

1. INTRODUCCIÓN

Actualmente los avances de la tecnología han ocasionado un gran impacto en todas las áreas de trabajo de nuestra sociedad, ya que estos avances demandan respuestas rápidas a cualquier estímulo de cambio lo que ocasiona que estas deben ser más eficientes. Contar con una información actualizada o específica, en el momento adecuado y acorde a las necesidades del negocio, se convierte en un recurso muy valioso para las organizaciones, esto obliga a que las empresas necesiten cambiar y rediseñar sus sistemas actuales, para poder competir en una sociedad donde los clientes necesitan y exigen una mejor atención, por lo que optan por contar con un sistema automatizado que brinde seguridad y disponibilidad de la información en la toma de decisiones.

El procesamiento manual de datos está siendo reemplazado por el procesamiento automático, mediante el uso de sistemas de información adecuados a las necesidades y requerimientos de cada empresa¹.

La Cooperativa de Transporte Público "Virgen de Chaguaya", es una institución que presta sus servicios públicos a la Ciudad de Tarija, mediante sus medios de transportes y de igual forma prestando servicios a los socios que participan dentro de tal institución.

Actualmente la Cooperativa no cuenta con un sistema automatizado en el registro de socios, control de la hoja de ruta y control de las cuentas (ahorro y préstamos), el cual hace que su trabajo dentro de la misma se vuelva lento.

Para ello se necesita una buena administración de sus recursos y atraer clientes por medio de un excelente servicio, mediante el análisis y diseño de un sistema de información automatizada, que permita a la institución reducir el retardo en el

¹Colección de tomos de Informática distribuidos por el periódico la razón

desarrollo de sus distintas actividades, mejorando de esta manera la administración de sus recursos, y facilitando al usuario información necesaria para la toma de decisiones para elevar sus estándares de desempeño, proyectando una mejor imagen a la sociedad y capturando una mayor clientela a través de una mejor atención.

El trabajo que se realizará estará enfocado a dar solución a los problemas ya mencionados dentro de la Cooperativa, tomando en cuenta el diseño de reportes adecuados a la necesidad de la cooperativa que se obtendrá mediante una Base de Datos eficiente.

2. ANTECEDENTES

2.1. Cooperativa de Transportes “Virgen de Chaguaya”

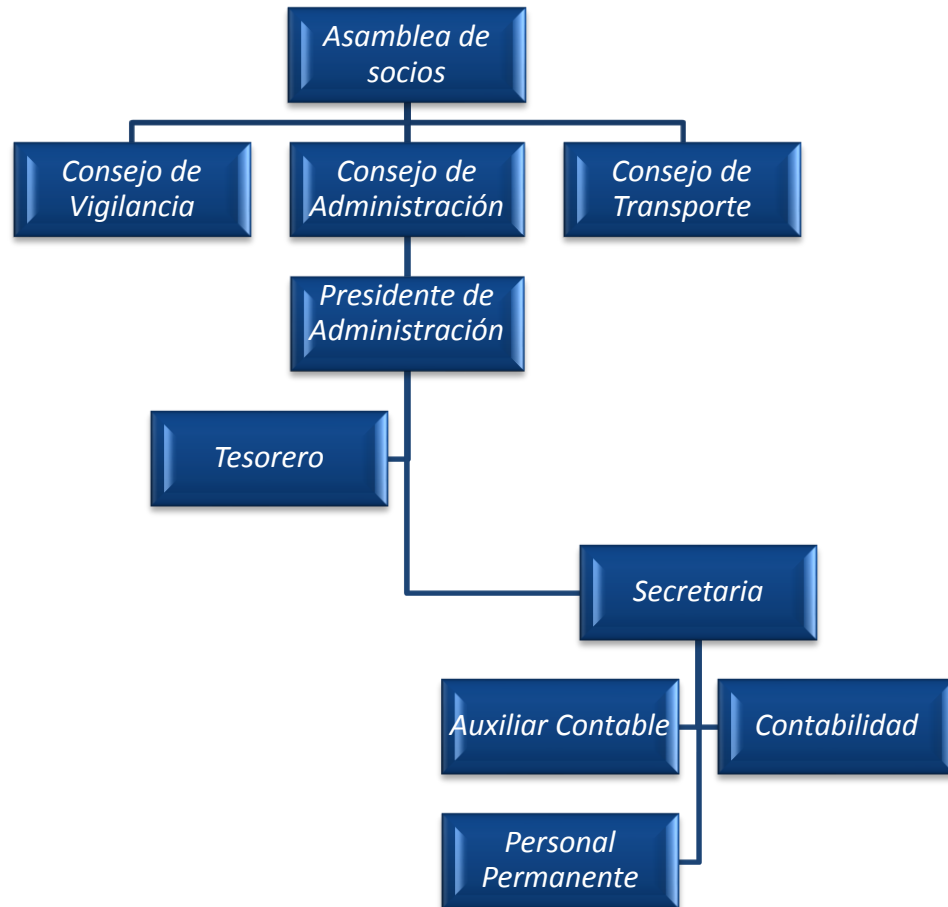
La Cooperativa de Transportes “Virgen de Chaguaya”, fue creada el 2 de mayo de 1986 ubicada en la ciudad de Tarija, en el momento de su creación fue constituida como una asociación económico / social de responsabilidad limitada, de fondo social y número de socios variable, de duración indefinida, desde su creación hasta la actualidad, fueron varios los socios con los que se contó para su administración pero todos ellos con una sola visión, la de prestar servicios de transporte público a través de sus diferentes líneas las cuales hacen distintos recorridos.

Debido a un constante incremento de socios la parte administrativa vio por conveniente descentralizar el manejo de la Cooperativa y fue así que se crearon algunos Departamentos los cuales ayudan en la toma de decisiones.

Dentro del Departamento de Ventas, el manejo de la información ha aumentado en razón al incremento de socios y consumo poblacional, a consecuencia de esto se hace difícil manipular la información, proceso que en la actualidad se lo realiza manualmente, lo cual ocasiona la falta de control y confiabilidad de la misma, igualmente limita la atención de tiempo a los socios.

Para su mejor manejo y entendimiento de sus diferentes Departamentos se ha establecido un organigrama institucional donde se aclara la dependencia de cada uno de ellos con respecto de los otros.

ORGANIGRAMA COOPERATIVA VIRGEN DE CHAGUAYA



La Cooperativa, actualmente cuenta con equipos de computación para su personal:

Pentium IV: manejado por el contador.

Pentium IV: manejada por la secretaria.

Pentium III: manejado por el encargado de ventas.

2.2. Antecedentes de Proyectos y Sistemas para el Área de Transporte

Los software que fueron creados para ser utilizados en diversas organizaciones tienen un carácter de tipo estándar, por lo que estas no satisfacen todas las necesidades de dichas organizaciones, para que una institución preste una mejor atención es conveniente crear un sistema con los requerimientos de la institución es decir un sistema hecho a medida.

En el programa de Ingeniería Informática existe el siguiente sistema de similares características:

➤ *Sindicato de Transporte La Tablada*

Características: La Cooperativa de Transporte La Tablada cuenta con diferentes líneas como ser:

- Línea A
- Línea B
- Línea C
- Línea S
- Línea CH
- Línea G
- Línea D
- Línea K

El sistema creado a medida efectúa operaciones tales como; registro de socios, registro de los aportes diarios, control del manejo de la información.

3. PLANTEAMIENTO DEL PROBLEMA

Efectuando un análisis de las actividades que se realiza dentro de la Cooperativa de Transporte Público “Virgen de Chaguaya”, se consiguió ver que el problema más prominente es la demanda de tiempo en razón a los procesos obsoletos, el control de los socios y micros demora pérdida de tiempo al asignarles un turno de acuerdo a la rotación, no existe un eficiente control de la hoja de ruta y reportes donde se muestren los ingresos de dinero de los relojes tampoco información sobre el estado de préstamos y ahorros de los socios. Aunque esta información se la hace de manera electrónica con el uso de Excel, no tiene un nivel de seguridad adecuado pues cualquier momento se puede perder un archivo ocasionando problemas al encargado, lo ideal sería almacenar esta información en una base de datos a la cual solo pueden ingresar ciertos usuarios privilegiados.

Se requiere contar con un sistema mediante el cual se resuelva el problema presentado anteriormente, que sea de fácil manejo, intuitivo para el usuario y que mantenga niveles de acceso a los trabajadores para que la información de la empresa esté segura ante personas que no deben manipularla. Además de que debe contar con una persona que lo administre y haga el debido mantenimiento.

4. OBJETIVOS

4.1. Objetivo General

Desarrollar el análisis y diseño de un sistema de información automatizado siguiendo la metodología y herramientas adecuadas, que permita gestionar, controlar y optimizar los procesos de registro en las hojas de ruta de la Cooperativa de Transporte Público “Virgen de Chaguaya”.

4.2. Objetivos Específicos

- Hacer una investigación preliminar acerca del manejo de la información en la Cooperativa a través del método ISAAC (Análisis de Cambio y Estudio de Actividades) de la Ingeniería de Requerimientos.
- Desarrollar el sistema en base a la Metodología de Desarrollo de Sistemas RUP.
- Aplicar los diagramas de UML para representar el proceso de análisis y diseño del sistema utilizando como herramientas de trabajo Enterprise para el diseño de diagramas.
- Diseñar una Base de Datos, que permita el almacenamiento seguro de la información aplicando el modelo Entidad-Relación para así poder ver claramente las relaciones que se presentan entre los diferentes Departamentos, en la etapa de diseño se aprovechará las herramientas PHP y MYSQL.
- Elaborar interfaces amigables para que los usuarios puedan interrelacionar con el sistema, estableciendo claves de acceso y asignación de roles en las diferentes áreas de la Cooperativa para un mejor manejo del sistema.
- Mejorar el proceso de seguimiento de información mediante la elaboración de informes y reportes que ayuden en la toma de decisiones.
- Realizar copias de seguridad en el sistema mediante un módulo amigable que posibilite restablecer la base de datos.

5. JUSTIFICACIÓN

Después de observar los resultados de las entrevistas realizadas al Departamento de Administración y otras áreas afectadas de la institución, se pudo percibir que el sistema a ejecutar se justifica dentro de un ámbito social, económico, tecnológico y académico.

5.1. Justificación Social

El sistema propuesto aportará grandes beneficios, tanto para el asociado en general, como para cada uno de los Departamento que se contemplan en la Cooperativa tomando en cuenta que a los socios les beneficiará en el momento de solicitar la información sobre su asignación de línea, horas del recorrido de su ruta y del mismo modo en el momento de cumplir los pagos de su hoja de ruta para poder continuar con su trabajo al día siguiente; en cuanto a la parte administrativa les favorecería sobre todo en el momento de la toma de decisiones para poder efectuar un mejor manejo de los bienes con los que cuenta la Cooperativa.

5.2. Justificación Económica

Se obtendría un beneficio en el ahorro de dinero por la disminución de horas de trabajo del vendedor de la hoja de ruta, al no tener que transcribir la información que realizó manualmente, porque la información ya se encontraría almacenada en la base de datos propuesta para la Cooperativa, del mismo modo contribuirá a controlar el ingreso diario de dinero a la Cooperativa.

5.3. Justificación Técnica

Hoy en día la necesidad de contar con información automatizada es un recurso imprescindible para toda institución con el objeto de ofrecer la posibilidad de utilizar una información precisa y oportuna para poder controlar los datos registrados y así optimizar un buen funcionamiento y un alto grado en la toma de decisiones.

El desarrollo de este sistema es justificado tecnológicamente porque se cuenta con todos los medios tecnológicos idispensables para su implementación dentro de esta Institución.

Además que los operadores no necesitan tener conocimientos avanzados de informática eso hace que el manejo sea sencillo y gestionable.

6. ALCANCES Y LIMITACIONES.

El sistema estará dirigido al personal encargado de cada uno de los Departamentos los cuáles pueden tener acceso a los datos del sistema y parámetros de manejo del mismo.

6.1. Alcances

- Ingresar su nombre de usuario (login) y contraseña (password) para poder entrar al sistema previa verificación de dichos datos.
- Cerrar sesión para mantener la seguridad en el sistema.
- Se realizará la rotación automática de líneas.
- Se elaborará el registro de los socios introduciendo sus datos personales, del mismo modo, se podrá modificar los datos si es que así se lo requiere.
- Se registrará el micro² de cada socio, siendo factible a cambios o renovación de los mismos.
- Se efectuará el registro de los conductores y de esta manera asignarles un vehículo para relacionarlo con un socio.
- Se generará el registro de licencias, sanciones aquellos socios que así lo soliciten o caso contrario sanciones que hayan cometido de acuerdo a las políticas de la Cooperativa.
- El sistema podrá facilitar la tarea de emitir las hojas de ruta (recibos) a los diferentes socios, del mismo modo se efectuará el registro de ingreso de dinero por concepto de relojes.
- El sistema deberá ser capaz de prestar toda información (detalle de los ingresos de cada uno de los socios) que se requiera saber al momento de solicitarla.
- El sistema podrá ser capaz de listar toda la información de los socios Activos³, Pasivos⁴ y Retirados⁵ de la Cooperativa.

² Micro: Vehículo que sirve para el transporte de pasajeros

6.2. Limitaciones

- Estará dedicada estrictamente a información de socios de la Cooperativa y no así a personas particulares.
- No se realizará la puesta en marcha del sistema en el entorno previsto de uso, pero se llevará a cabo las pruebas para corroborar que el software funciona correctamente.
- No se plasmará la contabilidad del sistema ya que se lo tomará como un módulo independiente, pudiendo ser propuesto como otro trabajo de grado por la complejidad del mismo.
- El sistema sólo podrá ser manejado por personal autorizado y el cual tenga asignado un rol respectivo en la institución.

³ Activos: Socios que trabajan y realizan aportes a la cooperativa de manera continua.

⁴ Pasivos: Socios que al no tener su vehículo de trabajo se encuentran imposibilitados de trabajar y realizar aportes continuos a la Cooperativa

⁵ Retirados: Socios que realizan el retiro total de sus aportes y que no pertenecen más a la Cooperativa.

2. MARCO TEÓRICO

En nuestro contexto social, donde la tecnología de la información desempeña un papel muy importante, la combinación de componentes como las computadoras, bases de datos y sus servicios, el modelo cliente / servidor y metodologías orientadas a objetos hacen posible solucionar principalmente los problemas de nuestro entorno, es por tanto que estas innovaciones proveen diferentes métodos para dar solución a problemas de manera más rápida y segura evitando posibles errores en diferentes campos.

Debido al crecimiento de la población Tarijeña y de la misma manera el crecimiento de asociados dentro de la Cooperativa de Transporte Público “Virgen de Chaguaya” se puede concluir que el manejo de la información de cada uno de los socios se ha hecho tedioso y muy difícil de poder manipular.

Para llevar a cabo el adelanto del sistema automatizado para el control de registros de los socios, el Control de la hoja de ruta¹ y el Control de las cuentas de ahorro y préstamos, se considerará aplicaciones de metodologías y herramientas informáticas adecuados para realizar el Análisis y Diseño del Sistema, Modelado de la Base de Datos, Ingeniería de Software y los Lenguajes de Programación.

Por tal razón el sistema a ejecutar utilizará el desarrollo del método ISAAC (Análisis de Cambio y Estudio de Actividades) y técnicas que nos ayudaran para el proceso del sistema. Las metodologías a emplear para el desarrollo del sistema se plantean a continuación:

¹ Hoja de ruta: Recibo emitido por la Cooperativa sin el cual no podrá estar habilitado para trabajar.

2.1. Metodología ISAAC

La abreviación del método ISAAC proviene de “Information System Work and Analysis of Changes”, herramienta para el desarrollo del “Análisis de cambio y planeación de la estrategia de la información” de un sistema.

ISAAC es un método orientado al cliente en ambientes de sistemas de información, desarrollado en orden, partiendo de un análisis de los problemas y proponiendo soluciones reales, se caracteriza por un alto grado de participación de usuarios y desarrolladores, además asegura que los negocios comprendan las necesidades de sistemas de información.

ISAAC comienza analizando los problemas en una organización y encuentra soluciones reales para problemas reales. Si una solución involucra el desarrollo de un sistema de información, entonces ISAAC continua desarrollando este sistema, lo pone en marcha y estudia detalladamente la empresa, de forma que sea el soporte claro de actividades del sistema. ISAAC exhibe cuatro estados, los cuales de citan a continuación:

- 1) Análisis de Cambio (primera etapa)
- 2) Estudio de Actividades
- 3) Análisis de Información
- 4) Implementación

El análisis de cambio es en general un método de solución de problemas. Sin embargo, si la solución involucra la construcción de uno o más sistemas de información, entonces el estudio de actividades delimita la propiedad de requerimiento del sistema.

Asimismo presenta un modelo de actividades especificados en niveles, luego se determina las propiedades y requerimientos del sistema de información para luego precisar el control del sistema.

Para la elaboración del sistema solo se tomará en cuenta la fase 1, con el cual determinaremos los requerimientos del problema partiendo de un análisis y proponiendo una solución.

2.2. Ingeniería de Software

Ingeniería de Software: La Ingeniería de software es una disciplina que surge de la ingeniería de sistemas y de Hardware, abarca un conjunto de tres elementos claves: *métodos, herramientas y procedimientos* que facilitan al gestor controlar el proceso de desarrollo de software y suministrar a los que practiquen dicha ingeniería, es la base para construir software de alta calidad de una forma productiva.

Metodología: Todo proceso de desarrollo de software necesita tener bases metodológicas para garantizar la calidad del mismo. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Sin embargo, muchas veces no se toma en cuenta el aprovechar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo.

Cuando los proyectos que se van a desplegar son de mayor envergadura, ahí si toma sentido el basarnos en una metodología de desarrollo, y empezamos a buscar cual sería la más apropiada para nuestro caso. Lo cierto es que muchas veces no encontramos la más adecuada y terminamos por hacer o diseñar nuestra propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el propósito planteado.

Para dar una idea de qué metodología podemos utilizar y cual se adapta más a nuestro medio, mencionaré tres de ellas de las que considero las más importantes, tal como: RUP, XP y MSF.

La Metodología RUP es más adaptable para proyectos de largo plazo.

La Metodología XP en cambio, se aconseja para proyectos de corto plazo.

La Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.²

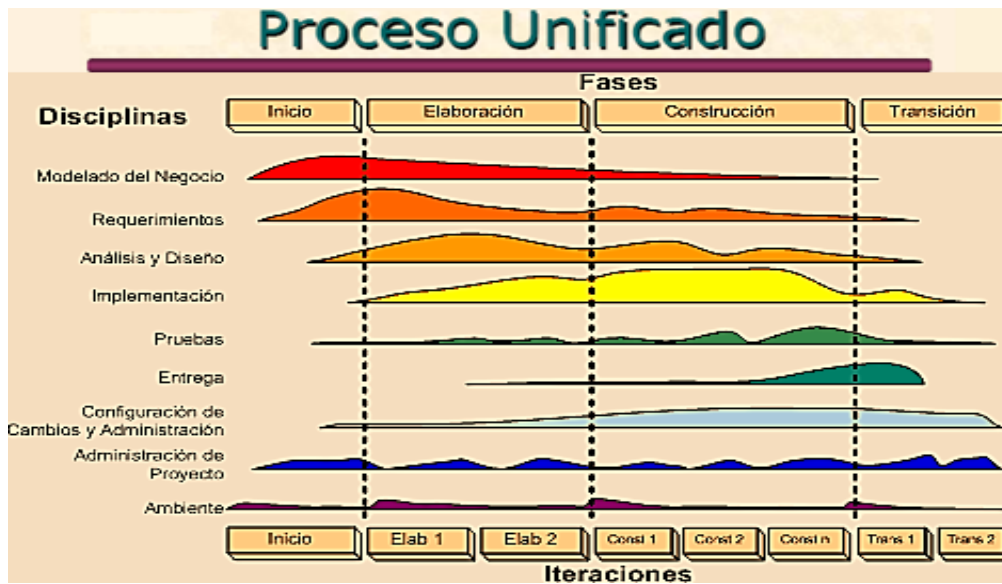
2.3 Metodología RUP

³La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process (Proceso Unificado Racional), es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada por el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP se caracteriza por ser interactivo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son 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).

² Roger S. Presuman, **Ingeniería del Software**, 3° Edición 1993

³ Luis A. Guerrero. Universidad de Chile. Departamento de Ciencias de la Computación. Taller de UML



FASES Y CICLOS

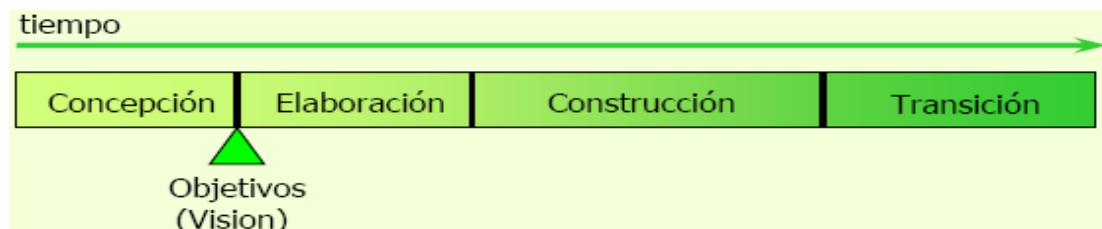
El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante.

FASE 1 Inicio

En el que se establece la oportunidad y alcance del proyecto (mediante la metodología ISAAC) Ver anexo No.5 y 6. Se reconocen todas las actividades externas con las que se trata (actores) y se define la interacción a un alto nivel de abstracción (identificando todos los casos de uso).

Producto Un documento de visión general: Requerimientos generales del proyecto, características principales, restricciones, modelo inicial de casos de uso, glosario.

Hito Las partes interesadas deben acordar el alcance y estimación de tiempo y costo, comprensión de los requerimientos plasmados en casos de uso.

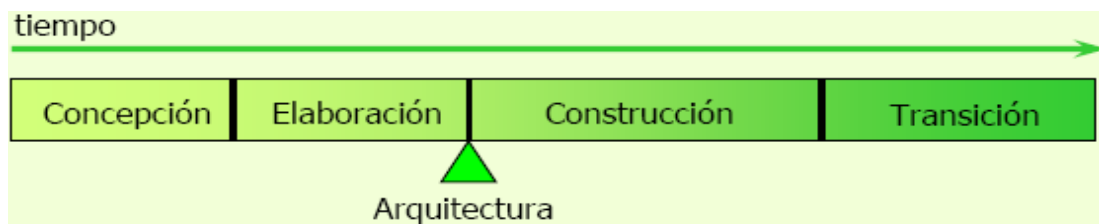


FASE 2 Elaboración

Se analiza el dominio del problema, se establece una arquitectura base sólida, se desarrolla un plan de proyecto y se eliminan los elementos de mayor riesgo para el desarrollo exitoso del proyecto.

Producto La elaboración es la parte más crítica del proceso, al final toda la ingeniería “dura” está hecha, se puede decidir si vale la pena seguir adelante, a partir de aquí la arquitectura, los requerimientos y los planes de desarrollo son estables. Ya son menos riesgosos y se puede planificar el resto del proyecto con menor incertidumbre. Se construye una arquitectura ejecutable que contemple los casos de uso críticos y los riesgos identificados, modelos de casos de uso con descripciones detalladas, un prototipo ejecutable de la arquitectura, plan de desarrollo para el resto del proyecto, un manual de usuario preliminar, descripción de la arquitectura del software (Diagrama de clases, Secuencias y Actividades)

Hito Condiciones de éxito de la elaboración: se establece la visión del producto, arquitectura, las pruebas de ejecución demuestran que los riesgos han sido abordados y resueltos, es el plan del proyecto algo realista, están de acuerdo con el plan todas las personas involucradas.

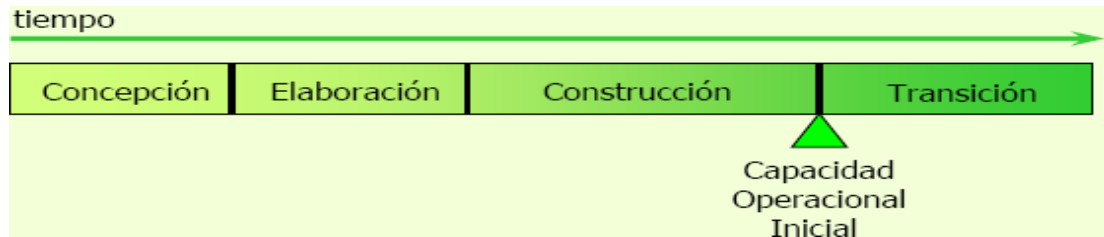


FASE 3 Construcción

En esta fase todas las componentes restantes se despliegan e incorporan al producto, todo es probado en profundidad, el énfasis está en la producción eficiente y no ya en la creación intelectual, puede hacerse construcción en paralelo, pero esto exige una planificación detallada y una arquitectura muy estable.

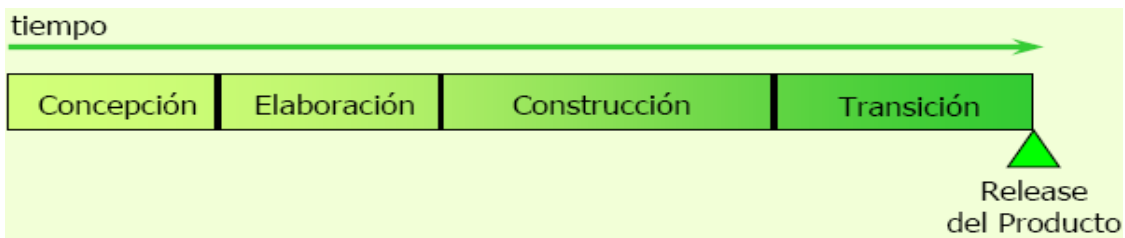
Producto El producto de software integrado y corriendo en la plataforma adecuada y con manuales de usuario.

Hito Se obtiene un producto Beta que debe decidirse si puede ponerse en ejecución sin mayores riesgos. Condiciones de éxito: el producto está maduro y estable para instalarlo en el ambiente del cliente, están los interesados listos para recibirlo.



FASE 4 Transición

El objetivo es traspasar el software desarrollado, una vez instalado surgirán nuevos elementos que impliquen nuevos ciclos de desarrollo, incluye pruebas Beta para validar el producto con las expectativas del cliente, ejecución paralela con sistemas antiguos, conversión de datos, entrenamiento de usuarios, distribución del producto.



Actividades

Y el flujo de trabajo (workflow) entre ellas en base a los llamados diagramas de actividades. El proceso define una serie de roles que se distribuyen entre los miembros del proyecto y que explican las tareas de cada uno y el resultado (artefactos) que se espera de ellos.

RUP define nueve actividades a realizar en cada fase del proyecto:

1. Modelado del negocio.
2. Análisis de requisitos.
3. Análisis y diseño.
4. Implementación.

5. Test.
6. Distribución.
7. Gestión y configuración de cambios.
8. Gestión del proyecto
9. Gestión del entorno.

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.

2.4 Modelo Orientado a Objetos

La técnica que será utilizada es UML (Lenguaje de Modelado Unificado)

2.4.1 Lenguaje de Modelado Unificado (UML)

El Lenguaje de Modelamiento Unificado (UML – Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar los elementos creados mediante un lenguaje común describiendo modelos.

UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de la base datos y componentes de software reutilizables.

UML puede ser utilizado por cualquier metodología de análisis y diseño orientada por objetos para expresar los diseños.

2.4.1.1 Diagrama de Casos de Usos

Un Diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuario u otras aplicaciones).

Es una herramienta esencial para la captura de requerimientos y para la planificación y control de un proyecto interactivo.

Los Casos de Uso se representa en el diagrama por una elipse que denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema.

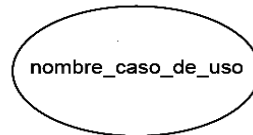
ELEMENTOS DE LOS CASOS DE USO

- **Actor:** Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Un solo actor puede actuar en muchos casos de uso; recíprocamente, un caso de uso puede tener varios actores. Los actores no necesitan ser humanos pueden ser sistemas externos que necesitan alguna información del sistema actual *ver Anexo No.1.*



- **Caso de Uso:** Es una operación / tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso *ver Anexo No.1.*



- **Relaciones**

- **Asociación** Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple *ver Anexo No.1.*
- **Dependencia o Instanciación** Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada *ver Anexo No.1.*
- **Generalización** Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>). Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores) *ver Anexo No.1.*

Extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

Uses: Se aconseja aprovechar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la misma.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelamiento de clases, en donde está la duda clásica de **usar** o **heredar**.

2.4.1.2 Diagrama de Clases

Un diagrama de clases o estructura estática muestra el conjunto de clases y objeto importante que forman parte de un sistema, junto con las relaciones existentes entre clases y objetos. Muestra de una manera estática la estructura de información del sistema y la visibilidad que tiene cada una de las clases, dada por sus relaciones con los demás en el modelo.

Elementos del Diagrama de Clases

En UML la clase se encuentra representada por un rectángulo con tres divisiones internas, son los elementos fundamentales del diagrama. Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

Un diagrama de clases está compuesto por los siguientes elementos:

- Clase: Atributos, métodos y visibilidad.
- Relaciones: Herencia, composición, agregación, asociación y uso.

Elementos

- **Clase** Es la unidad básica que encapsula toda la información de un objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.). En UML, una clase es representada por un rectángulo que posee tres divisiones: Ver Anexo No.2.

- **Atributos y Métodos:**

- **Atributos:** Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno. Ver Anexo No.2.
- **Métodos:** Los métodos u operaciones de una clase son la forma en cómo ésta interactúa con su entorno, éstos pueden tener características. Ver Anexo No.2.

- **Relaciones entre Clases:** En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación. Ver Anexo No.3.

2.4.1.3 Diagrama de Interacción

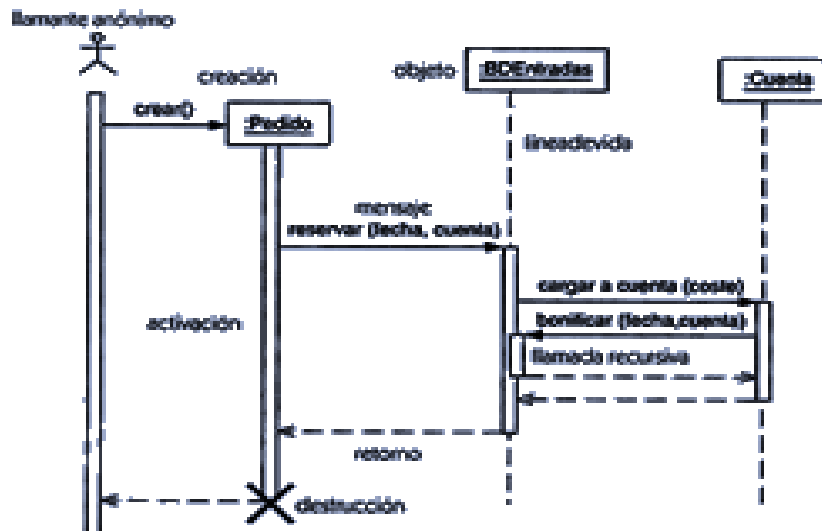
Los diagramas de interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento. Habitualmente, un diagrama de interacción capta el comportamiento de un solo caso de uso. Hay dos tipos de diagrama de interacción:

2.4.1.4 Diagrama de secuencia

Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. En los diagramas de este tipo intervienen objetos, que tienen un significado parecido al de los objetos representados en los diagramas de colaboración, es decir son instancias concretas de una clase que participa en la interacción.

En un diagrama de secuencia un objeto se modela como caja en la parte superior de la línea superior punteada, esta línea se llama línea de vida del objeto y representa la vida del objeto durante la interacción. En este tipo de diagramas asimismo intervienen los mensajes, que son la forma en que se

comunican los objetos: El objeto origen solicita (llama a) una operación del objeto destino, permiten indicar cuál es el momento en el que se envía o se completa un mensaje mediante el tiempo de transición, que se especifica en el diagrama.



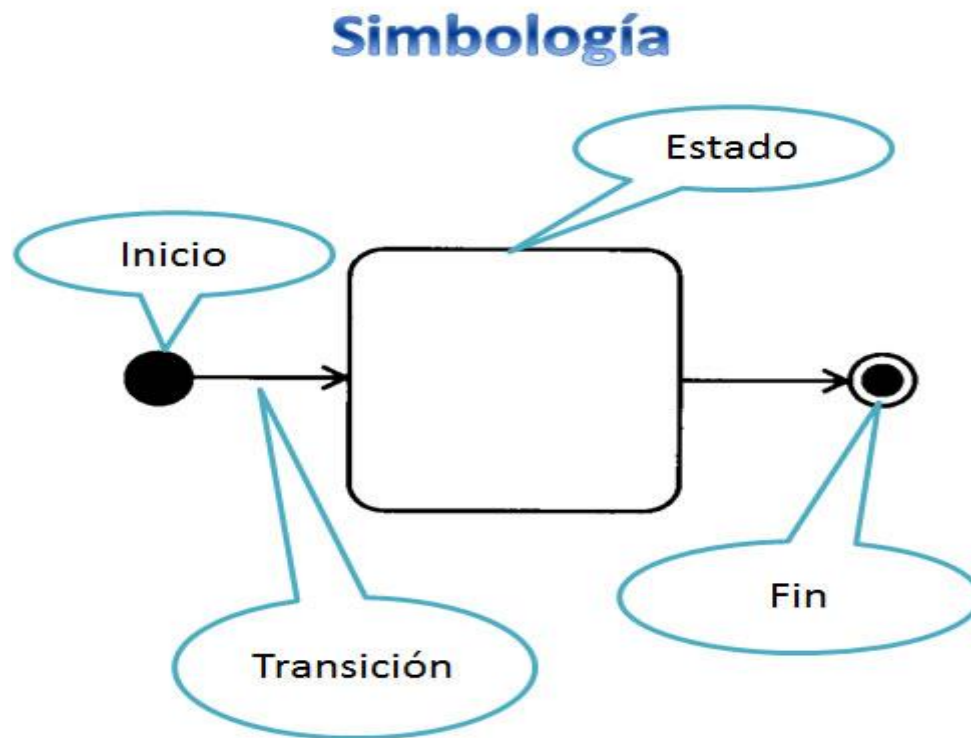
2.4.1.5 Diagrama de colaboración

Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. A diferencia de un diagrama de secuencias, un diagrama de colaboraciones prueba las relaciones entre los objetos, no la secuencia en el tiempo en que se producen los mensajes. Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en una forma diferente.

Formando parte de los diagramas de colaboración nos encontramos con objetos, enlaces y mensajes. Un objeto es una instancia de una clase que participa como una interacción, existen objetos simples y complejos. Un objeto es activo si posee un thread o hilo de control y es capaz de iniciar la actividad de control, mientras que un objeto es pasivo si mantiene datos pero no inicia la actividad.

Un enlace es una instancia de una asociación que conecta dos objetos de un diagrama de colaboración. El enlace puede ser reflexivo si conecta a un elemento consigo mismo. La existencia de un enlace entre dos objetos indica que puede existir un intercambio de mensajes entre los objetos conectados.

Los diagramas de interacción indican el flujo de mensajes entre elementos del modelo, el flujo de mensajes representa el envío de un mensaje desde un objeto a otro si entre ellos existe un enlace. Los mensajes que se envían entre objetos pueden ser de distintos tipos, también según como se producen en el tiempo; existen mensajes simples, sincrónicos, balking, timeout y asíncronos. Durante la ejecución de un diagrama de colaboración se crean y destruyen objetos y enlaces⁴.




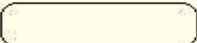


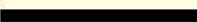


⁴ <http://www.desic.upv.es/uml> , 2008

2.4.1.6 Diagrama de Actividades

⁵En UML un diagrama de actividades se usa para mostrar la secuencia de actividades, muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad.

Los diagramas de actividades son útiles para el modelado de negocios donde se usan para detallar el proceso involucrado en las actividades del negocio, son útiles para describir procesos complicados, pueden ser usados para describir un Caso de Uso.

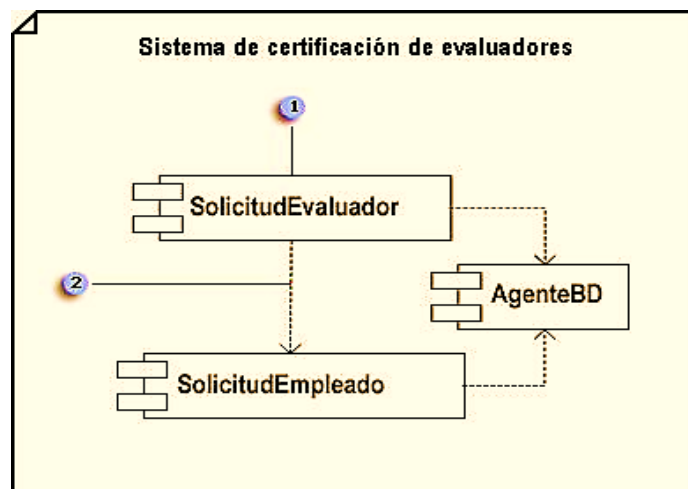
Un diagrama de actividad permite seleccionar el orden en que se harán las cosas y pueden manejar procesos paralelos.

SÍMBOLO	SIGNIFICADO
	Punto de inicio del proceso
	Actividad
	Condicionales
	Flujo de secuencia
	Bifurcación o entrada
	Punto final del proceso
	Swimlanes ("Calles")

⁵ http://www.sparxsystems.com.ar/resources/tutorial/uml2_activitydiagram.html

2.4.1.7 Diagrama de componentes

Muestra la dependencia entre los distintos componentes de software, incluyendo componentes de código fuente, binario y ejecutable. Un componente es un fragmento de código software (un fuente, binario o ejecutable) que se utiliza para mostrar dependencias en tiempo de compilación.

