → Información

Nombre	Dar_ok		
Tipo	Entidad		
Descripción	Representa la información relacionada con los tramites que ya han sido dados oks y son registrados en el sistema, por el administrador.		

Atributo	Tipo de dato	visibilidad	Valor inicial
cod_ok	serial	Privado	
ok	varchar	Privado	
fecha_ok	date	Privado	
hora_ok	time	Privado	

Tabla Nª 67: Descripción → Dar_Ok

Nombre	imagenes		
Tipo	Entidad		
Descripción	Representa la información relacionada a las imágenes de las capacitaciones, información o eventos registrados en el sistema, ya sea por el administrador, técnico o veterinario.		
Atributo	Tipo de dato	visibilidad	Valor inicial
cod_img	serial	Privado	
imagen	varchar	Privado	

Tabla Nª 68: Descripción → Imagenes

IV.4. MODELADO DE DATOS

IV.4.1 Introducción

A continuación se describe la creación de la base de datos del software, apuntando su característica. Para expresar este modelo se utiliza un Diagrama de Clases y luego su correspondiente implementación en el lenguaje utilizado para el desarrollo del proyecto el cual es JAVA.

Clase: Administrador

```
public class Administrador implements java.io.Serializable {

    private int codAdm;
    private Persona persona;
    private Date fechaNac;
    private Date fechaIng;
    private String email;
    private String estado;
    private Set reasignacions = new HashSet(0);
    private Set ingresoses = new HashSet(0);
    private Set mensajesForCodAdmDes = new HashSet(0);
    private Set personalBajas = new HashSet(0);
    private Set recepcions = new HashSet(0);
    private Set detalleAdtrams = new HashSet(0);
    private Set salidasas = new HashSet(0);
    private Set observacioneses = new HashSet(0);
    private Set aprobars = new HashSet(0);
    private Set asignacionsForCodAdmTec = new HashSet(0);
    private Set eventoses = new HashSet(0);
    private Set numArchivos = new HashSet(0);
    private Set informacions = new HashSet(0);
    private Set usuarios = new HashSet(0);
    private Set mensajesForCodAdmEnv = new HashSet(0);
    private Set bajasAltases = new HashSet(0);
    private Set bloqueoDesbloques = new HashSet(0);
    private Set asignacionsForCodAdmAsig = new HashSet(0);
    private Set envioses = new HashSet(0);

    public Administrador() {
    }
    public Administrador(int codAdm, Persona persona, String email)
    {
        this.codAdm = codAdm;
        this.persona = persona;
        this.email = email;
    }
    public Administrador(int codAdm, Persona persona, Date fechaNac,
    Date fechaIng, String email, String estado, Set reasignacions,
    Set ingresoses, Set mensajesForCodAdmDes, Set personalBajas,
    Set recepcions, Set detalleAdtrams, Set salidasas,
```

```

Set observacioneses, Set aprobars, Set asignacionsForCodAdmTec,
Set eventoses, Set numArchivos, Set informacions, Set usuarios,
Set mensajesForCodAdmEnv, Set bajasAltases, Set
bloqueoDesbloques,Set asignacionsForCodAdmAsig, Set envioses) {
    this.codAdm = codAdm;
    this.persona = persona;
    this.fechaNac = fechaNac;
    this.fechaIng = fechaIng;
    this.email = email;
    this.estado = estado;
    this.reasignacions = reasignacions;
    this.ingresoses = ingresoses;
    this.mensajesForCodAdmDes = mensajesForCodAdmDes;
    this.personalBajas = personalBajas;
    this.recepcions = recepcions;
    this.detalleAdtrams = detalleAdtrams;
    this.salidases = salidases;
    this.observacioneses = observacioneses;
    this.aprobars = aprobars;
    this.asignacionsForCodAdmTec = asignacionsForCodAdmTec;
    this.eventoses = eventoses;
    this.numArchivos = numArchivos;
    this.informacions = informacions;
    this.usuarios = usuarios;
    this.mensajesForCodAdmEnv = mensajesForCodAdmEnv;
    this.bajasAltases = bajasAltases;
    this.bloqueoDesbloques = bloqueoDesbloques;
    this.asignacionsForCodAdmAsig = asignacionsForCodAdmAsig;
    this.envioses = envioses;
}
public int getCodAdm() {
    return this.codAdm;
}
public void setCodAdm(int codAdm) {
    this.codAdm = codAdm;
}
public Persona getPersona() {
    return this.persona;
}
public void setPersona(Persona persona) {
    this.persona = persona;
}

public Date getFechaNac() {
    return this.fechaNac;
}
public void setFechaNac(Date fechaNac) {
    this.fechaNac = fechaNac;
}
public Date getFechaIng() {
    return this.fechaIng;
}
public void setFechaIng(Date fechaIng) {
    this.fechaIng = fechaIng;
}

```

```

}
public String getEmail() {
    return this.email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getEstado() {
    return this.estado;
}
public void setEstado(String estado) {
    this.estado = estado;
}
public Set getReasignacions() {
    return this.reasignacions;
}
public void setReasignacions(Set reasignacions) {
    this.reasignacions = reasignacions;
}
public Set getIngresoses() {
    return this.ingresoses;
}
public void setIngresoses(Set ingresoses) {
    this.ingresoses = ingresoses;
}
public Set getMensajesForCodAdmDes() {
    return this.mensajesForCodAdmDes;
}
public void setMensajesForCodAdmDes(Set mensajesForCodAdmDes) {
    this.mensajesForCodAdmDes = mensajesForCodAdmDes;
}
public Set getPersonalBajas() {
    return this.personalBajas;
}
public void setPersonalBajas(Set personalBajas) {
    this.personalBajas = personalBajas;
}
public Set getRecepcions() {
    return this.recepcions;
}
public void setRecepcions(Set recepcions) {
    this.recepcions = recepcions;
}
public Set getDetalleAdtrams() {
    return this.detalleAdtrams;
}
public void setDetalleAdtrams(Set detalleAdtrams) {
    this.detalleAdtrams = detalleAdtrams;
}
public Set getSalidases() {
    return this.salidases;
}
public void setSalidases(Set salidases) {
    this.salidases = salidases;
}

```

```

    }
    public Set getObservacioneses() {
        return this.observacioneses;
    }
    public void setObservacioneses(Set observacioneses) {
        this.observacioneses = observacioneses;
    }
    public Set getAprobars() {
        return this.aprobars;
    }
    public void setAprobars(Set aprobars) {
        this.aprobars = aprobars;
    }
    public Set getAsignacionsForCodAdmTec() {
        return this.asignacionsForCodAdmTec;
    }
    public void setAsignacionsForCodAdmTec(Set asignacionsForCodAdmTec)
{
        this.asignacionsForCodAdmTec = asignacionsForCodAdmTec;
    }
    public Set getEventoses() {
        return this.eventoses;
    }
    public void setEventoses(Set eventoses) {
        this.eventoses = eventoses;
    }
    public Set getNumArchivos() {
        return this.numArchivos;
    }
    public void setNumArchivos(Set numArchivos) {
        this.numArchivos = numArchivos;
    }
    public Set getInformacions() {
        return this.informacions;
    }
    public void setInformacions(Set informacions) {
        this.informacions = informacions;
    }
    public Set getUsuarios() {
        return this.usuarios;
    }
    public void setUsuarios(Set usuarios) {
        this.usuarios = usuarios;
    }
    public Set getMensajesForCodAdmEnv() {
        return this.mensajesForCodAdmEnv;
    }
    public void setMensajesForCodAdmEnv(Set mensajesForCodAdmEnv) {
        this.mensajesForCodAdmEnv = mensajesForCodAdmEnv;
    }
    public Set getBajasAltases() {
        return this.bajasAltases;
    }

```

```

    }
    public void setBajasAltases(Set bajasAltases) {
        this.bajasAltases = bajasAltases;
    }

    public Set getBloqueoDesbloques() {
        return this.bloqueoDesbloques;
    }
    public void setBloqueoDesbloques(Set bloqueoDesbloques) {
        this.bloqueoDesbloques = bloqueoDesbloques;
    }
    public Set getAsignacionesForCodAdmAsig() {
        return this.asignacionesForCodAdmAsig;
    }
    public void setAsignacionesForCodAdmAsig(Set
asignacionesForCodAdmAsig) {
        this.asignacionesForCodAdmAsig = asignacionesForCodAdmAsig;
    }
    public Set getEnvioses() {
        return this.envioses;
    }
    public void setEnvioses(Set envioses) {
        this.envioses = envioses;
    }
}

```

Clase: Apoderado

```

public class Apoderado implements java.io.Serializable {

    private int codApod;
    private Persona persona;
    private Set propietarios = new HashSet(0);
    public Apoderado() {
    }
    public Apoderado(int codApod, Persona persona) {
        this.codApod = codApod;
        this.persona = persona;
    }
    public Apoderado(int codApod, Persona persona, Set
propietarios) {
        this.codApod = codApod;
        this.persona = persona;
        this.propietarios = propietarios;
    }
    public int getCodApod() {
        return this.codApod;
    }

    public void setCodApod(int codApod) {
        this.codApod = codApod;
    }
    public Persona getPersona() {
        return this.persona;
    }
}

```

```

    }
    public void setPersona(Persona persona) {
        this.persona = persona;
    }
    public Set getPropietarios() {
        return this.propietarios;
    }
    public void setPropietarios(Set propietarios) {
        this.propietarios = propietarios;
    }
}

```

Clase: Aprobar

```

public class Aprobar implements java.io.Serializable {

    private int codAprob;
    private Tramite tramite;
    private Administrador administrador;
    private Date fechaAprob;
    private Date horaAprob;

    public Aprobar() {
    }
    public Aprobar(int codAprob, Tramite tramite, Administrador
administrador) {
        this.codAprob = codAprob;
        this.tramite = tramite;
        this.administrador = administrador;
    }
    public Aprobar(int codAprob, Tramite tramite, Administrador
administrador, Date fechaAprob, Date horaAprob) {
        this.codAprob = codAprob;
        this.tramite = tramite;
        this.administrador = administrador;
        this.fechaAprob = fechaAprob;
        this.horaAprob = horaAprob;
    }
    public int getCodAprob() {
        return this.codAprob;
    }
    public void setCodAprob(int codAprob) {
        this.codAprob = codAprob;
    }
    public Tramite getTramite() {
        return this.tramite;
    }

    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
}

```



```

    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Date getFechaAprob() {
        return this.fechaAprob;
    }
    public void setFechaAprob(Date fechaAprob) {
        this.fechaAprob = fechaAprob;
    }
    public Date getHoraAprob() {
        return this.horaAprob;
    }
    public void setHoraAprob(Date horaAprob) {
        this.horaAprob = horaAprob;
    }
}

```

Clase: Asignación

```

public class Asignacion implements java.io.Serializable {

    private int codAsig;
    private Administrador administradorByCodAdmTec;
    private Administrador administradorByCodAdmAsig;
    private Date fechaAsig;
    private Date horaAsig;
    private String observacion;
    private String estado;
    public Asignacion() {
    }
    public Asignacion(int codAsig, Administrador
administradorByCodAdmTec,
        Administrador administradorByCodAdmAsig) {
        this.codAsig = codAsig;
        this.administradorByCodAdmTec = administradorByCodAdmTec;
        this.administradorByCodAdmAsig = administradorByCodAdmAsig;
    }
    public Asignacion(int codAsig, Administrador
administradorByCodAdmTec, Administrador
administradorByCodAdmAsig, Date fechaAsig,
Date horaAsig, String observacion, String estado) {
        this.codAsig = codAsig;
        this.administradorByCodAdmTec = administradorByCodAdmTec;
        this.administradorByCodAdmAsig =
administradorByCodAdmAsig;
        this.fechaAsig = fechaAsig;
        this.horaAsig = horaAsig;
        this.observacion = observacion;
        this.estado = estado;
    }
    public int getCodAsig() {
        return this.codAsig;
    }
    public void setCodAsig(int codAsig) {

```

```

        this.codAsig = codAsig;
    }
    public Administrador getAdministradorByCodAdmTec() {
        return this.administradorByCodAdmTec;
    }
    public void setAdministradorByCodAdmTec(
        Administrador administradorByCodAdmTec) {
        this.administradorByCodAdmTec = administradorByCodAdmTec;
    }
    public Administrador getAdministradorByCodAdmAsig() {
        return this.administradorByCodAdmAsig;
    }
    public void setAdministradorByCodAdmAsig(
        Administrador administradorByCodAdmAsig) {
        this.administradorByCodAdmAsig =
            administradorByCodAdmAsig;
    }
    public Date getFechaAsig() {
        return this.fechaAsig;
    }
    public void setFechaAsig(Date fechaAsig) {
        this.fechaAsig = fechaAsig;
    }
    public Date getHoraAsig() {
        return this.horaAsig;
    }
    public void setHoraAsig(Date horaAsig) {
        this.horaAsig = horaAsig;
    }
    public String getObservacion() {
        return this.observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }

    public String getEstado() {
        return this.estado;
    }
    public void setEstado(String estado) {
        this.estado = estado;
    }
}

```

Clase: BajasAltas

```

public class BajasAltas implements java.io.Serializable {

    private int codBajAlt;
    private Tramite tramite;
    private Administrador administrador;
    private Date fechaBA;
    private Date horaBA;
    private String tipo;

```

```

public BajasAltas() {
}
public BajasAltas(int codBajAlt, Tramite tramite,
    Administrador administrador) {
    this.codBajAlt = codBajAlt;
    this.tramite = tramite;
    this.administrador = administrador;
}
public BajasAltas(int codBajAlt, Tramite tramite,
    Administrador administrador, Date fechaBA, Date
    horaBA, String tipo) {
    this.codBajAlt = codBajAlt;
    this.tramite = tramite;
    this.administrador = administrador;
    this.fechaBA = fechaBA;
    this.horaBA = horaBA;
    this.tipo = tipo;
}
public int getCodBajAlt() {
    return this.codBajAlt;
}
public void setCodBajAlt(int codBajAlt) {
    this.codBajAlt = codBajAlt;
}
public Tramite getTramite() {
    return this.tramite;
}
public void setTramite(Tramite tramite) {
    this.tramite = tramite;
}
public Administrador getAdministrador() {
    return this.administrador;
}
public void setAdministrador(Administrador administrador) {
    this.administrador = administrador;
}
public Date getFechaBA() {
    return this.fechaBA;
}
public void setFechaBA(Date fechaBA) {
    this.fechaBA = fechaBA;
}
public Date getHoraBA() {
    return this.horaBA;
}
public void setHoraBA(Date horaBA) {
    this.horaBA = horaBA;
}
public String getTipo() {
    return this.tipo;
}
public void setTipo(String tipo) {
    this.tipo = tipo;
}

```

```

    }
}

```

Clase: DarOk

```

public class DarOk implements java.io.Serializable {

    private int codOk;
    private Observaciones observaciones;
    private Tramite tramite;
    private Date fechaOk;
    private Date horaOk;
    private String ok;

    public DarOk() {
    }
    public DarOk(int codOk, Observaciones observaciones, Tramite
        tramite) {
        this.codOk = codOk;
        this.observaciones = observaciones;
        this.tramite = tramite;
    }
    public DarOk(int codOk, Observaciones observaciones, Tramite
        tramite, Date fechaOk, Date horaOk, String ok) {
        this.codOk = codOk;
        this.observaciones = observaciones;
        this.tramite = tramite;
        this.fechaOk = fechaOk;
        this.horaOk = horaOk;
        this.ok = ok;
    }
    public int getCodOk() {
        return this.codOk;
    }
    public void setCodOk(int codOk) {
        this.codOk = codOk;
    }
    public Observaciones getObservaciones() {
        return this.observaciones;
    }
    public void setObservaciones(Observaciones observaciones) {
        this.observaciones = observaciones;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Date getFechaOk() {
        return this.fechaOk;
    }
    public void setFechaOk(Date fechaOk) {