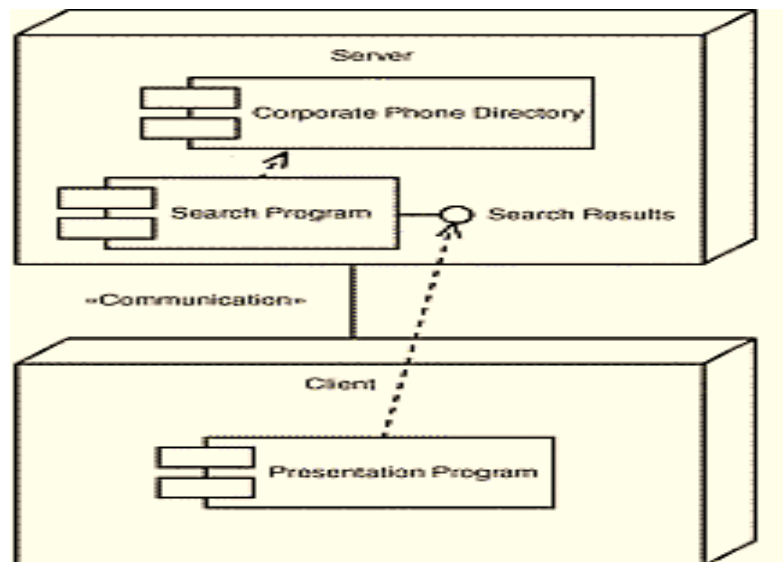


2.4.1.8 Diagrama de plataformas o despliegue:

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución.

En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes⁶.

⁶ <http://www.desic.upv.es/uml>, 2008



2.5 Base de Datos

Las Bases de Datos son conjunto de archivos o tablas formadas por registros; cada línea horizontal de las tablas constituye un registro: conjunto de datos referidos a un mismo elemento las columnas en cambio forman los campos. Estos pueden contener datos numéricos, alfabéticos, alfanuméricos, lógicos (verdadero o falso), fechas, etc.

Una base de datos es una forma de organizar un conjunto de datos que están lógicamente relacionados entre sí es una estructura que facilita su relacionamiento y recuperación de datos y nos muestra un resultado en relaciones y entidades.⁷

Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulan ese conjunto de datos.

⁷ James Martin, Jades Odell (Análisis y Diseño orientado a Objetos).

2.5.1 Sistemas Gestores De Bases De Datos

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.⁸

En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

2.5.2 Acceso a Base de Datos

Nos permite combinar la potencia de los datos organizados lógicamente y almacenados en forma persistente con la facilidad de uso de un navegador. PHP provee de funciones para abrir, y cerrar conexiones a la base de datos, para enviar sentencias SQL, y recibir el resultado de la consulta.

Entre las muchas bases de datos que se pueden usar tenemos: Postgres SQL, MySQL, Oracle, Informix, Access, Sql Server.

2.5.3 Modelo Relacional

El modelo relacional representa los datos y las relaciones entre los datos mediante una colección de tablas cada una de las cuales tiene un número de columnas con nombres únicos. Es importante por dos razones, primero debido a que las estructuras del modelo relacional son amplias y generales y pueden usarse para expresar diseños independientes de DBMS⁹.

⁸ “Diagramas y Algoritmos”. Sandoval, Carlos El País / 2001

⁹ TSAL, ALICE Y H Sistema de base de datos- administración y uso.

2.5.4 MySQL

MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Las características principales de MySQL son:

- **Es un gestor de base de datos.** Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- **Es una base de datos relacional.** Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- **Es Open Source.** El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- **Es una base de datos muy rápida,** segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.

Ventajas

- MySQL software es Open Source
- Velocidad al realizar las operaciones, lo que lo hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.

- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL usa la licencia GPL.

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

2.5.5 PostgreSQL

Es un servidor de base de datos relacional orientada a objetos de software libre. Como muchos otros proyectos el desarrollo de Postgresql es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo¹⁰.

Entre sus principales características tenemos:

- De almacenamiento confiable, consistente, robusto.
- De manipulación potente, flexible y eficiente.
- Considerado el competidor de Oracle.
- Utiliza Control de Concurrencia Multi-Versión (MVCC) para manejar los accesos a la base de datos, lo que permite a diferentes usuarios realizar consultas simultáneamente evitando de esta forma el bloqueo.
- Resistencia a fallas Write Ahead Logging (WAL). Lo que garantiza que en el caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual puedes restaurarla. Es decir soporta caídas del sistema sin perder ningún dato.

¹⁰ <http://es.wikipedia.org/wiki/Postgresql>

- Usa una arquitectura proceso por usuario Cliente / Servidor. En donde existe un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a Postgresql.

Características Técnicas

Postgresql tiene todo lo que se exige de un Gestor de Bases de Datos Relacional:

- Procedimientos almacenados
- Vistas
- Integridad Referencial
- Reglas (rules)
- Restricciones (Constraints)
- Orientación a objetos con herencia de tablas

Postgresql incorpora los cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema.

- Clases
- Herencia
- Tipos
- Funciones

Estas características colocan a Postgresql en la categoría de las Bases de Datos identificadas como objetos-relacionales. Postgresql tiene algunas características que son propias del mundo de las Bases de Datos orientadas a objetos.

Ventajas

- Mayor seguridad
- Bajo costo
- Instalación ilimitada
- Multiplataforma
- Diseñado para ambientes de alto volumen.

Desventajas

- Consume bastante más recursos y carga más el sistema.
- Es dos o tres veces más lenta que Mysql.
- Menos funciones en PHP.

2.6 PHP (Hypertext Pre-Processor).

PHP es un Pre-procesador de hipertexto y como tal, se ejecuta en un servidor Web remoto para procesar páginas Web antes que sean cargadas en el navegador. Además de sus potentes características. PHP es un lenguaje que ha sido diseñado específicamente para el desarrollo y la producción de páginas Web. Sin embargo, el conocimiento de ambos lenguajes no es un prerequisite necesario para empezar con él, PHP es un software de código abierto^{11 12}.

Es uno de los lenguajes del lado servidor más extendido en la Web. Permite embeber pequeños fragmentos de código dentro de las paginas HTML y realizar determinadas acciones de forma fácil y eficaz sin tener que generar programas en un lenguaje distinto al HTML. Ofrece varias funciones para la explotación de bases de datos de una manera sencilla.

Puede ser usado para crear páginas complejas, su lenguaje de programación es de estilo clásico, es un lenguaje de programación con variables, sentencias condicionales, bucles y funciones.

¹¹ Fundamentos de programación algoritmos estructuras de datos y objetos. Joyanes A., Luis. McGrawHill / 2003.

¹² “Programación Orientada a Objetos”. Joyanes, Luis. McGraw-Hill 2000.

Características fundamentales de PHP

PHP es conocido por sus características avanzadas, las fundamentales se deben considerar primero.

- Es un lenguaje de script de código abierto para servidores.
- Es independiente del sistema operativo y puede ser manejado en cualquiera de ellos, incluyendo Microsoft Windows, Linux por nombrar algunos.
- Ocupa una amplia gama de servidores Web, útiles como Apache.
- Se conecta a gran cantidad de bases de datos, como MySQL, Oracle, Informix, etc. Una de las características que distingue a PHP es el que proporciona a sitios Web, de comercio electrónico que manejan bases de datos.

2.7 Herramientas para Graficar

2.7.1 Herramientas CASE

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación de UML.

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, usando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de Software.

Desarrollo Iterativo:

Rational Rose emplea un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha transformado, se efectúa la siguiente iteración.

Trabajo en Grupo

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para efectuar el control de proyectos que permite mantener la integridad de dichas unidades.

Generador de Código

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Ingeniería Inversa

Rational Rose proporciona mecanismos para ejecutar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede conseguir información sobre su diseño.

2.7.2 Enterprise

Es una herramienta comprensible de diseño y análisis UML, que cubre el desarrollo de software desde la captura de requerimientos a través de las etapas de análisis, modelos de diseño, pruebas y mantenimiento.

Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener¹³.

2.8 JavaScript

¹³ <http://www.sparxsystems.com.ar/new/products/ea.php>

JavaScript es básicamente un lenguaje de creación de secuencias de comandos en el lado del cliente que se aprovecha en navegadores Web. En la actualidad se usa para que los programadores puedan interactuar con páginas Web y con la propia ventana del navegador.

JavaScript se basa en Java, un lenguaje de programación orientado a objetos que se utiliza en la Web a través de subprogramas incrustados. JavaScript para desarrolladores Web ofrece una introducción de nivel de programador junto con las características más útiles y avanzadas de JavaScript.

2.9 Html

Es un lenguaje muy sencillo que permite describir hipertexto, es decir texto presentado en forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas.

La descripción se basa en especificar en el texto la estructura lógica del contenido así como los diferentes efectos que se quieren dar y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado como Internet Explorer, Mozilla.

2.10 CSS

Las hojas de estilo las usamos para formatear las páginas Web en todo aquello que deseamos. CSS es complicado, pero no es nada si comprendemos que podemos definir, en una sola línea de código, un estilo de texto o títulos para todo el sitio. Una vez que modificamos esta línea en segundos, todo el sitio cambiará automáticamente.

2.11 Apache

Apache HTTP Server es uno de los más robustos y rápidos servidores web multiplataforma que existen, ha sido creado desde la idea de *open-source* demostrando una vez más que ir asociado a esta idea no es signo de fracaso y lo avala el ser el servidor web más manejado en todo el mundo (desde pequeña y grandes empresas, pasando por instituciones y universidades). Con Apache HTTP Server podremos ejecutar CGI, Perl, Php3 + Bases de datos, SSL, soporte para host virtuales, soporte IPv6, etc., en fin, casi todo lo que le pidamos a cualquier servidor web. Es un Servidor web de código abierto. [7]

2.12 Estructura del Sistema Propuesto

2.12.1 Sistema

El sistema estará compuesto por el Análisis y Diseño del mismo, que se lograra utilizando las aplicaciones del lenguaje propuesto con la plataforma ya citada, y realizando metodologías para sus diferentes actividades. Ver Anexo No.4

2.12.2 Base de Datos

Se empleara una Base de Datos para resolver los problemas que se suscitan en la Cooperativa dando como resultado informes eficientes y correctos respecto a la información que se desea obtener. Ver Anexo No.4

2.12.3 Archivos

Contiene la información necesaria de la Cooperativa, lo cual nos permitirá implantar un buen sistema, aprovechando datos confiables para la base de datos. Ver Anexo No.4

2.12.4 Operador(es)

Es el encargado de manejar el sistema, de una forma adecuada, responsable y favorable para el trabajo que desempeña. Ver Anexo No.4.

2.13 Copias de respaldo y recuperación

El Reglamento establece para los ficheros de datos de nivel alto que deberá conservarse una copia de respaldo y de los procedimientos de recuperación de los datos en un lugar diferente a aquél en que se encuentran los equipos informáticos que los tratan cumpliendo, en todo caso, las medidas de seguridad exigidas.

El almacenamiento y conservación de juegos de copias de seguridad y de los procedimientos de restauración de datos fuera de la ubicación principal (por ejemplo, en cámaras de seguridad de bancos, que nos ofrecen altas medidas de seguridad) nos garantiza la continuidad de la actividad y la disponibilidad de la información ante cualquier incidencia grave o muy grave, sea física o lógica, que afecte a los equipos y servidores centrales (incendios, inundaciones, etc.).¹⁴

2.13.1 Manejo de cuentas o roles de usuario

Como administradores podemos definir el entorno de trabajo de un usuario para restringir o personalizar lo que un usuario ve y tiene disponible cuando inicia una sesión. Por ejemplo, podemos establecer los colores de la pantalla, la configuración del ratón, y que las conexiones de red y de impresoras sean siempre las mismas cuando un usuario inicie una sesión. Para administrar los entornos de trabajo de los usuarios manejan normas que regulan qué usuarios pueden aprovechar un determinado recurso compartido del sistema, esto incluye archivos, carpetas, impresoras, etc.¹⁵

¹⁴ Por: Andrés Veyrat Marqués de Manaca Consulting, S.L. “<http://www.microsoft.com/business/es-es/Content/Paginas/article.aspx?cbcid=326>”

¹⁵ http://fmc.axarnet.es/winnt4svr/administracion/tema_04.htm

3.1 Análisis de Requerimientos

Siguiendo la metodología RUP, la determinación de requerimientos es desplegar un modelo del sistema que se va a construir, y la utilización de los casos de uso es una forma adecuada de crear ese modelo.

3.2 Vista de Casos de Uso

Los casos de uso son un mecanismo ampliamente aprovechado para descubrir y registrar los requisitos, influyen mucho respecto de un proyecto. La escritura de caso de uso es una técnica excelente para entender y describir los requisitos.

3.2.1 Identificación de Actores

Para iniciar el análisis se comienza con la identificación de los actores que interactúan con el sistema, un actor es alguien o algo que tiene que interactuar con el sistema que se está desarrollando.




- **Administrador del sistema.**_ El actor encargado de administrar el sistema es el que se ocupa de las tareas administrativas del sistema es decir el registro de usuarios, modificación de datos del usuario, asignar roles, modificación de roles, asignar permisos, modificar permisos, de igual forma la modificación de login y password.
- **Actor socio.**_ El actor socio no interactúa de manera directa con el sistema, sino que es la persona que se encuentra afiliada a la cooperativa, asimismo puede efectuar una solicitud de préstamo, comprar hoja de ruta, solicitud e impresión de kardex; para cumplir todas las actividades mencionadas anteriormente debe presentarse ante los actores abajo descritos.



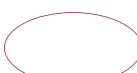





- **Actor Encargado de ventas.**_ El actor encargado de ventas es el usuario que interactúa directamente con el sistema, y tiene ciertas atribuciones en las páginas de registro de cuota de préstamos, registro aporte de ahorro, registro aporte de administración, registro cuota de ingreso, además realizar el registro de ingreso por concepto de alquileres de los relojes, asimismo puede realizar la modificación de los mismos.
- **Actor Secretaria.**_ El actor secretaria es el usuario que interactúa directamente con el sistema, y tiene ciertas atribuciones en las páginas de registro de nuevos socios, además la modificación de los datos de socios, registro de vehículo o herramienta de trabajo, igualmente la modificación de datos del vehículo, registrar conductor, al igual que la modificación de los datos del conductor, registro de sanciones a conductores, de igual forma la modificación de datos de sanción, registro de licencias o permisos a los socios, además la modificación de datos de la licencia, generar reporte de sanciones a conductores, generar reporte de licencias o permisos a los socios, registro de cuota de ingreso al contado, registro por concepto de ahorro adicional, y registro por concepto de solicitudes.
- **Actor Auxiliar de transporte.** El actor auxiliar de transporte es el usuario que interactúa directamente con el sistema y tiene ciertas atribuciones en las páginas de registro de ascensos de los socios, como así la modificación de los ascensos, registro de asignación de línea, registro de nueva línea, registro de ampliaciones, y la modificación de los datos de los mismos, generar reporte de ascensos, reporte de rotación de grupo, reporte de rotación de socios, actualización de rol tomando en cuenta las licencias y sanciones, generar rotaciones de línea y en línea.
- **Actor Presidente.**_ El actor presidente no interactúa directamente con el sistema, los actores arriba mencionados hacen llegar los diferentes reportes









concernientes a la información financiera de la empresa, generados por el sistema, y otros reportes como ser de préstamos, reporte de socios activos, pasivos. Toda la documentación enviada a presidencia es utilizada en la toma de decisiones, para lograr una mejor administración de la cooperativa.




- **Actor conductor.**_ El actor conductor no interactúa directamente con el sistema, es la persona encargada de la conducción del microbús de un determinado socio, siendo su función el de transportar a los pasajeros, sus datos se encuentran registrados en el sistema.
- **Actor consejero de vigilancia.**_ El actor consejero de vigilancia no interactúa directamente con el sistema, es la persona encargada de llevar un control del rendimiento óptimo de la cooperativa (personal de trabajo, cumplimiento de las obligaciones de los socios).

3.2.2 Descripción de Casos de Uso

Caso de Uso	Descripción
 Conectarse	Con el nombre y contraseña del usuario, se permite el ingreso del mismo al sistema.
 Gestionar usuarios	Posibilita registrar, modificar y asignar los datos de los usuarios.
 Crear Copia de Seguridad	Admite crear una copia de seguridad de la base de datos del sistema en cualquier momento.

 Restaurar Copia de Seguridad	Permite restaurar una copia de seguridad de la base de datos del sistema en cualquier momento.
 Gestionar Empresa	Accede agregar los datos de la empresa en el sistema.
 Gestionar Ingreso Hoja Ruta	Aprueba ingresar una hoja de ruta donde se tiene que adjuntar un aporte de ahorro, cuota de préstamo, aporte de administración y cuota de ingreso.
 Gestionar Modificar Hoja ruta	Admite ingresar una hoja de ruta donde se tiene que modificar un aporte de ahorro, cuota de préstamo, aporte de administración y cuota de ingreso.
 Gestionar Ingreso Reloj	Permite adicionar y modificar los ingresos por reloj que controla las rutas de las líneas.
 Gestionar socio	Consiente registrar, transformar y excluir los socios que cuenta el sindicato.
 Gestionar Vehiculo	Faculta registrar, cambiar y eliminar los vehículos que cuenta un socio.
 Gestionar Conductor	Accede registrar, modificar, suprimir y asignar un vehículo a un conductor.

 Asignar Licencias y permisos	Permite el registro y luego la asignación de licencias y permisos a un señalado socio.
 Aplicar Sanciones	Aprueba el registro y luego la asignación de sanciones a un estipulado socio.
 Generar reportes Sanciones	Aprueba generar el reporte de sanciones que se tenga para un definido socio.
 Generar Reportes Licencias	Posibilita generar el reporte de licencias y permisos que se tenga para un socio.
 Generar Reporte Ingresos	Genera el reporte de ingresos (ahorro, aporte administración, crédito) que se tenga para un delimitado socio.
 Gestionar Ingresos Adicionales	Autoriza adicionar y modificar ingresos adicionales que se tenga para un determinado socio.
 Gestionar Cargos	Accede registrar, modificar y eliminar datos de los cargos de cada uno de los usuarios.
 Gestionar Ascensos	Admite registrar, modificar y asignar un ascenso para un señalado socio.

 Gestionar Línea	Permite registrar, modificar y asignar una línea para un determinado socio.
 Gestionar Rol	Admite generar rotaciones en línea, rotaciones de línea y asignar esta rotación a un socio.
 Generar Reporte	Accede generar reporte de ascensos y mostrar las rotaciones de líneas para los socios.

NOTA: Cabe recalcar que la rotación de línea y rotación en línea tiene las siguientes diferencias

Nombre del caso de uso: Rotación de línea

Descripción: Caso de uso que es utilizado para la rotación de línea, para la generación de la rotación la cooperativa en la actualidad cuenta con el grupo A (el mismo está conformado por grupo A-1 la línea 1, grupo A-2 línea 7 y grupo A-3 línea 4), grupo B(el mismo está conformado por la línea 6 , línea 11 y línea 9) y la línea 2, línea 8 .La rotación de línea se la efectúa cada inicio de semana y consiste en el intercambio de posición entre las diferentes líneas de un mismo grupo.

Por ejemplo rotación del grupo A

- *Primer semana del mes las posiciones serian línea 1, línea 4 y línea 7.*
- *Segunda semana del mes las posiciones serian línea 4, línea 7 y línea 1.*
- *Tercer semana del mes las posiciones serian línea 7, línea 1 y línea 4.*
- *Cuarta semana del mes vuelve a la posición inicial línea 1, línea 4 y línea 7.*

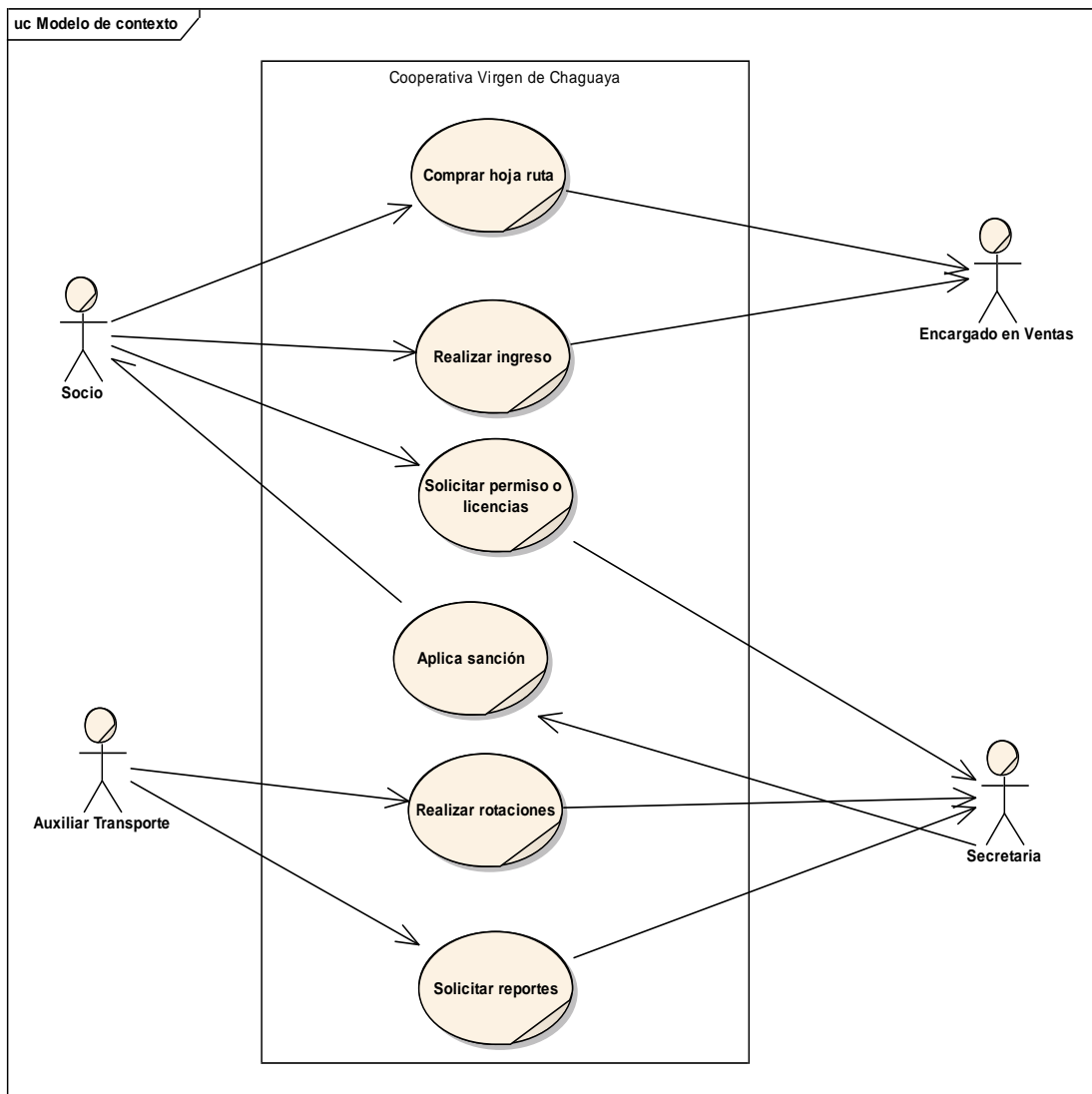
Del mismo modo se realiza la rotación del grupo B. Quedando excluidas la línea 2 y línea 8 (por ser de reciente creación).

Nombre del caso de uso: Rotación en línea

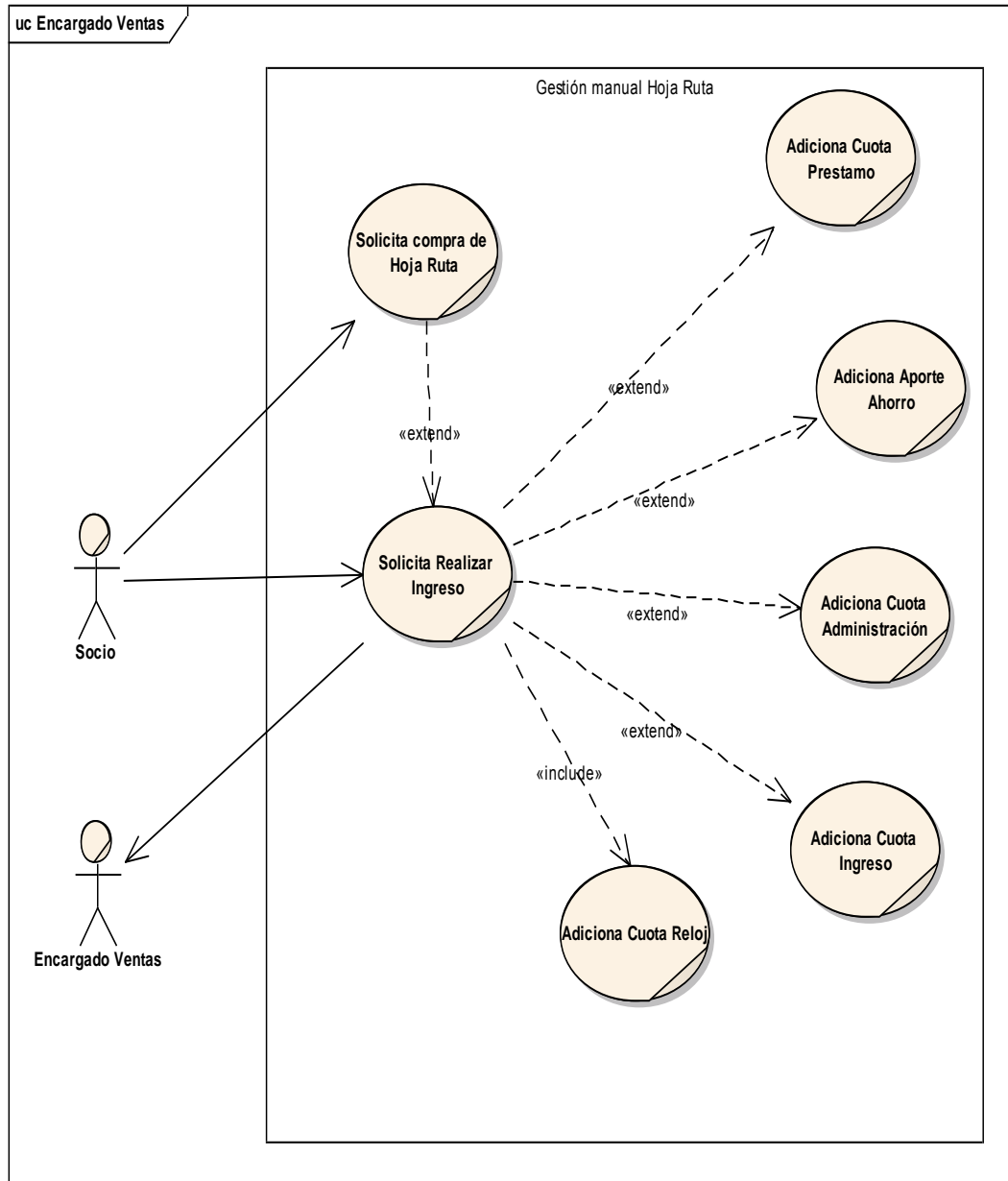
Descripción: Para la generación de la rotación de socios se debe tomar en cuenta la actualización de rol realizada, con los socios que se hallan activos.

En la rotación de socios se toma en cuenta la hora de salida la posición y la fecha, un determinado socio no puede salir dos días consecutivos de la misma parada, en una misma hora.

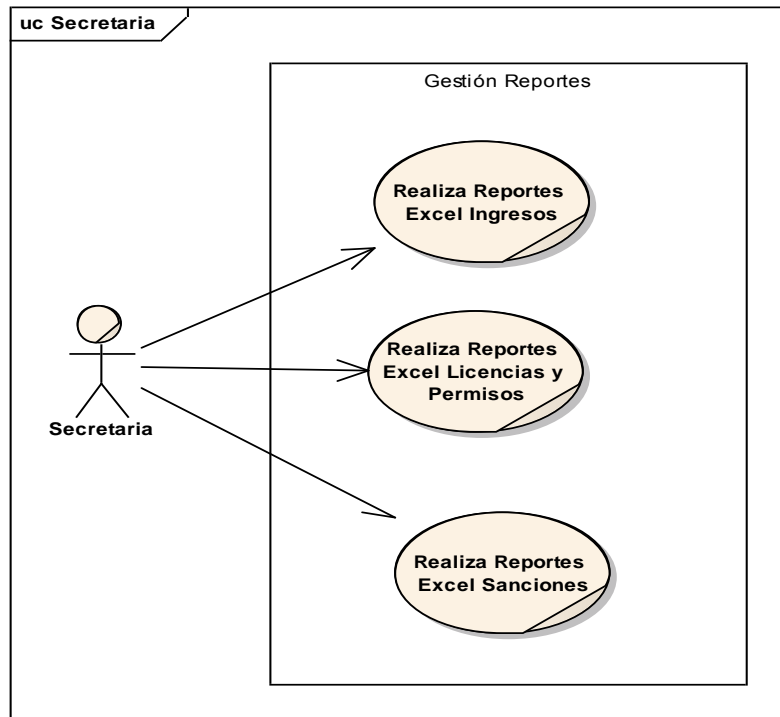
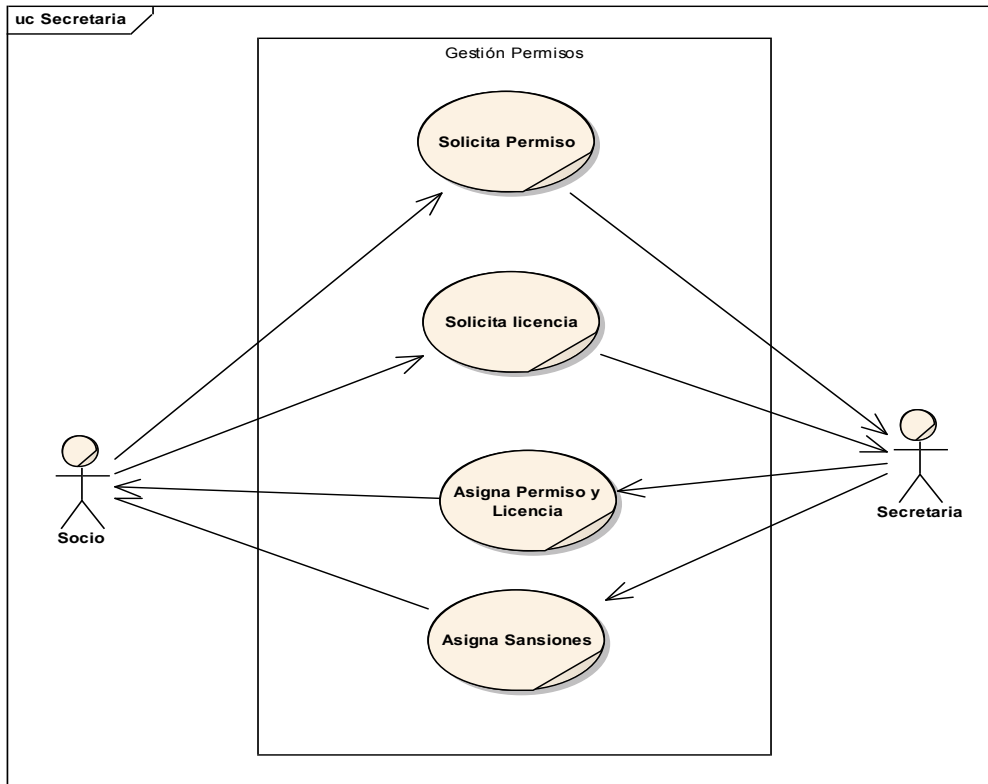
3.2.3 Diagrama de casos de uso del Negocio



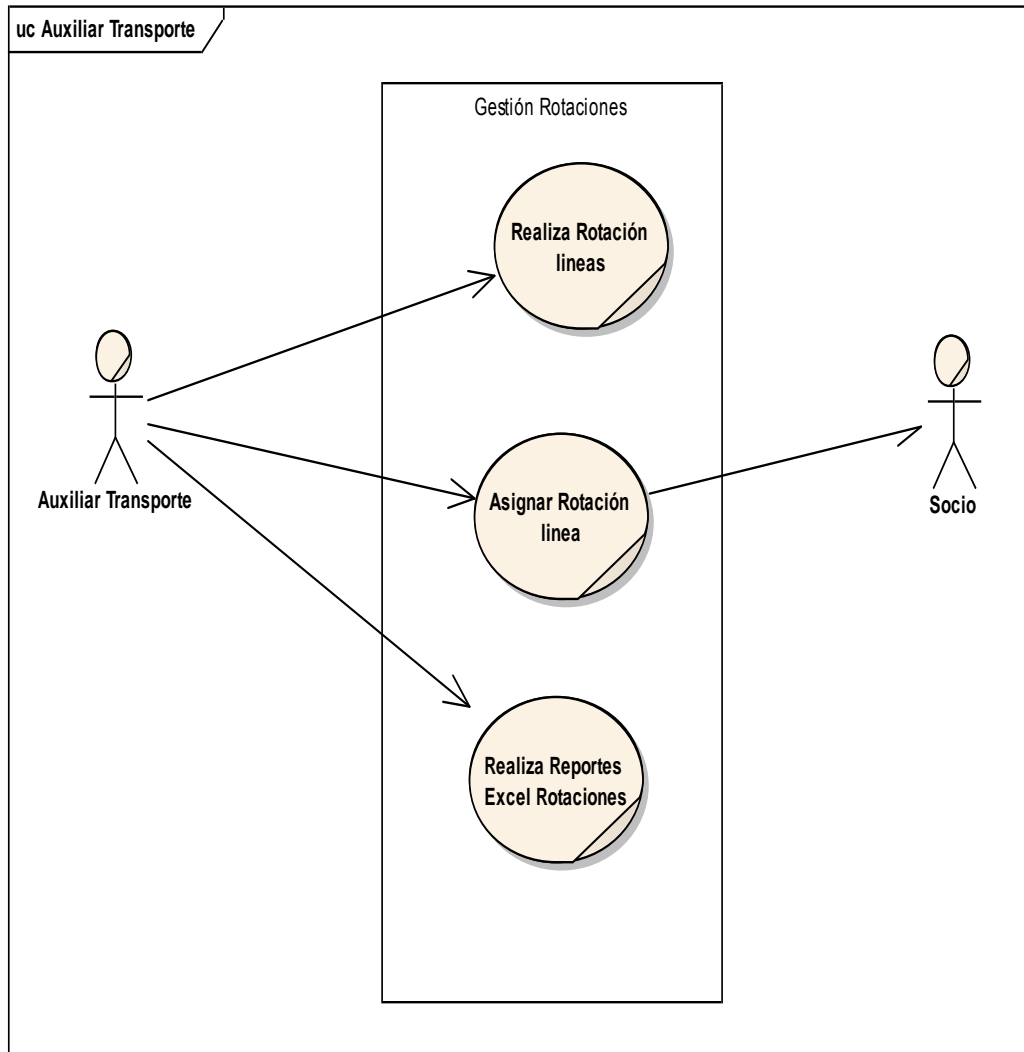
3.2.4 Diagrama de casos de uso de Encargado de Ventas: Gestión Hoja Ruta



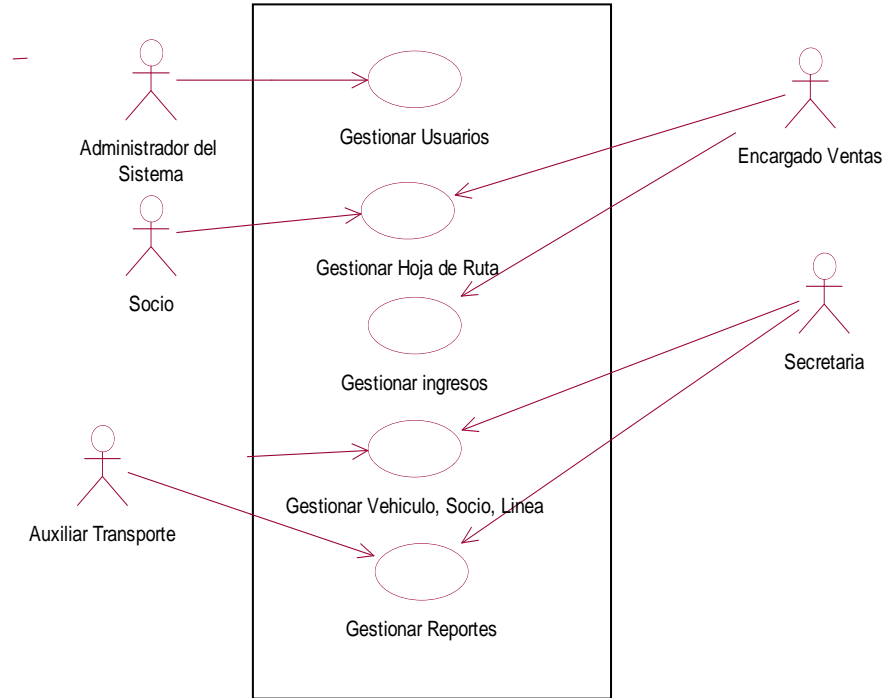
3.2.5 Diagrama de casos de uso Secretaria: Gestión, Permisos, Reportes



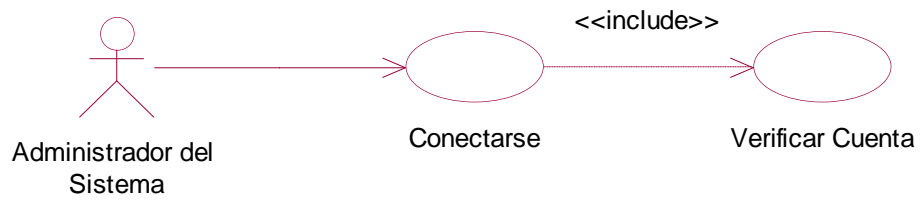
3.2.6 Diagrama de casos de uso Auxiliar de Transporte: Gestión Rotaciones, Reportes.



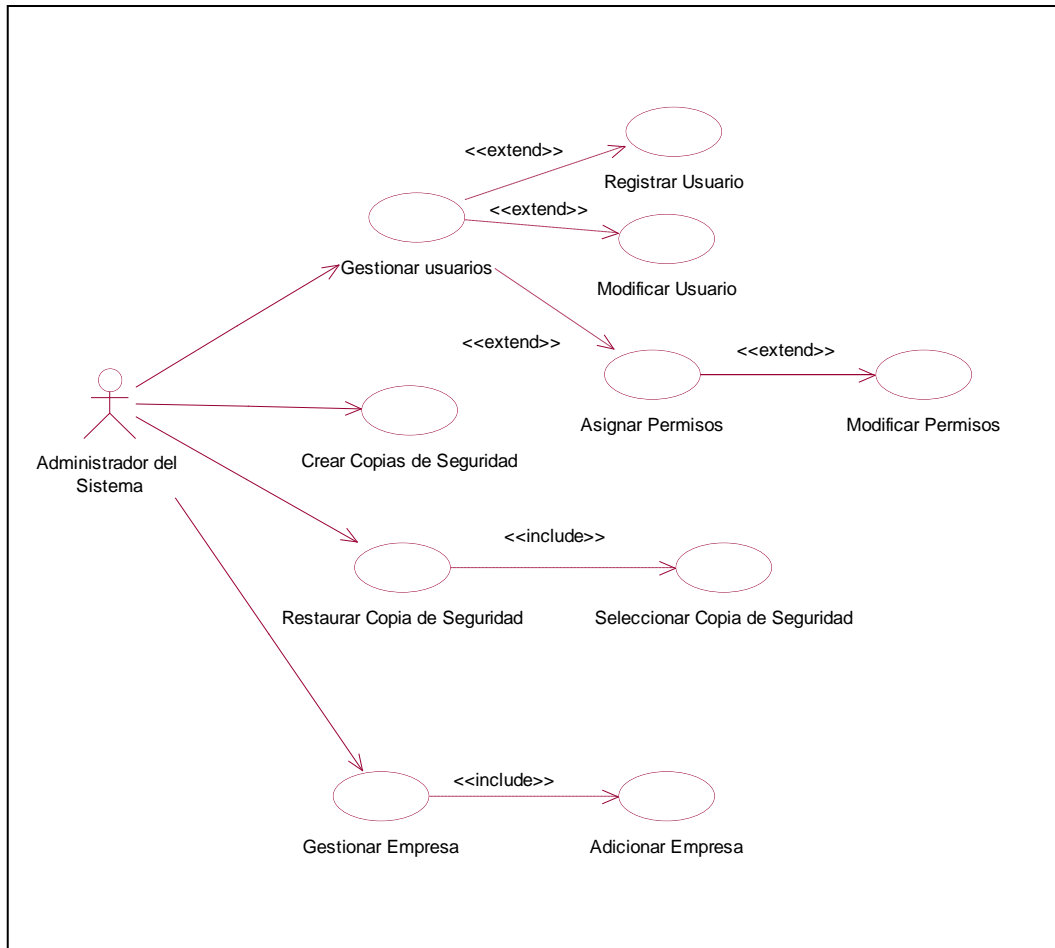
3.2.7 Diagrama de contexto del sistema



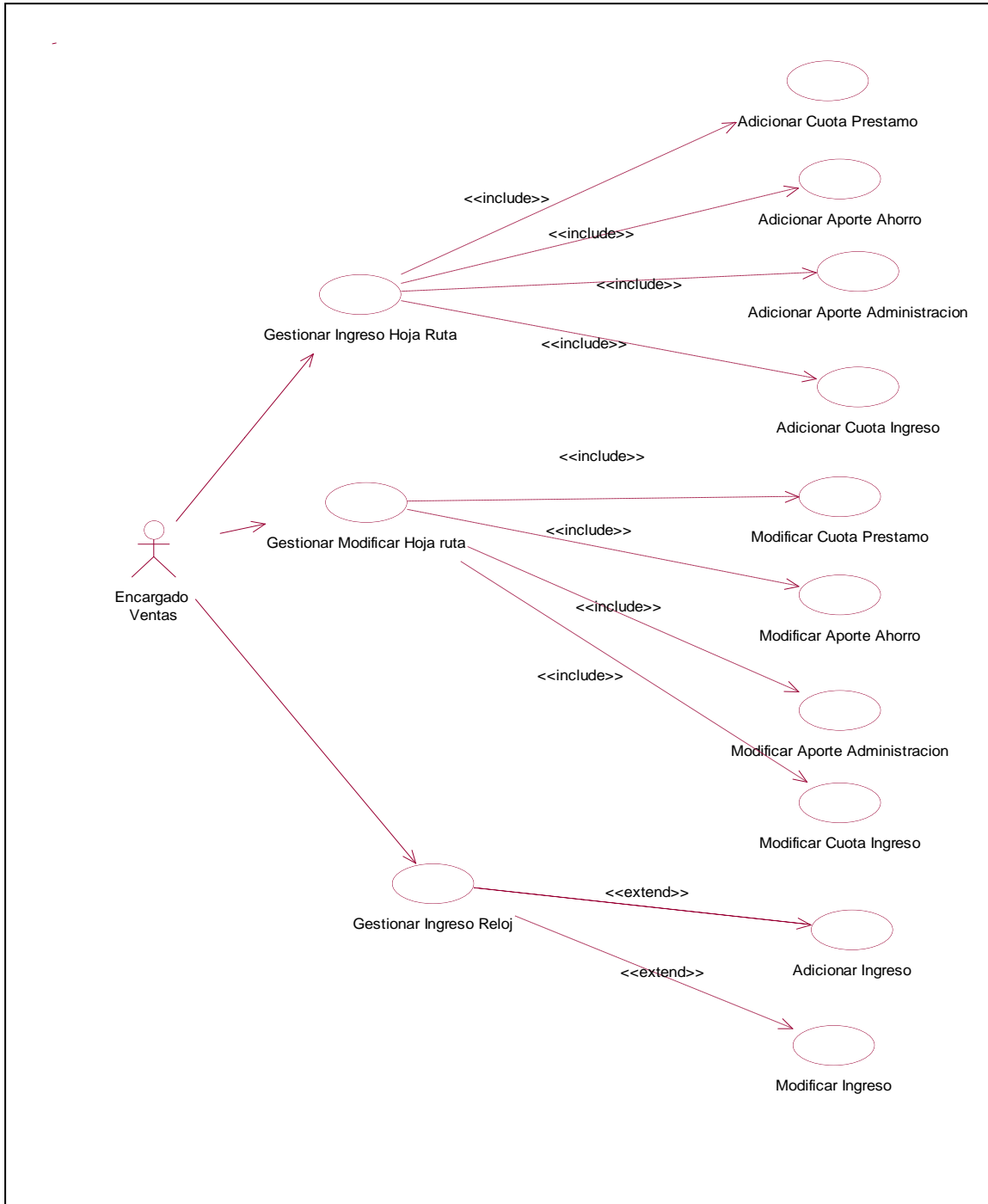
3.2.8 Diagrama de Casos de Uso: Conectarse



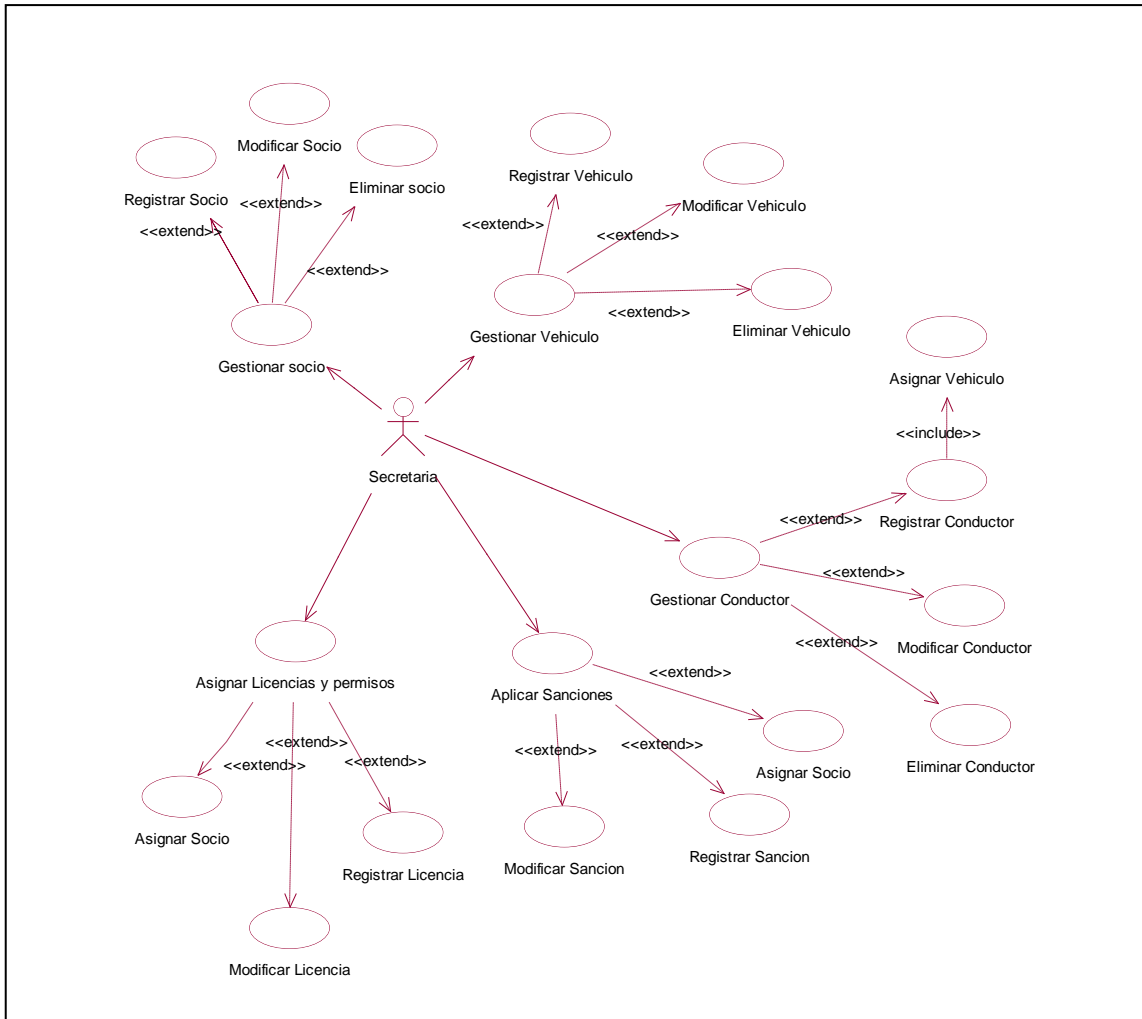
3.2.9 Diagrama de Casos de Uso: Administrador Sistema.- Es el encargado de registrar, modificar y asignar usuarios, crear copias de seguridad, restaurar copias de seguridad, adicionar datos de la cooperativa o empresa.



3.2.10 Diagrama de Casos de Uso: Encargado de Ventas.- Es el encargado de realizar los ingresos y modificaciones por cuota préstamo, aporte de ahorro, aporte administración, cuota de ingreso en una hoja de ruta.

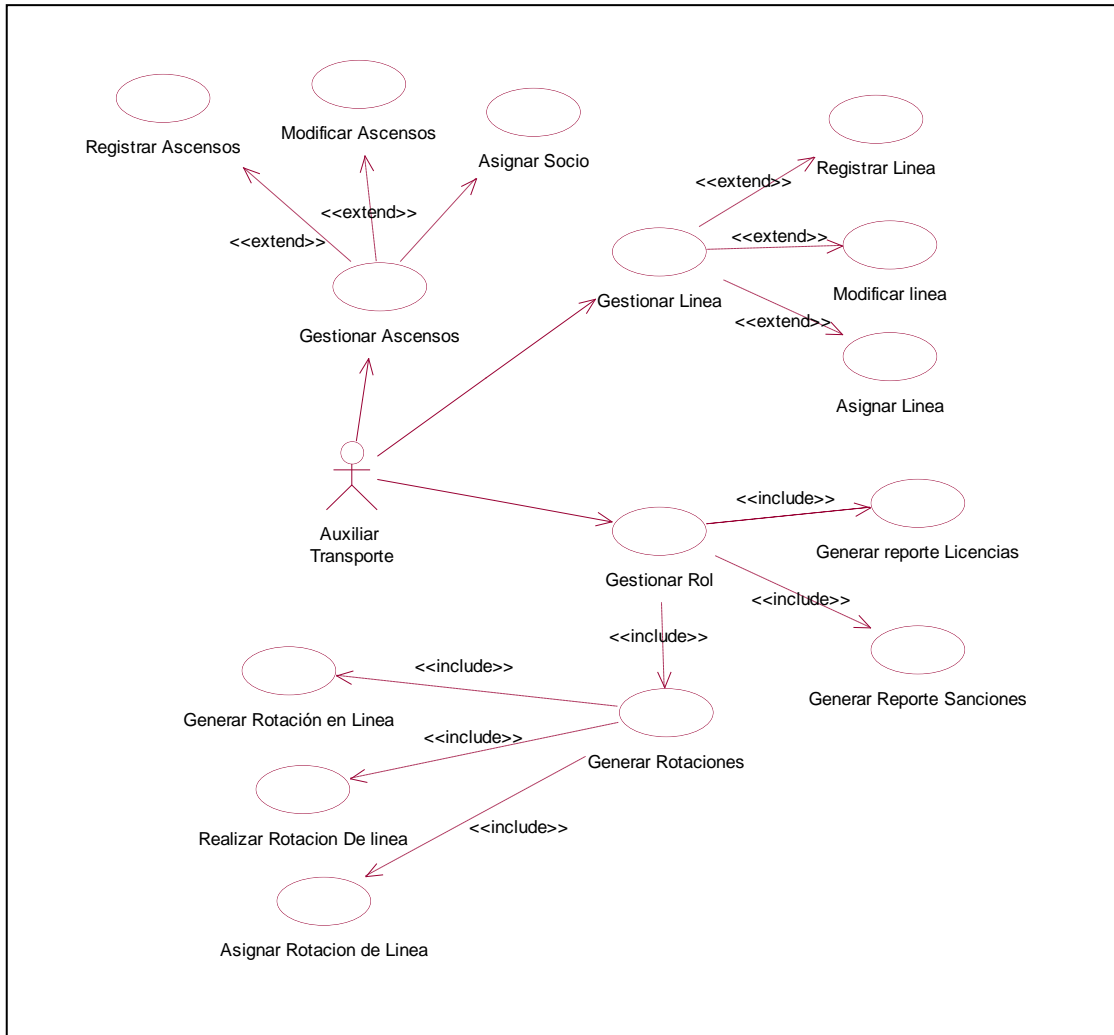


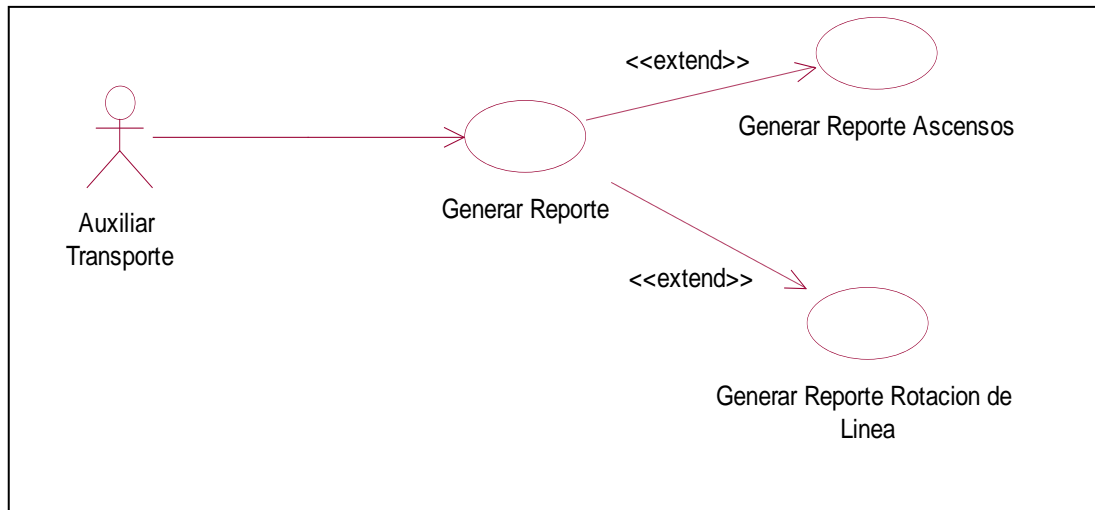
3.2.11 Diagrama de Casos de Uso: Secretaria.- Es el encargado de registrar, modificar y eliminar socio, vehículo, conductor, aplicar sanciones y aplicar licencias y permisos a un socio, además de generar reportes.





3.2.12 Diagrama de Casos de Uso: Auxiliar de Transporte.- Es el encargado de registrar, modificar y eliminar ascensos, línea, rotaciones y aplicar estas rotaciones a los grupos de socios.





3.2.13 Especificaciones de los Casos de Uso

Casos de Uso	Conectarse
Descripción	El usuario ingresa su usuario y clave
Actor	Usuario
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Tener un usuario ✓ Ingresado al sistema
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema despliega el formulario para introducir el usuario y clave. 2. El usuario ingresa el usuario y clave. 3. El usuario hace click en el botón ingresar
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario hace click en cerrar cancelando el ingreso al sistema 2. El usuario no cuenta con usuario y hace click en registrar usuario.
Post condición	Validar Usuario

Casos de Uso	Gestionar Empresa
Descripción	El Administrador Sistema ingresa los datos de la empresa.
Actor	Administrador Sistema
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar de empresa.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra los datos de la empresa. 2. El Administrador Sistema selecciona Adicionar. 3. El sistema le señala el formulario para Adicionar. 4. El Administrador Sistema introduce los datos de la empresa. 5. El Administrador Sistema hace click en Aceptar. 6. El sistema guarda en la base de datos el registro. 7. El sistema ejecuta mensaje datos registrados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Administrador Sistema hace click en el botón cancelar cancelando la adición. 2. El administrador sistema introduce datos inválidos. 3. El sistema despliega mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Usuarios
Descripción	El Administrador Sistemaregistrar, modificar o eliminar los datos de los usuarios.
Actor	Administrador Sistema
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Usuarios.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de los usuarios. 2. El Administrador sistema selecciona registrar. 3. El sistema le señala el formulario para registrar. 4. El Administrador Sistema introduce los datos de los usuarios. 5. El Administrador Sistema hace click en Aceptar. 6. El Sistema guarda los datos en la base de datos 7. El Sistema despliega un mensaje datos registrados correctamente. 8. El Administrador Sistema selecciona modificar. 9. El sistema le expande el formulario con los datos a modificar. 10. El Administrador Sistema introduce los datos a modificar. 11. El Administrador Sistema hace click en Aceptar. 12. El Sistema guarda los datos en la base de datos 13. El Sistema despliega un mensaje datos modificados correctamente. 14. El Administrador Sistema selecciona eliminar. 15. El sistema le muestra un mensaje con la confirmación de la eliminación de los usuarios. 16. El Administrador Sistema hace click en Aceptar. 17. El Sistema cambia el estado en la base de datos

Flujo Alternativo	<ol style="list-style-type: none"> 1. El Administrador Sistema hace click en el botón cancelar cancelando el registro, modificación o eliminación. 2. El Administrador Sistema introduce datos inválidos. 3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Asignar permisos
Descripción	El Administrador Sistema elige el usuario para asignarle un permiso.
Actor	Administrador Sistema
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar asignar permisos
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le señala un campo de texto para seleccionar el permiso. 2. El sistema le muestra un campo de texto para seleccionar usuario. 3. El Administrador Sistema selecciona el permiso y el usuario. 4. El sistema asigna al usuario el permiso. 5. El Administrador Sistema hace click en guardar.
Flujo Alternativo	<ol style="list-style-type: none"> 6. El Administrador Sistema hace click en el botón cerrar.
Post condición	Adicionar Acción

Casos de Uso	Crear Copia de Seguridad
Descripción	El Administrador Sistema realiza copias de seguridad para poder luego restablecer si es necesario.
Actor	Administrador Sistema
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar administrar. ✓ Pulsado en copias de seguridad.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un campo de texto y la opción examinar para elegir la ubicación donde se efectuará la copia de seguridad 2. El Administrador Sistema ingresa la ubicación donde se ejecutara la nueva copia de seguridad. 3. El Administrador Sistema hace click en el botón Aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Administrador Sistema hace click en el botón cancelar.
Post condición	Se guardan los datos.

Casos de Uso	Restaurar Copias de Seguridad
Descripción	El Administrador Sistema restablece una copia de seguridad que vea conveniente realizarlo.
Actor	Administrador Sistema
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar administrar. ✓ Pulsado en restaurar copias de seguridad.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un campo de texto y la opción examinar para elegir la ubicación donde se encuentra la copia de seguridad a restablecer 2. El Administrador Sistema ingresa la ubicación donde se encuentra la copia a restablecer. 3. El Administrador Sistema hace click en el botón Aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Administrador Sistema hace click en el botón cancelar.
Post condición	Se restablecen los datos de la base de datos.

Casos de Uso	Gestionar Ingreso Hoja Ruta
Descripción	El Encargado Ventas adicionar los datos de la cuota de préstamo, Aporte Ahorro, Aporte Administración, Ingreso en la hoja de ruta.
Actor	Encargado Ventas
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Ingreso Hoja Ruta.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de todas las hojas de ruta. 2. El Encargado Ventas selecciona adicionar. 3. El sistema le despliega el formulario para adicionar. 4. El Encargado Ventas introduce los datos de la hoja de ruta. 5. El Encargado Ventas hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema señala un mensaje datos registrados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Encargado Ventas hace click en el botón cancelar cancelando el ingreso de la hoja de ruta. 2. El Encargado Ventas introduce datos inválidos. 3. El sistema da mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Modificar Hoja Ruta
Descripción	El Encargado Ventas modificar los datos de la cuota de préstamo, Aporte Ahorro, Aporte Administración, Ingreso en la hoja de ruta.
Actor	Encargado Ventas
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Ingreso Hoja Ruta.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de todas las hojas de ruta. 2. El Encargado Ventas selecciona modificar. 3. El sistema le señala el formulario con los datos a modificar. 4. El Encargado Ventas introduce los datos a modificar. 5. El Encargado Ventas hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema evidencia un mensaje datos modificados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Encargado Ventas hace click en el botón cancelar cancelando la modificación. 2. El Encargado Ventas introduce datos inválidos. 3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Aporte Ingreso Reloj
Descripción	El Encargado Ventas adicionar, modificar los datos de la ingreso por reloj.
Actor	Encargado Ventas
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Ingreso Reloj.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado ingresos por reloj. 2. El Encargado Ventasselecciona adicionar. 3. El sistema le demuestra el formulario para adicionar. 4. El Encargado Ventas introduce los datos de la ingreso. 5. El Encargado Ventas hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema señala un mensaje datos registrados correctamente. 8. El Encargado Ventas selecciona modificar. 9. El sistema le prueba el formulario con los datos a modificar. 10. El Encargado Ventas introduce los datos a modificar. 11. El Encargado Ventas hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema evidencia un mensaje datos modificados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Encargado Ventas hace click en el botón cancelar cancelando la adición, modificación. 2. El Encargado Ventas introduce datos inválidos. 3. El sistema indica mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Socio
Descripción	La Secretaria registrar, modificar,eliminar los datos del Socio.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Socio.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado Socios. 2. La Secretaria selecciona registrar. 3. El sistema le demuestra el formulario para registrar. 4. La Secretaria introduce los datos de los socios. 5. La Secretaria hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema indica un mensaje datos registrados correctamente. 8. La Secretaria selecciona modificar. 9. El sistema le prueba el formulario con los datos a modificar. 10. La Secretaria introduce los datos a modificar. 11. La Secretaria hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema evidencia un mensaje datos modificados correctamente. 14. La Secretaria selecciona el socio eliminar. 15. El sistema le señala un mensaje con la confirmación de la eliminación del socio. 16. La secretaria hace click en Aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La Secretaria hace click en el botón cancelar cancelando el registro, modificación y eliminación de los datos del socio. 2. La Secretaria introduce datos inválidos.

	3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Vehículo
Descripción	La Secretaria registrar, modificar,eliminar los datos del vehículo.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Vehículo.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de Vehículos. 2. La Secretaria selecciona registrar. 3. El sistema le prueba el formulario para registrar. 4. La Secretaria introduce los datos de los vehículos. 5. La Secretaria hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema indica un mensaje datos registrados correctamente. 8. La Secretaria selecciona modificar. 9. El sistema despliega el formulario con los datos a modificar. 10. La Secretaria introduce los datos a modificar. 11. La Secretaria hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema demuestra un mensaje datos modificados correctamente. 14. La Secretaria selecciona el vehículo eliminar. 15. El sistema le indica un mensaje con la confirmación de la eliminación del vehículo.

	16. La secretaria hace click en Aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La Secretaria hace click en el botón cancelar cancelando el registro, modificación y eliminación de los datos del vehículo. 2. La Secretaria introduce datos inválidos. 3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Conductor
Descripción	La Secretaria registrar, modificar,eliminar los datos del conductor.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Conductor.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de conductores. 2. La Secretaria selecciona registrar. 3. El sistema le demuestra el formulario para registrar. 4. La Secretaria introduce los datos de los conductores 5. La Secretaria selecciona el vehículo. 6. La Secretaria hace click en Aceptar. 7. El sistema guarda los datos en la base de datos. 8. El sistema indica un mensaje datos registrados correctamente. 9. La Secretaria selecciona modificar. 10. El sistema le prueba el formulario con los datos a modificar. 11. La Secretaria introduce los datos a modificar. 12. La Secretaria hace click en Aceptar. 13. El sistema guarda los datos en la base de datos. 14. El sistema evidencia un mensaje datos modificados correctamente. 15. La Secretaria selecciona el conductor eliminar. 16. El sistema le señala un mensaje con la confirmación de la eliminación del conductor. 17. La secretaria hace click en Aceptar.

Flujo Alternativo	<ol style="list-style-type: none">1. La Secretaria hace click en el botón cancelar cancelando el registro, modificación y eliminación de los datos del conductor.2. La Secretaria introduce datos inválidos.3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Aplicar Sanciones
Descripción	La Secretaria registrar, modificar, las sanciones a los socios que incumplan alguna regla.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Aplicar Sanción.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de sanciones. 2. La Secretaria selecciona aplicar sanción. 3. El sistema le indica el formulario para asignar sanción. 4. La Secretaria asigna la sanción al socio. 5. La Secretaria hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema prueba un mensaje datos asignados correctamente. 8. La Secretaria selecciona modificar. 9. El sistema le demuestra el formulario con los datos a modificar. 10. La Secretaria selecciona socio a modificar. 11. La Secretaria hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema evidencia un mensaje datos modificados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La Secretaria hace click en el botón cancelar cancelando el registro, modificación de la sanción a un socio. 2. La Secretaria introduce datos inválidos. 3. El sistema señala mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Aplicar Licencias y permisos
Descripción	La Secretaria registrar, modificar, las licencias y permisos a los socios que así lo solicitan.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Aplicar Permiso.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de licencias y permisos. 2. La Secretaria selecciona aplicar permiso. 3. El sistema le prueba el formulario para asignar permiso. 4. La Secretaria asigna el permiso al socio. 5. La Secretaria hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema evidencia un mensaje datos asignados correctamente. 8. La Secretaria selecciona modificar. 9. El sistema le demuestra el formulario con los datos a modificar. 10. La Secretaria selecciona socio a modificar. 11. La Secretaria hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema indica un mensaje datos modificados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La Secretaria hace click en el botón cancelar suprimiendo el registro, modificación del permiso a un socio. 2. La Secretaria introduce datos inválidos. 3. El sistema señala mensaje de error.
Post condición	Adicionar Acción.

Caso de Uso	Generar Reporte de Sanciones
Descripción	El sistema le muestra el reporte de las sanciones de acuerdo a cada socio.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Generar reporte de sanciones.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le señala formulario para elegir el socio. 2. La secretaria elige el socio y hace click en Aceptar. 3. El sistema el reporte de sanciones que corresponde a cada socio 4. La secretaria selecciona imprimir. 5. El sistema visualiza el reporte a imprimir. 6. La secretaria selecciona exportar. 7. El sistema visualiza el reporte de Excel.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La secretaria hace click en el botón cancelar
Post condición	Validar Datos

Caso de Uso	Generar Reporte de Licencias
Descripción	El sistema le muestra el reporte de las licencias y permisos de acuerdo a cada socio.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Generar reporte de Licencias.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le demuestra formulario para elegir el socio. 2. La secretaria elige el socio y hace click en Aceptar. 3. El sistema el reporte de licencias y permisos que corresponde a cada socio 4. La secretaria selecciona imprimir. 5. El sistema visualiza el reporte a imprimir. 6. La secretaria selecciona exportar. 7. El sistema visualiza el reporte de Excel.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La secretaria hace click en el botón cancelar
Post condición	Validar Datos

Caso de Uso	Generar Reporte de Ingresos
Descripción	El sistema le muestra el reporte de Ingresos de acuerdo al tipo de ingreso como ser por hoja de ruta, por reloj y ingresos adicionales.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Generar reporte de Ingresos.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le señala el formulario para elegir el tipo de ingreso. 2. La secretaria elige el tipo de ingreso y hace click en Aceptar. 3. El sistema el reporte de ingresos de acuerdo al tipo de ingreso. 4. La secretaria selecciona imprimir. 5. El sistema visualiza el reporte a imprimir. 6. La secretaria selecciona exportar. 7. El sistema visualiza el reporte de Excel.
Flujo Alternativo	<ol style="list-style-type: none"> 8. La secretaria hace click en el botón cancelar
Post condición	Validar Datos

Casos de Uso	Gestionar Ingresos Adicionales
Descripción	La secretaria adicionar, modificar los datos de la ingreso por ahorros adicionales o ingresos por solicitudes.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Ingreso Adicionales.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado ingresos adicionales. 2. La secretaria selecciona adicionar Ahorros adicionales o ingresos por solicitudes. 3. El sistema le indica el formulario para adicionar. 4. La secretaria introduce los datos del ingreso. 5. La secretaria hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema prueba un mensaje datos registrados correctamente. 8. La secretaria selecciona modificar Ahorros adicionales o ingresos por solicitudes. 9. El sistema evidencia el formulario con los datos a modificar. 10. La secretaria introduce los datos a modificar. 11. La secretaria hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema despliega un mensaje datos transformados correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La secretaria hace click en el botón cancelar cancelando la adición, modificación. 2. La secretaria introduce datos inválidos.

	3. El sistema demuestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Cargos
Descripción	La secretaria registra, modifica o elimina los datos de los cargos.
Actor	Secretaria
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Cargos.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de los cargos. 2. La secretaria selecciona registrar. 3. El sistema prueba el formulario para registrar. 4. La secretaria introduce los datos del cargo. 5. La secretaria selecciona modificar. 6. El sistema le indica el formulario con los datos a modificar. 7. La secretaria introduce los datos a modificar. 8. La secretaria selecciona eliminar. 9. El sistema le evidencia un mensaje con la confirmación de la eliminación del cargo. 10. La secretaria hace click en Aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 1. La secretaria hace click en el botón cancelar cancelando el registro, modificación o eliminación. 2. La secretaria introduce datos inválidos. 3. El sistema señala mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Ascensos
Descripción	El Auxiliar Transporte registrar, modificar los datos del ascenso y asigna a un socio.
Actor	Auxiliar Transporte
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Ascensos.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado ascensos. 2. El Auxiliar Transporte selecciona Registrar 3. El sistema demuestra el formulario para registrar. 4. El Auxiliar Transporte introduce los datos del 5. El Auxiliar Transporte hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema prueba un mensaje datos registrados correctamente. 8. El Auxiliar Transporte selecciona modificar. 9. El sistema despliega el formulario con los datos a modificar. 10. El Auxiliar Transporte introduce los datos del 11. El Auxiliar Transporte hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema indica un mensaje datos modificados correctamente. 14. El Auxiliar Transporte seleccionar Asignar Ascensos. 15. El sistema le muestra los ascensos y socios para seleccionar. 16. El Auxiliar Transporte selecciona el ascenso, el socio y hace click en Aceptar.

	<p>17. El sistema guarda los datos en la base de datos.</p> <p>18. El sistema evidencia un mensaje datos asignados correctamente.</p>
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Auxiliar Transporte hace click en el botón cancelar cancelando el registro y modificación del ascenso. 2. El Auxiliar Transporte introduce datos inválidos. 3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Línea
Descripción	El Auxiliar Transporte registrar, modificar los datos de una línea y asigna a un socio.
Actor	Auxiliar Transporte
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Líneas.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado Líneas. 2. El Auxiliar Transporte selecciona Registrar 3. El sistema le muestra el formulario para registrar. 4. El Auxiliar Transporte introduce los datos de la Línea. 5. El Auxiliar Transporte hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema demuestra un mensaje datos registrados correctamente. 8. El Auxiliar Transporte selecciona modificar. 9. El sistema le muestra el formulario con los datos a modificar. 10. El Auxiliar Transporte introduce los datos de la línea. 11. El Auxiliar Transporte hace click en Aceptar. 12. El sistema guarda los datos en la base de datos. 13. El sistema muestra un mensaje datos modificados correctamente. 14. El Auxiliar Transporte seleccionar Asignar Línea. 15. El sistema le indica líneas y socios para seleccionar. 16. El Auxiliar Transporte selecciona la línea, el socio y hace click en Aceptar. 17. El sistema guarda los datos en la base de datos. 18. El sistema señala un mensaje datos asignados

	correctamente.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Auxiliar Transporte hace click en el botón cancelar cancelando el registro, modificación y asignación de la Línea. 2. El Auxiliar Transporte introduce datos inválidos. 3. El sistema muestra mensaje de error.
Post condición	Adicionar Acción.

Casos de Uso	Gestionar Rol
Descripción	El Auxiliar Transporte genera las rotaciones de las líneas y asigna a un socio.
Actor	Auxiliar Transporte
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Gestionar Rol.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le muestra un listado de rotaciones. 2. El Auxiliar Transporte Generar Rotaciones. 3. El sistema le indica el formulario para registrar rotación de línea. 4. El Auxiliar Transporte introduce los datos de la Rotación de Línea. 5. El Auxiliar Transporte hace click en Aceptar. 6. El sistema guarda los datos en la base de datos. 7. El sistema demuestra un mensaje datos registrados correctamente. 8. El sistema despliega el formulario para registrar rotación en línea. 9. El Auxiliar Transporte introduce los datos de la Rotación en Línea. 10. El Auxiliar Transporte hace click en Aceptar. 11. El sistema guarda los datos en la base de datos. 12. El sistema prueba un mensaje datos registrados correctamente. 13. El Auxiliar Transporte seleccionar Asignar Rotación Línea. 14. El sistema le señala rotación y socios para seleccionar. 15. El Auxiliar Transporte selecciona la rotación, el socio y hace click en Aceptar.

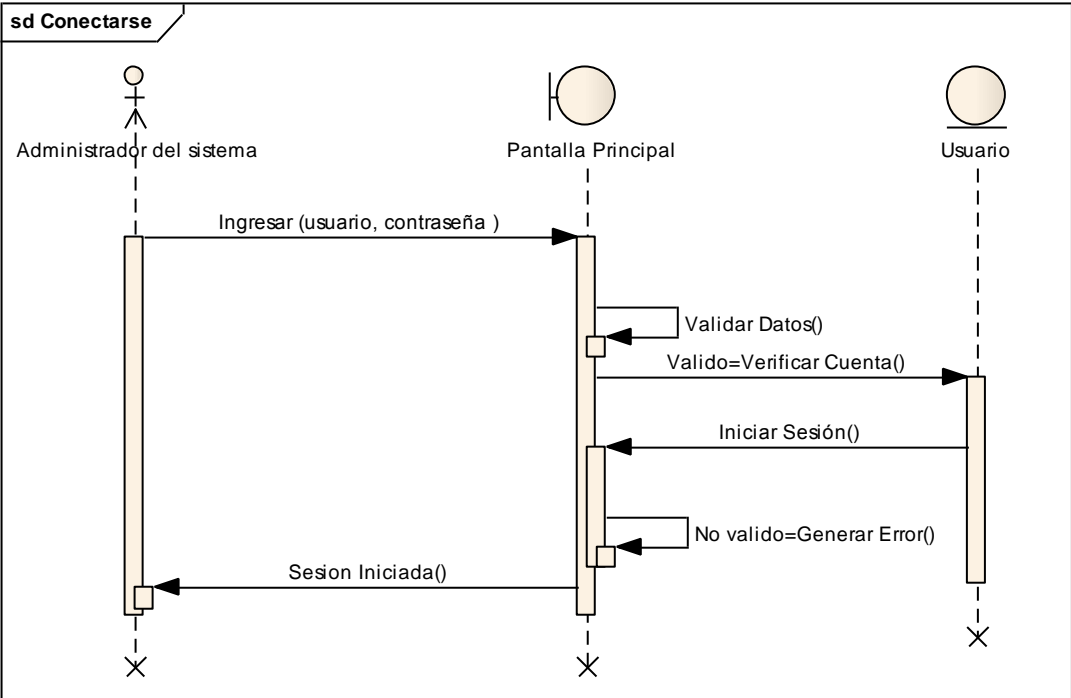
	<p>16. El sistema guarda los datos en la base de datos.</p> <p>17. El sistema muestra un mensaje datos asignados correctamente.</p>
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Auxiliar Transporte hace click en el botón cancelar cancelando la generación y asignación de la Línea. 2. El Auxiliar Transporte introduce datos inválidos. 3. El sistema señala mensaje de error.
Post condición	Adicionar Acción.