```

```

        this.fechaOk = fechaOk;
    }
    public Date getHoraOk() {
        return this.horaOk;
    }
    public void setHoraOk(Date horaOk) {
        this.horaOk = horaOk;
    }
    public String getOk() {
        return this.ok;
    }
    public void setOk(String ok) {
        this.ok = ok;
    }
}

```

Clase: DatosTramite

```

public class DatosTramite implements java.io.Serializable {

    private int codDatTram;
    private Tramite tramite;
    private Zona zona;
    private String calle;
    private String numero;
    private String descripcion;

    public DatosTramite() {
    }
    public DatosTramite(int codDatTram, Tramite tramite, Zona zona
    ){
        this.codDatTram = codDatTram;
        this.tramite = tramite;
        this.zona = zona;
    }
    public DatosTramite(int codDatTram, Tramite tramite, Zona zona,
    String calle, String numero, String descripcion) {
        this.codDatTram = codDatTram;
        this.tramite = tramite;
        this.zona = zona;
        this.calle = calle;
        this.numero = numero;
        this.descripcion = descripcion;
    }
    public int getCodDatTram() {
        return this.codDatTram;
    }
    public void setCodDatTram(int codDatTram) {
        this.codDatTram = codDatTram;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {

```

```

        this.tramite = tramite;
    }
    public Zona getZona() {
        return this.zona;
    }
    public void setZona(Zona zona) {
        this.zona = zona;
    }
    public String getCalle() {
        return this.calle;
    }
    public void setCalle(String calle) {
        this.calle = calle;
    }
    public String getNumero() {
        return this.numero;
    }
    public void setNumero(String numero) {
        this.numero = numero;
    }
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}

```

Clase: Destino

```

public class Destino implements java.io.Serializable {

    private int codDes;
    private String destino;
    private String detalle;
    private Set envioses = new HashSet(0);

    public Destino() {
    }
    public Destino(int codDes) {
        this.codDes = codDes;
    }
    public Destino(int codDes, String destino, String detalle, Set
        envioses) {
        this.codDes = codDes;
        this.destino = destino;
        this.detalle = detalle;
        this.envioses = envioses;
    }
    public int getCodDes() {
        return this.codDes;
    }
    public void setCodDes(int codDes) {
        this.codDes = codDes;
    }
}

```

```

    }
    public String getDestino() {
        return this.destino;
    }
    public void setDestino(String destino) {
        this.destino = destino;
    }
    public String getDetalle() {
        return this.detalle;
    }
    public void setDetalle(String detalle) {
        this.detalle = detalle;
    }
    public Set getEnvioses() {
        return this.envioses;
    }
    public void setEnvioses(Set envioses) {
        this.envioses = envioses;
    }
}

```

Clase: Dirección

```

public class Direccion implements java.io.Serializable {

    private static final long serialVersionUID = 1L;
    private int codDirec;
    private Zona zona;
    private Persona persona;
    private String calle;
    private int numCasa;

    public Direccion() {
    }
    public Direccion(int codDirec, Zona zona, Persona persona,
        String calle, int numCasa) {
        this.codDirec = codDirec;
        this.zona = zona;
        this.persona = persona;
        this.calle = calle;
        this.numCasa = numCasa;
    }
    public int getCodDirec() {
        return this.codDirec;
    }

    public void setCodDirec(int codDirec) {
        this.codDirec = codDirec;
    }
    public Zona getZona() {
        return this.zona;
    }
    public void setZona(Zona zona) {
        this.zona = zona;
    }
}

```

```

    }
    public Persona getPersona() {
        return this.persona;
    }
    public void setPersona(Persona persona) {
        this.persona = persona;
    }
    public String getCalle() {
        return this.calle;
    }
    public void setCalle(String calle) {
        this.calle = calle;
    }
    public int getNumCasa() {
        return this.numCasa;
    }
    public void setNumCasa(int a) {
        this.numCasa = a;
    }
}

```

Clase: Envíos

```

public class Envios implements java.io.Serializable {

    private int codEnv;
    private Tramite tramite;
    private Administrador administrador;
    private Destino destino;
    private Date fechaEnv;
    private Date horaEnv;
    private String observacion;
    private Set recepciones = new HashSet(0);

    public Envios() {
    }
    public Envios(int codEnv, Tramite tramite, Administrador
administrador,
        Destino destino) {
        this.codEnv = codEnv;
        this.tramite = tramite;
        this.administrador = administrador;
        this.destino = destino;
    }
    public Envios(int codEnv, Tramite tramite, Administrador
administrador, Destino destino, Date fechaEnv, Date horaEnv,
String observacion, Set recepciones) {
        this.codEnv = codEnv;
        this.tramite = tramite;
        this.administrador = administrador;
        this.destino = destino;
        this.fechaEnv = fechaEnv;
        this.horaEnv = horaEnv;
        this.observacion = observacion;
    }
}

```



```

        this.recepcions = recepcions;
    }
    public int getCodEnv() {
        return this.codEnv;
    }
    public void setCodEnv(int codEnv) {
        this.codEnv = codEnv;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Destino getDestino() {
        return this.destino;
    }
    public void setDestino(Destino destino) {
        this.destino = destino;
    }
    public Date getFechaEnv() {
        return this.fechaEnv;
    }
    public void setFechaEnv(Date fechaEnv) {
        this.fechaEnv = fechaEnv;
    }
    public Date getHoraEnv() {
        return this.horaEnv;
    }
    public void setHoraEnv(Date horaEnv) {
        this.horaEnv = horaEnv;
    }
    public String getObservacion() {
        return this.observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
    public Set getRecepcions() {
        return this.recepcions;
    }
    public void setRecepcions(Set recepcions) {
        this.recepcions = recepcions;
    }
}

```

Clase: Ingresos

```

public class Ingresos implements java.io.Serializable {

    private int codIng;
    private Tramite tramite;
    private Administrador administrador;
    private Salidas salidas;
    private Integer numIng;
    private Date fechaIng;
    private Date horaIng;
    private String observacion;

    public Ingresos() {
    }
    public Ingresos(int codIng, Tramite tramite, Administrador
        administrador, Salidas salidas) {
        this.codIng = codIng;
        this.tramite = tramite;
        this.administrador = administrador;
        this.salidas = salidas;
    }
    public Ingresos(int codIng, Tramite tramite, Administrador
        administrador, Salidas salidas, Integer numIng, Date
        fechaIng, Date horaIng, String observacion) {
        this.codIng = codIng;
        this.tramite = tramite;
        this.administrador = administrador;
        this.salidas = salidas;
        this.numIng = numIng;
        this.fechaIng = fechaIng;
        this.horaIng = horaIng;
        this.observacion = observacion;
    }
    public int getCodIng() {
        return this.codIng;
    }
    public void setCodIng(int codIng) {
        this.codIng = codIng;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Salidas getSalidas() {
        return this.salidas;
    }
}

```

```

    }
    public void setSalidas(Salidas salidas) {
        this.salidas = salidas;
    }
    public Integer getNumIng() {
        return this.numIng;
    }
    public void setNumIng(Integer numIng) {
        this.numIng = numIng;
    }
    public Date getFechaIng() {
        return this.fechaIng;
    }
    public void setFechaIng(Date fechaIng) {
        this.fechaIng = fechaIng;
    }
    public Date getHoraIng() {
        return this.horaIng;
    }
    public void setHoraIng(Date horaIng) {
        this.horaIng = horaIng;
    }
    public String getObservacion() {
        return this.observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
}

```

Clase: Motivo

```

public class Motivo implements java.io.Serializable {

    private int codMot;
    private String motivo;
    private String descripcion;
    private Set bloqueoDesbloques = new HashSet(0);

    public Motivo() {
    }
    public Motivo(int codMot, String motivo, String descripcion) {
        this.codMot = codMot;
        this.motivo = motivo;
        this.descripcion = descripcion;
    }
    public Motivo(int codMot, String motivo, String descripcion,
        Set bloqueoDesbloques) {
        this.codMot = codMot;
        this.motivo = motivo;
        this.descripcion = descripcion;
        this.bloqueoDesbloques = bloqueoDesbloques;
    }
}

```

```

    public int getCodMot() {
        return this.codMot;
    }
    public void setCodMot(int codMot) {
        this.codMot = codMot;
    }
    public String getMotivo() {
        return this.motivo;
    }
    public void setMotivo(String motivo) {
        this.motivo = motivo;
    }
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    public Set getBloqueoDesbloques() {
        return this.bloqueoDesbloques;
    }
    public void setBloqueoDesbloques(Set bloqueoDesbloques) {
        this.bloqueoDesbloques = bloqueoDesbloques;
    }
}

```

Clase: Observaciones

```

public class Observaciones implements java.io.Serializable {

    private int codObs;
    private Tramite tramite;
    private Administrador administrador;
    private Date fechaObs;
    private Date horaObs;
    private String observacion;
    private Set darOks = new HashSet(0);

    public Observaciones() {
    }
    public Observaciones(int codObs, Tramite tramite,
        Administrador administrador) {
        this.codObs = codObs;
        this.tramite = tramite;
        this.administrador = administrador;
    }
    public Observaciones(int codObs, Tramite tramite,
        Administrador administrador, Date fechaObs, Date
        horaObs, String observacion, Set darOks) {
        this.codObs = codObs;
        this.tramite = tramite;
        this.administrador = administrador;
        this.fechaObs = fechaObs;
        this.horaObs = horaObs;
    }
}

```

```

        this.observacion = observacion;
        this.darOks = darOks;
    }
    public int getCodObs() {
        return this.codObs;
    }
    public void setCodObs(int codObs) {
        this.codObs = codObs;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Date getFechaObs() {
        return this.fechaObs;
    }
    public void setFechaObs(Date fechaObs) {
        this.fechaObs = fechaObs;
    }
    public Date getHoraObs() {
        return this.horaObs;
    }
    public void setHoraObs(Date horaObs) {
        this.horaObs = horaObs;
    }
    public String getObservacion() {
        return this.observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
    public Set getDarOks() {
        return this.darOks;
    }
    public void setDarOks(Set darOks) {
        this.darOks = darOks;
    }
}

```

Clase: Persona

```

public class Persona implements java.io.Serializable {

    private int codPer;
    private int ci;
    private String nombre;

```

```

private String ap;
private String am;
private Set direcciones = new HashSet(0);
private Set proyectistas = new HashSet(0);
private Set apoderados = new HashSet(0);
private Set administradores = new HashSet(0);
private Set propietarios = new HashSet(0);
private Set telefonos = new HashSet(0);

public Persona() {
}
public Persona(int codPer, int ci, String nombre) {
    this.codPer = codPer;
    this.ci = ci;
    this.nombre = nombre;
}
public Persona(int codPer, int ci, String nombre, String ap,
String am, Set direcciones, Set proyectistas, Set apoderados,
Set administradores, Set propietarios, Set telefonos) {
    this.codPer = codPer;
    this.ci = ci;
    this.nombre = nombre;
    this.ap = ap;
    this.am = am;
    this.direcciones = direcciones;
    this.proyectistas = proyectistas;
    this.apoderados = apoderados;
    this.administradores = administradores;
    this.propietarios = propietarios;
    this.telefonos = telefonos;
}
public int getCodPer() {
    return this.codPer;
}
public void setCodPer(int codPer) {
    this.codPer = codPer;
}

public int getCi() {
    return this.ci;
}
public void setCi(int ci) {
    this.ci = ci;
}
public String getNombre() {
    return this.nombre;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public String getAp() {
    return this.ap;
}
public void setAp(String ap) {

```

```

        this.ap = ap;
    }
    public String getAm() {
        return this.am;
    }
    public void setAm(String am) {
        this.am = am;
    }
    public Set getDirecciones() {
        return this.direcciones;
    }
    public void setDirecciones(Set direcciones) {
        this.direcciones = direcciones;
    }
    public Set getProyectistas() {
        return this.proyectistas;
    }
    public void setProyectistas(Set proyectistas) {
        this.proyectistas = proyectistas;
    }
    public Set getApoderados() {
        return this.apoderados;
    }
    public void setApoderados(Set apoderados) {
        this.apoderados = apoderados;
    }
    public Set getAdministradores() {
        return this.administradores;
    }
    public void setAdministradores(Set administradores) {
        this.administradores = administradores;
    }
    public Set getPropietarios() {
        return this.propietarios;
    }
    public void setPropietarios(Set propietarios) {
        this.propietarios = propietarios;
    }
    public Set getTelefonos() {
        return this.telefonos;
    }
    public void setTelefonos(Set telefonos) {
        this.telefonos = telefonos;
    }
}

```

Clase: PersonalBaja

```

public class PersonalBaja implements java.io.Serializable {

    private int codPerBaj;
    private Administrador administrador;
    private Date fechaPB;
    private Date horaPB;
}

```

```

private String motivo;
private String detalle;
private String estado;

public PersonalBaja() {
}
public PersonalBaja(int codPerBaj, Administrador administrador)
{
    this.codPerBaj = codPerBaj;
    this.administrador = administrador;
}
public PersonalBaja(int codPerBaj, Administrador administrador,
    Date fechaPB, Date horaPB, String motivo, String
    detalle,String estado) {
    this.codPerBaj = codPerBaj;
    this.administrador = administrador;
    this.fechaPB = fechaPB;
    this.horaPB = horaPB;
    this.motivo = motivo;
    this.detalle = detalle;
    this.estado = estado;
}
public int getCodPerBaj() {
    return this.codPerBaj;
}
public void setCodPerBaj(int codPerBaj) {
    this.codPerBaj = codPerBaj;
}
public Administrador getAdministrador() {
    return this.administrador;
}
public void setAdministrador(Administrador administrador) {
    this.administrador = administrador;
}
public Date getFechaPB() {
    return this.fechaPB;
}
public void setFechaPB(Date fechaPB) {
    this.fechaPB = fechaPB;
}
public Date getHoraPB() {
    return this.horaPB;
}
public void setHoraPB(Date horaPB) {
    this.horaPB = horaPB;
}
public String getMotivo() {
    return this.motivo;
}
public void setMotivo(String motivo) {
    this.motivo = motivo;
}
public String getDetalle() {
    return this.detalle;
}

```



```

    }
    public void setDetalle(String detalle) {
        this.detalle = detalle;
    }
    public String getEstado() {
        return this.estado;
    }
    public void setEstado(String estado) {
        this.estado = estado;
    }
}