Caso de Uso	Generar Reporte de Ascensos
Descripción	El sistema le muestra el reporte de Ascensos de acuerdo al socio.
Actor	Auxiliar Transporte
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Generar reporte de Ascensos.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema le aprueba el formulario para elegir Ascensos. 2. El Auxiliar Transporte elige el Ascenso y hace click en Aceptar. 3. El sistema el reporte de ascensos de acuerdo al socio. 4. El Auxiliar Transporte selecciona imprimir. 5. El sistema visualiza el reporte a imprimir. 6. El Auxiliar Transporte selecciona exportar. 7. El sistema visualiza el reporte de Excel.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Auxiliar Transporte hace click en el botón cancelar
Post condición	Validar Datos

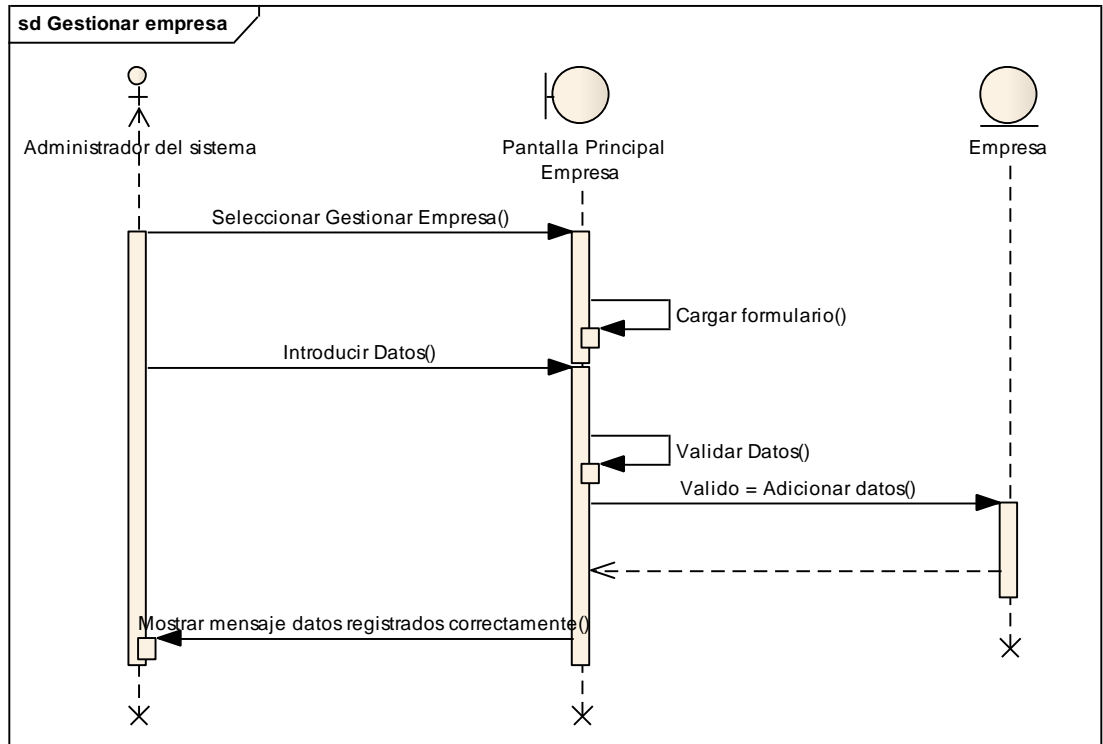
Caso de Uso	Generar Reporte de Rotaciones
Descripción	El sistema le muestra el reporte de rotaciones de acuerdo al socio.
Actor	Auxiliar Transporte
Tipo	Primario
Precondición	<ul style="list-style-type: none"> ✓ Conectarse ✓ Seleccionar Generar reporte de Rotaciones.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema despliega el formulario para elegir las rotaciones. 2. El Auxiliar Transporte eligelas rotaciones y hace click en Aceptar. 3. El sistema el reporte de rotaciones de acuerdo al socio. 4. El Auxiliar Transporte selecciona imprimir. 5. El sistema visualiza el reporte a imprimir. 6. El Auxiliar Transporte selecciona exportar. 7. El sistema visualiza el reporte de Excel.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Auxiliar Transporte hace click en el botón cancelar
Post condición	Validar Datos

3.3 Diagramas de Secuencia

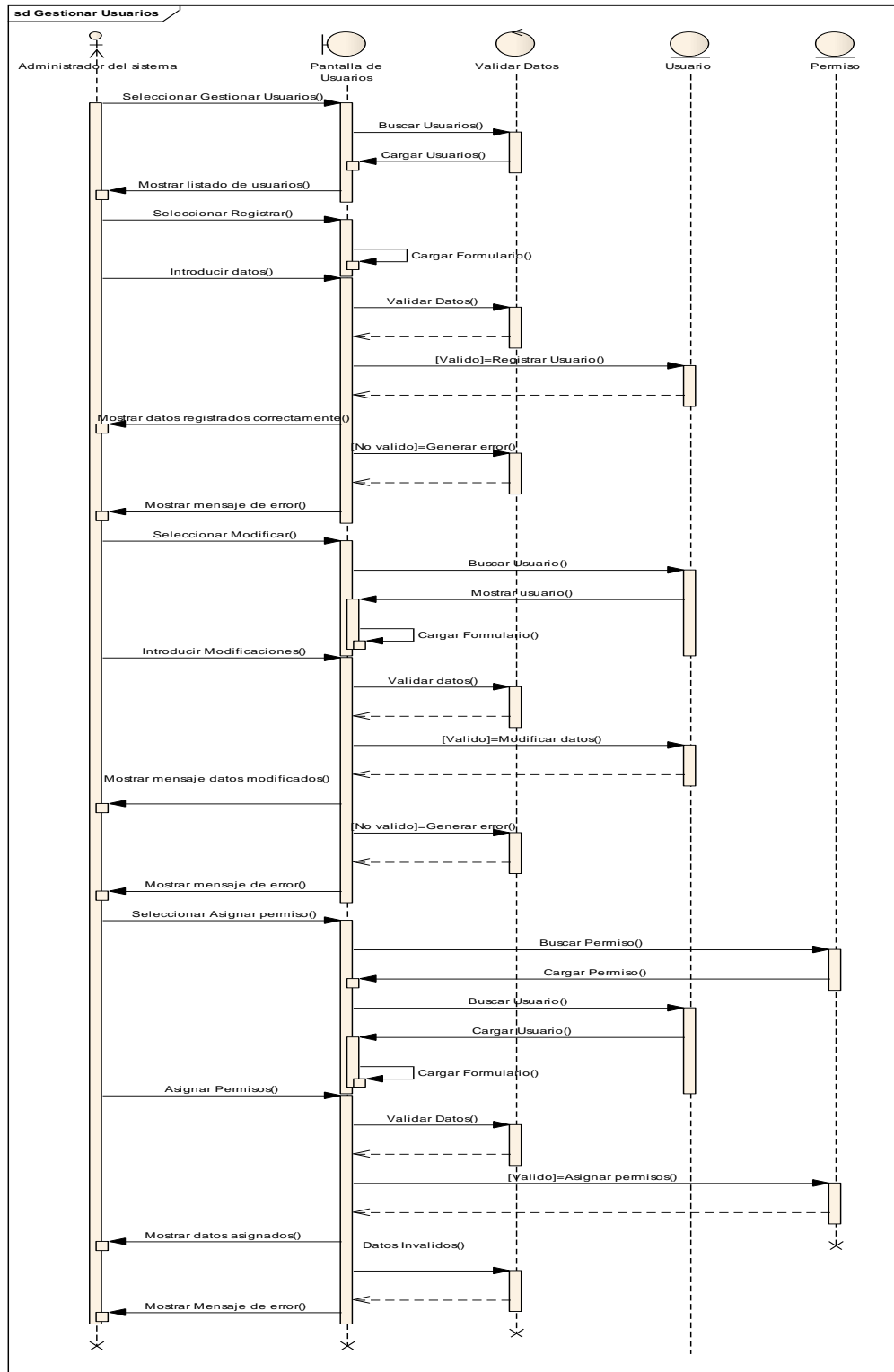
3.3.1 Diagrama de Secuencia: Conectarse



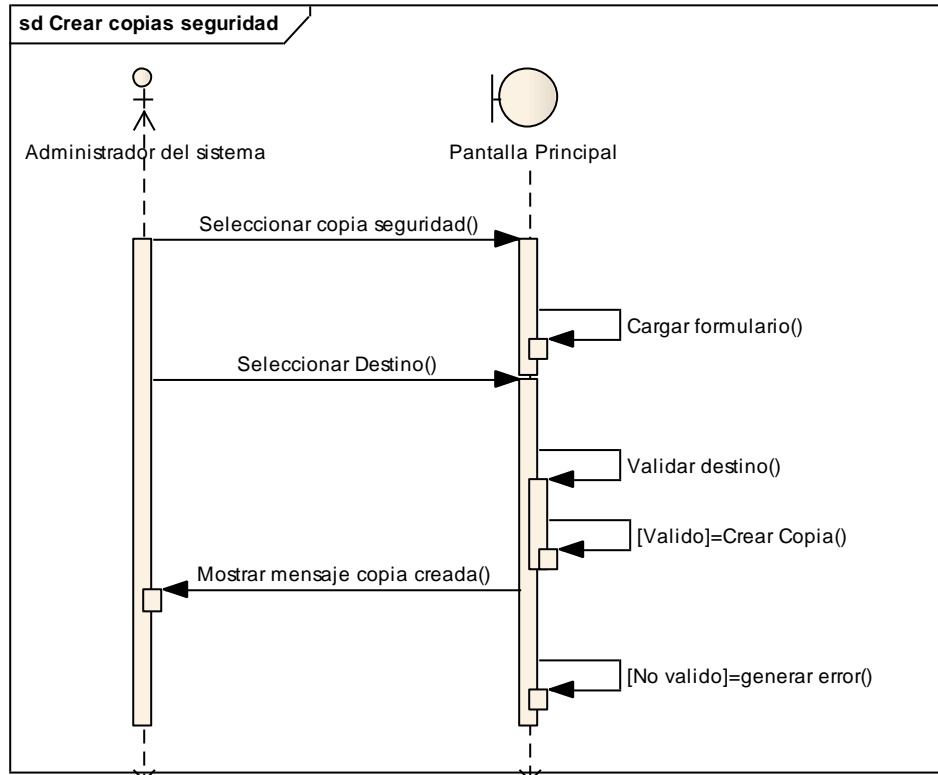
3.3.2 Diagrama de Secuencia: Gestionar Empresa



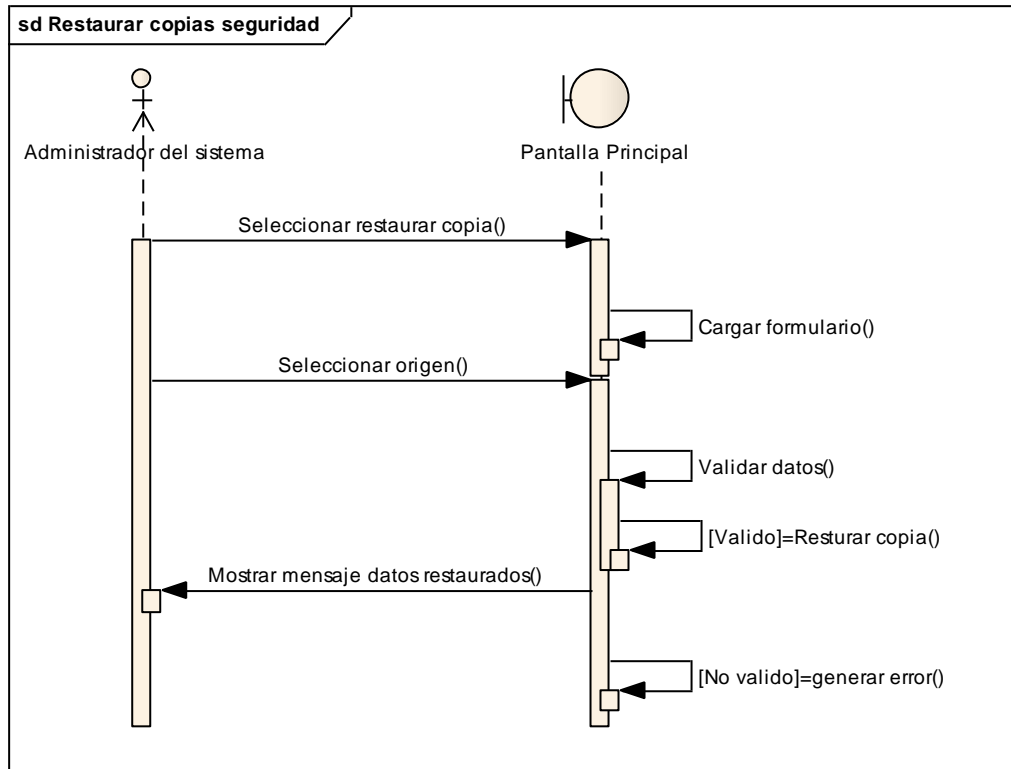
3.3.3 Diagrama de Secuencia: Gestionar Usuarios



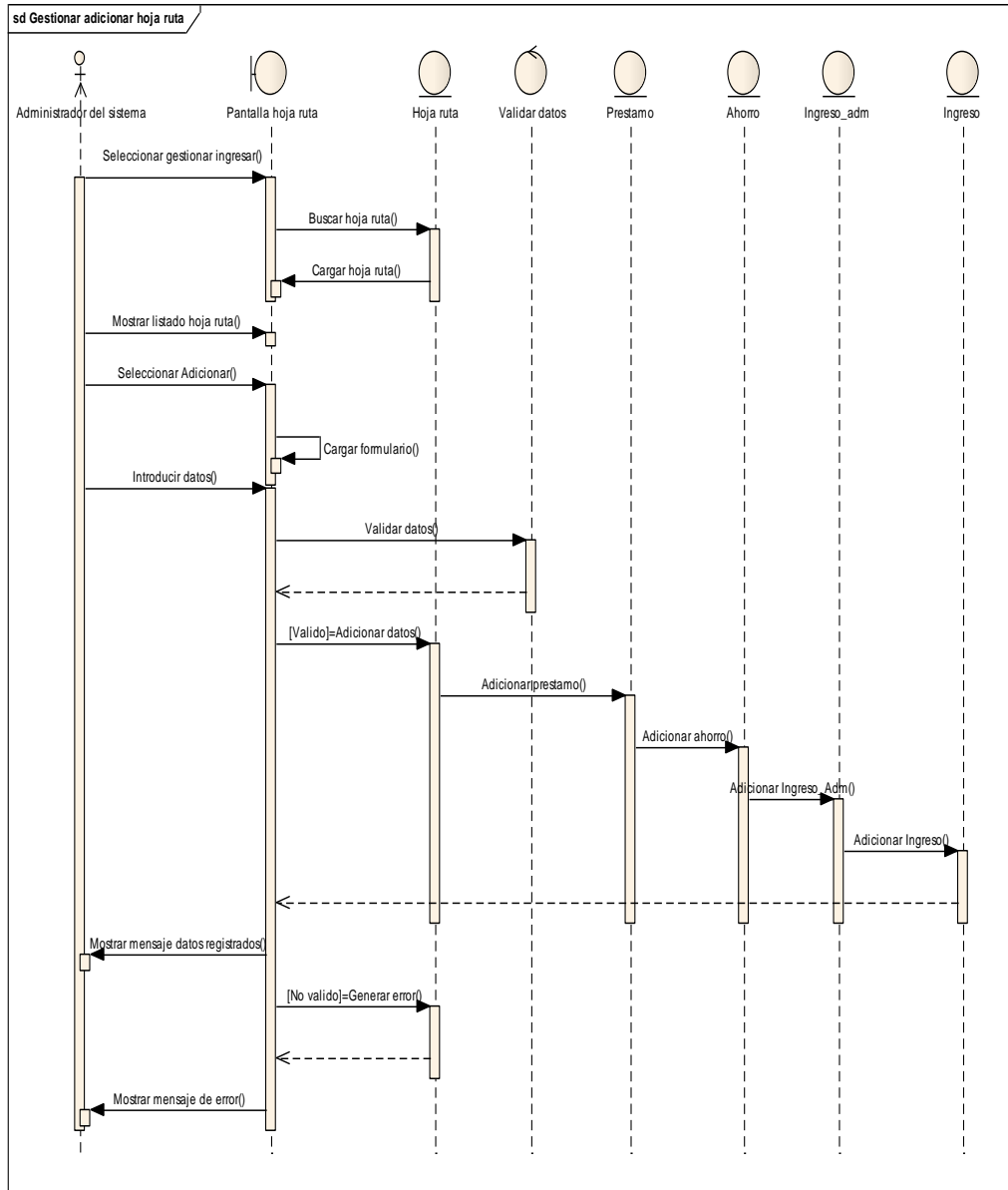
3.3.4 Diagrama de Secuencia: Crear Copias de Seguridad



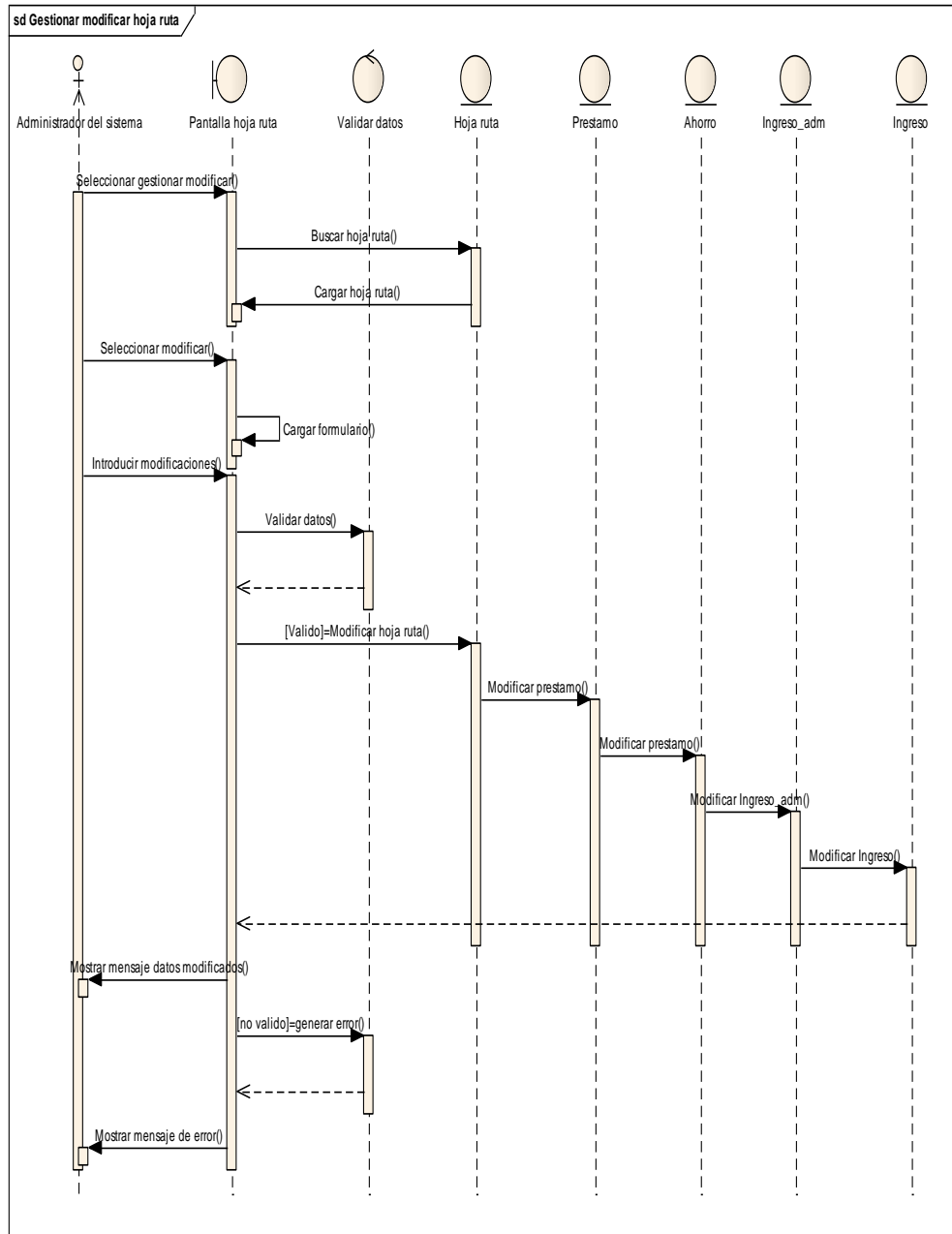
3.3.5 Diagrama de Secuencia: Restaurar Copias de Seguridad



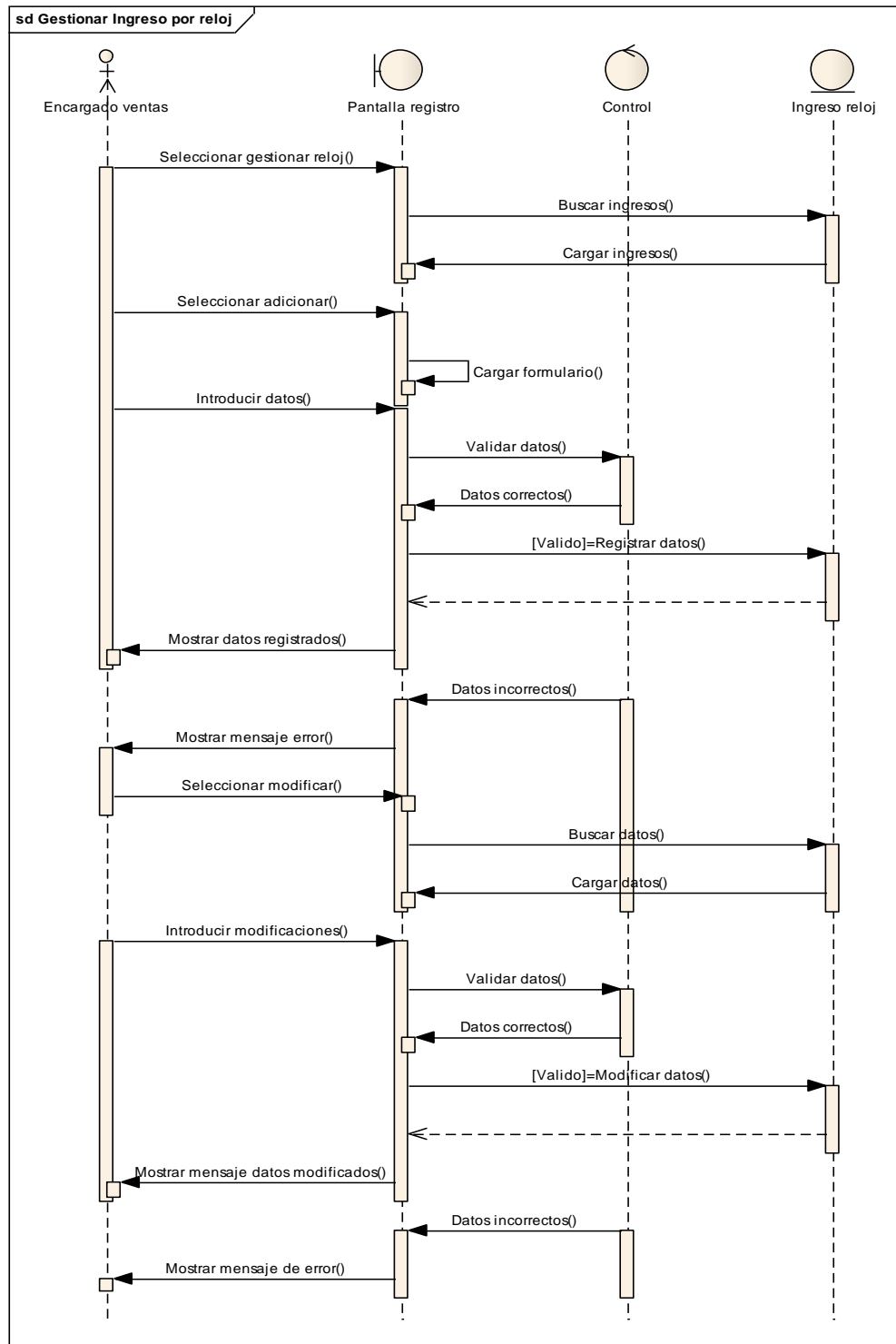
3.3.6 Diagrama de Secuencia: Gestionar Ingreso por Hoja de Ruta



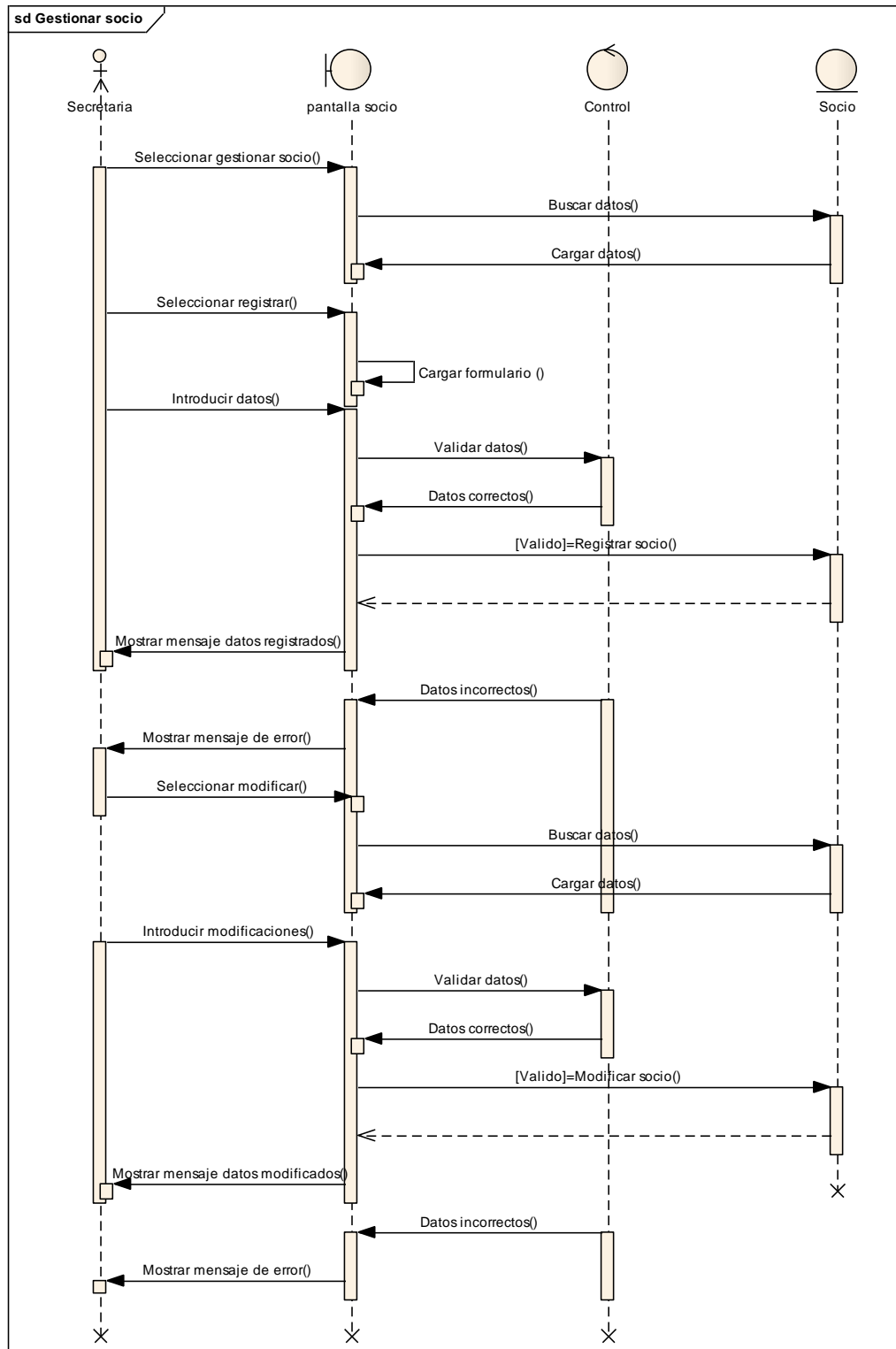
3.3.7 Diagrama de Secuencia: Gestionar Modificar por Hoja de Ruta



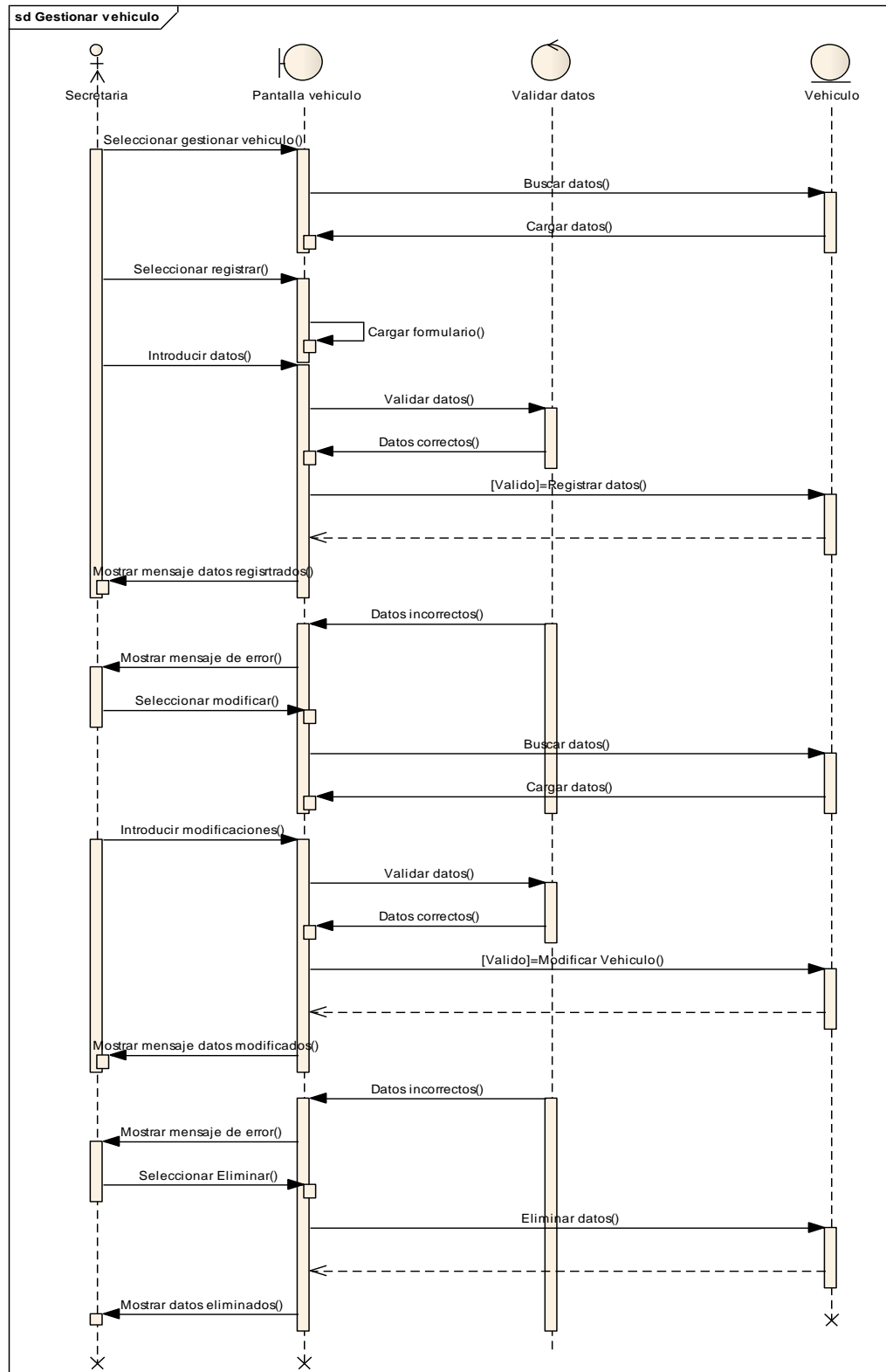
3.3.8 Diagrama de Secuencia: Gestionar Ingreso por Reloj



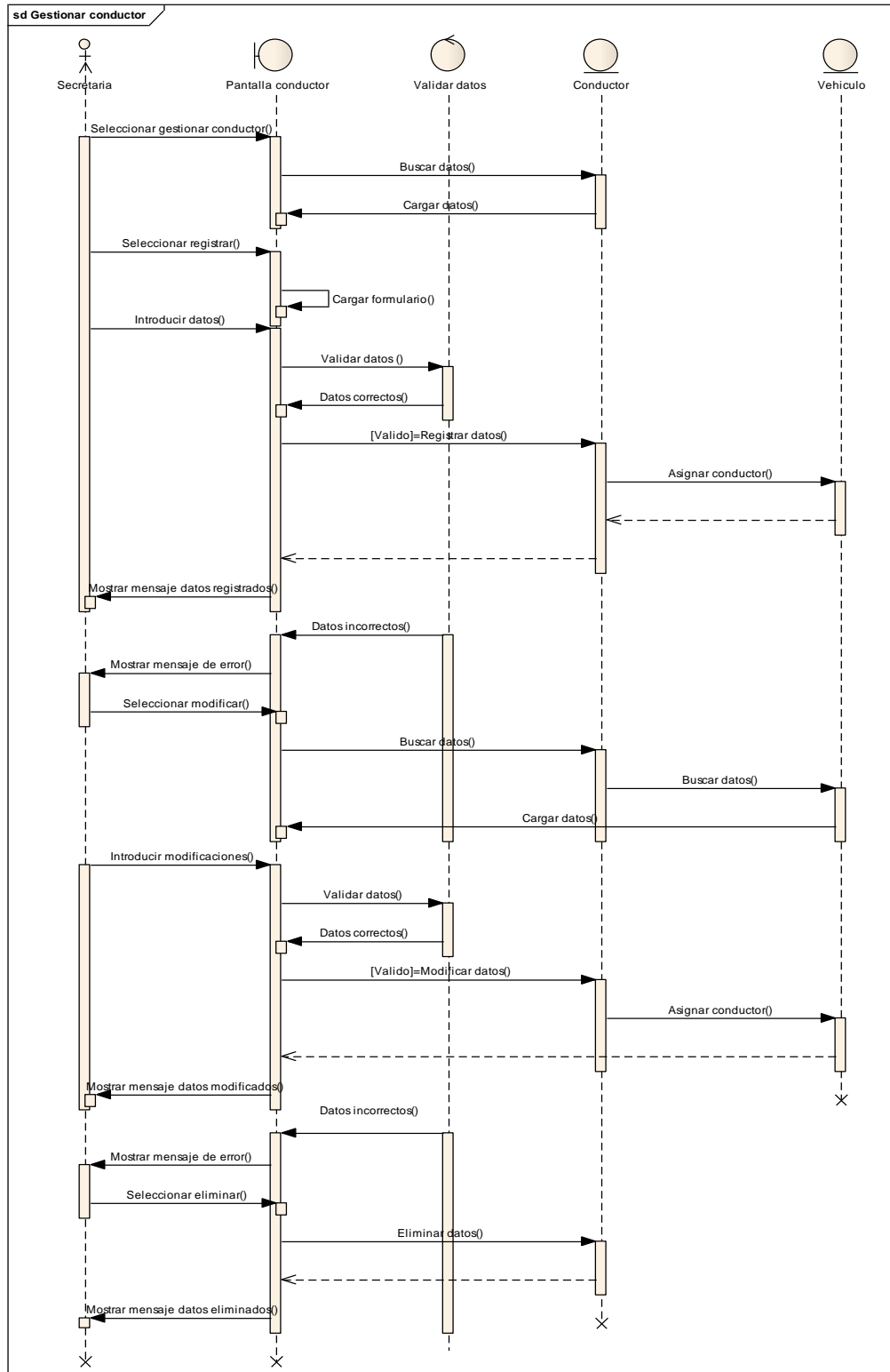
3.3.9 Diagrama de Secuencia: Gestionar Socio



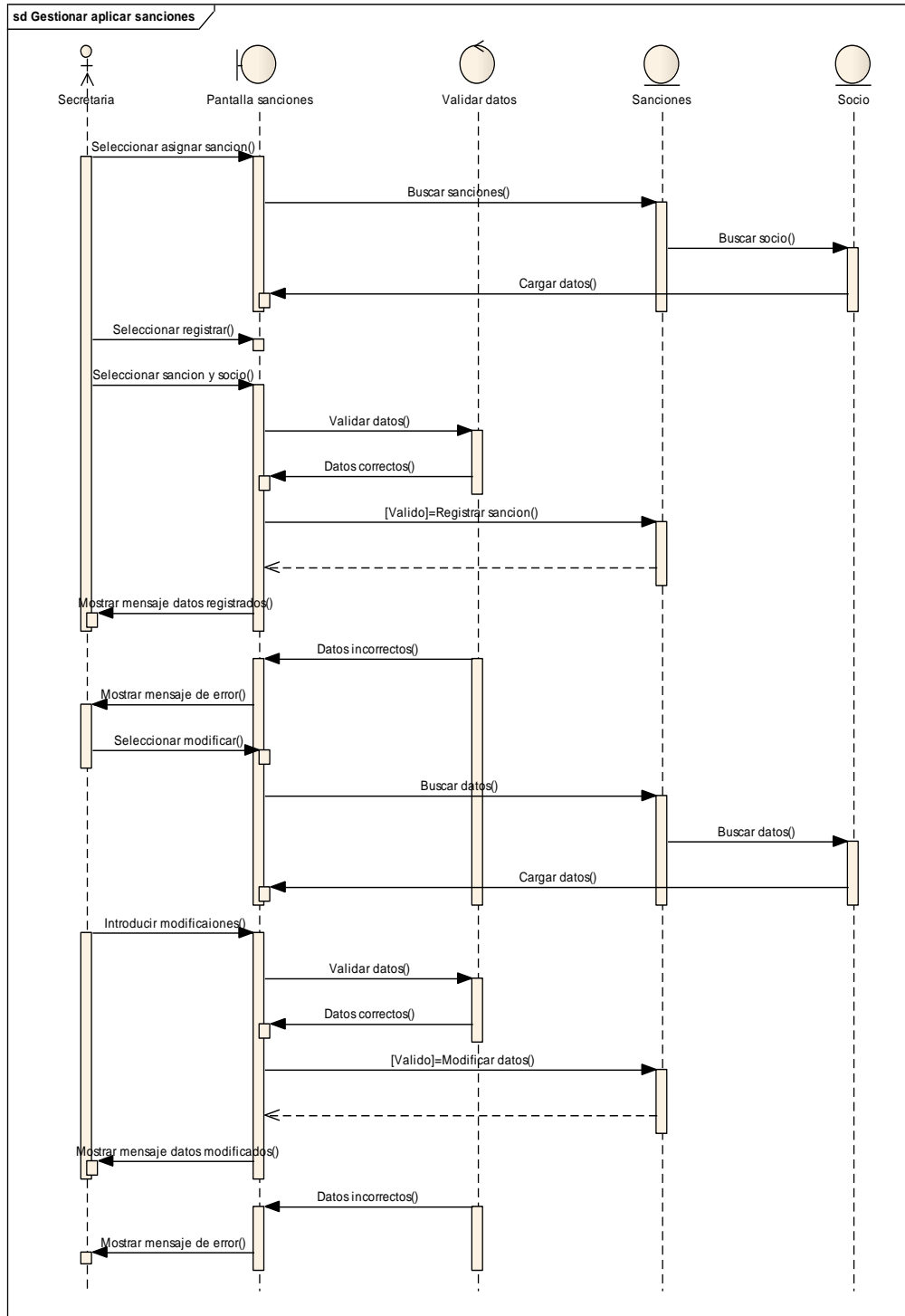
3.3.10 Diagrama de Secuencia: Gestionar Vehículo