```

Clase: Propietario

```

public class Propietario implements java.io.Serializable {

    private int codProp;
    private Proyectoista proyectoista;
    private Apoderado apoderado;
    private Persona persona;
    private Set tramites = new HashSet(0);

    public Propietario() {
    }
    public Propietario(int codProp, Proyectoista proyectoista,
        Apoderado apoderado, Persona persona) {
        this.codProp = codProp;
        this.proyectoista = proyectoista;
        this.apoderado = apoderado;
        this.persona = persona;
    }
    public Propietario(int codProp, Proyectoista proyectoista,
        Apoderado apoderado, Persona persona, Set tramites)
    {
        this.codProp = codProp;
        this.proyectoista = proyectoista;
        this.apoderado = apoderado;
        this.persona = persona;
        this.tramites = tramites;
    }

    public int getCodProp() {
        return this.codProp;
    }
    public void setCodProp(int codProp) {
        this.codProp = codProp;
    }
    public Proyectoista getProyectoista() {
        return this.proyectoista;
    }
    public void setProyectoista(Proyectoista proyectoista) {
        this.proyectoista = proyectoista;
    }
    public Apoderado getApoderado() {

```

```

        return this.apoderado;
    }
    public void setApoderado(Apoderado apoderado) {
        this.apoderado = apoderado;
    }
    public Persona getPersona() {
        return this.persona;
    }
    public void setPersona(Persona persona) {
        this.persona = persona;
    }
    public Set getTramites() {
        return this.tramites;
    }
    public void setTramites(Set tramites) {
        this.tramites = tramites;
    }
}

```

Clase: Projectista

```

public class Projectista implements java.io.Serializable {

    private int codProy;
    private Persona persona;
    private Set propietarios = new HashSet(0);

    public Projectista() {
    }
    public Projectista(int codProy, Persona persona) {
        this.codProy = codProy;
        this.persona = persona;
    }
    public Projectista(int codProy, Persona persona, Set
        propietarios) {
        this.codProy = codProy;
        this.persona = persona;
        this.propietarios = propietarios;
    }
    public int getCodProy() {
        return this.codProy;
    }
    public void setCodProy(int codProy) {
        this.codProy = codProy;
    }
    public Persona getPersona() {
        return this.persona;
    }
    public void setPersona(Persona persona) {
        this.persona = persona;
    }
    public Set getPropietarios() {
        return this.propietarios;
    }
}

```

```

        public void setPropietarios(Set propietarios) {
            this.propietarios = propietarios;
        }
    }
}

```

Clase: Reasignación

```

public class Reasignacion implements java.io.Serializable {

    private int codRea;
    private Tramite tramite;
    private Administrador administrador;
    private Date fechaRea;
    private Date horaRea;
    private String detalleRea;

    public Reasignacion() {
    }
    public Reasignacion(int codRea, Tramite tramite, Administrador
        administrador) {
        this.codRea = codRea;
        this.tramite = tramite;
        this.administrador = administrador;
    }
    public Reasignacion(int codRea, Tramite tramite,
        Administrador administrador, Date fechaRea, Date
        horaRea, String detalleRea) {
        this.codRea = codRea;
        this.tramite = tramite;
        this.administrador = administrador;
        this.fechaRea = fechaRea;
        this.horaRea = horaRea;
        this.detalleRea = detalleRea;
    }
    public int getCodRea() {
        return this.codRea;
    }
    public void setCodRea(int codRea) {
        this.codRea = codRea;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Date getFechaRea() {
        return this.fechaRea;
    }
}

```

```

    }
    public void setFechaRea(Date fechaRea) {
        this.fechaRea = fechaRea;
    }
    public Date getHoraRea() {
        return this.horaRea;
    }
    public void setHoraRea(Date horaRea) {
        this.horaRea = horaRea;
    }
    public String getDetalleRea() {
        return this.detalleRea;
    }
    public void setDetalleRea(String detalleRea) {
        this.detalleRea = detalleRea;
    }
}

```

Clase: Recepción

```

public class Recepcion implements java.io.Serializable {

    private int codRecep;
    private Envios envios;
    private Tramite tramite;
    private Administrador administrador;
    private Date fechaRecep;
    private Date horaRecep;
    private String observacion;

    public Recepcion() {
    }
    public Recepcion(int codRecep, Envios envios, Tramite tramite,
        Administrador administrador) {
        this.codRecep = codRecep;
        this.envios = envios;
        this.tramite = tramite;
        this.administrador = administrador;
    }
    public Recepcion(int codRecep, Envios envios, Tramite tramite,
        Administrador administrador, Date fechaRecep, Date
        horaRecep, String observacion) {
        this.codRecep = codRecep;
        this.envios = envios;
        this.tramite = tramite;
        this.administrador = administrador;
        this.fechaRecep = fechaRecep;
        this.horaRecep = horaRecep;
        this.observacion = observacion;
    }
    public int getCodRecep() {
        return this.codRecep;
    }
    public void setCodRecep(int codRecep) {

```

```

        this.codRecep = codRecep;
    }
    public Envios getEnvios() {
        return this.envios;
    }
    public void setEnvios(Envios envios) {
        this.envios = envios;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Date getFechaRecep() {
        return this.fechaRecep;
    }
    public void setFechaRecep(Date fechaRecep) {
        this.fechaRecep = fechaRecep;
    }
    public Date getHoraRecep() {
        return this.horaRecep;
    }
    public void setHoraRecep(Date horaRecep) {
        this.horaRecep = horaRecep;
    }
    public String getObservacion() {
        return this.observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
}

```

Clase: Requisitos

```

public class Requisitos implements java.io.Serializable {

    private int codReq;
    private TipoTramite tipoTramite;
    private String requisitos;
    private Set detalleRequisitoses = new HashSet(0);

    public Requisitos() {
    }
    public Requisitos(int codReq, TipoTramite tipoTramite) {
        this.codReq = codReq;
        this.tipoTramite = tipoTramite;
    }
}

```

```

    }
    public Requisitos(int codReq, TipoTramite tipoTramite, String r
        equisitos,Set detalleRequisitoses) {
        this.codReq = codReq;
        this.tipoTramite = tipoTramite;
        this.requisitos = requisitos;
        this.detalleRequisitoses = detalleRequisitoses;
    }

    public int getCodReq() {
        return this.codReq;
    }
    public void setCodReq(int codReq) {
        this.codReq = codReq;
    }
    public TipoTramite getTipoTramite() {
        return this.tipoTramite;
    }
    public void setTipoTramite(TipoTramite tipoTramite) {
        this.tipoTramite = tipoTramite;
    }
    public String getRequisitos() {
        return this.requisitos;
    }
    public void setRequisitos(String requisitos) {
        this.requisitos = requisitos;
    }
    public Set getDetalleRequisitoses() {
        return this.detalleRequisitoses;
    }
    public void setDetalleRequisitoses(Set detalleRequisitoses) {
        this.detalleRequisitoses = detalleRequisitoses;
    }
}

```

Clase: Rol

```

public class Rol implements java.io.Serializable {

    private int codRol;
    private String rol;
    private String descripcion;
    private Set usuarios = new HashSet(0);

    public Rol() {
    }
    public Rol(int codRol) {
        this.codRol = codRol;
    }
    public Rol(int codRol, String rol,String descripcion, Set
        usuarios) {
        this.codRol = codRol;
        this.rol = rol;
        this.descripcion = descripcion;
    }
}

```

```

        this.usuarios = usuarios;
    }
    public int getCodRol() {
        return this.codRol;
    }
    public void setCodRol(int codRol) {
        this.codRol = codRol;
    }
    public String getRol() {
        return this.rol;
    }
    public void setRol(String rol) {
        this.rol = rol;
    }
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    public Set getUsuarios() {
        return this.usuarios;
    }
    public void setUsuarios(Set usuarios) {
        this.usuarios = usuarios;
    }
}

```

Clase: Salidas

```

public class Salidas implements java.io.Serializable {

    private int codSal;
    private Tramite tramite;
    private Administrador administrador;
    private Date fechaSal;
    private Date horaSal;
    private String observacion;
    private Set ingresoses = new HashSet(0);

    public Salidas() {
    }

    public Salidas(int codSal, Tramite tramite, Administrador
administrador) {
        this.codSal = codSal;
        this.tramite = tramite;
        this.administrador = administrador;
    }
    public Salidas(int codSal, Tramite tramite, Administrador
administrador, Date fechaSal, Date horaSal, String
observacion, Set ingresoses) {
        this.codSal = codSal;
        this.tramite = tramite;
    }
}

```

```

        this.administrador = administrador;
        this.fechaSal = fechaSal;
        this.horaSal = horaSal;
        this.observacion = observacion;
        this.ingresos = ingresos;
    }
    public int getCodSal() {
        return this.codSal;
    }
    public void setCodSal(int codSal) {
        this.codSal = codSal;
    }
    public Tramite getTramite() {
        return this.tramite;
    }
    public void setTramite(Tramite tramite) {
        this.tramite = tramite;
    }
    public Administrador getAdministrador() {
        return this.administrador;
    }
    public void setAdministrador(Administrador administrador) {
        this.administrador = administrador;
    }
    public Date getFechaSal() {
        return this.fechaSal;
    }
    public void setFechaSal(Date fechaSal) {
        this.fechaSal = fechaSal;
    }
    public Date getHoraSal() {
        return this.horaSal;
    }
    public void setHoraSal(Date horaSal) {
        this.horaSal = horaSal;
    }
    public String getObservacion() {
        return this.observacion;
    }
    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
    public Set getIngresos() {
        return this.ingresos;
    }
    public void setIngresos(Set ingresos) {
        this.ingresos = ingresos;
    }
}

```

Clase: Teléfono

```

public class Telefono implements java.io.Serializable {

```



```

private static final long serialVersionUID = 1L;
private int codTelf;
private Persona persona;
private String numero;

public Telefono() {
}
public Telefono(int codTelf, Persona persona) {
    this.codTelf = codTelf;
    this.persona = persona;
}

public Telefono(int codTelf, Persona persona, String numero) {
    this.codTelf = codTelf;
    this.persona = persona;
    this.numero = numero;
}
public int getCodTelf() {
    return this.codTelf;
}
public void setCodTelf(int codTelf) {
    this.codTelf = codTelf;
}
public Persona getPersona() {
    return this.persona;
}
public void setPersona(Persona persona) {
    this.persona = persona;
}
public String getNumero() {
    return this.numero;
}
public void setNumero(String numero) {
    this.numero = numero;
}
}

```

Clase: TipoTramite

```

public class TipoTramite implements java.io.Serializable {

    private int codTipTram;
    private String tramite;
    private String area;
    private Set requisitoses = new HashSet(0);
    private Set tramites = new HashSet(0);

    public TipoTramite() {
    }
    public TipoTramite(int codTipTram ) {
        this.codTipTram = codTipTram;
        //this.tramite = tramite; , String tramite
    }
}

```

```

@SuppressWarnings("unchecked")
public TipoTramite(int codTipTram, String tramite, String area,
    Set requisitoses, Set tramites) {
    this.codTipTram = codTipTram;
    this.tramite = tramite;
    this.area = area;
    this.requisitoses = requisitoses;
    this.tramites = tramites;
}
public int getCodTipTram() {
    return this.codTipTram;
}
public void setCodTipTram(int codTipTram) {
    this.codTipTram = codTipTram;
}
public String getTramite() {
    return this.tramite;
}
public void setTramite(String tramite) {
    this.tramite = tramite;
}
public String getArea() {
    return this.area;
}
public void setArea(String area) {
    this.area = area;
}
public Set getRequisitoses() {
    return this.requisitoses;
}
public void setRequisitoses(Set requisitoses) {
    this.requisitoses = requisitoses;
}

public Set getTramites() {
    return this.tramites;
}
public void setTramites(Set tramites) {
    this.tramites = tramites;
}
}

```

Clase: Trámite

```

public class Tramite implements java.io.Serializable {

    private int codTram;
    private TipoTramite tipoTramite;
    private Propietario propietario;
    private Date fecha;
    private Date hora;
    private int gestion;
    private int numPrevio;
    private int numOficial;
}

```

```

private String proceso;
private String estado;
private Set datosTramites = new HashSet(0);
private Set numArchivos = new HashSet(0);
private Set detalleRequisitoses = new HashSet(0);
private Set recepciones = new HashSet(0);
private Set salidas = new HashSet(0);
private Set observacioneses = new HashSet(0);
private Set envioses = new HashSet(0);
private Set reasignaciones = new HashSet(0);
private Set aprobars = new HashSet(0);
private Set ingresoses = new HashSet(0);
private Set bloqueoDesbloques = new HashSet(0);
private Set darOks = new HashSet(0);
private Set bajasAltases = new HashSet(0);
private Set detalleAdtrams = new HashSet(0);

public Tramite() {
}

public Tramite(int codTram, TipoTramite tipoTramite,
    Propietario propietario, int gestion, int numPrevio, int
numOficial) {
    this.codTram = codTram;
    this.tipoTramite = tipoTramite;
    this.propietario = propietario;
    this.gestion = gestion;
    this.numPrevio = numPrevio;
    this.numOficial = numOficial;
}

public Tramite(int codTram, TipoTramite tipoTramite,
    Propietario propietario, Date fecha, Date hora, int gestion,
int numPrevio, int numOficial, String proceso, String estado,
Set datosTramites, Set numArchivos, Set detalleRequisitoses,
Set recepciones, Set salidas, Set observacioneses, Set envioses,
    Set reasignaciones, Set aprobars, Set ingresoses,
    Set bloqueoDesbloques, Set darOks, Set bajasAltases,
    Set detalleAdtrams) {
    this.codTram = codTram;
    this.tipoTramite = tipoTramite;
    this.propietario = propietario;
    this.fecha = fecha;
    this.hora = hora;
    this.gestion = gestion;
    this.numPrevio = numPrevio;
    this.numOficial = numOficial;
    this.proceso = proceso;
    this.estado = estado;
    this.datosTramites = datosTramites;
    this.numArchivos = numArchivos;
    this.detalleRequisitoses = detalleRequisitoses;
    this.recepciones = recepciones;
    this.salidas = salidas;
}

```

```

        this.observacioneses = observacioneses;
        this.envioses = envioses;
        this.reasignacions = reasignacions;
        this.aprobars = aprobars;
        this.ingresoses = ingresoses;
        this.bloqueoDesbloques = bloqueoDesbloques;
        this.darOks = darOks;
        this.bajasAltases = bajasAltases;
        this.detalleAdtrams = detalleAdtrams;
    }

    public int getCodTram() {
        return this.codTram;
    }
    public void setCodTram(int codTram) {
        this.codTram = codTram;
    }
    public TipoTramite getTipoTramite() {
        return this.tipoTramite;
    }
    public void setTipoTramite(TipoTramite tipoTramite) {
        this.tipoTramite = tipoTramite;
    }
    public Propietario getPropietario() {
        return this.propietario;
    }
    public void setPropietario(Propietario propietario) {
        this.propietario = propietario;
    }
    public Date getFecha() {
        return this.fecha;
    }
    public void setFecha(Date fecha) {
        this.fecha = fecha;
    }
    public Date getHora() {
        return this.hora;
    }
    public void setHora(Date hora) {
        this.hora = hora;
    }
    public int getGestion() {
        return this.gestion;
    }
    public void setGestion(int gestion) {
        this.gestion = gestion;
    }
    public int getNumPrevio() {
        return this.numPrevio;
    }
    public void setNumPrevio(int numPrevio) {
        this.numPrevio = numPrevio;
    }
    public int getNumOficial() {

```

```

        return this.numOficial;
    }
    public void setNumOficial(int numOficial) {
        this.numOficial = numOficial;
    }
    public String getProceso() {
        return this.proceso;
    }
    public void setProceso(String proceso) {
        this.proceso = proceso;
    }
    public String getEstado() {
        return this.estado;
    }
    public void setEstado(String estado) {
        this.estado = estado;
    }
    public Set getDatosTramites() {
        return this.datosTramites;
    }
    public void setDatosTramites(Set datosTramites) {
        this.datosTramites = datosTramites;
    }
    public Set getNumArchivos() {
        return this.numArchivos;
    }

    public void setNumArchivos(Set numArchivos) {
        this.numArchivos = numArchivos;
    }
    public Set getDetalleRequisitoses() {
        return this.detalleRequisitoses;
    }
    public void setDetalleRequisitoses(Set detalleRequisitoses) {
        this.detalleRequisitoses = detalleRequisitoses;
    }
    public Set getRecepcions() {
        return this.recepcions;
    }
    public void setRecepcions(Set recepcions) {
        this.recepcions = recepcions;
    }
    public Set getSalidases() {
        return this.salidases;
    }
    public void setSalidases(Set salidases) {
        this.salidases = salidases;
    }
    public Set getObservacioneses() {
        return this.observacioneses;
    }
    public void setObservacioneses(Set observacioneses) {
        this.observacioneses = observacioneses;
    }
}

```

```

    public Set getEnvioses() {
        return this.envioses;
    }
    public void setEnvioses(Set envioses) {
        this.envioses = envioses;
    }
    public Set getReasignaciones() {
        return this.reasignaciones;
    }
    public void setReasignaciones(Set reasignaciones) {
        this.reasignaciones = reasignaciones;
    }
    public Set getAprobars() {
        return this.aprobars;
    }
    public void setAprobars(Set aprobars) {
        this.aprobars = aprobars;
    }
    public Set getIngresoses() {
        return this.ingresoses;
    }
    public void setIngresoses(Set ingresoses) {
        this.ingresoses = ingresoses;
    }
    public Set getBloqueoDesbloques() {
        return this.bloqueoDesbloques;
    }
    public void setBloqueoDesbloques(Set bloqueoDesbloques) {
        this.bloqueoDesbloques = bloqueoDesbloques;
    }
    public Set getDarOks() {
        return this.darOks;
    }
    public void setDarOks(Set darOks) {
        this.darOks = darOks;
    }
    public Set getBajasAltases() {
        return this.bajasAltases;
    }
    public void setBajasAltases(Set bajasAltases) {
        this.bajasAltases = bajasAltases;
    }
    public Set getDetalleAdtrams() {
        return this.detalleAdtrams;
    }
    public void setDetalleAdtrams(Set detalleAdtrams) {
        this.detalleAdtrams = detalleAdtrams;
    }
}

```

Clase: Usuario

```

public class Usuario implements java.io.Serializable {

```

```

private int codUs;
private Administrador administrador;
private Rol rol;
private String login;
private String clave;

public Usuario() {
}

public Usuario(int codUs, Administrador administrador, Rol rol,
               String usuario, String clave) {
    this.codUs = codUs;
    this.administrador = administrador;
    this.rol = rol;
    this.login = login;
    this.clave = clave;
}

public int getCodUs() {
    return this.codUs;
}

public void setCodUs(int codUs) {
    this.codUs = codUs;
}

public Administrador getAdministrador() {
    return this.administrador;
}

public void setAdministrador(Administrador administrador) {
    this.administrador = administrador;
}

public Rol getRol() {
    return this.rol;
}

public void setRol(Rol rol) {
    this.rol = rol;
}

public String getLogin() {
    return this.login;
}

public void setLogin(String login) {
    this.login = login;
}

public String getClave() {
    return this.clave;
}

public void setClave(String clave) {
    this.clave = clave;
}
}

```

Clase: Zona

```

public class Zona implements java.io.Serializable {

```

```

private static final long serialVersionUID = 1L;
private int codZona;
private String zona;
@SuppressWarnings("unchecked")
private Set datosTramites = new HashSet(0);
@SuppressWarnings("unchecked")
private Set direcciones = new HashSet(0);

public Zona() {
}
public Zona(int codZona) {
    this.codZona = codZona;
}
@SuppressWarnings("unchecked")
public Zona(int codZona, String zona, Set datosTramites, Set
    direcciones) {
    this.codZona = codZona;
    this.zona = zona;
    this.datosTramites = datosTramites;
    this.direcciones = direcciones;
}
public int getCodZona() {
    return this.codZona;
}
public void setCodZona(int codZona) {
    this.codZona = codZona;
}
public String getZona() {
    return this.zona;
}
public void setZona(String zona) {
    this.zona = zona;
}
@SuppressWarnings("unchecked")
public Set getDatosTramites() {
    return this.datosTramites;
}
@SuppressWarnings("unchecked")
public void setDatosTramites(Set datosTramites) {
    this.datosTramites = datosTramites;
}
@SuppressWarnings("unchecked")
public Set getDirecciones() {
    return this.direcciones;
}
@SuppressWarnings("unchecked")
public void setDirecciones(Set direcciones) {
    this.direcciones = direcciones;
}
}
}

```


IV.5. MODELADO DE IMPLEMENTACIÓN

IV.5.1. 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).