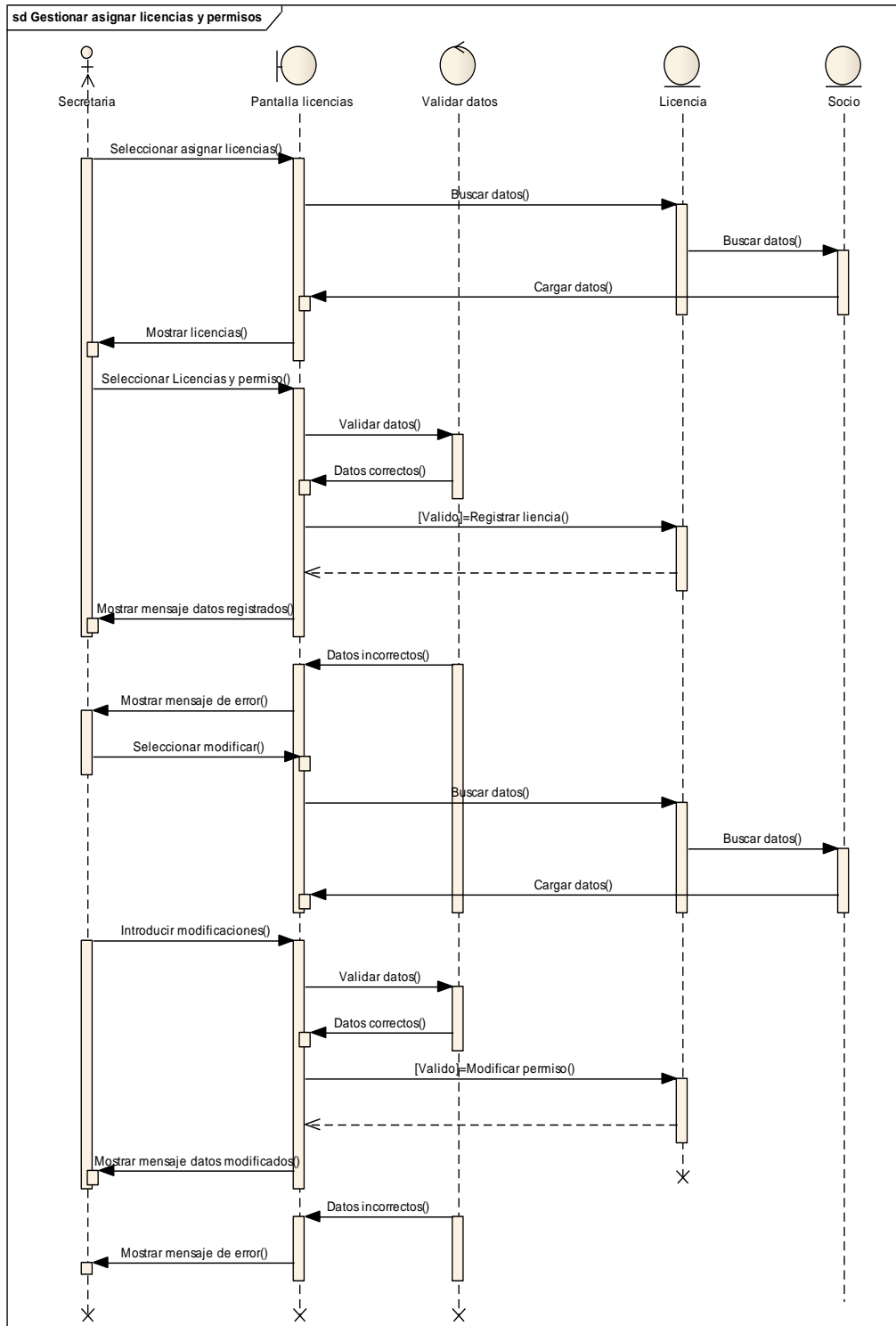
3.3.11 Diagrama de Secuencia: Gestionar Conductor



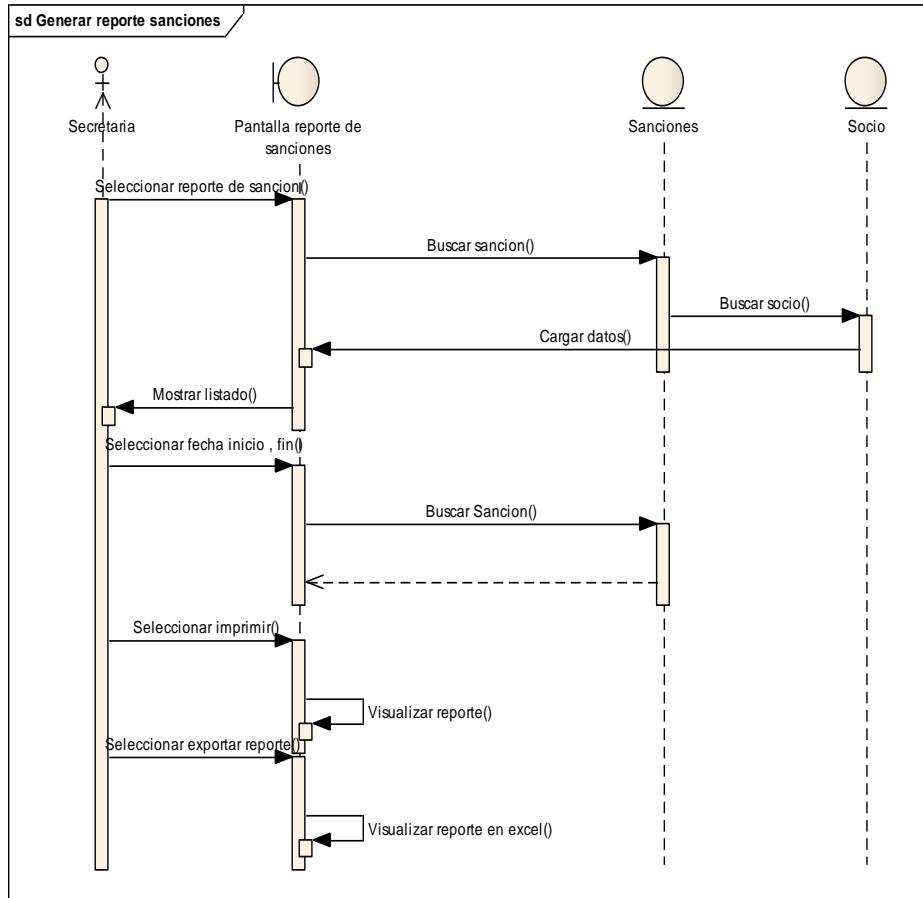
3.3.12 Diagrama de Secuencia: Gestionar Aplicar Sanciones



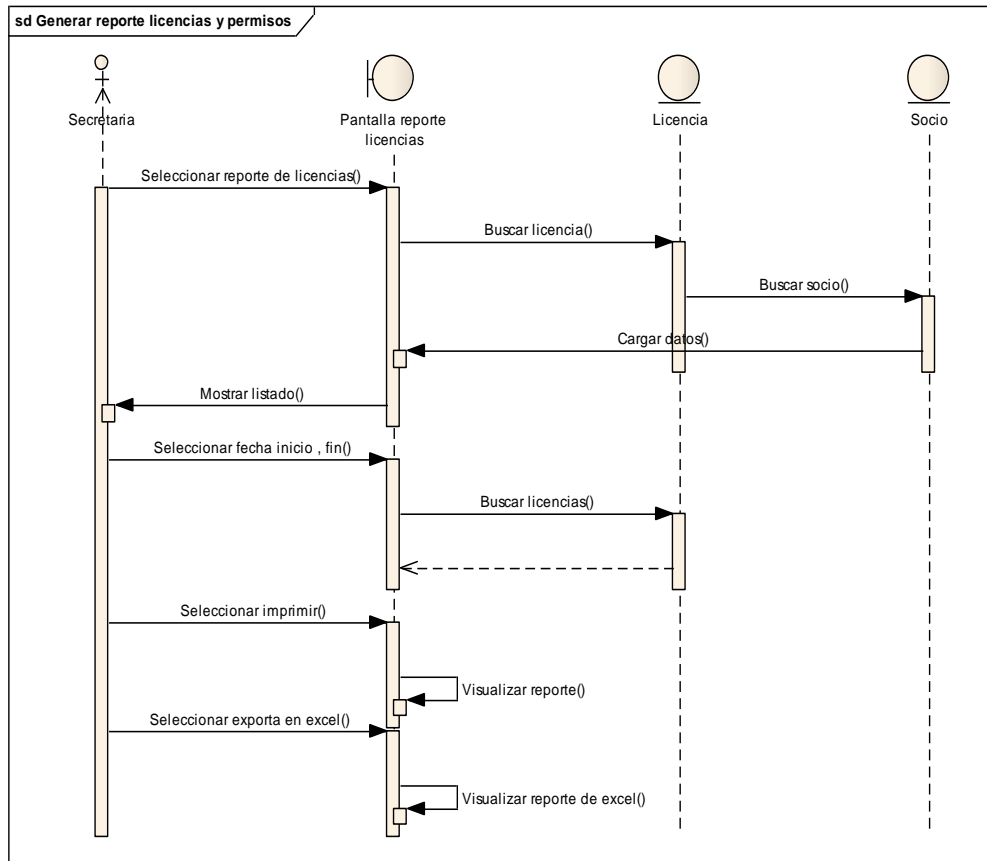
3.3.13 Diagrama de Secuencia: Gestionar Asignar Licencias y Permisos



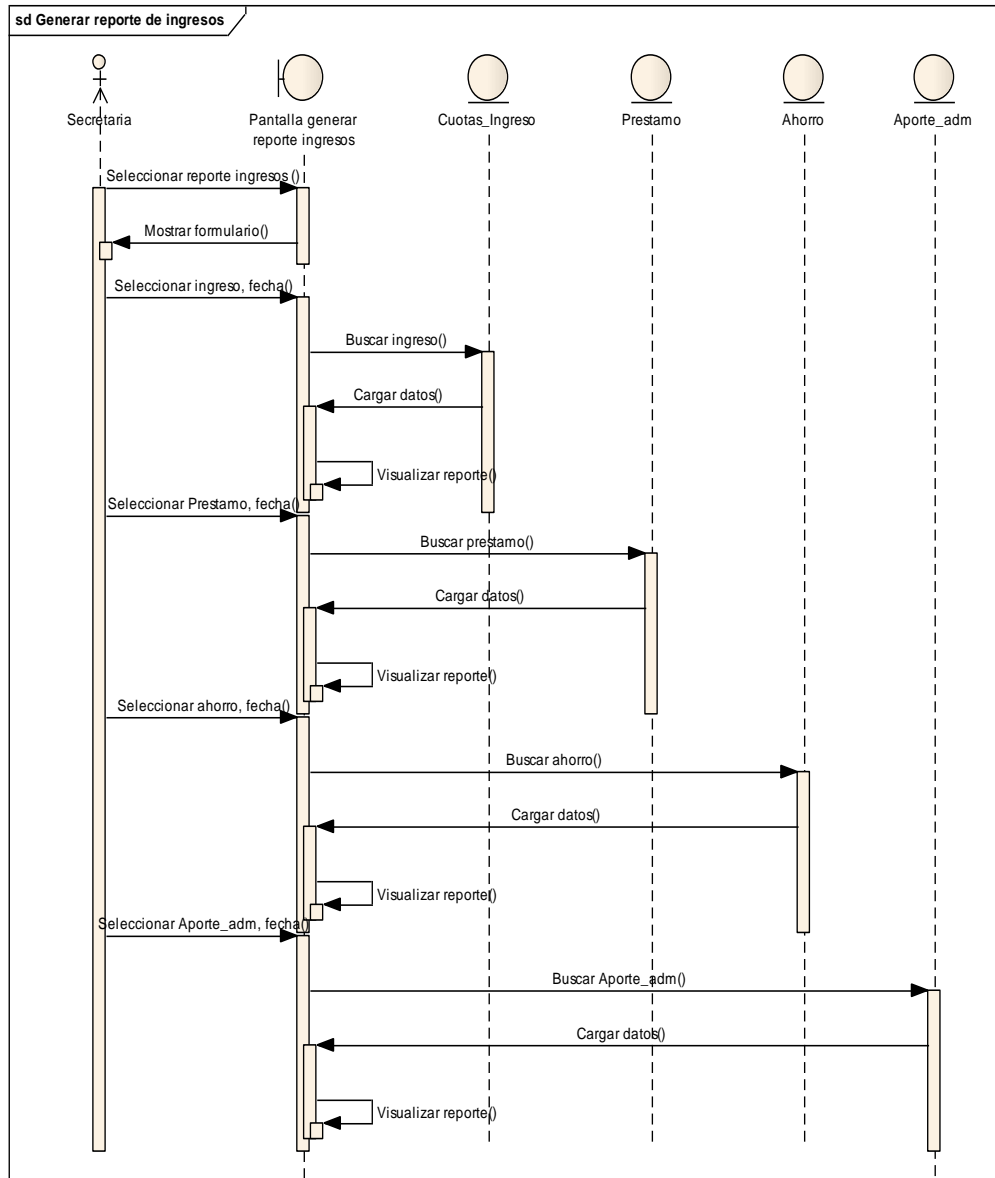
3.3.14 Diagrama de Secuencia: Generar Reporte de Sanciones



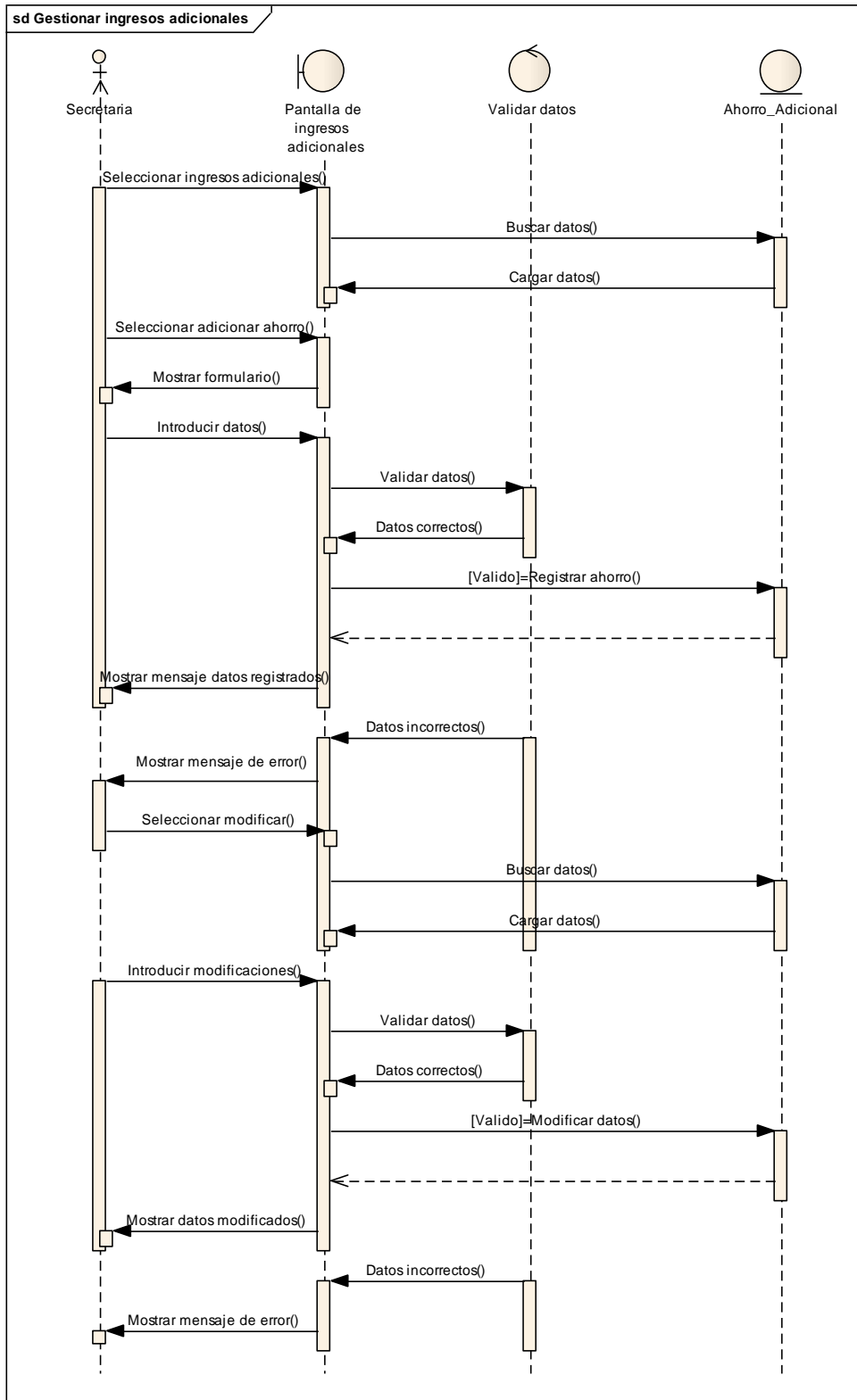
3.3.15 Diagrama de Secuencia: Generar Reporte de Licencias y permisos



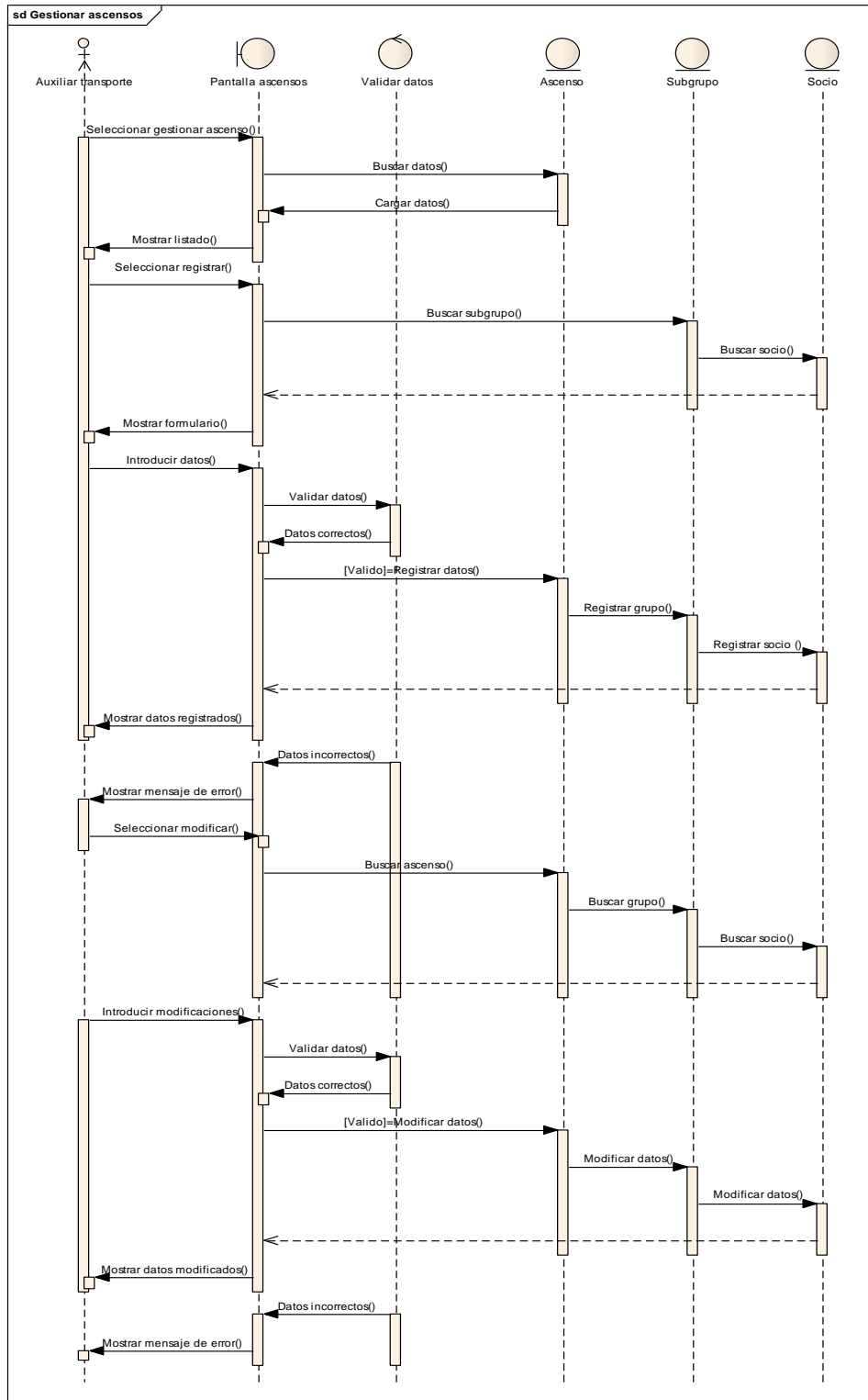
3.3.16 Diagrama de Secuencia: Generar Reporte de Ingresos



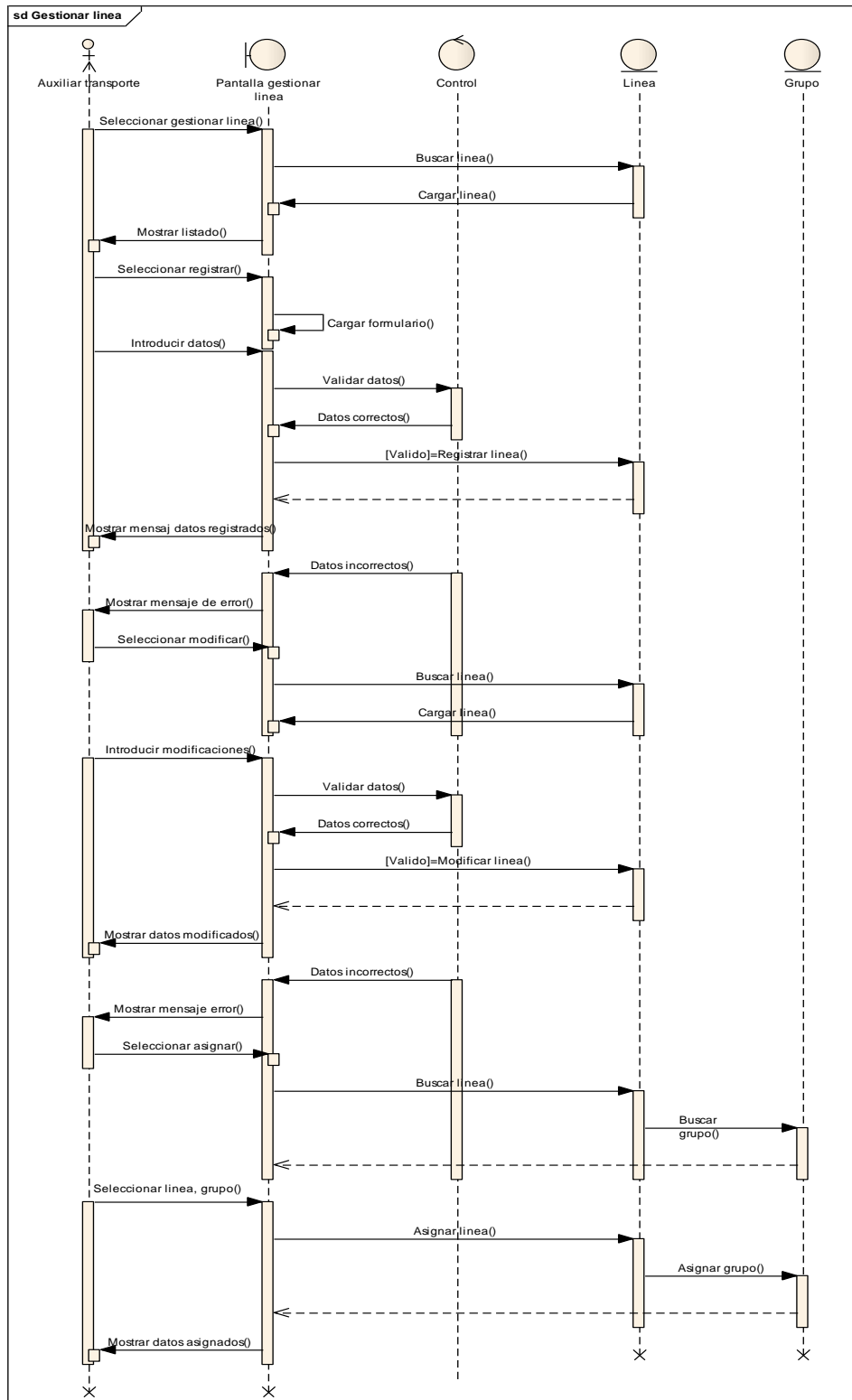
3.3.17 Diagrama de Secuencia: Gestionar Ingresos Adicionales



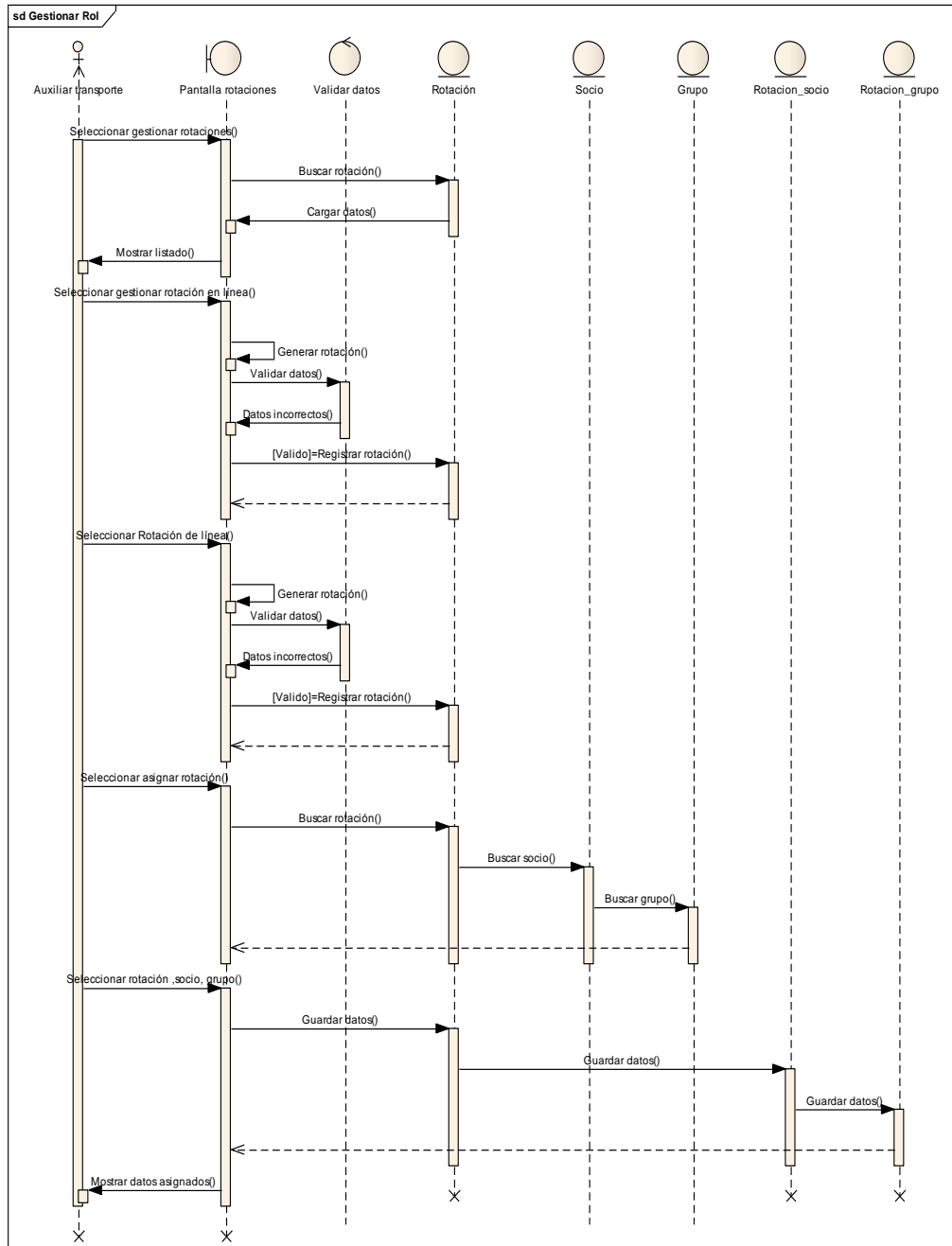
3.3.19 Diagrama de Secuencia: Gestionar Ascensos



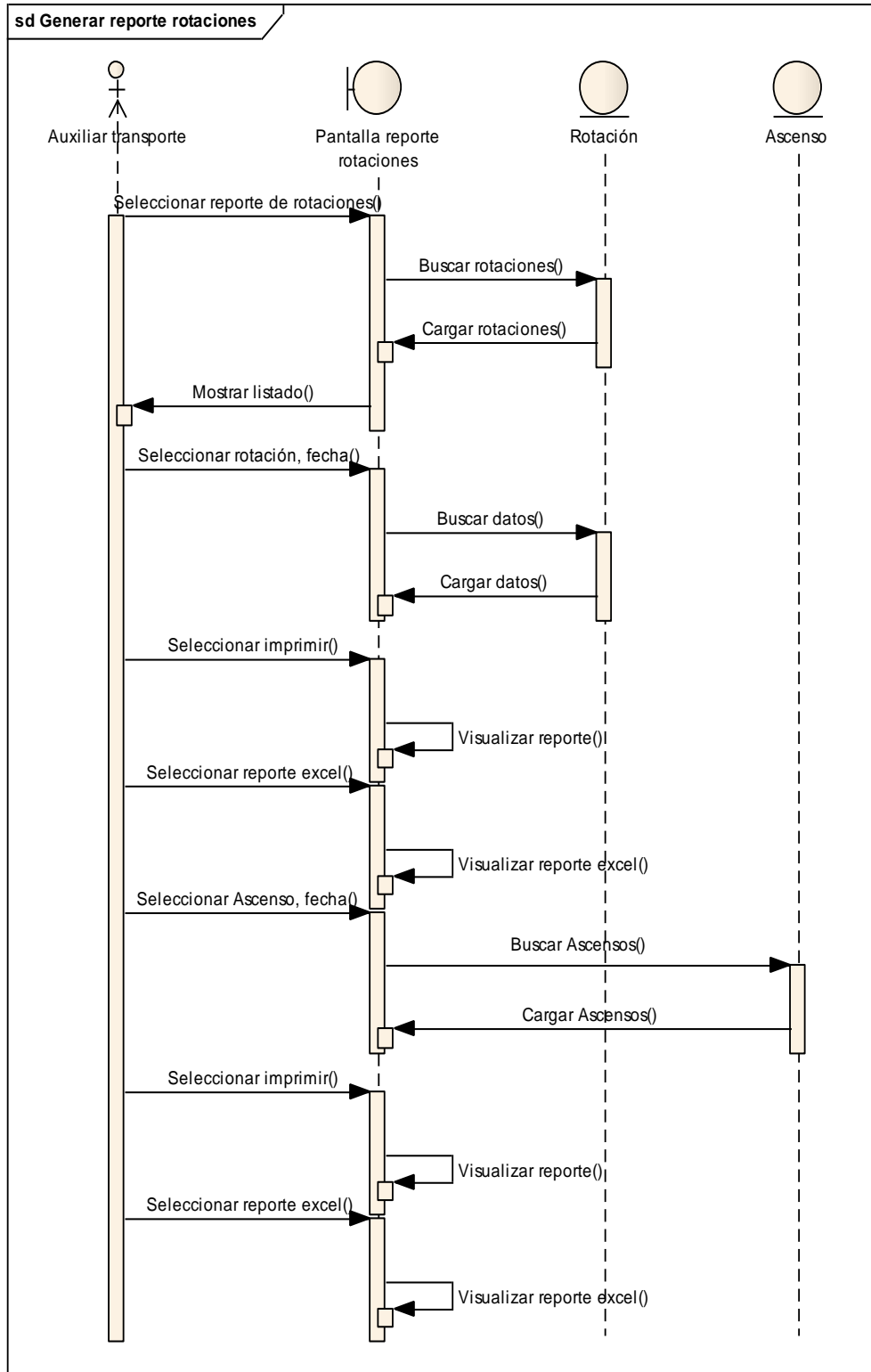
3.3.20 Diagrama de Secuencia: Gestionar Línea



3.3.21 Diagrama de Secuencia: Gestionar Rol

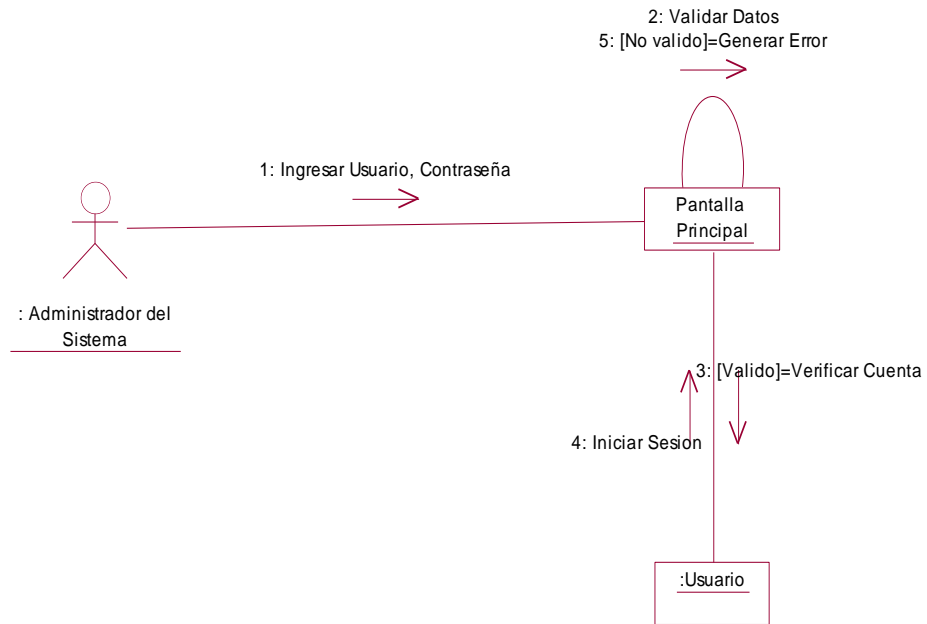


3.3.2 Diagrama de Secuencia: Generar Reporte Rotaciones

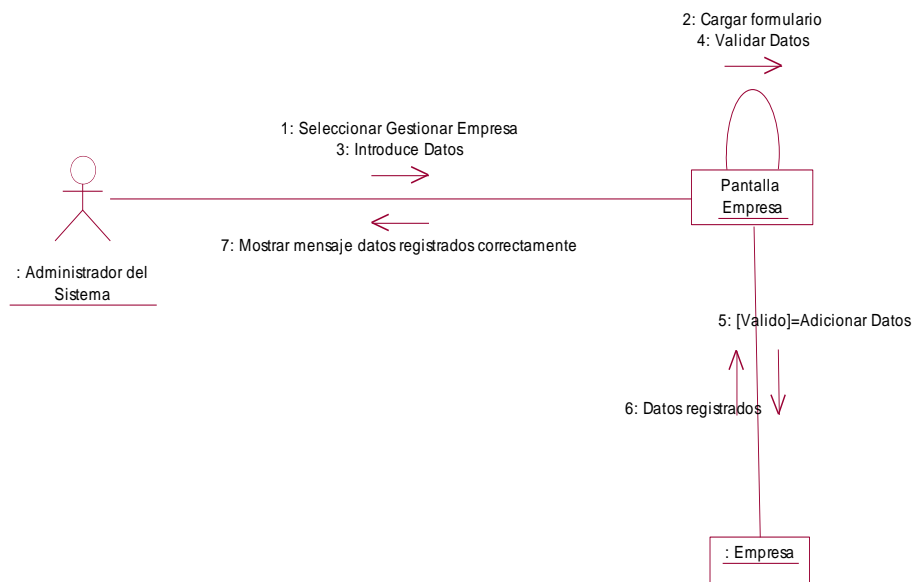


3.4 Diagramas de Colaboración

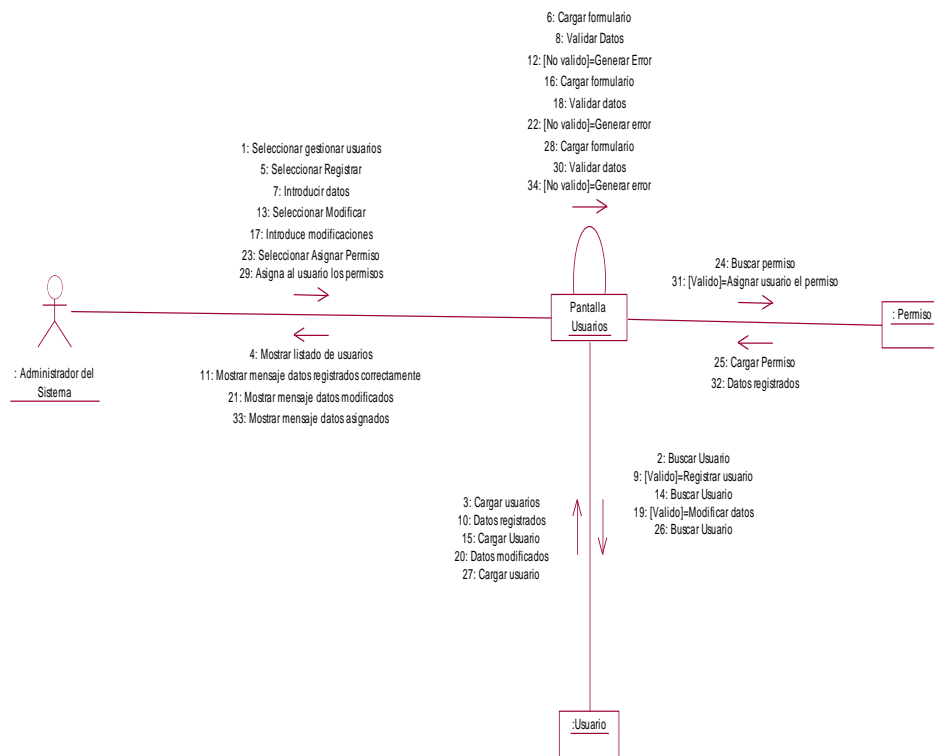
3.4.1 Diagrama de Colaboración: Conectarse



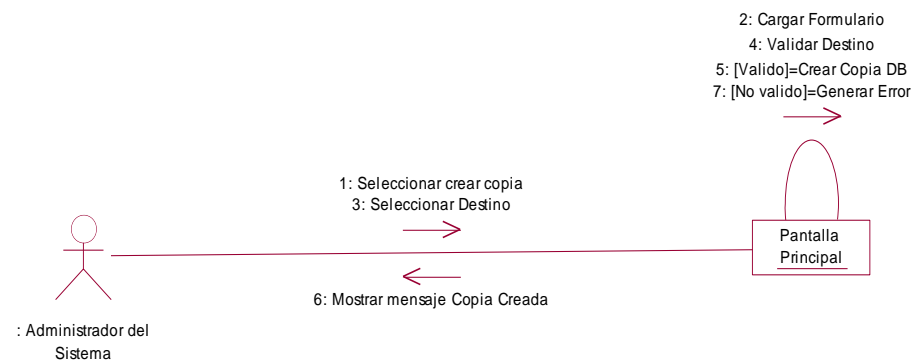
3.4.2 Diagrama de Colaboración: Gestionar Empresa



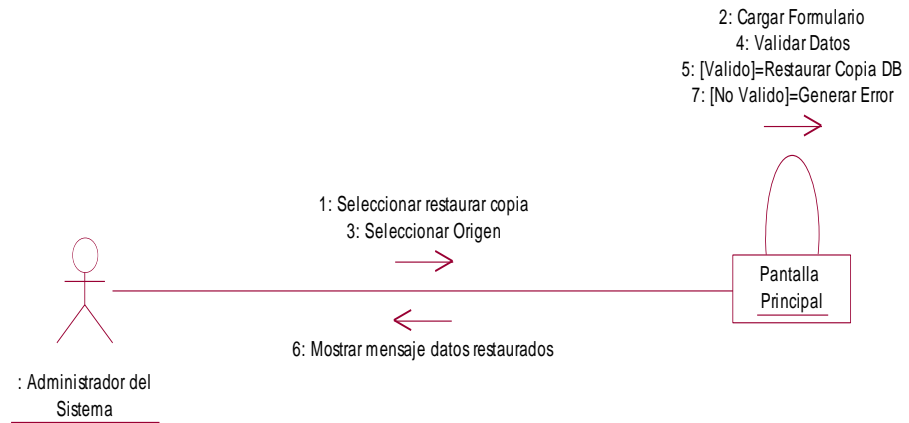
3.4.3 Diagrama de Colaboración: Gestionar Usuarios



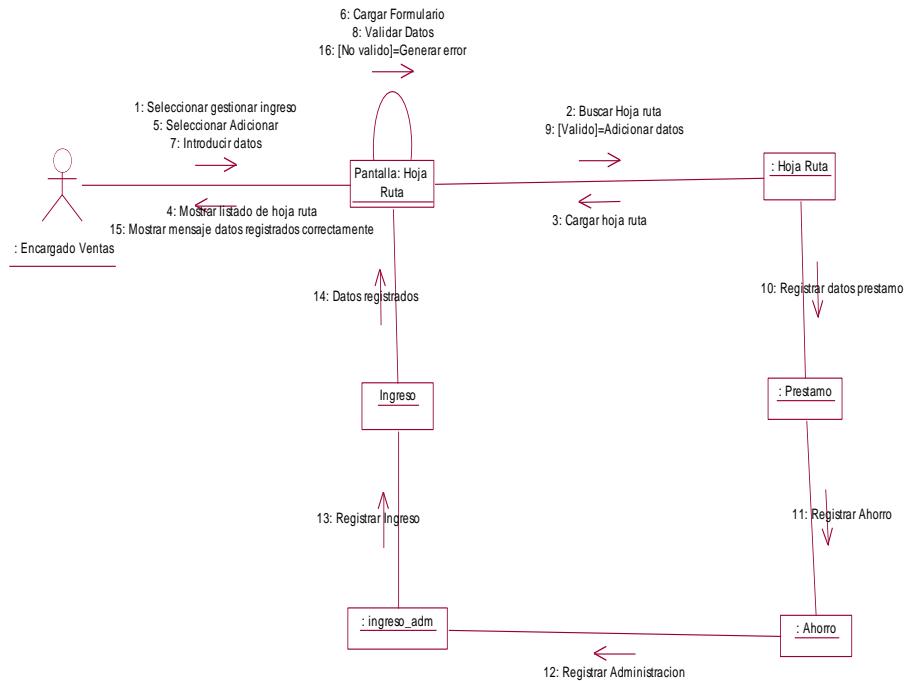
3.4.4 Diagrama de Colaboración: Crear Copias de Seguridad



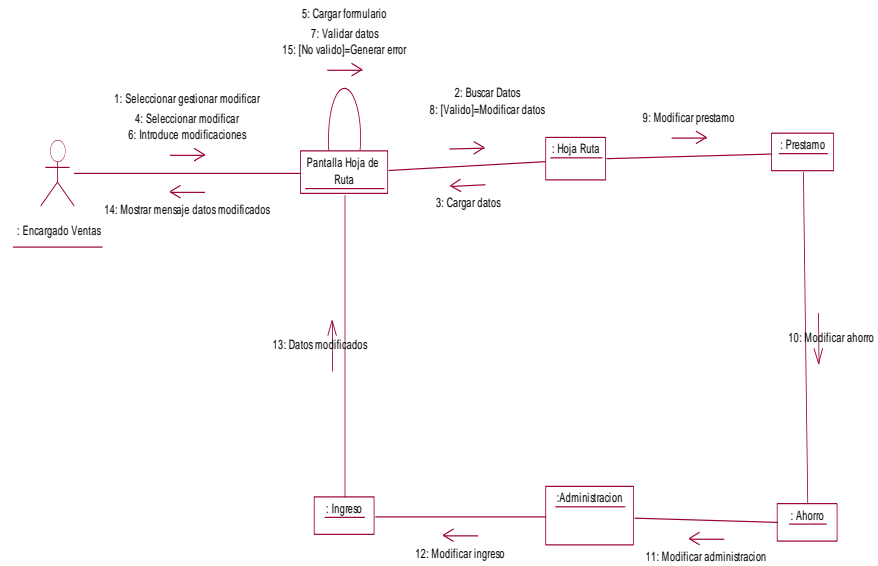
3.4.5 Diagrama de Colaboración: Restaurar Copias de Seguridad



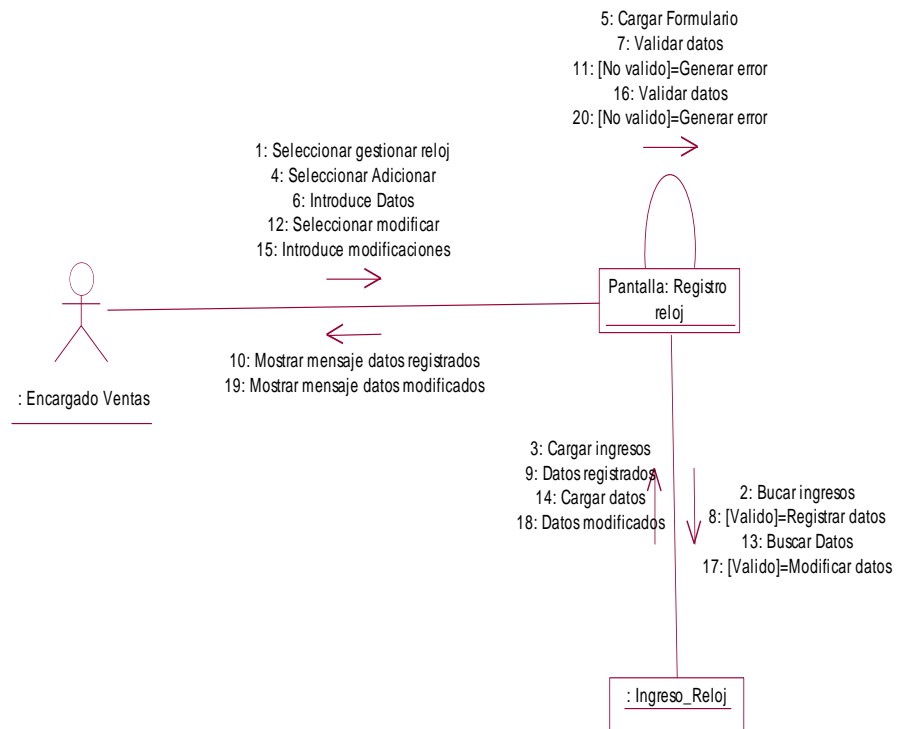
3.4.6 Diagrama de Colaboración: Gestionar Ingreso por Hoja de Ruta



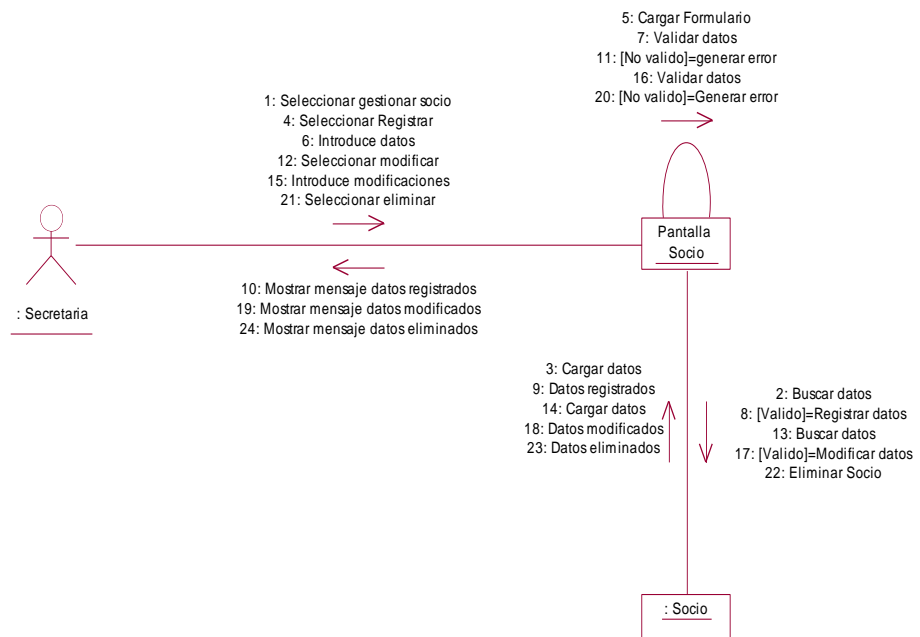
3.4.7 Diagrama de Colaboración: Gestionar Modificar por Hoja de Ruta



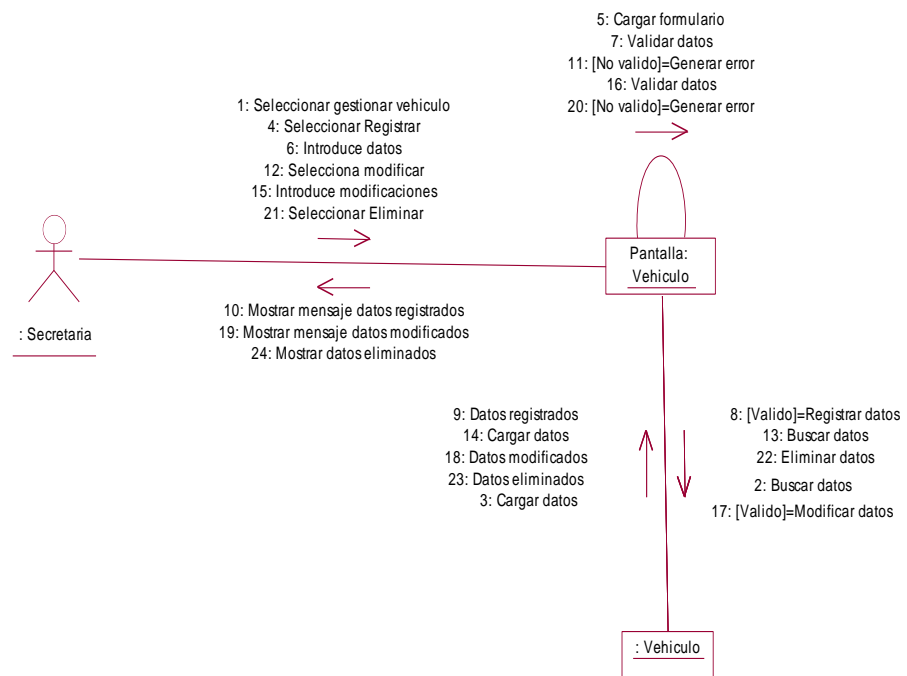
3.4.8 Diagrama de Colaboración: Gestionar Ingreso por Reloj



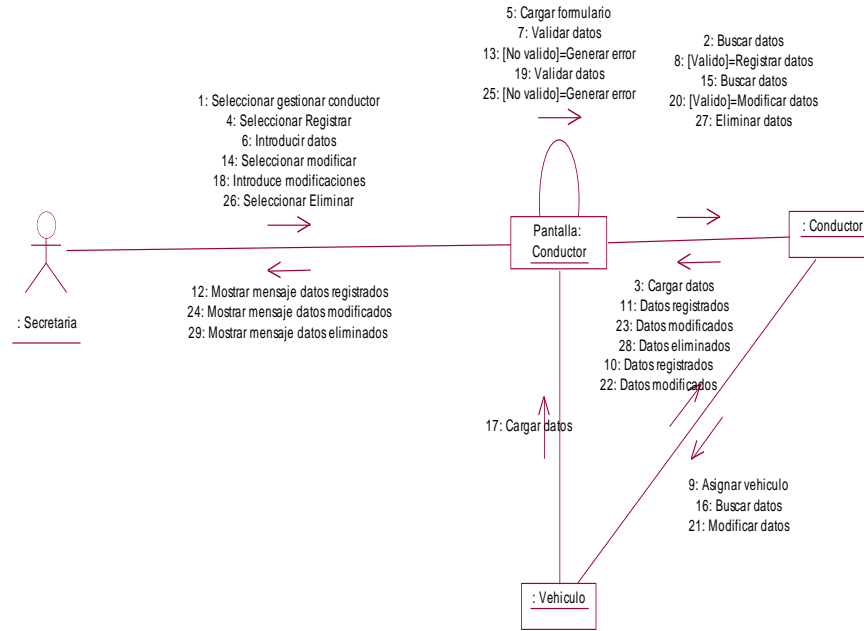
3.4.9 Diagrama de Colaboración: Gestionar Socio



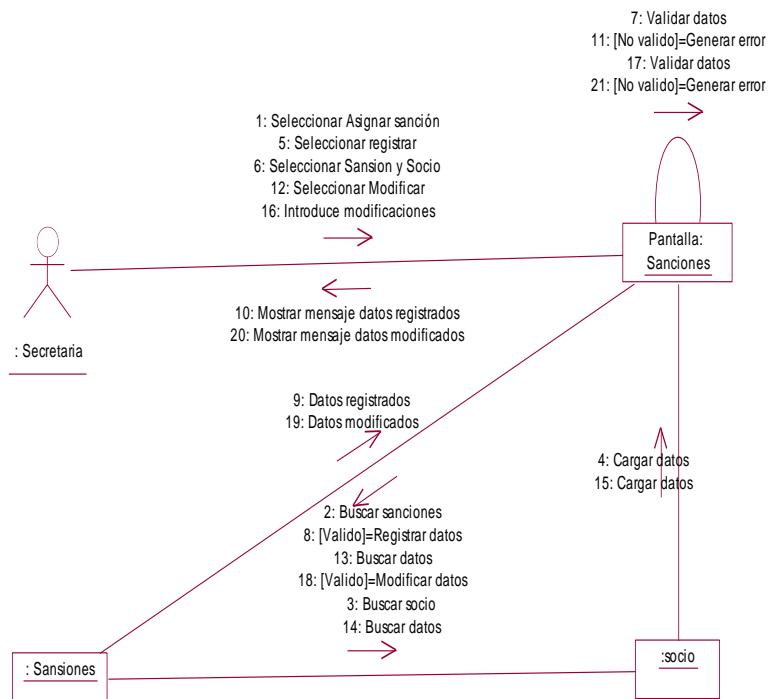
3.4.10 Diagrama de Colaboración: Gestionar Vehículo



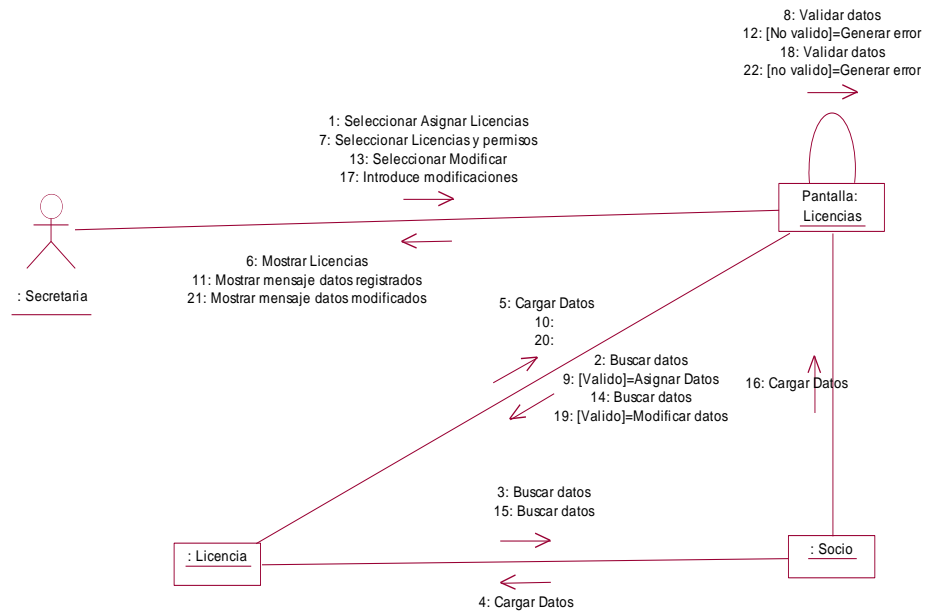
3.4.11 Diagrama de Colaboración: Gestionar Conductor



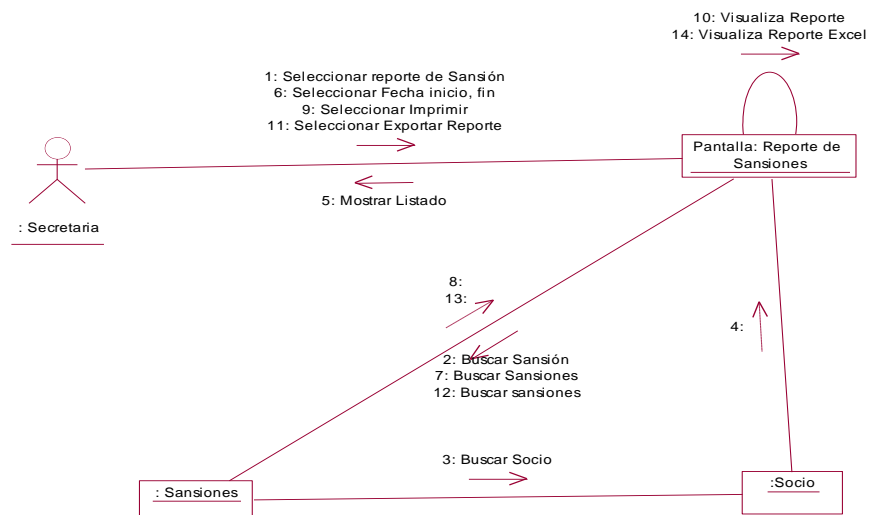
3.4.12 Diagrama de Colaboración: Gestionar Aplicar Sanciones



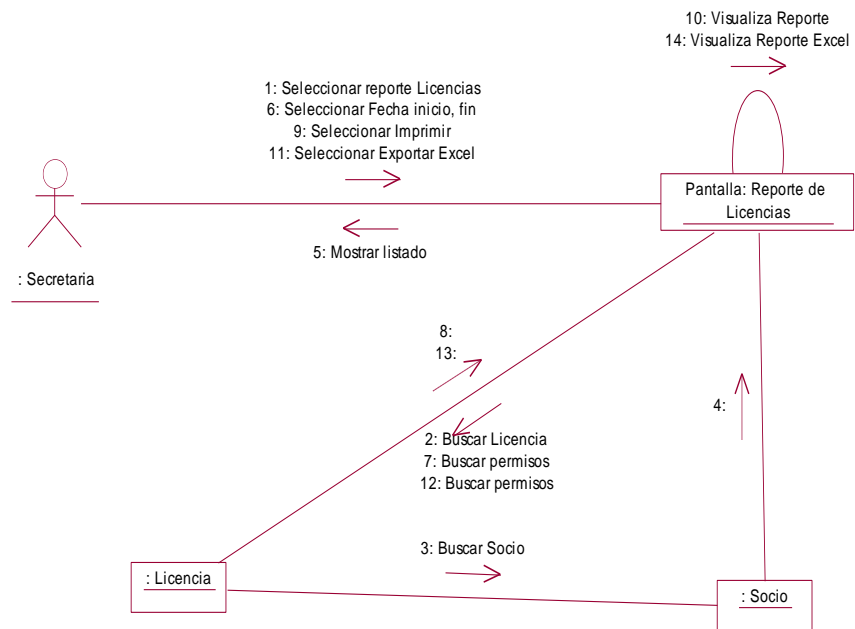
3.4.13 Diagrama de Colaboración: Gestionar Asignar Licencias y Permisos



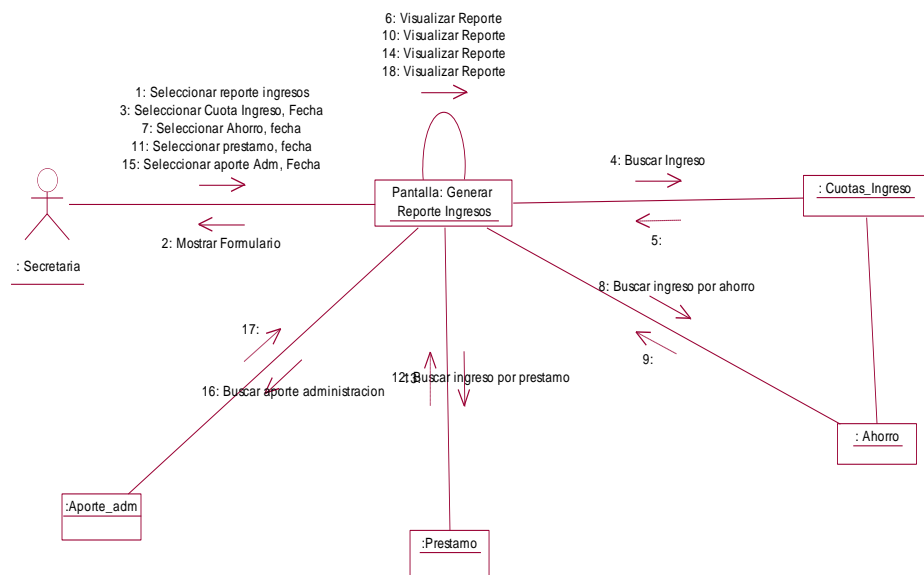
3.4.14 Diagrama de Colaboración: Generar Reporte de Sanciones



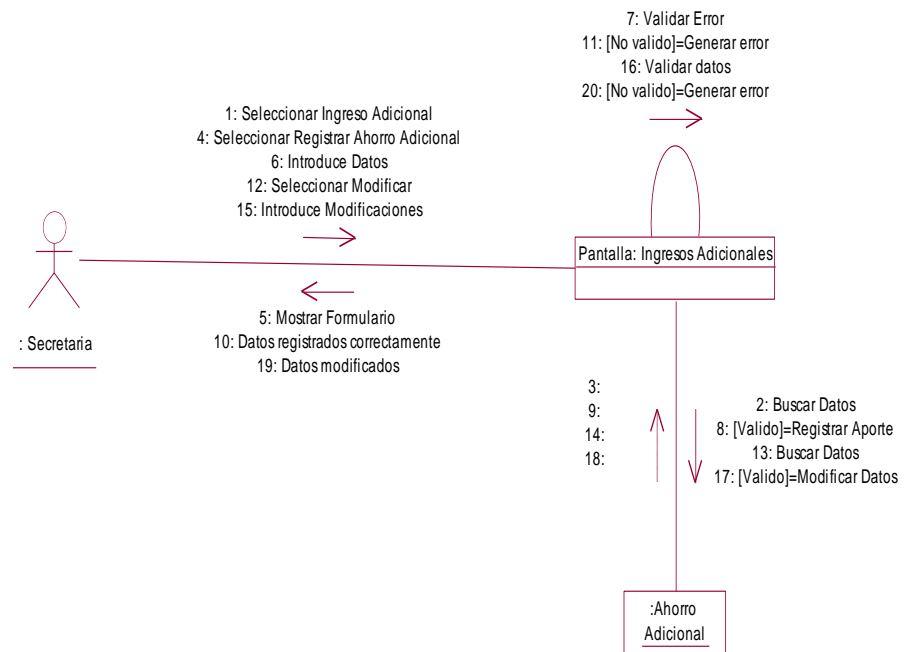
3.4.15 Diagrama de Colaboración: Generar Reporte de Licencias y permisos



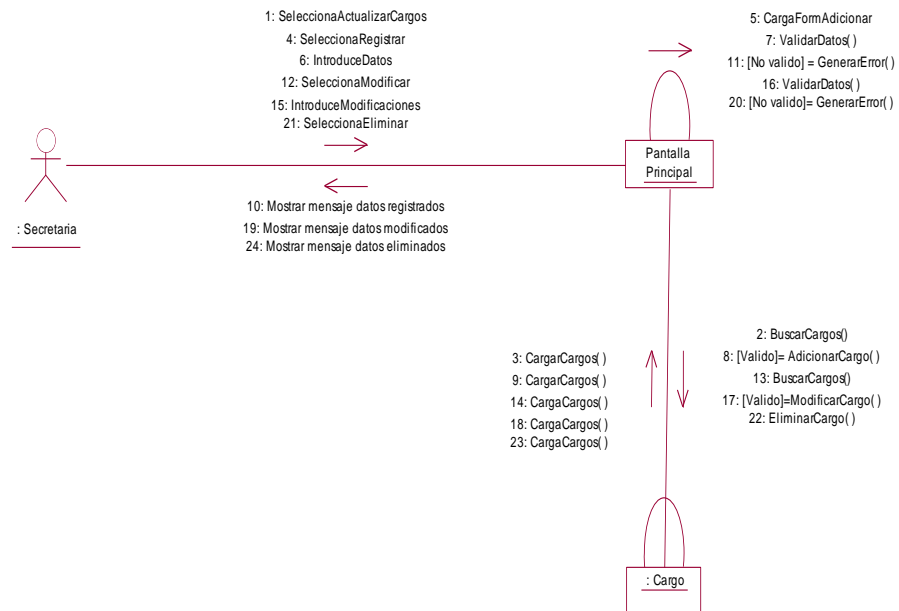
3.4.16 Diagrama de Colaboración: Generar Reporte de Ingresos



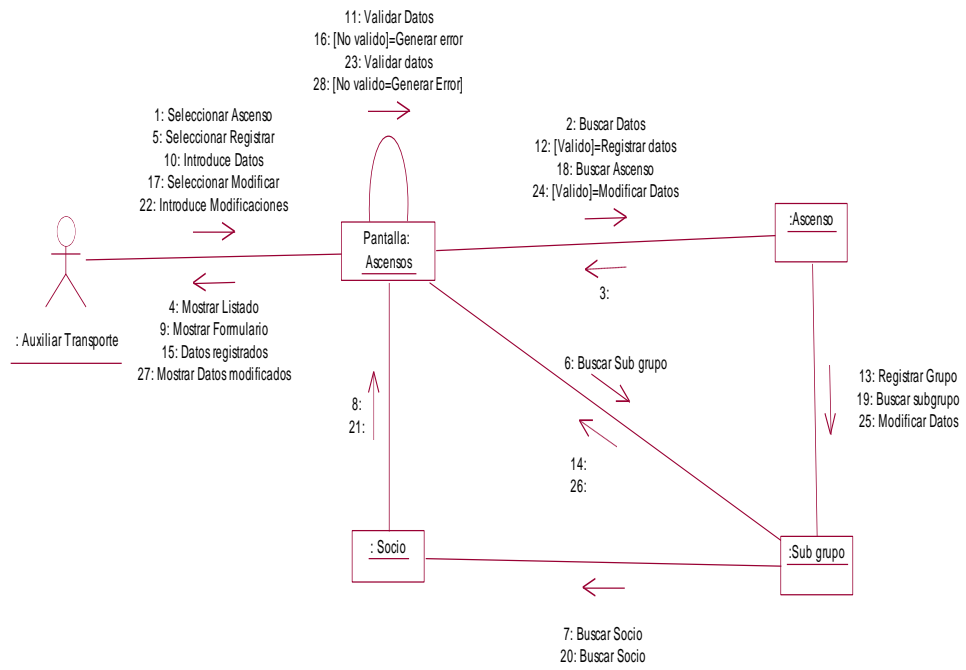
3.4.17 Diagrama de Colaboración: Gestionar Ingresos Adicionales



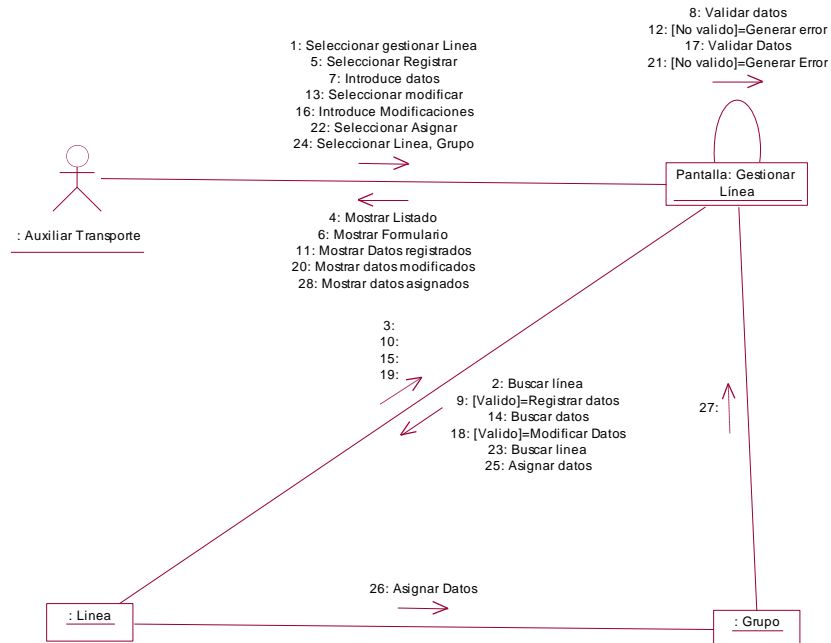
3.4.18 Diagrama de Colaboración: Gestionar Cargos



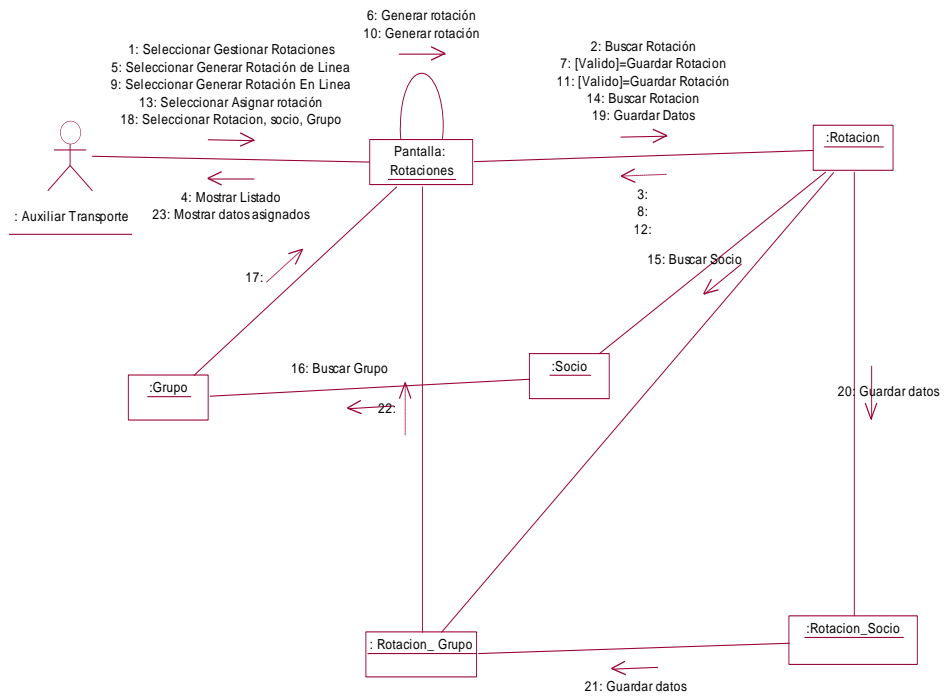
3.4.19 Diagrama de Colaboración: Gestionar Ascensos



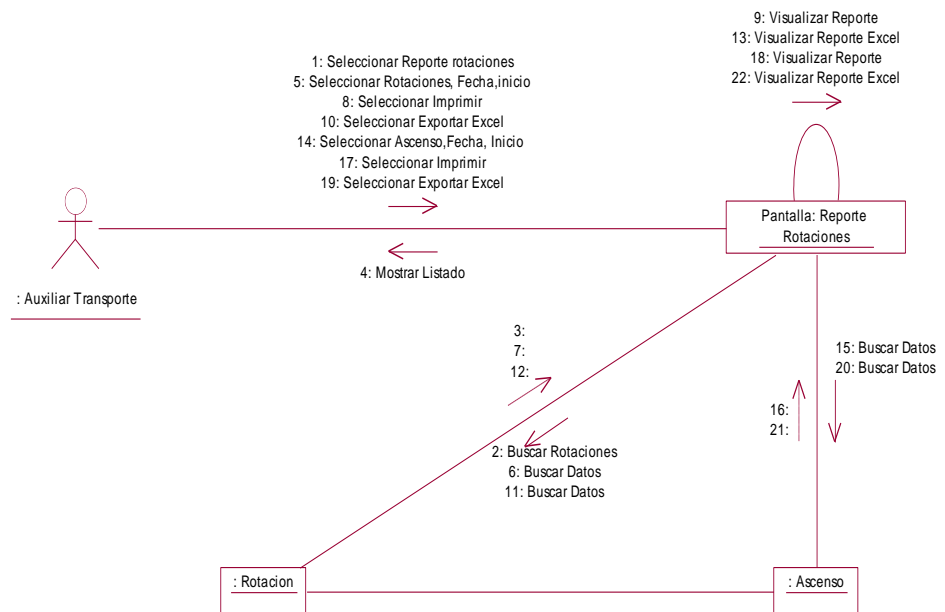
3.4.20 Diagrama de Colaboración: Gestionar Línea