IV.5.2 Diagrama de Componentes

IV.5.1.1 Ingresar al Sistema

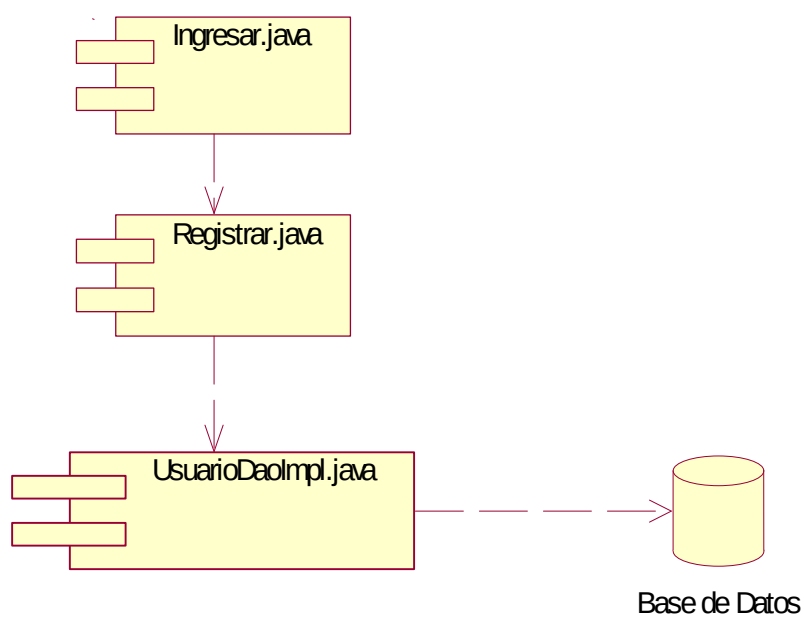


Figura 437: Diagrama de Componente: Ingresar al Sistema

IV.5.1.2 Resguardar y Restaurar Base de Datos

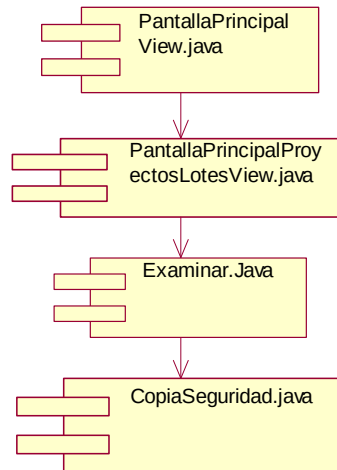


Figura 438: Diagrama de Componente: Resguardar y Restaurar Base de Datos

IV.5.1.3 Adminstrar Propietarios

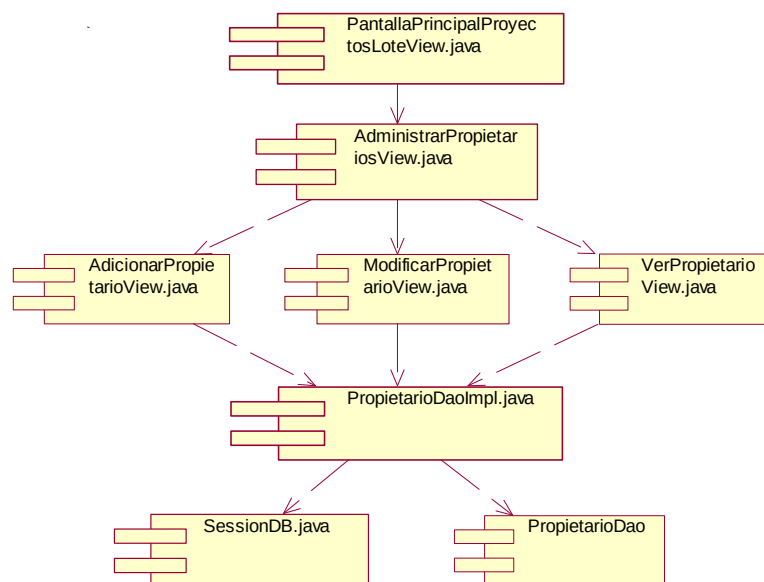


Figura 439: Diagrama de Componente: Adminstrar Propietarios

IV.5.1.4 Adminstrar Ingresos

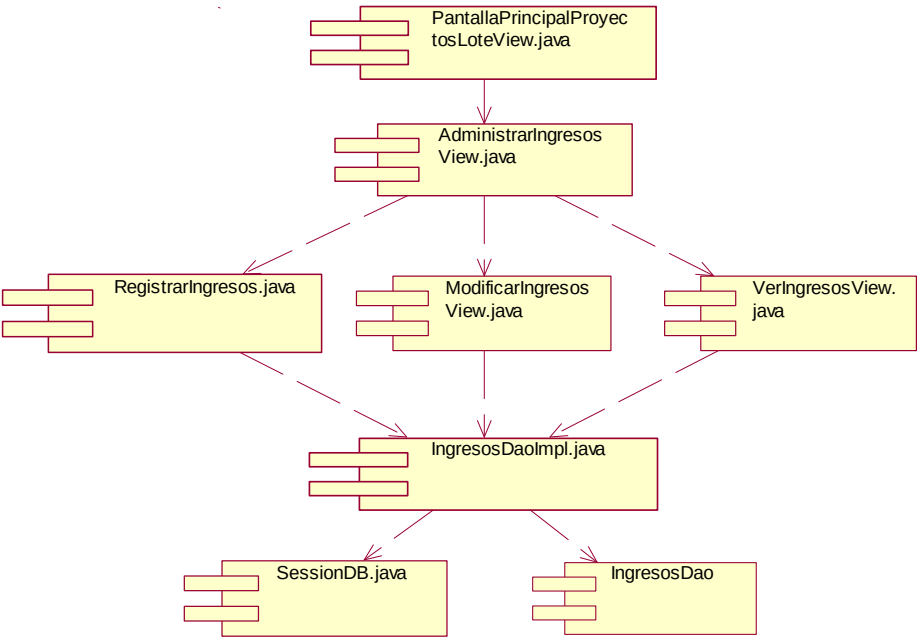


Figura 440: Diagrama de Componente: Adminstrar Ingresos

IV.5.1.5 Adminstrar Salidas

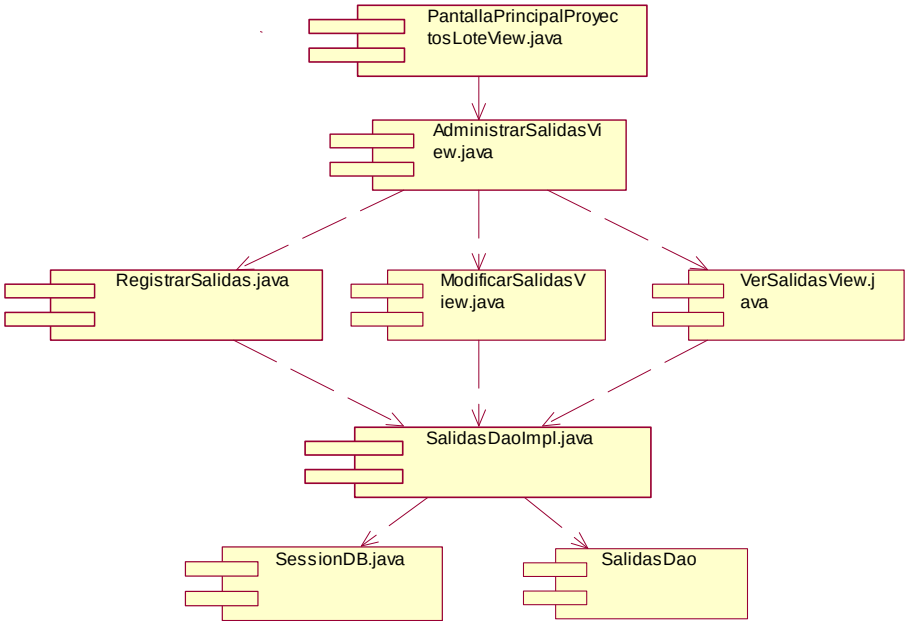


Figura 441: Diagrama de Componente: Adminstrar Salidas

IV.5.1.7 Adminstrar Envios

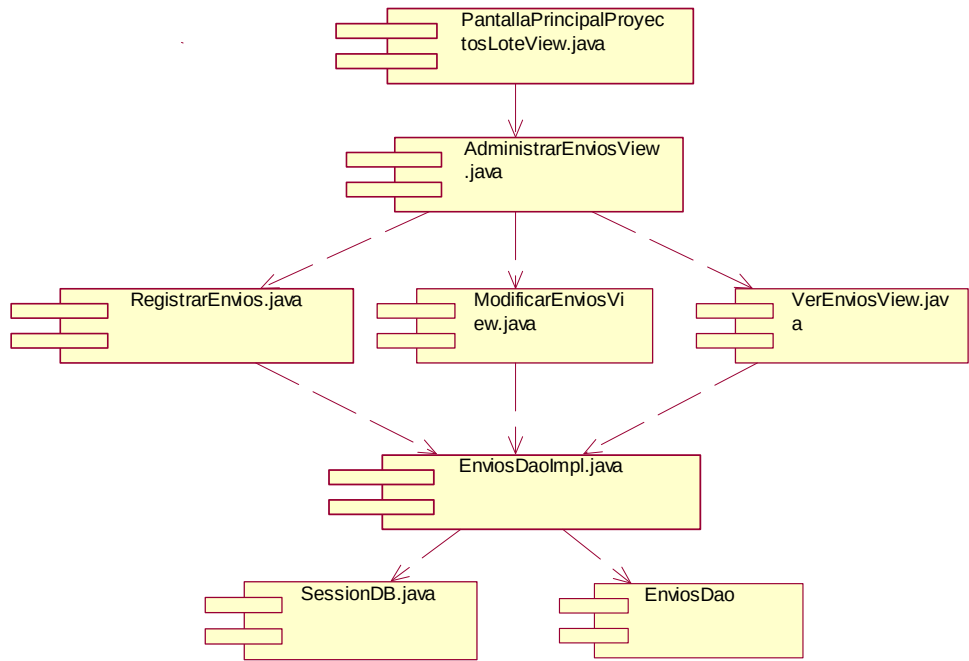


Figura 442: Diagrama de Componente: Adminstrar Envios

IV.5.1.7 Adminstrar Recepción

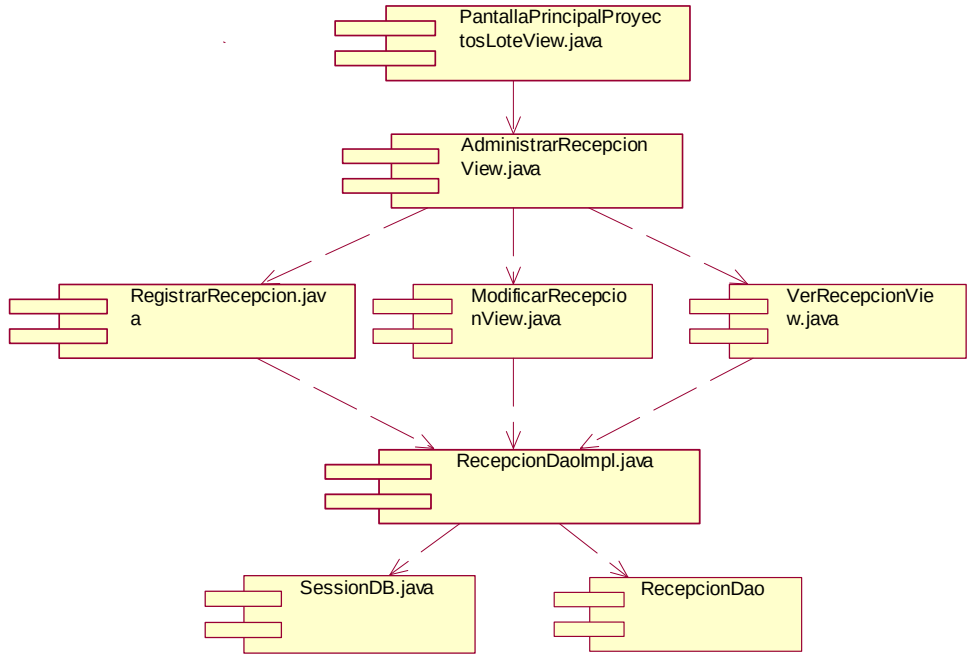


Figura 443: Diagrama de Componente: Adminstrar Recepción

IV.5.1.8 Reasignar Trámite

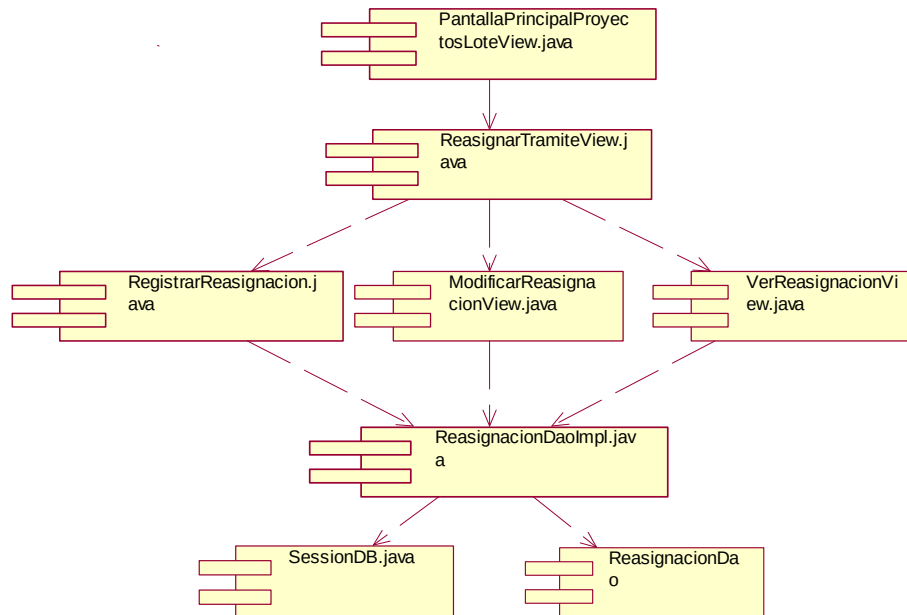


Figura 444: Diagrama de Componente: Reasignar Trámite

IV.5.1.9 Administrar Observaciones Generales

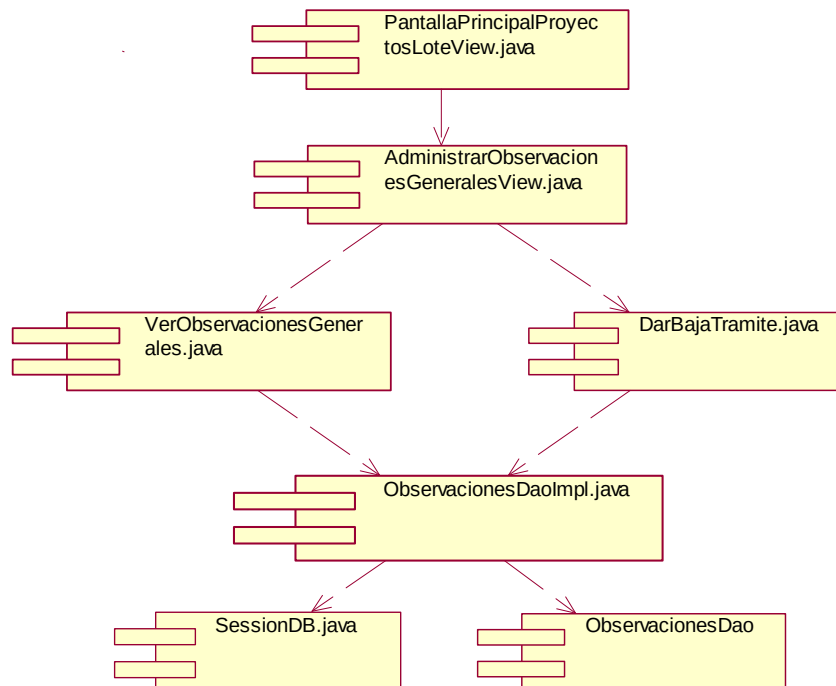


Figura 445: Diagrama de Componente: Administrar Observaciones Generales

IV.5.1.10 Administrar Personal

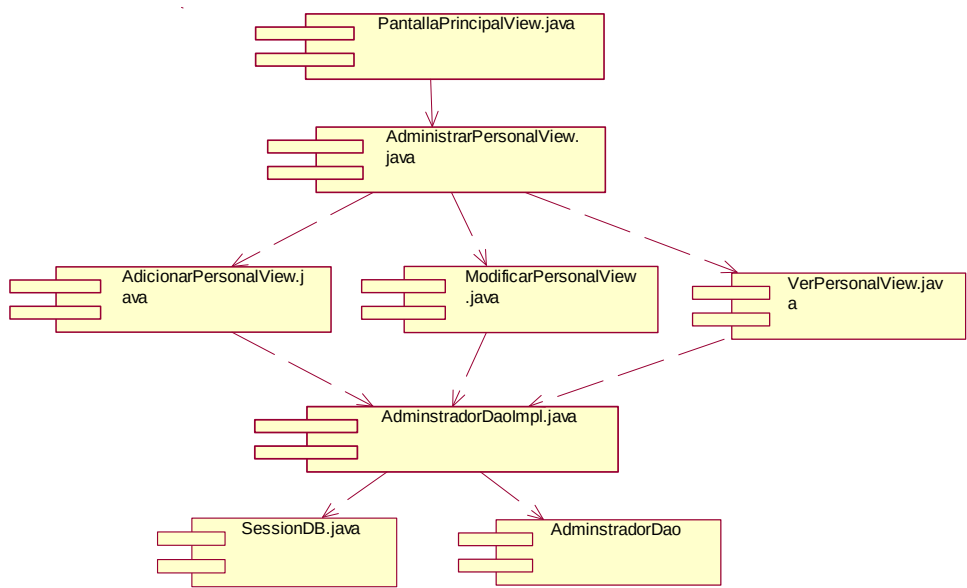


Figura 446: Diagrama de Componente: Resguardar y Restaurar Base de Datos

IV.6. CASOS DE PRUEBA

IV.6.1. Introducción

Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba.

IV.6.2. Definición

La prueba es un proceso de ejecución de un programa con la intención de descubrir errores.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

IV.6.3. Prueba de Caja Blanca

Permite examinar la estructura interna del programa. Se diseñan casos de prueba para examinar la lógica del programa.

Código de los Casos de Prueba

Técnica: Camino Básico

Administrar PropietariosController: cargardatos ()

```
public Object[][] cargardatos(){
    System.out.println("entro aqui");
    List lista=this.tramiteDaoImpl.getAllTramite();
    //List listaProp=this.propietarioDaoImpl.getAllPropietario();
    Object[] objetosArray = lista.toArray();
    //Object[] objetosArray1 = listaProp.toArray();
    Object[][] resultado = new Object[objetosArray.length][6];
    for (int i = 0; i < objetosArray.length; i++) {
        System.out.println("aaaaa1");
        Tramite tramite = (Tramite) objetosArray[i];
        //Propietario propietario = (Propietario)
objetosArray1[i];
        System.out.println("aaaaa2");
        resultado[i][0] =tramite.getPropietario().getCodProp();
        resultado[i][1] =tramite.getNumPrevio();
        System.out.println("aaaaa3"+tramite.getNumPrevio());
        resultado[i][2]
```

```

=tramite.getPropietario().getPersona().getNombre();
    System.out.println("aaaaa3");
    resultado[i][3]
=tramite.getPropietario().getPersona().getAp();
    resultado[i][4]
=tramite.getPropietario().getPersona().getAm();
    resultado[i][5]
=tramite.getPropietario().getProyectista().getPersona().getNombre()+" " +
    ""+tramite.getPropietario().getProyectista().getPersona().getAp();
    System.out.println("aaaaa4");
}
return resultado;
}
}

```

Complejidad: $V(G) = A - N + 2$

Donde A es el número de aristas y N es el número de nodos contenidos en el grafo.

$$V(G) = 9 - 8 + 2 = 3$$

$$V(G) = 3 \text{ Regiones}$$

$$V(G) = 2 \text{ Nodos Predicados} + 1 = 3$$

Caminos:

C1: 1-2-3-4-5-7-8

C2: 1-2-3-4-6-7-8

C3: 1-2-8

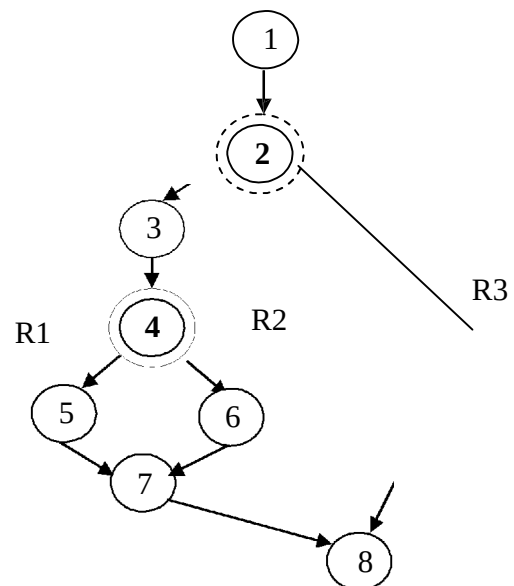


Fig. 447: Caso de Prueba: AdministrarPropietariosController

Administrar PersonalController: cargarDatosByEstado()

```

public Object[][] cargarDatosByEstado(){
    String datoEstado = "activo";
    if (administrarPersonalView.Pasivos.isSelected() == true){
        datoEstado = "inactivo";
    }
    // List lista=this.usuarioDaoImpl.getAllUsuarioByEstado(datoEstado,
    xcodAsociacion);
    List
    lista=this.administradorDaoImpl.getAllAdministradorByEstado(datoEstado);
    List usu =this.usuarioDaoImpl.getAllUsuario();
    //nombreRol=usuario.getRol().getRol();
    Object[] objetosArray = lista.toArray();
    Object[] objetosArray1 = usu.toArray();
    Object[][] resultado = new Object[objetosArray.length][6];
    for (int i = 0; i < objetosArray.length; i++) {
        Administrador administrador = (Administrador)
objetosArray[i];
        System.out.println("aaaaa1");
        Usuario usuario = (Usuario) objetosArray1[i];
        System.out.println("aaaaa2");
        resultado[i][0] =administrador.getCodAdm();
        resultado[i][1] =administrador.getPersona().getCi();
        resultado[i][2] =administrador.getPersona().getNombre();
        resultado[i][3] =administrador.getPersona().getAp();
        resultado[i][4] =administrador.getPersona().getAm();
        System.out.println("aaaaa3");
        resultado[i][5] =usuario.getRol().getRol();
        System.out.println("aaaaa4");
    }
    return resultado;
}

```

$$V(G) = 5 - 5 + 2 = 2$$

$$V(G) = 2 \text{ Regiones}$$

$$V(G) = 1 \text{ Nodo Predicado} + 1 = 2$$

Caminos:

C1: 1-2-3-5

C2: 1-2-4-5

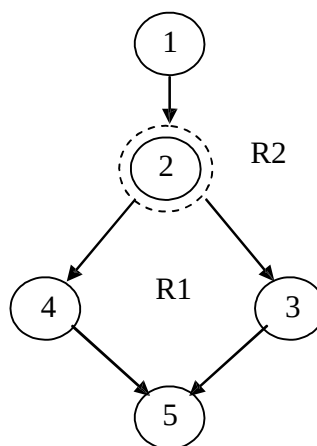


Fig. 448: Caso de Prueba: AdministrarPersonalController

DarBajaController: aceptar()

```

public void aceptar(){
    this.administrarPersonalView=administrarPersonalView;

    try {
        if (validarDatos().equals("Correcto")) {
            Administrador administrador = new
Administrador();//this.empleadoDaoImpl.getEmpleadoByCiPersona(adicionarEmp
pleadoView.jTCi.getText());
            PersonalBaja personal_baja = new
PersonalBaja();//this.personaDaoImpl.getPersonaByCi(adicionarEmpleadoView.
jTCi.getText());
            //System.out.println(persona.getNombre());
            @SuppressWarnings("unused")
            boolean motivo, detalle;
            motivo = darBajaView.textEmail.getText().isEmpty();
            detalle = darBajaView.jTextArea1.getText().isEmpty();

            if(!(darBajaView.textEmail.getText().isEmpty())){
                if (!(motivo || detalle)) {
                    Administrador administrador1 =
administradorDaoImpl.getAdministradorBycodAdm(Integer.parseInt(codigoadmin
));
                    administrador1.setEstado("inactivo");

                this.administradorDaoImpl.updateAdministrador(administrador1);

                    personal_baja = new PersonalBaja();

personal_baja.setMotivo(darBajaView.textEmail.getText());
personal_baja.setDetalle(darBajaView.jTextArea1.getText());
personal_baja.setFechaPB(darBajaView.fechaIngreso.getDate());
personal_baja.setHoraPB(darBajaView.fechaIngreso.getDate());
                    personal_baja.setEstado("inactivo");
                    personal_baja.setAdministrador(administrador1);

                this.personalBajaDaoImpl.savePersonalBaja(personal_baja);

                    JOptionPane.showMessageDialog(null, " El Administrador
"+nombreAdm+" a Sido Dado De Baja Satisfactoriamente ", "Informe del
Sistema", JOptionPane.INFORMATION_MESSAGE);
                    administrarPersonalView.borrarView();
                    administrarPersonalView.cargarView();
                    this.darBajaView.setVisible(false);
                }
            }
            else{
                JOptionPane.showMessageDialog(null, " Introduzca todos
los campos", "Informe del Sistema", JOptionPane.INFORMATION_MESSAGE);
                System.out.println(administrador.getCodAdm() + "-----
else" + administrador.toString());
            }
        }
    }
}