3.4.21 Diagrama de Colaboración: Gestionar Rol

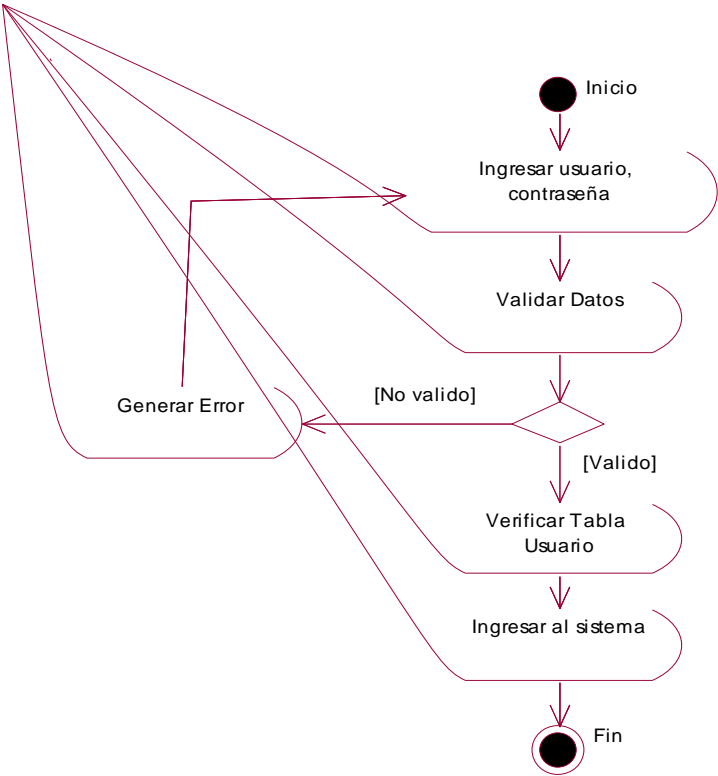


3.4.22 Diagrama de Colaboración: Generar Reporte Rotaciones

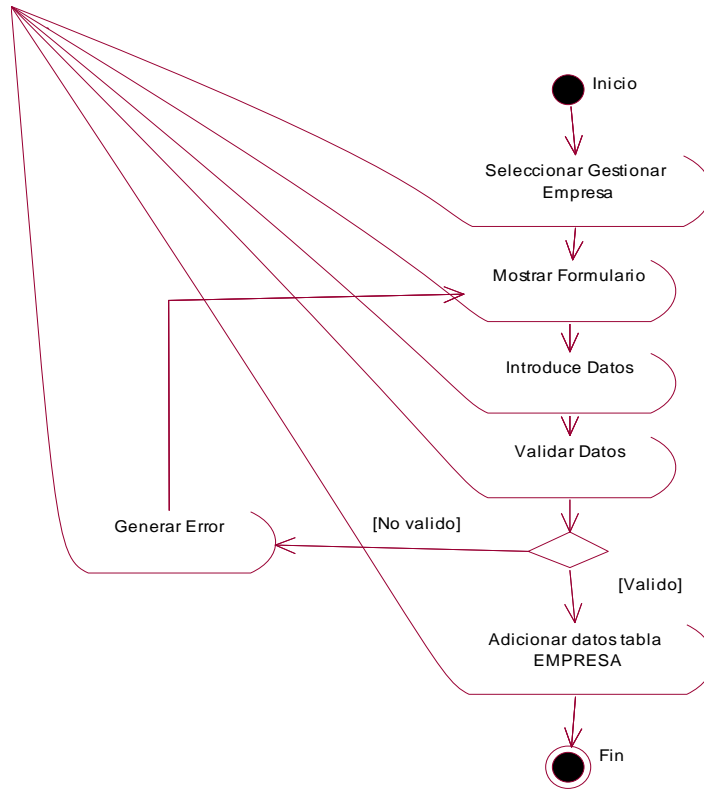


3.5 Diagramas de Actividades

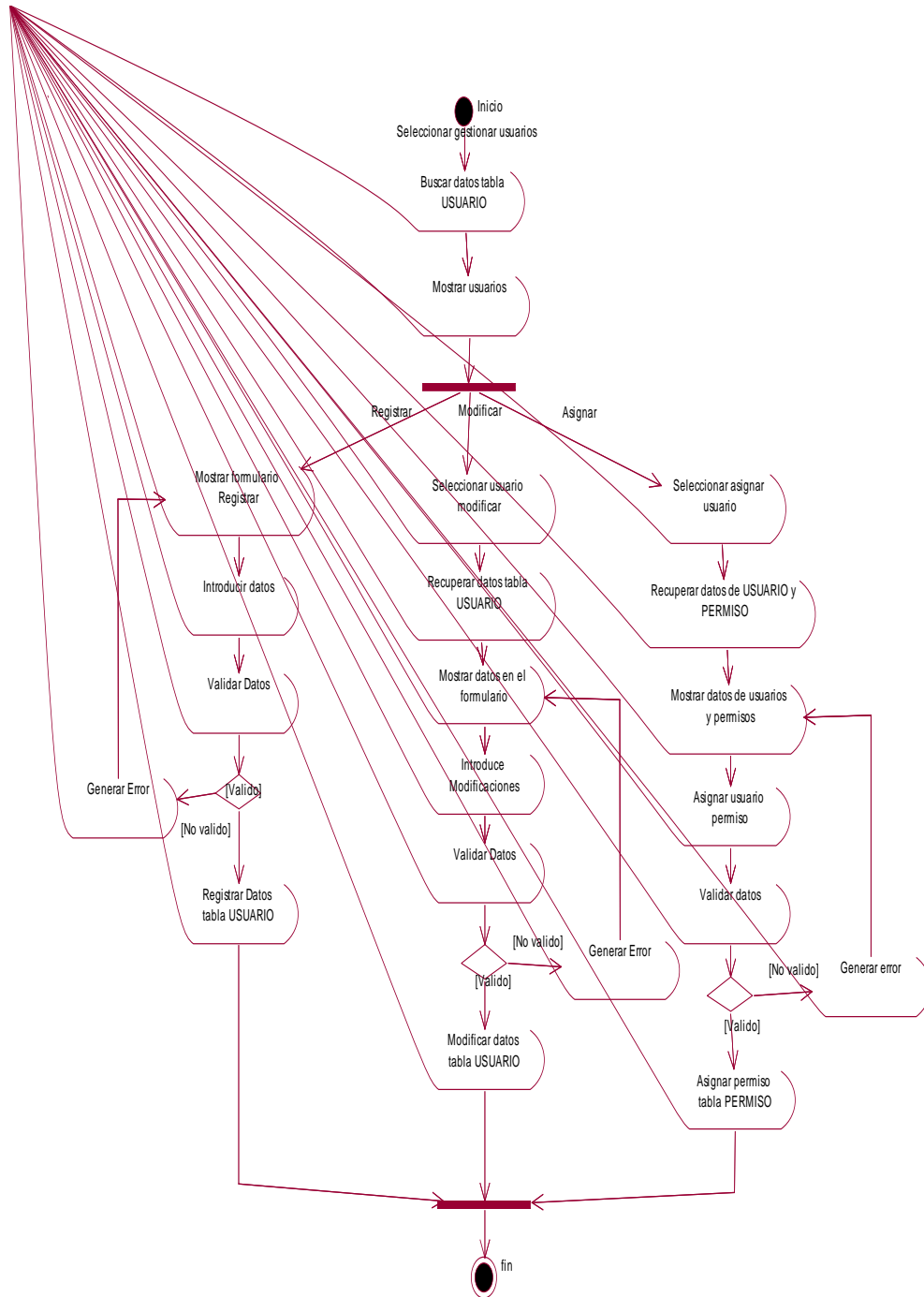
3.5.1 Diagrama de Actividades: Conectarse



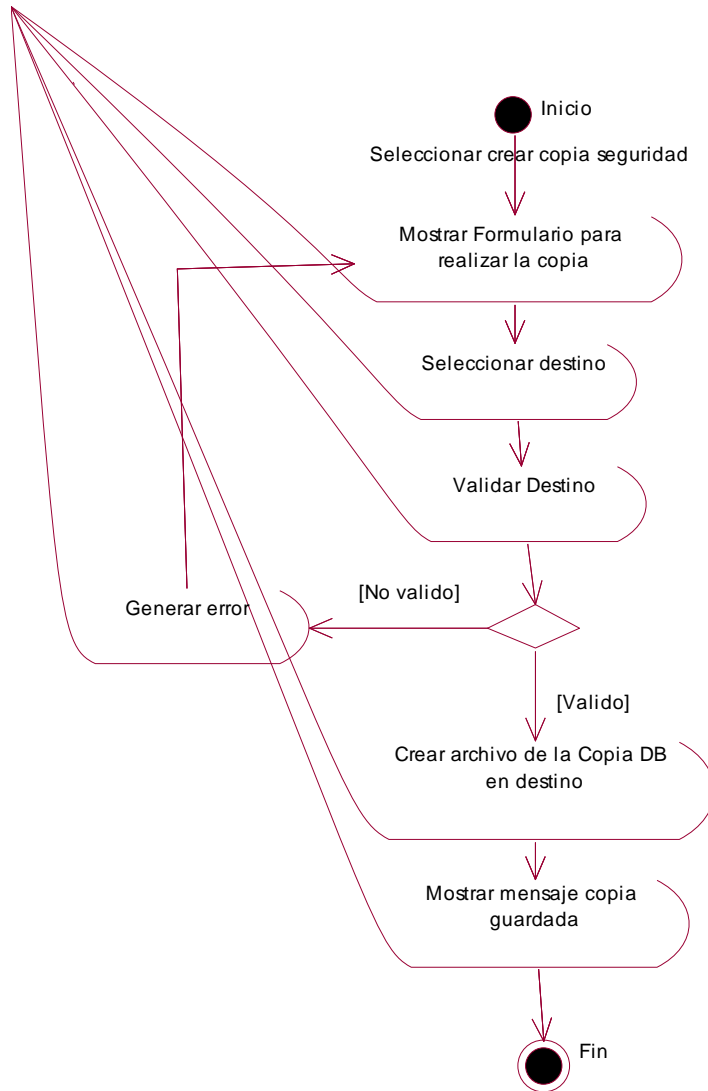
3.5.2 Diagrama de Actividades: Gestionar Empresa



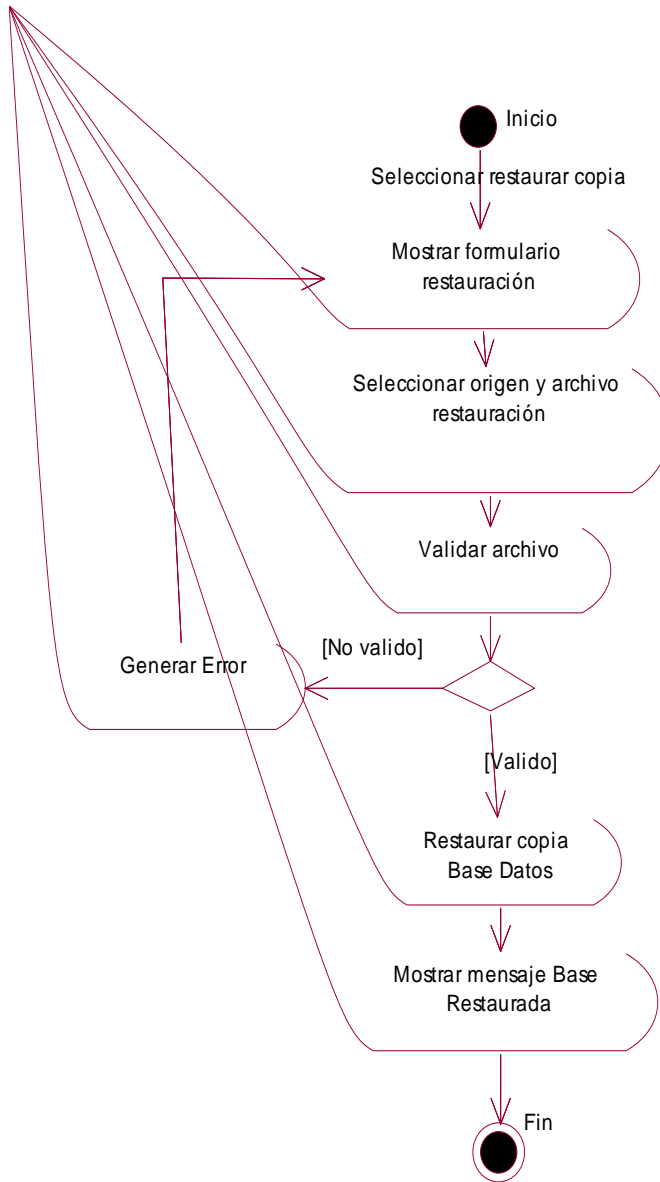
3.5.3 Diagrama de Actividades: Gestionar Usuarios



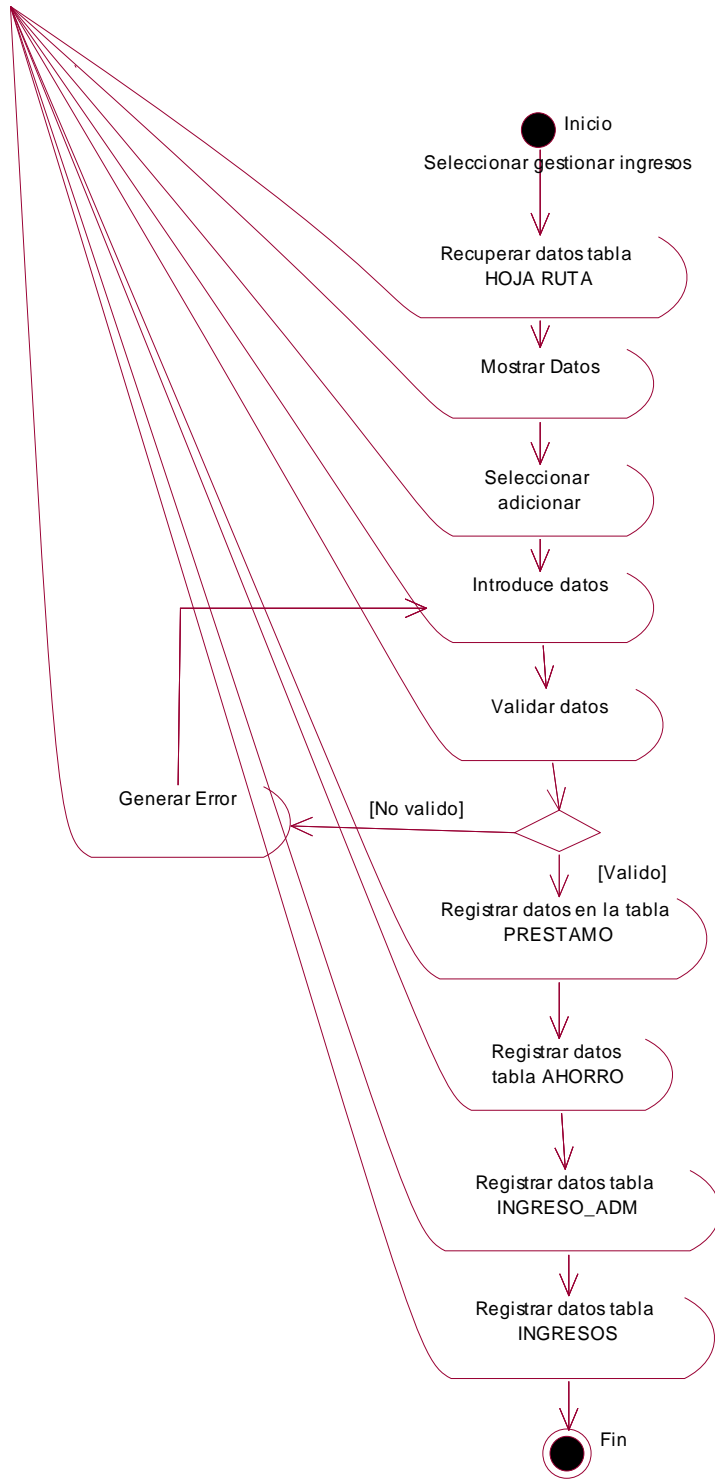
3.5.4 Diagrama de Actividades: Crear Copias de Seguridad



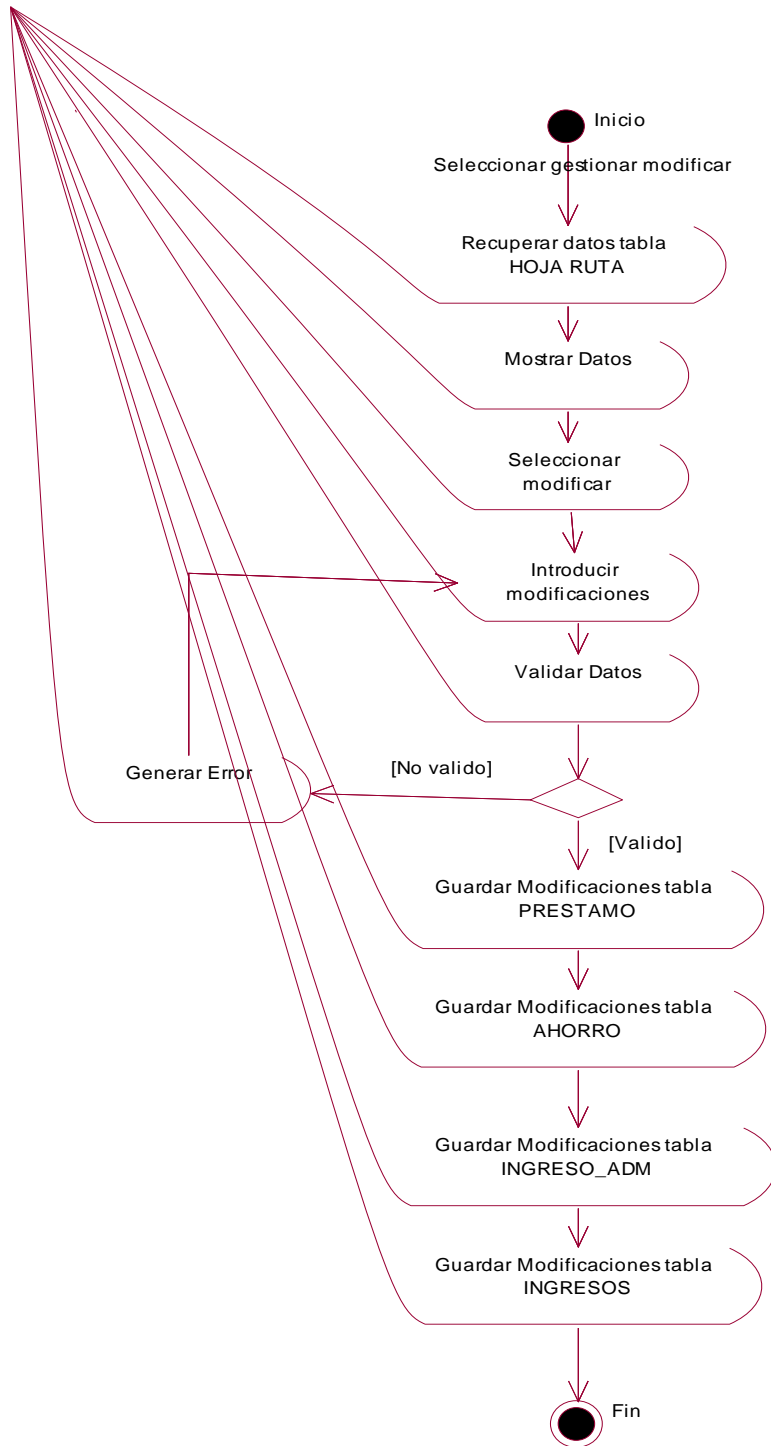
3.5.5 Diagrama de Actividades: Restaurar Copias de Seguridad



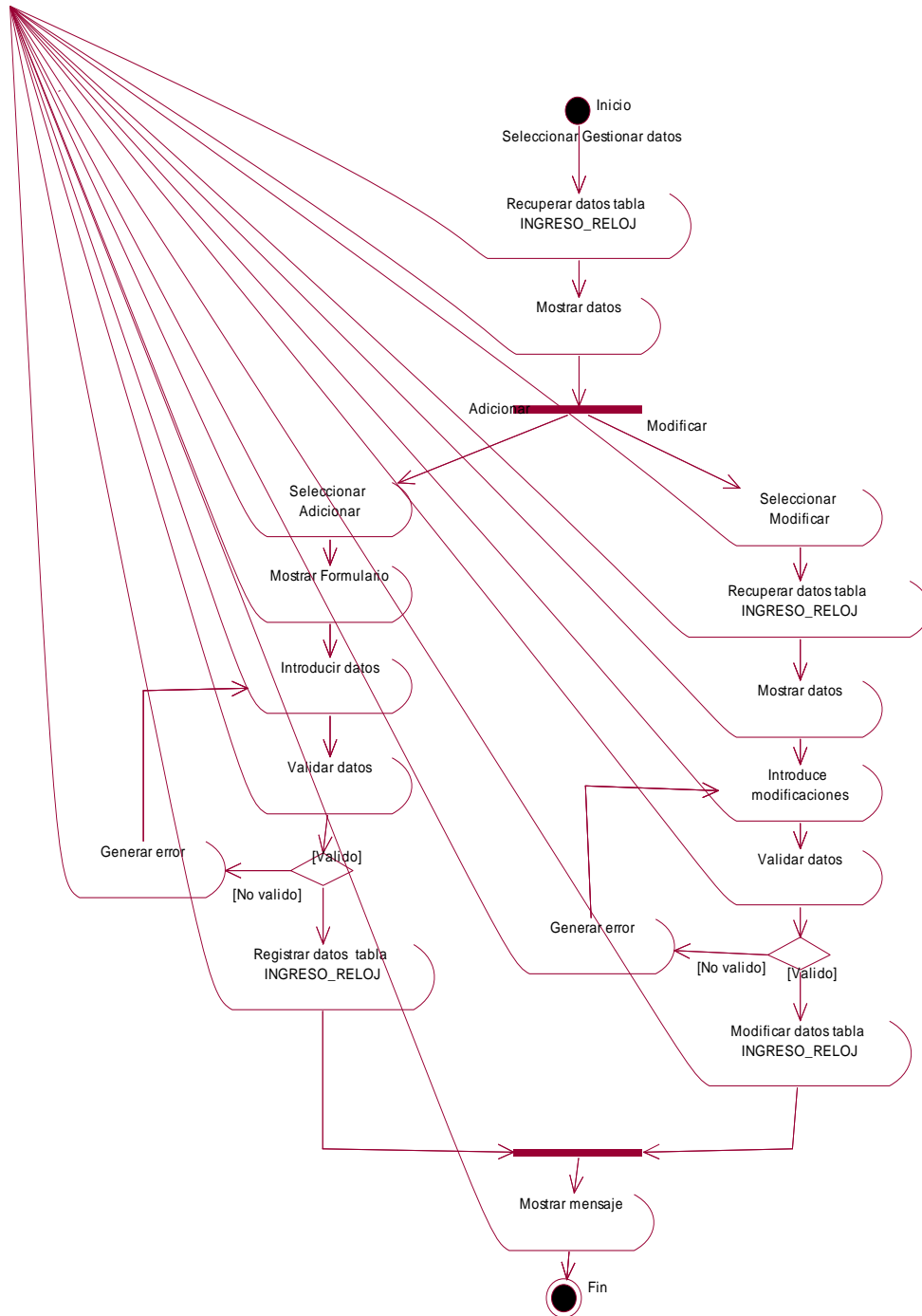
3.5.6 Diagrama de Actividades: Gestionar Ingreso por Hoja de Ruta



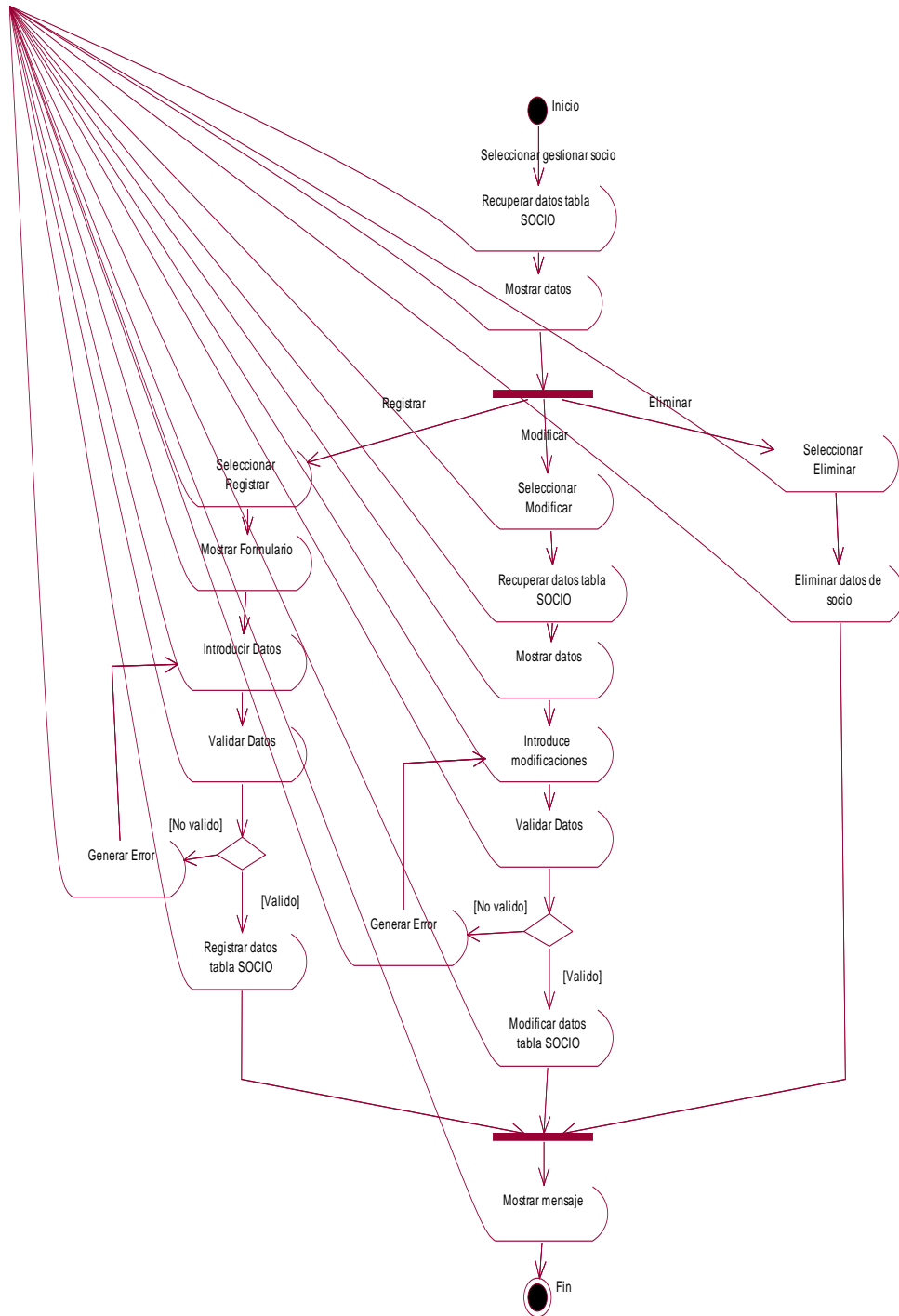
3.5.7 Diagrama de Actividades: Gestionar Modificar por Hoja de Ruta



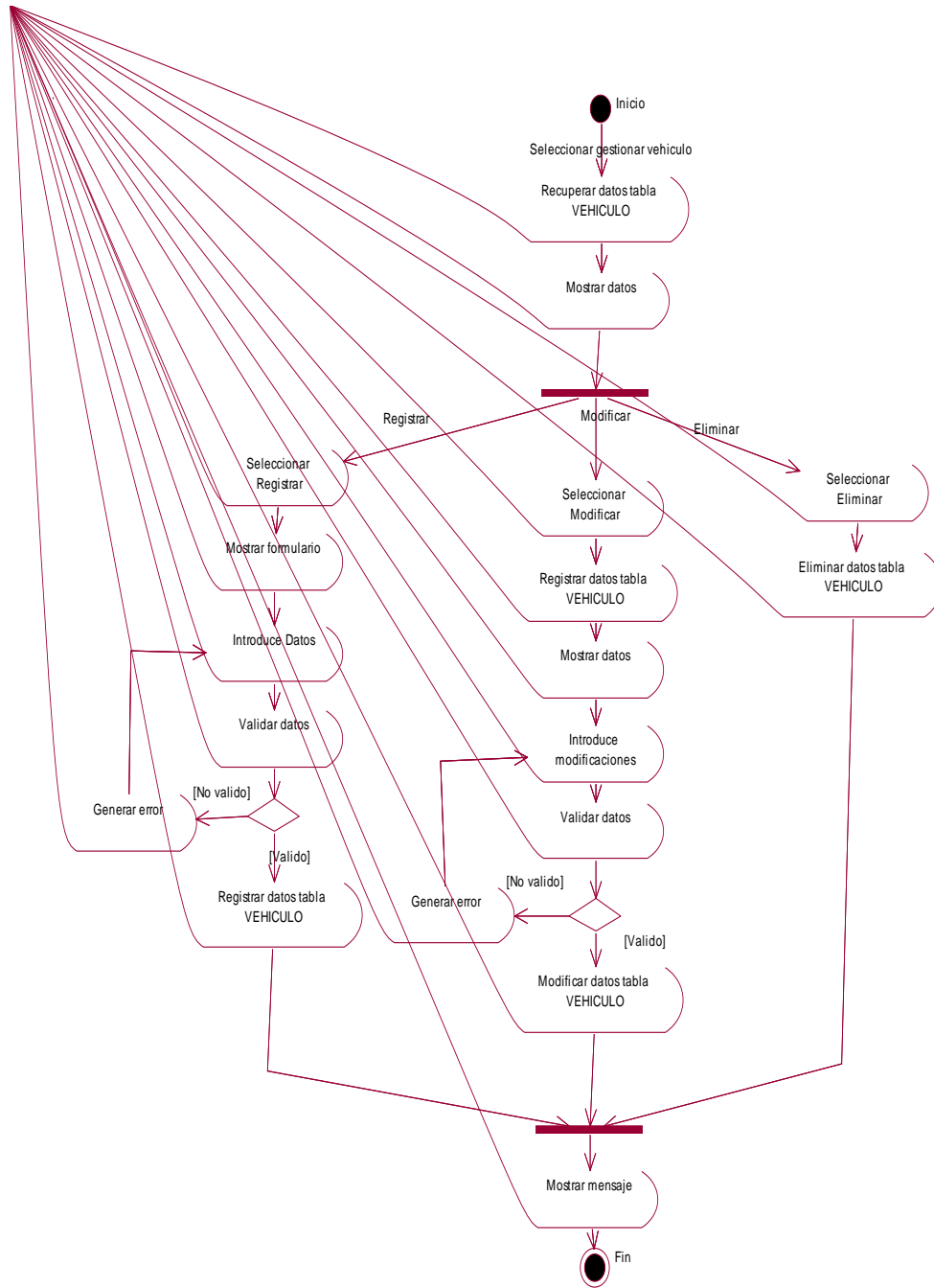
3.5.8 Diagrama de Actividades: Gestionar Ingreso por Reloj



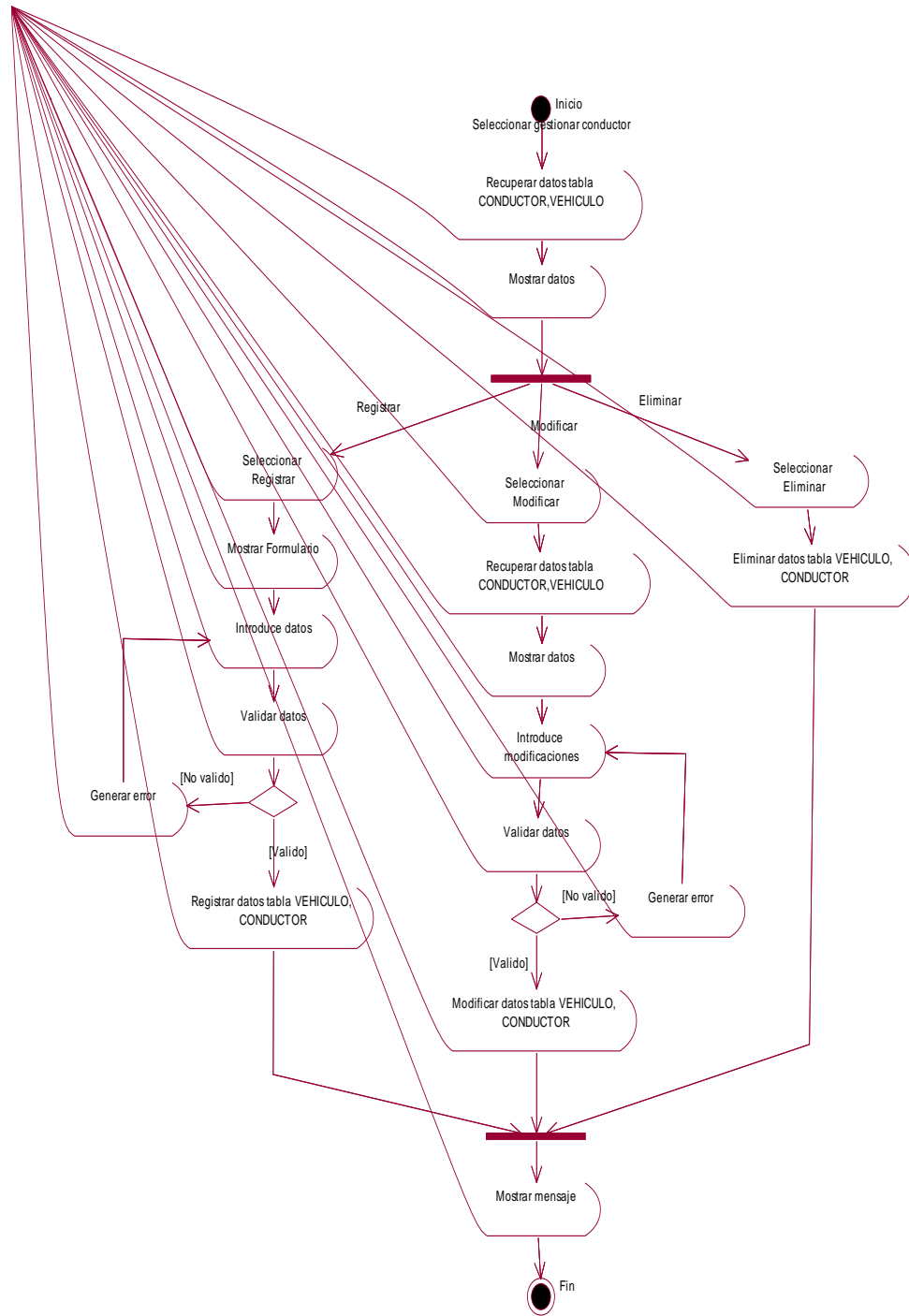
3.5.9 Diagrama de Actividades: Gestionar Socio



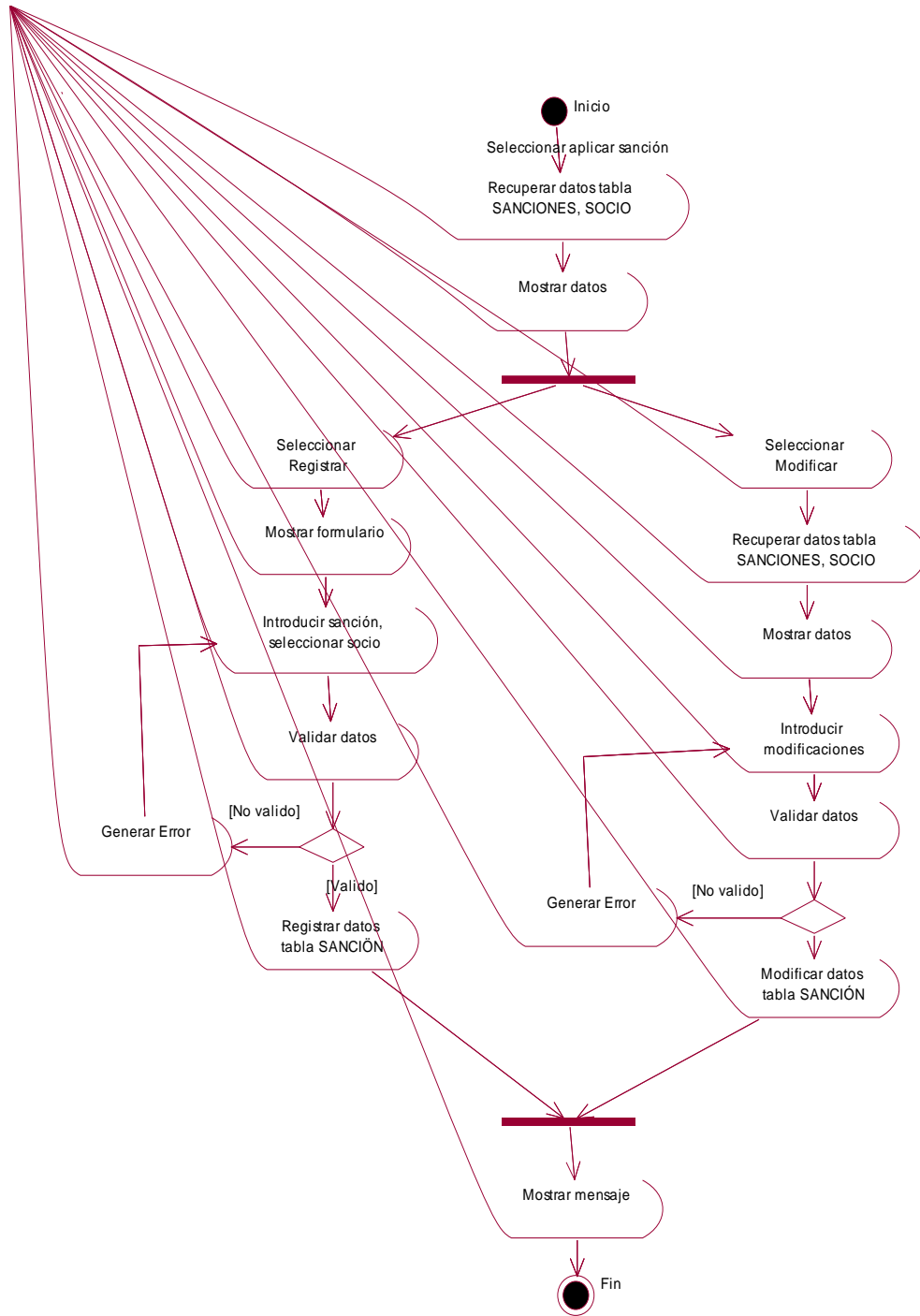
3.5.10 Diagrama de Actividades: Gestionar Vehículo