```

```
        }  
    }  
    else{  
        JOptionPane.showMessageDialog(null, "El Administrador ya  
ha sido dado de baja", "Informe del Sistema",  
JOptionPane.INFORMATION_MESSAGE);  
    }  
}  
  
} //fin del if validar  
catch(Exception e){  
    JOptionPane.showMessageDialog(null, " Introduzca Bien los  
Datos", "Informe del Sistema", JOptionPane.INFORMATION_MESSAGE);  
    e.printStackTrace();  
}
```

$$V(G) = 9 - 8 + 2 = 3$$

$$V(G) = 3 \text{ Regiones}$$

$$V(G) = 2 \text{ Nodos Predicados} + 1 = 3$$

Caminos:

C1: 1-2-3-4-5-7-8

C2: 1-2-3-4-6-7-8

C3: 1-2-8

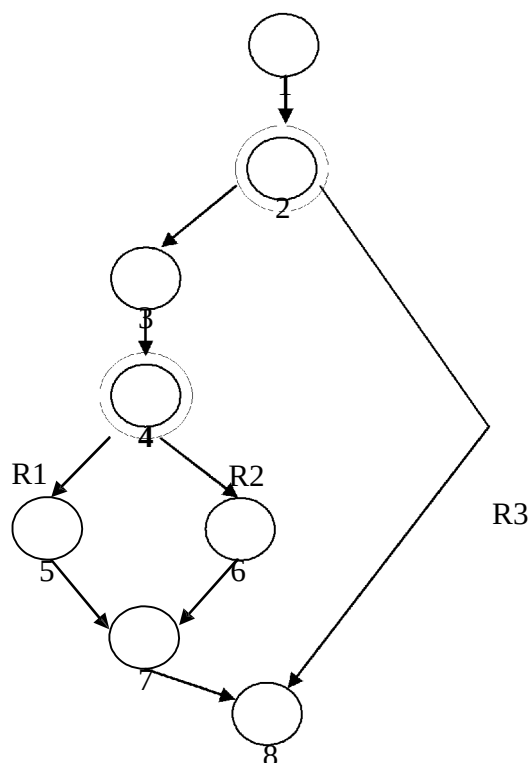


Fig. 449: Caso de Prueba: DarBajaController

AdministrarPersonalBajaController: cargardatos()

```

public Object[][] cargardatos(){
    List lista=this.administradorDaoImpl.getAllAdministrador2();
    List usu =this.usuarioDaoImpl.getAllUsuario();
    //nombreRol=usuario.getRol().getRol();
    Object[] objetosArray = lista.toArray();
    Object[] objetosArray1 = usu.toArray();
    Object[][] resultado = new Object[objetosArray.length][6];
    for (int i = 0; i < objetosArray.length; i++) {
        Administrador administrador = (Administrador)
objetosArray[i];
        System.out.println("aaaaa1");
        Usuario usuario = (Usuario) objetosArray1[i];
        System.out.println("aaaaa2");
        resultado[i][0] =administrador.getCodAdm();
        resultado[i][1] =administrador.getPersona().getCi();
        resultado[i][2] =administrador.getPersona().getNombre();
        resultado[i][3] =administrador.getPersona().getAp();
        resultado[i][4] =administrador.getPersona().getAm();
        System.out.println("aaaaa3");
        resultado[i][5] =usuario.getRol().getRol();
        System.out.println("aaaaa4");
    }
    return resultado;
}

```

}

$$V(G) = 11 - 10 + 2 = 3$$

$$V(G) = 3 \text{ Regiones}$$

$$V(G) = 2 \text{ Nodos Predicados} + 1 = 3$$

Caminos:

C1: 1-2-3-4-5-6-8-10

C2: 1-2-3-4-5-7-8-10

C3: 1-2-3-9-10

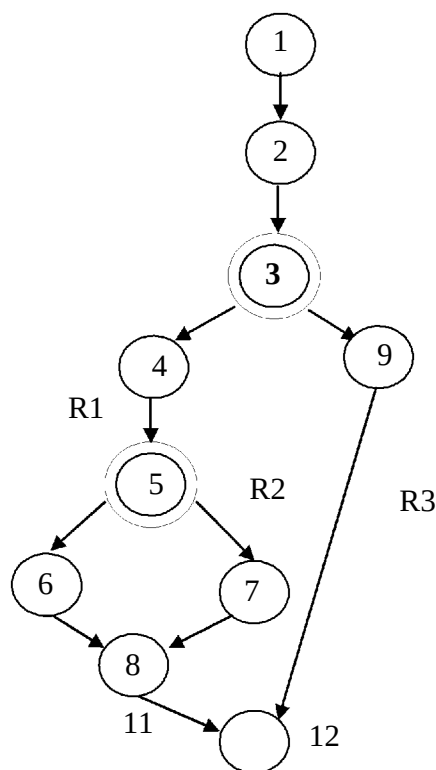


Fig. 450: Caso de Prueba: AdminstrarPersonalBajaController

DAltaController: aceptar().

```

public void aceptar(AdministrarPersonalBajaView
administrarPersonalBajaView, String codigoadmin){
    this.administrarPersonalBajaView=administrarPersonalBajaView;

    try {
        if (validarDatos().equals("Correcto")) {
            Administrador administrador = new
Administrador();//this.empleadoDaoImpl.getEmpleadoByCiPersona(adicionarEmp
pleadoView.jTCi.getText());
            PersonalBaja personal_baja = new
PersonalBaja();//this.personaDaoImpl.getPersonaByCi(adicionarEmpleadoView.
jTCi.getText());
            //System.out.println(persona.getNombre());
            @SuppressWarnings("unused")
            boolean motivo, detalle;
            motivo = darAltaView.textEmail.getText().isEmpty();
            detalle = darAltaView.jTextArea1.getText().isEmpty();

```

```

        if(! (darAltaView.textEmail.getText().isEmpty())){
            if (! (motivo || detalle)) {
                Administrador administrador1 =
administradorDaoImpl.getAdministradorBycodAdm(Integer.parseInt(codigoadmin
));
                administrador1.setEstado("activo");

this.administradorDaoImpl.updateAdministrador(administrador1);

                personal_baja = new PersonalBaja();
personal_baja.setMotivo(darAltaView.textEmail.getText());
personal_baja.setDetalle(darAltaView.jTextArea1.getText());
personal_baja.setFechaPB(darAltaView.fechaIngreso.getDate());
personal_baja.setHoraPB(darAltaView.fechaIngreso.getDate());
                personal_baja.setEstado("activo");
                personal_baja.setAdministrador(administrador1);

this.personalBajaDaoImpl.savePersonalBaja(personal_baja);

                JOptionPane.showMessageDialog(null, " El Administrador
"+nombreAdm+" a Sido Dado De Alta Satisfactoriamente ", "Informe del
Sistema", JOptionPane.INFORMATION_MESSAGE);
                administrarPersonalBajaView.borrarView();
                administrarPersonalBajaView.cargarView();
                this.darAltaView.setVisible(false);
            }
            else{
                JOptionPane.showMessageDialog(null, " Introduzca todos
los campos", "Informe del Sistema", JOptionPane.INFORMATION_MESSAGE);
                System.out.println(administrador.getCodAdm() + "-----
else" + administrador.toString());
            }
        }
        else{
            JOptionPane.showMessageDialog(null, "El Administrador ya
ha sido dado de baja", "Informe del Sistema",
JOptionPane.INFORMATION_MESSAGE);
        }
    }

} //fin del if validar
catch(Exception e){
    JOptionPane.showMessageDialog(null, " Introduzca Bien los
Datos", "Informe del Sistema", JOptionPane.INFORMATION_MESSAGE);
    e.printStackTrace();
}
}

```

V(G)= 4 - 5 + 2 = 1
V(G)= 1 Regiones
V(G)= 0 Nodos Predicados + 1 = 1
Caminos:
C1: 1-2-3-4-5

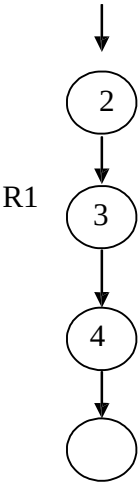


Fig. 451: Caso de Prueba: DarAltaController

COMPONENTE II:

**ESTRATEGIAS DE SOCIALIZACIÓN DEL
SISTEMA INFORMÁTICO DIRIGIDA A LA
POBLACIÓN TARIJEÑA Y A LOS
BENEFICIARIOS DEL PROYECTO**

II. INTRODUCCIÓN

II.1 INTRODUCCIÓN

La importancia creciente que se está dando a invertir en Socialización y Capacitación requiere que el profesional este altamente preparado ante las demandas generadas por los acontecimientos cotidianos en el contexto global en el que se compite. En la era de la globalización, es un desafío para las empresas de países en vías de desarrollo es diseñar Socializaciones y Capacitaciones contingentes y técnicamente abordables, que permitan competir mano a mano con países desarrollados y así lograr el crecimiento y desarrollo que les permita una apertura al mundo en condiciones de equivalencia.

El origen de las demandas de suelen ser diversas: nueva tecnología, presiones sindicales, gerentes operativos o los mismos trabajadores. Muchos exigen capacitarse en algunas materias que creen necesarias pero desconocen los verdaderos problemas de fondo o las soluciones que buscan.

Una de las claves a considerar es el entorno, comprender la situación en que actúan los trabajadores que van a ser capacitadas y la utilización que podrán hacer del aprendizaje obtenido. Esto requiere que el analista o consultor considere el enfoque estratégico de la empresa y el carácter técnico. Cuando se puede definir de mejor modo los roles que cada uno de los protagonistas tendrá, sea relator, socializador, capacitado o analista. De esta manera, el análisis de las relaciones entre cada uno de ellos permitirá definir el campo de acción, las zonas de consenso y de conflicto, en las que se requiere de la negociación para que la acción de la socialización y capacitación pueda desarrollarse y alcanzar sus objetivos.

II.2.1. Introducción

El componente –Estrategias de socialización del Sistema de Escritorio dirigida a la población tarijeña y a los beneficiarios del proyecto||, ha adoptado dos formas de llevar a cabo la realización de éste componente:

- A través de una Campaña Publicitaria.
- Y mediante un curso de capacitación sobre el uso de las TIC y del Sistema de Escritorio, donde el personal y/o representantes de Ordenamiento Territorial asistentes son capacitados.

II.2.2. Propósito

El propósito del componente –Estrategias de socialización del sistema de Escritorio dirigida a la población tarijeña y a los beneficiarios del Proyecto||, es socializar el sistema de Escritorio mediante campaña publicitaria y capacitación al personal de Ordenamiento Territorial de Tarija, en especial al departamento o área de Proyectos y Lotes quien nos está brindando la información necesaria para el desarrollo de éste proyecto. Ahora bien, a través de esta socialización se pretende alcanzar los siguientes Objetivos:

- Dar a conocer al público objetivo las funcionalidades y beneficios del Sistema de Escritorio desarrollado para la Gestión de Información en el Proceso de Seguimiento de Trámites para Ordenamiento Territorial.
- Capacitar a los directores de área y/o representantes de Ordenamiento Territorial de Tarija y en especial al área de Proyectos y Lotes quienes nos cooperaron con la información necesaria para la realización de éste proyecto.

II.2.3. Campaña Publicitaria

II.2.3.1. Comunicación Social

Comunicación social es un campo de estudio sociológico que explora especialmente aquello relacionado con la información, cómo esta es percibida, transmitida y entendida y su impacto social. [1]

II.2.3.1.1. Comunicación

La comunicación es un campo de estudio dentro de las ciencias sociales que trata de explicar cómo se realizan los intercambios comunicativos y cómo estos intercambios afectan a la sociedad y comunicación. Es decir, investiga el conjunto de principios, conceptos y regularidades que sirven de base al estudio de la comunicación como proceso social.

'La comunicación' es un fenómeno de carácter social que comprende todos los actos mediante los cuales los seres vivos se comunican con sus semejantes para transmitir o intercambiar información. Comunicar significa poner en común e implica compartir. [1]

La comunicación inicia con el surgimiento de la vida en nuestro planeta y su desarrollo ha sido simultáneo al progreso de la humanidad. Se manifestó primero a través de un lenguaje no verbal, evolucionando y complejizándose conforme el hombre mismo evolucionaba.

Todos los días los seres vivos se comunican de diferentes maneras, pero sólo los seres humanos podemos hacerlo racionalmente; llevando a cabo infinidad de actividades, tales como: conversar, reír, llorar, leer, callar, ver televisión entre otras; por ello se dice que la comunicación humana es un proceso [1]:

- **Dinámico:** porque está en continuo movimiento y no se limita a una relación

Emisor-->Receptor estático, pues los roles se
intercambian.

- **Inevitable:** Pues es imposible no comunicar, incluso el silencio comunica.
- **Irreversible:** porque una vez realizada, no puede regresar, borrarse o ignorarse.
- **Bidireccional:** porque existe una respuesta en ambas direcciones.
- **Verbal y no verbal:** porque implica la utilización de ambos lenguajes -en algunos casos

II.2.3.1.2. Publicidad

La publicidad es una técnica de comunicación masiva, destinada a difundir o informar al público sobre un bien o servicio a través de los medios de comunicación con el objetivo de motivar al público hacia una acción de consumo.

La publicidad llega al público a través de los medios de comunicación, dichos medios de comunicación emiten los anuncios a cambio de una contraprestación previamente fijada para adquirir dichos espacios en un contrato de compra y venta por la agencia de publicidad y el medio, emitiendo el anuncio en un horario dentro del canal que es previamente fijado por la agencia con el medio, y con el previo conocimiento del anunciante, dicho contrato es denominado contrato de emisión o difusión. [1]

La publicidad informa al consumidor sobre los beneficios de un determinado producto o servicio, resaltando la diferenciación por sobre otras marcas.

Uno de los principales objetivos de la publicidad es crear demanda o modificar las tendencias de demanda de un producto (bien o servicio). Entendiendo que la comercialización busca identificar el mercado apropiado para cada producto, la publicidad es la comunicación por la cual la información sobre el producto es transmitida a estos individuos (público objetivo).

II.2.3.1.3. Medios de Comunicación

Los medios de comunicación son los espacios donde lo público se hace transparente y donde los ciudadanos aprenden los hechos más elementales de la cultura, incluida la cultura democrática: modos de vivir, de comportarse, de relacionarse entre sí, de divertirse, de consumir (Pérez Tornero, 1994, p. 15).

Los medios de comunicación y los contenidos que ellos emiten o permiten acceder, conllevan a una efectiva socialización de cualquier producto o servicio en la población.

Funciones de los medios de medios de comunicación [2]:

Las funciones atribuidas a los medios de comunicación de masas son teóricamente tres:

- Entretener, que habitualmente se indica en tercer lugar, pero que suele ser la más importante; contribuye a la estabilidad social, pero también al embrutecimiento e insensibilización del público (telebasura, prensa del corazón).
- Formar, o educar, es decir, tener contenidos específicos cuya finalidad sea la adquisición de conocimientos por parte del espectador. La televisión educativa, especialmente los documentales de naturaleza, las grandes series de ambientación histórica, las adaptaciones literarias o el teatro televisado, fueron los programas estrella en las televisiones europeas en los años sesenta y setenta del siglo XX; pero declinaron de forma evidente cuando el público pudo escoger otras ofertas entre la variada programación que permitió la multiplicación de cadenas (públicas y privadas) a partir de los años ochenta y noventa. La televisión en América (sobre todo en Estados Unidos), estuvo desde su inicio más volcada en el entretenimiento (teleseries de ficción -policiacas, sentimentales y otros), lo que no impedía el éxito puntual de productos como la adaptación de Raíces.
- Informar, dar a conocer todo aquello que pueda ser relevante para el

público y éste daba conocer.

Entre los medios o canales que utiliza la publicidad para anunciar productos o servicios; tenemos:

- **Anuncios en televisión:** Publicidad realizada a través de cadenas de televisión, bien a través de spots, patrocinios, microespacios temáticos y otros. Es un medio caro y de gran impacto. Sólo utilizable para productos o servicios de amplio consumo. Se han introducido nuevas fórmulas como el patrocinio de programas o recomendación de presentadores.
- **Anuncios en radio:** Desplazada en relevancia por la televisión, mantiene un público que por necesidades concretas o preferencias subjetivas escuchan el medio fielmente.
- **Anuncios en prensa:** Medio muy segmentado por su naturaleza: existen revistas de niños, jóvenes, mujeres, profesionales, etc. Se trata de un medio leído por personas que gustan de información por lo que la publicidad puede ser más extensa y precisa.
- **Anuncios en Trípticos:** Es uno de los muchos espacios publicitarios, en el cual se puede agregar texto e imágenes. Generalmente se hacen con una apariencia muy llamativa para atraer al consumidor. Estos los vemos por todas partes y causan un gran impacto visual. Los trípticos se componen de una información primaria luego por alguna secundaria que pueden detallar esa información, y luego una imagen explícita que se relacione con el texto.
- **Anuncios en Afiches:** Los afiches se utilizan para colocar el nombre de departamentos y secciones con el fin de orientar a la clientela. Suelen incluir el logotipo y los colores corporativos de la cadena de distribución.

Los afiches se usan también para divulgar y apoyar campañas dentro del establecimiento tales como Rebajas, Semana Fantástica, Día del Padre, etc. Generalmente, se colocan pegados en las paredes, muebles o cabeceras de góndola, sobre pies metálicos o colgados del techo.

II.2.3.1.4. Paradigmas de las estrategias publicitarias

Algunas estrategias para la realización de una publicidad efectiva son:

Asociación sico-emotiva al consumidor. Por medio de [18]:

- **Estética:** imágenes, música, personas, etc.
- **Humor.**
- **Sentimientos:** amor materno, enamoramiento, etc.
- **Testimoniales:** de unas figuras o personas famosas o reconocidas de forma positiva, o de personajes de asociación proactiva.
- **Demostración:** Pruebas, tests, ensayos.

Oportunidad. El mensaje debería aprovechar el momento, coyuntura o situación del tiempo de referencia.

Frecuencia. El consumidor comienza a retener un mensaje cuando este es

repetitivo. **Sinceridad.** El fraude produce frustración en el consumidor. Lo hace sentir mal y conduce a sentimientos depresivos, que incluso pueden llevar al suicidio. Con respecto a esto existen muchos casos, así que se debe ser cuidadoso, y honesto, lo que produce mejores resultados

Propuesta Única de Venta.

- Todo anuncio debe hacer una proposición concreta al consumidor.
- La proposición debe distinguirse de la competencia (ventaja competitiva, elemento diferenciador o posicionamiento); esta es la condición más importante del USP.
- Debe ser tan atractiva que influya sobre la totalidad del mercado meta del producto.
- Actualmente la proposición de venta es de carácter emocional cambiando su denominación a (ESP)

Posicionamiento o ubicación

- Selecciona un segmento del público para convertirlo en el centro de la campaña.

De esta manera realiza una interpelación directa a ese tipo de consumidor y todo gira en torno a la relación con un consumidor.

- El segundo paso de selección consiste en la selección de facetas de la subjetividad de los consumidores (trabajo cualitativo).
- La selección del público va a ser interpelada pero el resto también puede serlo.

Todos pueden ser movilizados por la campaña ya que esta estrategia combina lo particular con lo general.

- El centro del mensaje es el público. El consumidor es tratado como diferente ya que esta estrategia necesita distinguir.
- Esta estrategia es utilizada para productos genéricos (no específicos) ya que compiten en un mercado saturado.

II.2.3.2. Propuesta del Proyecto referente a la Campaña Publicitaria.

II.2.3.2.1. Planteamiento del Problema

Actualmente el personal de Ordenamiento Territorial no conocen en la gran mayoría lo que son las TIC, su evolución y su capacidad de adaptación a nuevas situaciones, en especial situaciones del negocio empresarial, y lo beneficiosas que éstas pueden ser; por lo que siguen incurriendo en los mismos errores de siempre: mal manejo de información lo que conlleva a tomar decisiones erradas, una atención deficiente al cliente entre otros. Además de seguir permaneciendo bajo el epígrafe del subdesarrollo tecnológicamente hablando.

Son pocos los medios por los que se difunden la importancia, los beneficios de

las tecnologías de información y comunicación y de las eficientes soluciones que éstas le pueden dar a los problemas de información que enfrentan la mayoría de las instituciones.

II.2.3.2.1. Metas y objetivos Generales de la Socialización

Socializar y publicar los aspectos generales del Sistema de Escritorio: G.I.P.S.E.T (**Gestión de Información en el Proceso de Seguimiento de Trámites**), como también la importancia y beneficios que ofrece el mismo desarrollado en el proyecto.

II.2.3.2.2. Objetivos Específicos

- Diseñar afiches que muestren, que den a conocer el Proyecto, como así el Sistema de Escritorio G.I.P.S.E.T.
- Convertir al personal de Ordenamiento Territorial como el centro de la campaña publicitaria y de socialización.

II.2.3.2.3. Metodología

Se pretende adoptar el método de Asociación sico-emotiva a las personas involucradas. Por medio de:

- **Estética**

Mostrar en los afiches imágenes llamativas acerca del Sistema de Escritorio G.I.P.S.E.T, y lo referente al proyecto.

- **Humor**

Llegar al público objetivo mediante frases, imágenes que capturen su atención de forma humorística.

Además para llevar a cabo la socialización se tomará en cuenta los siguientes aspectos:

- **Oportunidad.** El mensaje debería aprovechar el momento, coyuntura o situación del tiempo de referencia.
- **Frecuencia.** El consumidor (microempresario) comienza a retener un mensaje cuando este es repetitivo.
- **Sinceridad.** El fraude produce frustración en el consumidor. Lo hace sentir mal y conduce a sentimientos depresivos, que incluso pueden llevar al suicidio. Con respecto a esto existen muchos casos, así que se debe ser cuidadoso, y honesto, lo que produce mejores resultados.

En síntesis:

La publicidad deberá ser creativa, atractiva, original, que sea capaz de captar el interés del público objetivo hacia este Software de Gestión.

La campaña se realizará a través:

- Propaganda radial.
- Distribución de trípticos e lugares estratégicos de la ciudad de Tarija.
- Afiches tabloides, los cuales colocados en los lugares más concurridos de la ciudad.

Para la producción de esta campaña en primer lugar realizaremos la cotización de los costos que implican la misma, tanto del diseño del spot publicitario, la difusión del mismo y el costo que produce la realización e impresión de los trípticos y afiches.

II.2.3.2.4. Público Objetivo

El público afectado o a quién va dirigido el desarrollo de la socialización son:

- El personal involucrado de Ordenamiento Territorial y en especial departamento o área de Proyectos y Lotes.
- La población tarijeña en general.

II.2.3.2.5. Resultados Esperados

- Al final de la ejecución del proyecto se tiene terminado los diseños de los afiches.
- Estos afiches deben mostrar la importancia el de Sistema de Escritorio G.I.P.S.E.T y el proyecto, en nuestra sociedad, por lo que tienen que estar adecuadamente diseñados.

II.2.3.2.6. Fechas de realización de la campaña de socialización

La campaña de socialización se realizará a partir del 18 de octubre de 2010 hasta el 25 de octubre de 2010, salvo algún contratiempo no planificado.

II.2.3.2.7. Presupuesto y Financiación para la Campaña Publicitaria

Para el desarrollo del sistema se consideran los siguientes precios:

-	Diseño de Afiches	240 Bs.
TOTAL.....		240 Bs.

II.2.3.2.8. Organizador de la campaña de publicitaria

El diseño de los trípticos y afiches se lo llevará a cabo por el personal del proyecto.

II.2.3.3. Medios de Verificación

II.2.3.3.1. Diseños de afiches terminados y disponibles.

El diseño de los afiches terminados, se encuentran en el CD adjunto a este documento.

Captura de las imágenes del diseño de los afiches:



Figura N° 521: Afiche N° 1

II.2.4. Capacitación al personal y/o representantes de Ordenamiento Territorial

de la Ciudad de Tarija

II.2.4.1. Técnicas y Métodos De Enseñanza

II.2.4.1.1. Principios Didácticos

Los métodos y técnicas de la enseñanza, independiente de las teorías que los originen deben sujetarse a algunos principios comunes, teniendo en cuenta el desarrollo y madurez pedagógica alcanzada hasta el presente. Algunos de ellos son [3]:

- **Principio de proximidad:** Integrar la enseñanza lo más cerca posible en la vida cotidiana del educando.
- **Principio de dirección:** Tornar claros y precisos los objetivos a alcanzar.
- **Principio de marcha propia y continúa:** Procura respetar las diferencias individuales, no exigiendo la misma realización de todos los educandos.
- **Principio de ordenamiento:** Con el establecimiento de un orden se busca facilitar la tarea de aprendizaje.
- **Principio de adecuación:** Es necesario que las tareas y objetivos de la enseñanza sean acordes con la necesidades del educando.
- **Principio de eficiencia:** El ideal: mínimo esfuerzo máxima eficiencia en el aprendizaje.
- **Principio de realidad psicológica:** Previene que no se debe perder de vista la edad evolutiva de los alumnos, así como tampoco sus

diferencias individuales.

- **Principio de dificultad o esfuerzo:** Es preciso tener el cuidado de no colocar al educando ante situaciones de las que tenga Posibilidades de salir bien. Pues el fracaso continuado es peor veneno para la criatura humana.
- **Principio de participación:** El educando es parte activa y dinámica del proceso.
- **Principio de espontaneidad:** Cualquier proceso emprendido debe favorecer las manifestaciones naturales del educando.
- **Principio de transparencia:** El conocimiento aprendido debe replicarse en otras situaciones de la vida diaria.
- **Principio de evaluación:** Con un proceso continuo de evaluación, el docente podrá identificar a tiempo dificultades en el proceso de aprendizaje.
- **Principio reflexión:** Inducir al pensamiento reflexivo en el alumno como parte integral de actuar del ser humano.
- **Principio de responsabilidad:** Encaminar todo el proceso de enseñanza de modo que el educando madure en cuanto a comportamiento responsable.

II.2.4.2. Métodos de Enseñanza

II.2.4.2.1. Métodos de Enseñanza individualizada

Su objetivo máximo es ofrecer oportunidades de desenvolvimiento individual más eficiente, y llevar al educando a un completo desarrollo de sus

posibilidades personales.

Método de Proyectos

Lleva al alumno a la realización efectiva de algo, es activo y lo lleva para que realice, actúe es en suma determinar una tarea y que el alumno la realice. Ofrece pasos para solucionar problemas con la solución para la realización que da experiencia al alumno.

- Solución de problemas por realización.
- Carácter general o global abarca conjunto de disciplinas.
- Restringido abarca una o dos disciplinas.
- Desenvuelve espíritu de iniciativa, responsabilidad, solidaridad y libertad.
- Cadena organizada de actividades para realizar algo.

Plan Dalton

Este plan se basa en la actividad individual y la libertad. El objetivo consiste en desenvolver la actividad individual e incentiva la iniciativa al dejar al alumno escoger los trabajos y el momento de realizarlos.

El alumno hace contratos semanales en lo que está interesado, trabajando inmediatamente a su ritmo y posibilidades.

Este método tiene como particularidades que se dan conferencias del profesor con alumnos para organizar trabajos, también un boletín mural donde el profesor consigna instrucciones y la hoja de tareas como documento esencial del plan que contiene explicaciones sobre trabajos, ejercicios e indicaciones de

fuentes de consulta.

Algunos inconvenientes que se presentan con este método es que acentúa exageradamente la individualidad. Tiene un carácter esencialmente individual. [3]

Técnica Winnetka

Busca conjugar las ventajas del trabajo individual con las del trabajo colectivo, sin perder las diferencias individuales.

Enseñanza por unidades o Plan Morrison

Las fases del plan empiezan por la exploración, donde se sondean los conocimientos, luego una presentación donde se expone el tema, sigue la fase de asimilación, en la cual el alumno amplía su conocimiento y continua con una fase de organización, que es el trabajo de integración para culminar con una recitación, que es la presentación oral de la unidad.

Enseñanza programada

Es el método más reciente para individualizar y permitir que cada alumno trabaje según su propio ritmo y posibilidades.

Enseñanza personalizada

Defiende la tesis que cada educando debe desarrollarse y estudiar a su propio ritmo de aprendizaje.

II.2.4.2.2. Métodos de Enseñanza Socializada

Tiene por objeto la integración social, sin descuidar la individualización.

El estudio en grupo

Se basa en los siguientes principios [3]:

- Crear un ambiente de estudio
- Creación de liderazgo distribuido
- Formulación de objetivos
- Permitir el consenso
- Facilitar la comprensión del proceso
- Asegurar una evaluación permanente

Método de la discusión

Sirve de orientación a la clase para realizar de forma cooperativa el estudio de una unidad o tema. Se designan un coordinador y un secretario y el resto de grupo de clase.

Método de la asamblea

Toma la misma forma de una discusión ampliada pero con la diferencia como si fuera un cuerpo colegiado gubernamental: por ejemplo asamblea de estudiantes por la paz.

Método de panel

Es una reunión de especialistas para la discusión general de un tema determinado, el cual es el área de dominio de los participantes. Hay tres formas básicas, panel simple, simple con alternativa y el panel de interrogadores.

II.2.4.3. Técnicas de Enseñanza

Hay muchas técnicas para hacer llegar nuestro conocimiento y lograr un aprendizaje apropiado [3]:

Técnica expositiva

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

Técnica del dictado

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

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

Técnica biográfica

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

Técnica exegetica

Consiste en la lectura comentada de textos relacionados con el asunto en estudio, requiere la consulta de obras de autores.

Su finalidad consiste en acostumbrar a leer las obras representativas de un autor, de un tema o una disciplina.

Técnica cronológica

Esta técnica consiste en presentar o desenvolver los hechos en el orden y la secuencia de su aparición en el tiempo.

Esta técnica puede ser progresiva o regresiva-progresiva cuando los hechos son abordados partiendo desde el pasado hasta llegar al presente.

Regresiva cuando esos mismos hechos parten desde el presente en sentido inverso hacia el pasado.

Técnica del interrogatorio

Uno de los mejores instrumentos del campo didáctico como auxiliar en la acción de educar, este permite conocer al alumno y resaltar sus aspectos positivos. Puede ser empleado para:

- Motivación de la clase.
- Estimulo para la reflexión.
- Recapitulación y síntesis de lo aprendido.

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 del debate

Puede versar sobre:

- Temas que hayan provocado divergencias durante el desarrollo de una clase
- Tópicos del programa
- Dudas surgidas y no aclaradas
- Temas de actualidad social.

Técnica del seminario

El seminario es una técnica más amplia que la discusión o el debate, pudiéndose incluir ambas en su desarrollo.

- El profesor expone lo fundamental del tema.
- Los estudiantes exponen los resultados de sus estudios, donde los llevan al debate.
- Cuando no se queda aclarado el profesor presta ayuda en el tema.
- Al final son coordinadas las conclusiones, con el auxilio del profesor
- Para un seminario eficiente todos los estudiantes deben prepararse para dicho tema.

Técnica del estudio de casos

Consiste en la presentación de un caso o problema para que la clase sugiera o presente soluciones.

- El profesor es orientador
- La presentación de un caso es presentado por el profesor, un alumno, o una autoridad.
- La participación puede llevarse: las opiniones pueden ser dadas individualmente, por los alumnos el tema es subdividido en

subtemas que serán dados a grupos para estudiarlos.

Técnica de problemas

Estudia los problemas de las diversas disciplinas en el orden en que van surgiendo y evolucionando, tiene por objeto desarrollar el razonamiento del alumno, a fin de prepararlo para enfrentar situaciones problemáticas que la vida puede presentarle a cada instante.

Técnica de la demostración

Es el procedimiento más deductivo y puede asociarse a cualquier otra técnica de enseñanza cuando sea necesario comprobar afirmaciones no muy evidentes o ver cómo funciona, en la práctica, lo que fue estudiado teóricamente.

Esta técnica tiene por objetivos:

- confirmar explicaciones orales o escritas
- ilustrar lo que fue expuesto teóricamente
- iniciar teóricamente una técnica para evitar errores
- propiciar un esquema de acción correcto para la ejecución de una tarea.
- convencer racionalmente en cuanto a la veracidad de proposiciones abstractas.

Técnica de la experiencia

La experiencia es un procedimiento eminentemente activo y que procura:

- repetir un fenómeno ya conocido
- explicar un fenómeno que no es conocido
- comprobar, con razones lo que va a suceder, partiendo de experiencias
- conferir confianza para actuar en el terreno de la realidad de

manera lógica

- convencer a cerca de la veracidad de la ley de causa y efecto
- Fortalecer la confianza en sí mismo
- Formar la mentalidad científica
- Orientar para solucionar problemas
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

II.2.4.2. Propuesta del Proyecto referente a la Capacitación

II.2.4.2.1. Descripción del curso

Este curso de 2 días está diseñado para preparar a los/las participantes (al personal de Ordenamiento Territorial) en el conocimiento y utilización de las TIC, esto con el fin de que puedan manejar y administrar el sistema de Escritorio desarrollado en el proyecto de manera individual y sin depender de algún otro experto.

Este curso está dirigido con preferencia y en especial al departamento o área de Proyectos y Lotes y sus respectivos Jefes de Departamento, pero puede adaptarse de acuerdo a las circunstancias (falta de participación, petición de otras áreas interesados en participar) a la inclusión de otros participantes, además queda abierta la posibilidad de extender el curso de acuerdo a la respuesta.

II.2.4.2.2. Metas y objetivos generales del Curso

Influenciar positivamente las actitudes de los/as participantes respecto a los beneficios de utilizar el Sistema de Escritorio desarrollado en el

proyecto y el uso y manejo de las TIC como medio alternativo para incrementar su competitividad. Proporcionarle al/la participante los conocimientos y las habilidades necesarios para manejar y administrar correctamente el software de gestión.

II.2.4.2.3. Objetivos del Aprendizaje del/la participante

Después de asistir a los cursos de capacitación sobre las TIC y manejo del sistema de Escritorio desarrollado en el proyecto, el/la participante estará en capacidad de administrar el sistema de gestión de su empresa.

II.2.4.2.4. Descripción de los métodos de Capacitación

El proyecto plantea llevar a cabo para la capacitación una:

✓ Metodología de Enseñanza Socializada

El cual pretende ejecutar una metodología de enseñanza socializada, dirigida a los gerentes o encargados de las pequeñas y medianas empresas, con lo que se busca una integración social sin descuidar la individualización, aplicando **métodos de asamblea y Panel**.

Para llevarlas a cabo se desarrollará el curso usando las siguientes técnicas:

✓ Técnica expositiva

Consiste en la exposición oral, por parte del capacitador, en la que se estimulará la participación de los microempresarios en los trabajos que se realicen, este requiere una buena motivación para atraer la atención de los participantes.

✓ Técnica de la experiencia

La experiencia es un procedimiento eminentemente activo y que

procura:

- Exponer de forma presencial los pasos a seguir para realizar alguna acción.
- Explicar el funcionamiento de algo mientras es observado.
- Comprobar, con razones lo que va a suceder, partiendo de experiencias.
- Conferir confianza para actuar en el terreno de la realidad de manera lógica.
- Convencer a cerca de la veracidad de la ley de causa y efecto.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

Demostración: Pruebas, test, ensayos del Sistema de Escritorio.

Para la Capacitación se empleará Manuales de usuarios que contienen toda la información detallada sobre el manejo de sistema de Escritorio y llevaremos un registro del personal que asiste a la capacitación.

II.2.4.2.5. Descripción de los materiales de Capacitación

Materiales impresos: Guías del curso, manuales de usuario.

Materiales audiovisuales: Equipo de computación, proyector, micrófono, parlantes

Otros materiales: Marcadores, pizarrón.

II.2.4.2.6. Criterios para la selección de los/las participantes

Los/as participantes de este curso deben ser personal de Ordenamiento Territorial preferentemente del departamento de Proyecto y Lotes, no se exige a los participantes

tener conocimientos acerca de computación. También, de acuerdo a la decisión del Director y Jefes de Departamento, pueden ser participantes del curso, representantes de Ordenamiento Territorial debidamente comprobada su identificación.

II.2.4.2.7. Métodos para la evaluación del curso

Cuestionario pre curso: para evaluar los conocimientos pre curso acerca de computación.

Guías de aprendizaje: los/as participantes evalúan sus propias habilidades

Verificación de lo aprendido: el/la capacitador/a evalúa las habilidades de los/as participantes en el manejo del sistema de gestión y su administración.

Evaluación del curso de capacitación: los/as participantes evalúan el curso.

II.2.4.2.8. Duración del curso

El curso tendrá una duración de dos días, cada clase tendrá una duración aproximada de 2 horas por día.

II.2.4.2.9. Número sugerido de participantes

Preferiblemente se recomienda no sobrepasar la cantidad de 15 participantes en el curso, esto debido a la falta de instalaciones para acoger a más personas, al presupuesto del proyecto y para favorecer al propio ambiente de aprendizaje.

II.2.4.2.10. Fechas del curso

De acuerdo al cronograma del proyecto, el curso se realizará a partir del 18 de octubre de 2010 hasta el 19 de octubre de 2010, salvo algún contratiempo no planificado.

II.2.4.2.11. Organizador del curso de capacitación

El curso será organizado por el director del proyecto, bajo asesoramiento de los docentes de la materia de Taller III y del Director de Ordenamiento Territorial Arq. Álvaro Orozco Herbas.

II.2.4.2.12. Contenido del Curso sobre “Las TIC”

▪ Que son Tic

Las Tecnologías de Información y Comunicación (Tics), se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

- Hardware
- Software
- Servicios de TI

- Telecomunicaciones

- **Globalización**

La globalización es el proceso por el que la creciente comunicación e interdependencia entre los distintos países del mundo unifica mercados, sociedades y culturas, a través de una serie de transformaciones sociales, económicas y políticas que les dan un carácter global. Así, los modos de producción y de movimientos de capital se configuran a escala planetaria, mientras los gobiernos van perdiendo atribuciones ante lo que se ha denominado la sociedad en red.

En este contexto el desarrollo tecnológico, y refiriéndome en sí a la incorporación de las TIC en cada nación, hará que el desarrollo en esta llegue sin tardar demasiado, la competitividad de un país en un sistema globalizado depende en gran medida de la tecnología que adquiera, por eso es tan estrecha la relación entre TIC y globalización. Esperemos que nuestros países comiencen a incorporar las TIC en sus procesos productivos y así reducir la dependencia de potencias mundiales.

- **Las Tic y la Competitividad**

La utilización de las TIC y el Comercio Electrónico por parte del sector empresarial, abre nuevas oportunidades de negocios que redundan en una serie de beneficios económicos y sociales, los cuales van desde el mejoramiento y la facilitación de la comunicación entre las empresas, hasta el manejo más eficiente de los recursos. En tal sentido, las tecnologías deben ser utilizadas por el sector en las diferentes etapas de la cadena de valor.

- Calidad
- Productivida
- Servicio

- Imagen

Los procesos centrados en la producción: En esta etapa, las TICs pueden ser utilizadas para diseñar y probar nuevos productos, e-procurement, procesos de pagos, sistemas de gestión automática de stocks, diferentes tipos de links electrónicos con proveedores, sistemas de control y procesos más relacionados con la producción, entre otros.

Los procesos internos: En lo referente a administración de personal, entrenamiento, reclutamiento interno, compartir y diseminar información de la compañía vía electrónica, entre otros.

Los procesos de compra on-line: Acceso a vendedores y catálogos de productos, compras y pagos electrónicos, utilización de marketplaces electrónicos, administración de inventarios, etc.

Los procesos centrados en el cliente: Marketing, solicitudes y sistemas de pagos, seguimiento y atención al cliente. El comercio electrónico es parte fundamental de esta categoría, esencial para la venta y la post-venta. Este proceso abre nuevas posibilidades de ganancia ampliando el acceso, la promoción y las ventas, mediante la reducción de las barreras geográficas. Surge también la posibilidad de enviar las mercaderías en forma electrónica, en el caso de productos digitales (tales como software, música, libros, asesoría, servicios digitales, entre otros).

▪ Software

Las soluciones a implementar deben tener influencia sobre la

productividad y la rentabilidad en muchos eslabones de la cadena productiva. La cartera de soluciones para una empresa debería estar formada al menos por:

- Datos Históricos de la Empresa para el análisis.
- Soluciones para gestionar la planificación, producción, comercialización, así como para el cumplimiento de las normas y estándares.
- Herramientas para la ingeniería de sistemas (Capacidad de análisis y evolución de los sistemas en el tiempo).
- Sistemas de Recursos Humanos.
- Controladores de Procesos Productivos.
- Optimización de la vida útil de los procesos (Revisión de procesos).

Que debemos hacer cuando un cliente nos hace un pedido: analizar y evaluar.

- La disponibilidad de inventarios de materias primas
- La capacidad para fabricar el producto solicitado y en el tiempo establecido.
- La disponibilidad de recursos de producción y de materiales para su fabricación.
- La adecuada logística para entregar el producto (ya sea propia o de terceros)
- La rentabilidad del producto.

El Software para la empresa es vital debido a:

- Necesidad de optimizar dinámicamente los procesos

- Optimización de la cadena de suministro en la empresa y entre empresas.

Establecer Procesos comerciales en colaboración:

- Soluciones para gestionar la cadena de suministro en colaboración.
- Conectividad comercial ilimitada.
- Modelización de procesos comerciales.
- Soluciones para mercados electrónicos privados o para la colaboración en el lado de las ventas.

▪ Conclusiones

Una característica importante de los países con mayor grado de desarrollo empresarial, es contar con un alto nivel de competitividad y con una tecnología avanzada, el cual a su vez está asociado positivamente con el uso de las Tics

El uso de las Tics por parte de las instituciones esta directamente asociado a la innovación, necesaria para incrementar su eficiencia y competitividad.

El uso de las computadoras está relacionado con la simplificación de labores administrativas y el mejor manejo de los procesos de planificación, y con todo lo relacionado al seguimiento de trámites.

El grado de adopción de las Tics por parte de las instituciones, tiende a reducir la brecha digital en los países en desarrollo.

El impacto de la adopción de Tics en el desempeño de las instituciones públicas, se observa en la eficiencia media del trabajo, (más eficiencia con igual número de empleados).

El uso correcto de las Tics en las instituciones, conlleva mejores controles

administrativos y operativos, por lo tanto logramos mejores niveles de eficiencia, satisfacción del cliente y oportunidades de negocio.

II.2.4.3. Medios de Verificación

II.2.4.3. 1. Registro de los participantes de los cursos de capacitación realizados por los desarrolladores del proyecto.

El registro de asistencia se lo presenta en ANEXOS.

CAPÍTULO III

CONCLUSIONES Y

RECOMENDACIONES

III CONCLUSIONES Y RECOMENDACIONES

III.1 Conclusiones

Durante la elaboración del presente trabajo se pudo observar que la información con la que trabaja **La Ordenamiento Territorial** del departamento de Tarija es de importancia considerable, tomando esto aspectos se concluye que la elaboración de un sistema de gestión para el proceso de seguimiento de tramites es muy útil, ya que traería beneficios a los propietarios que realizan trámites en dicha institución, desde ahorrar tiempo, esfuerzo, trabajo en el registro de los datos de los propietarios, el estado de los tramites, observaciones , en el registro de a dónde salió el trámite, hasta en la protección de la información de estos procesos.

Se debe exponer que el éxito del proyecto solo se realizará con la implementación completa de las estrategias de socialización, para dar a conocer los beneficios de este Sistema a la población tarijeña y mediante la capacitación lograr formar un personal capacitado a enfrentarse con las innovaciones de la tecnología.

Durante el tiempo de trabajo realizado este proyecto permite llegar a la conclusión que la metodología RUP nos permite lograr mejoras al software, gracias a su proceso de iteración para alcanzar mayores resultados. La utilización del UML nos permite lograr un sistema confiable, seguro. La implementación de la Base de Datos en Postgres nos garantiza la seguridad de la información.

El sistema contará con información confiable, segura, oportuna en el momento en que los propietarios lo requieran, brindando información controlada y organizada.

III.2 Recomendaciones

Entre las recomendaciones que destacan se podrían mencionar las siguientes:

- ⇒ Para el desarrollo de futuros proyectos con características similares a éste Se recomienda hacer uso de las metodologías utilizadas como (UML, XP, etc.), ya que éstas son estándares para el desarrollo de éste tipo de aplicaciones, ya que permitieron llegar a la conclusión del Proyecto con alta tecnología de punta.
- ⇒ Debido a la portabilidad de este Sistema, se recomienda a **Ordenamiento Territorial** hacer lo posible para que este trabajo pueda ser implementado.
- ⇒ Para optimizar el rendimiento del sistema, se recomienda a los administradores de cada área y a los técnicos de Ordenamiento Territorial implementar el sistema en equipos de última generación
- ⇒ Se recomienda a los administradores de cada área y a los técnicos de Ordenamiento Territorial consultar el manual de usuario y el manual de instalación para explotar al máximo el uso del sistema.
- ⇒ Las estrategias de campaña deben seguir después de terminado el proyecto para lograr un mayor éxito en la ejecución del proyecto.
- ⇒ Los componentes aquí implementados no resultan ser un estándar ya que se pueden adicionar otros que ayuden a obtener mejores resultados en futuros proyectos.
- ⇒ Se recomienda también profundizar más sobre la realización de campañas publicitaria, ya que la propuesta solo abarca una proporción de un estudio sobre campañas publicitarias.
- ⇒ Se recomienda para este proyecto a un largo plazo implementar un módulo SIG que Gestione la Información Geográfica puesto que la información que maneja el Área de Proyectos y Lotes de ORDENAMIENTO TERRITORIAL, está muy inmersa en este campo de los Sistemas de Información Geográfica.