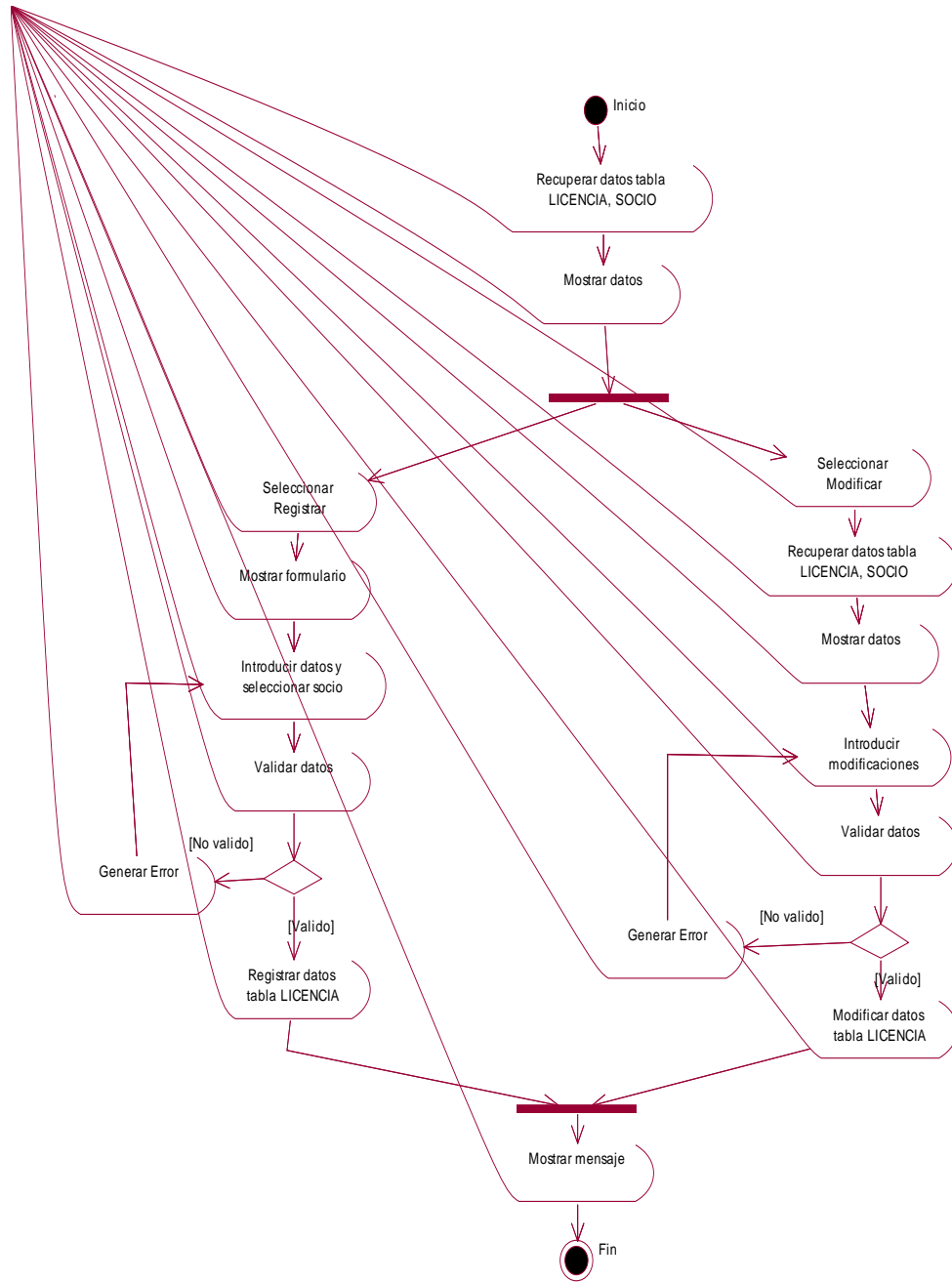
3.5.11 Diagrama de Actividades: Gestionar Conductor



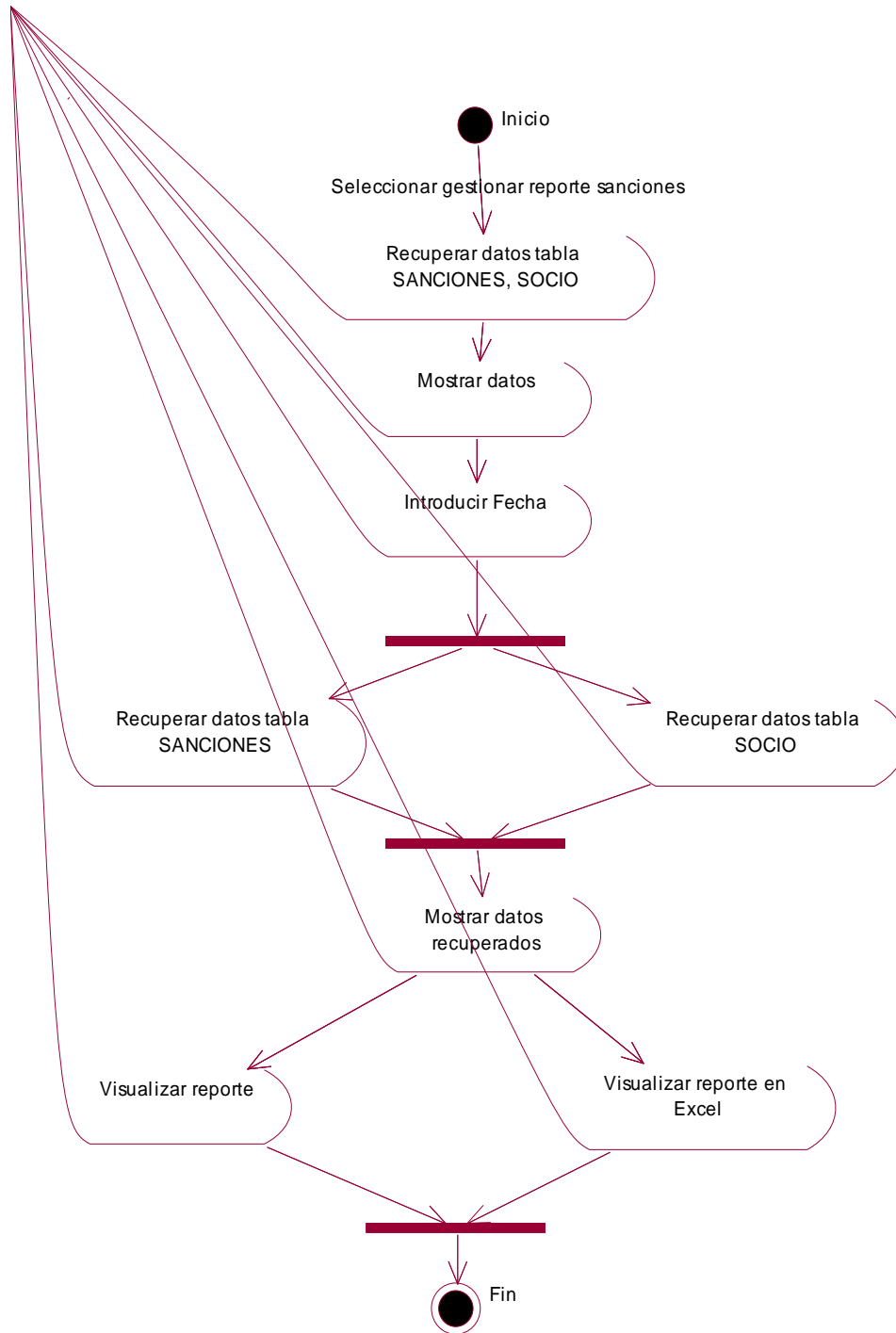
3.5.12 Diagrama de Actividades: Gestionar Aplicar Sanciones



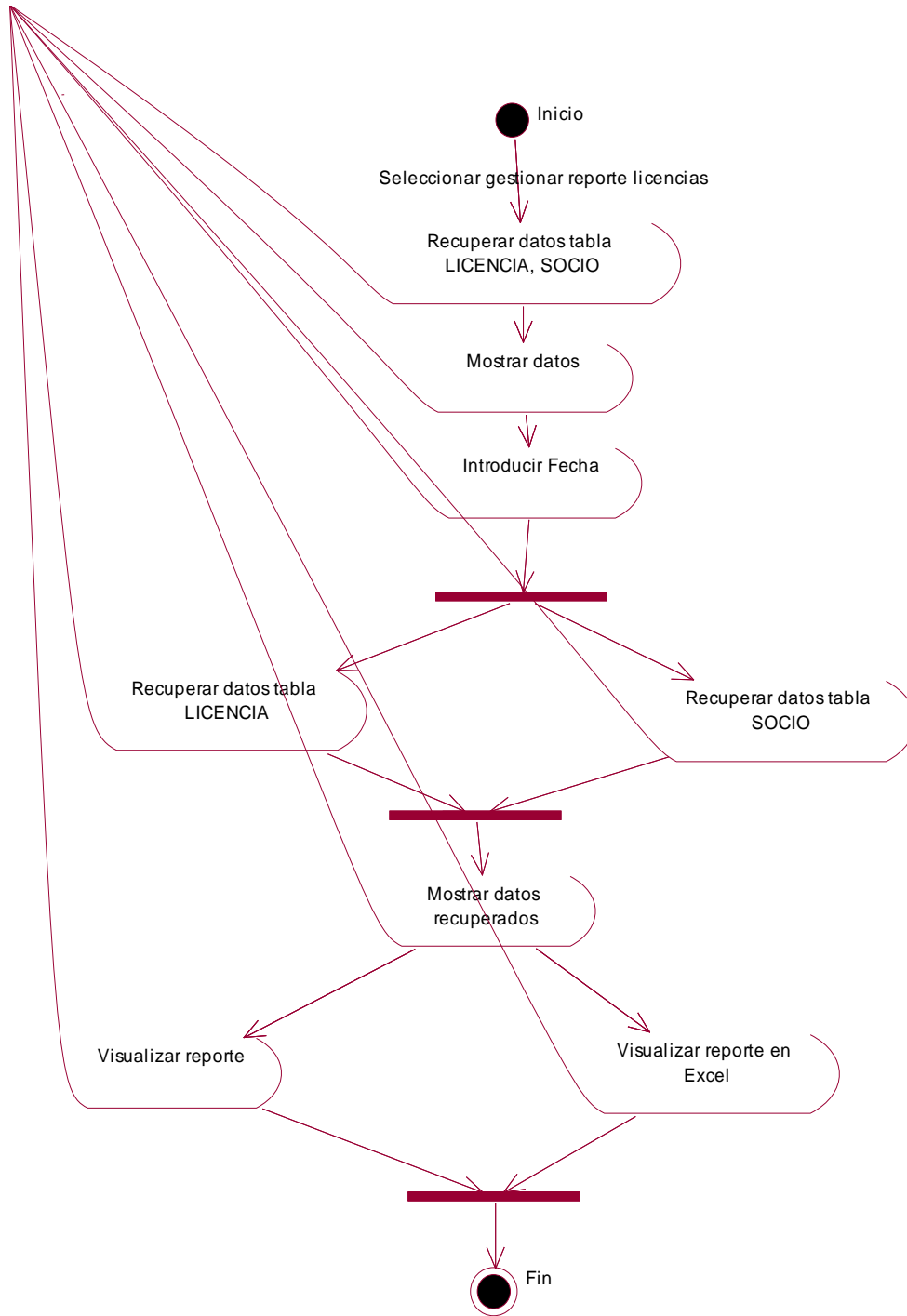
3.5.13 Diagrama de Actividades: Gestionar Asignar Licencias y Permisos



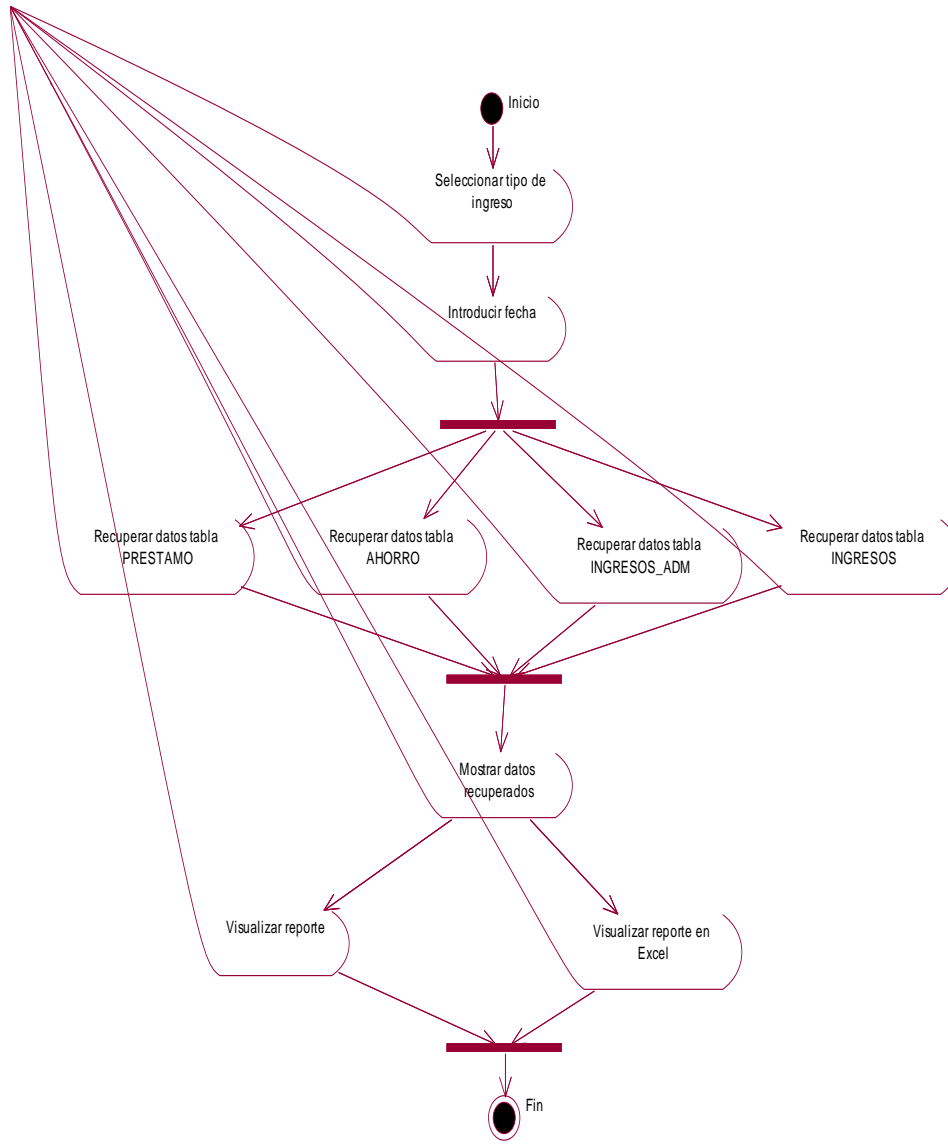
3.5.14 Diagrama de Actividades: Generar Reporte de Sanciones



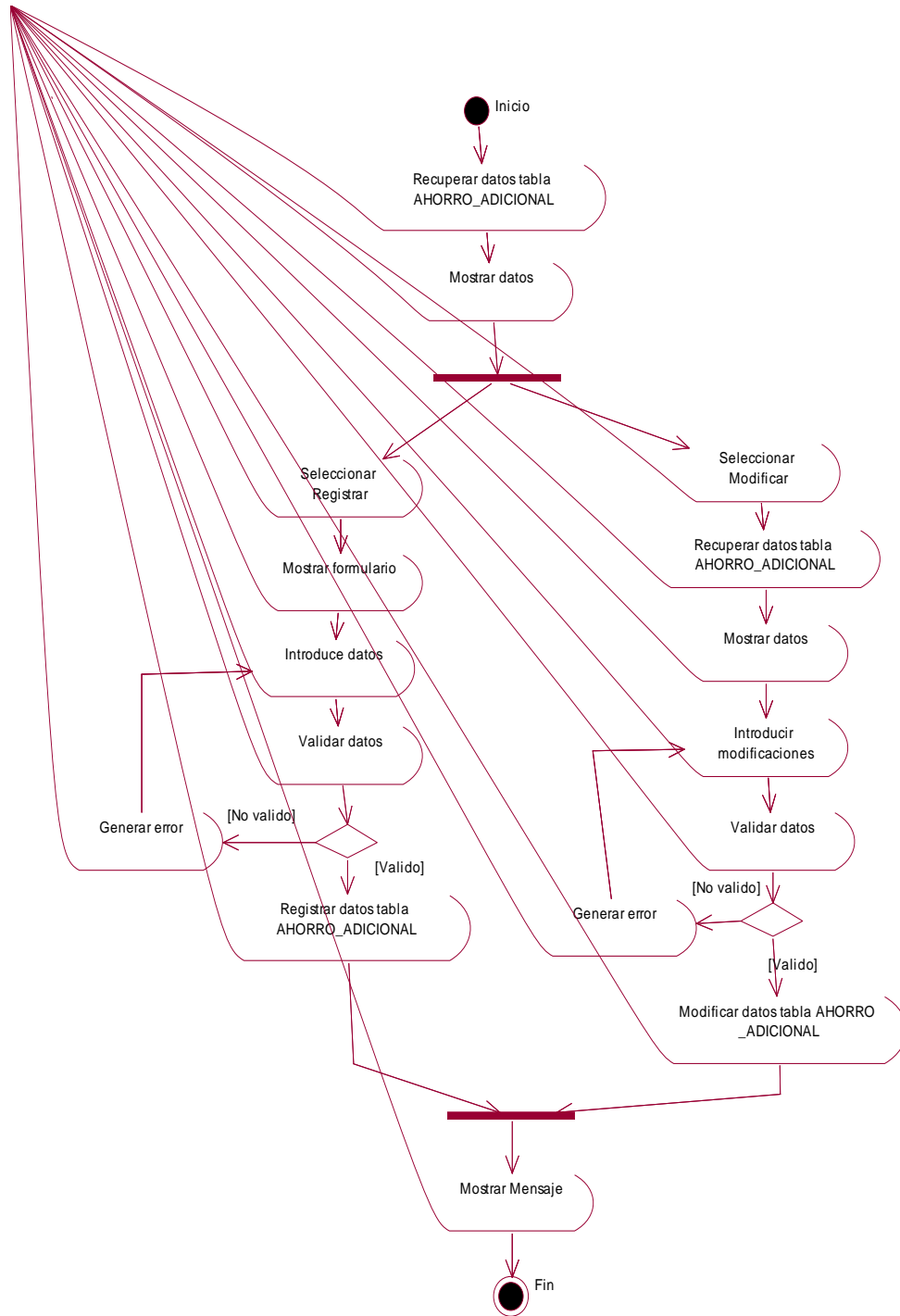
3.5.15 Diagrama de Actividades: Generar Reporte de Licencias y permisos



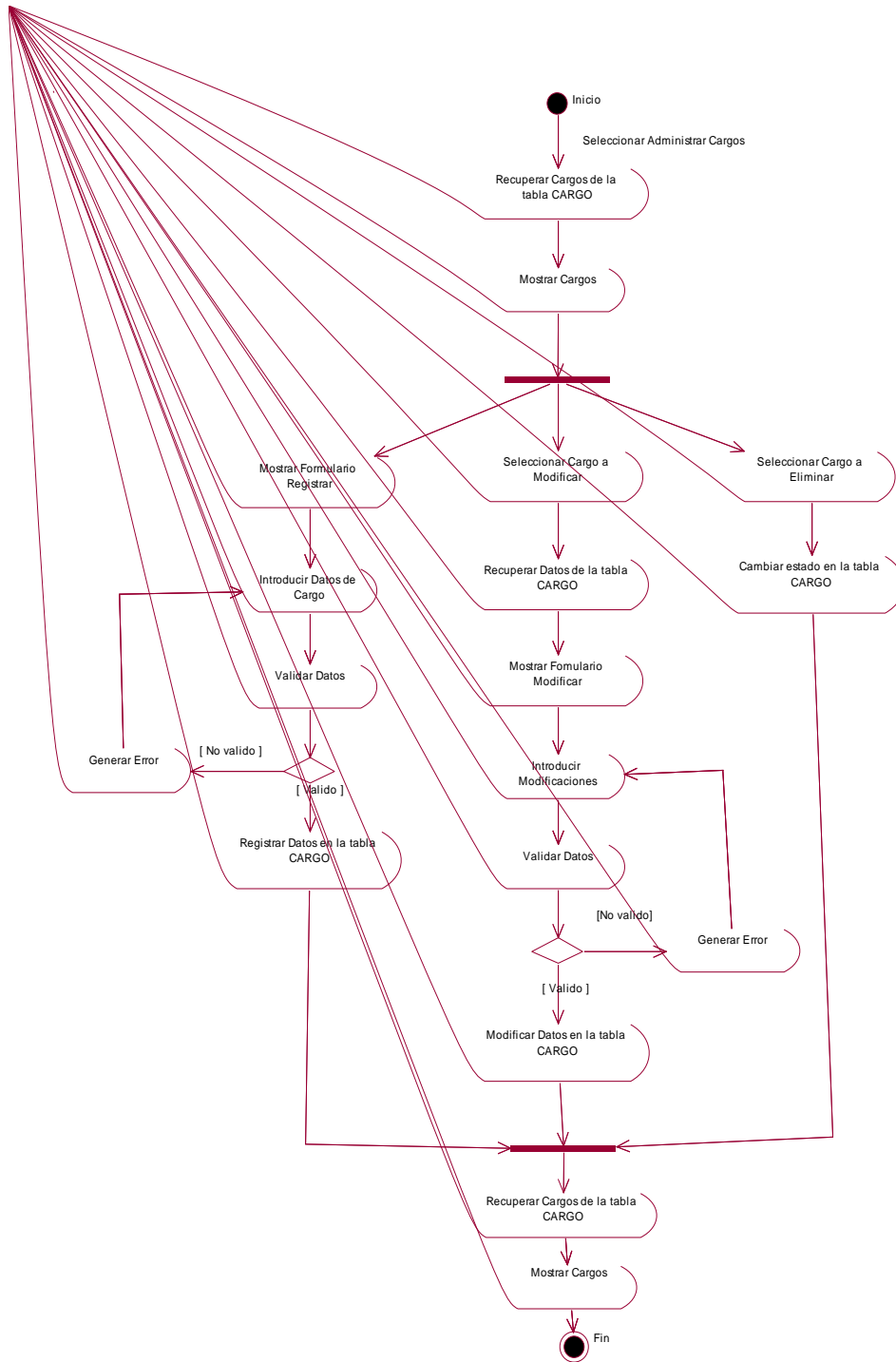
3.5.16 Diagrama de Actividades: Generar Reporte de Ingresos



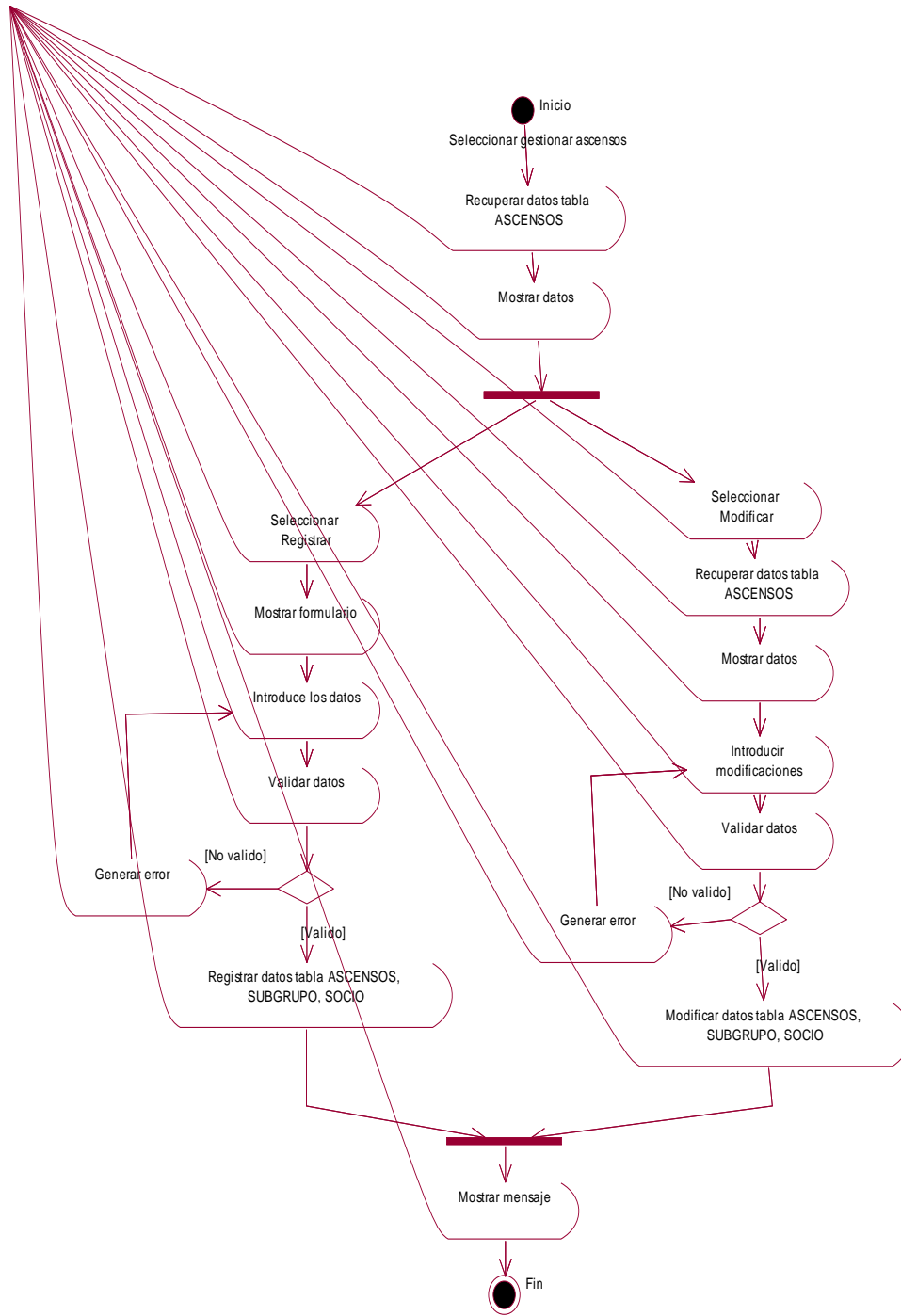
3.5.17 Diagrama de Actividades: Gestionar Ingresos Adicionales



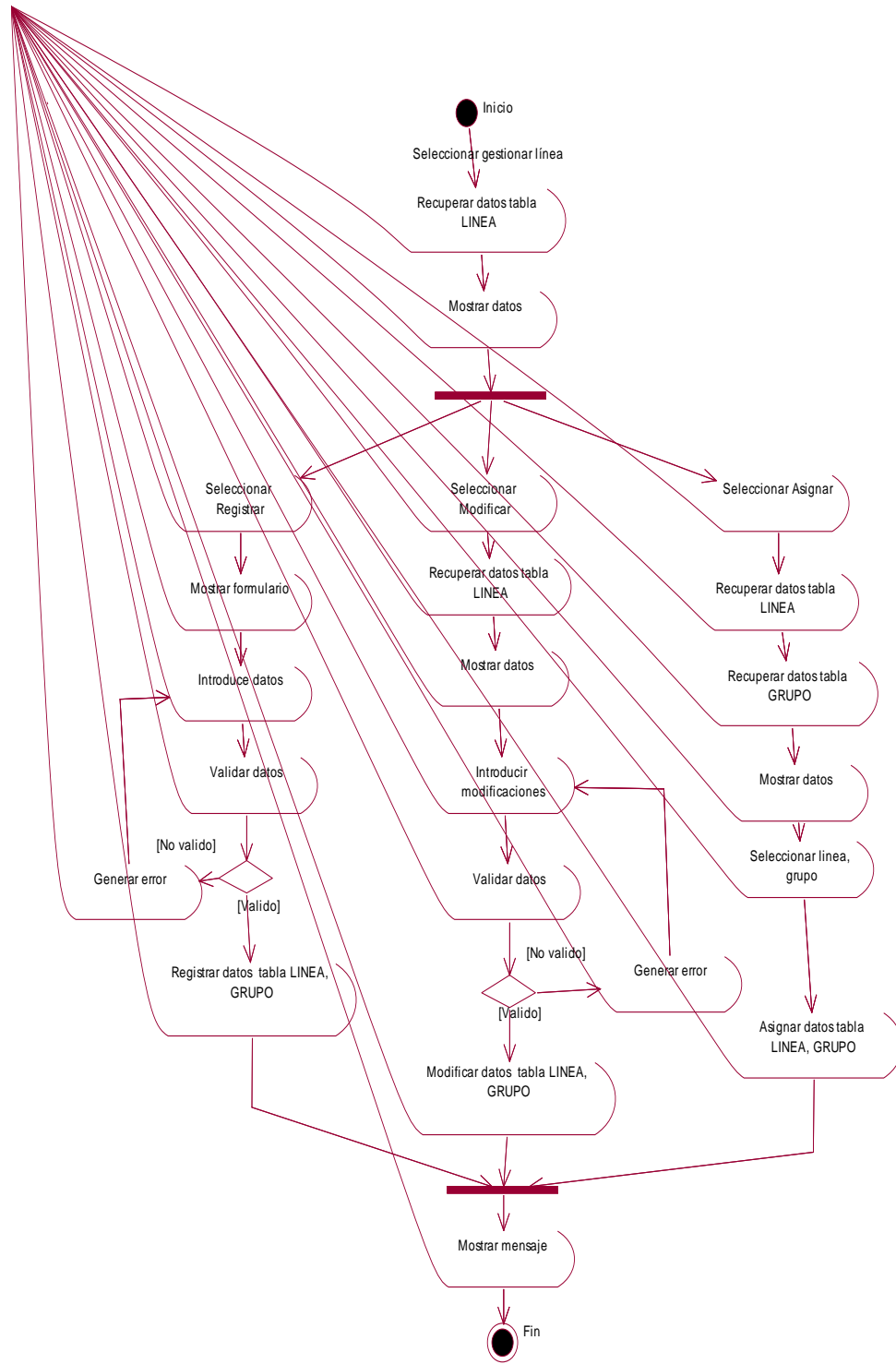
3.5.18 Diagrama de Actividades: Gestionar Cargos



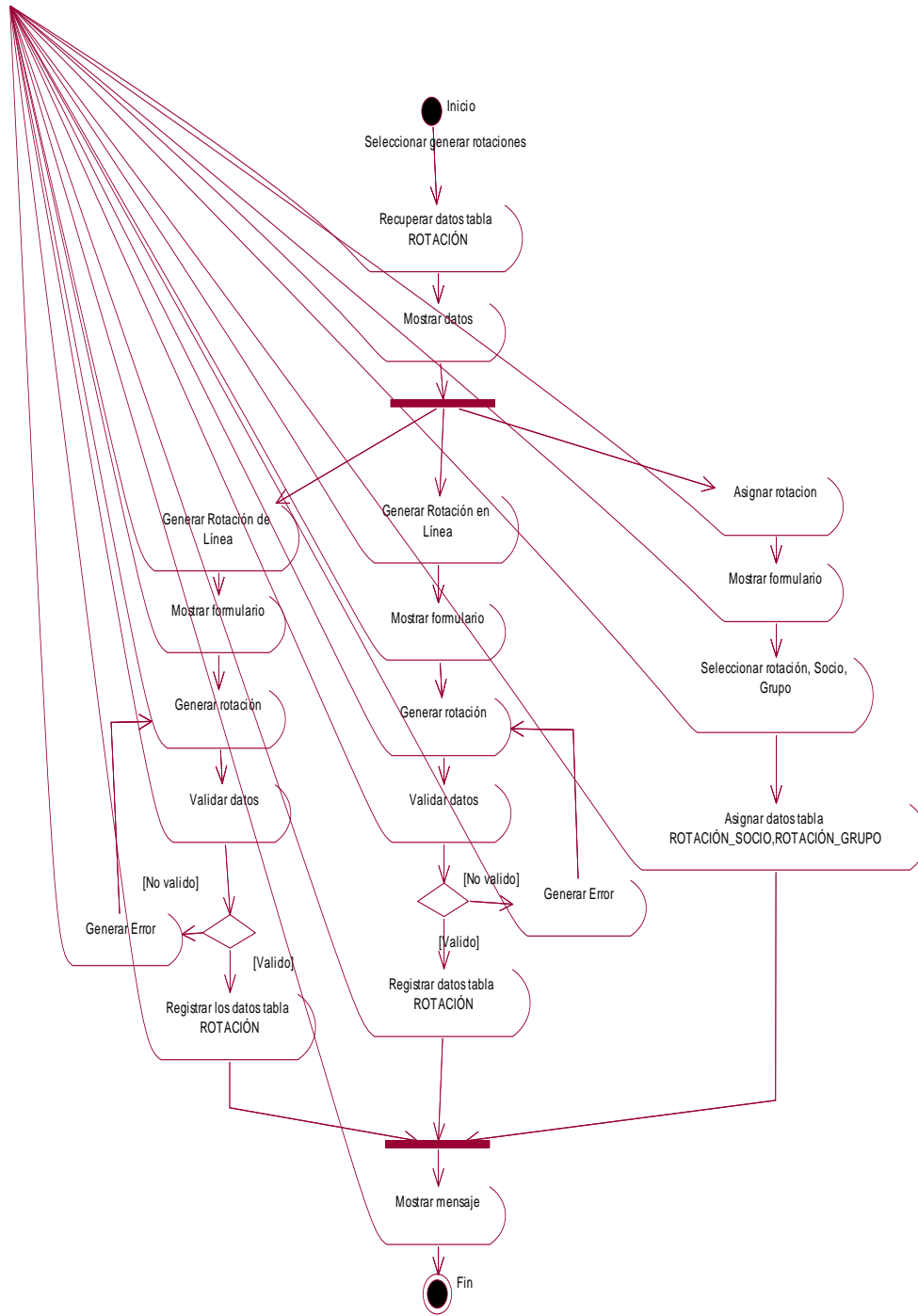
3.5.19 Diagrama de Actividades: Gestionar Ascensos



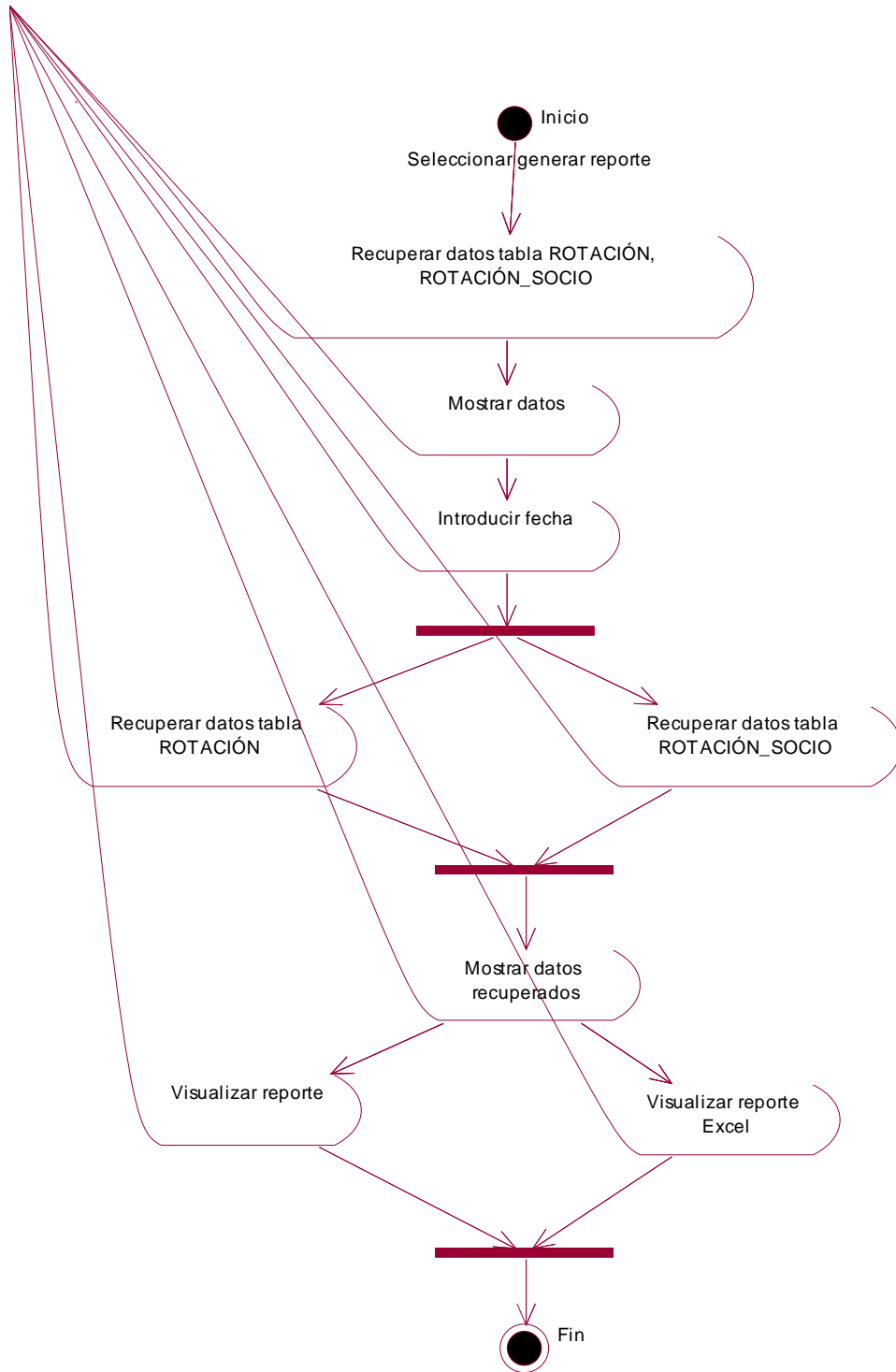
3.5.20 Diagrama de Actividades: Gestionar Línea



3.5.21 Diagrama de Actividades: Gestionar Rol

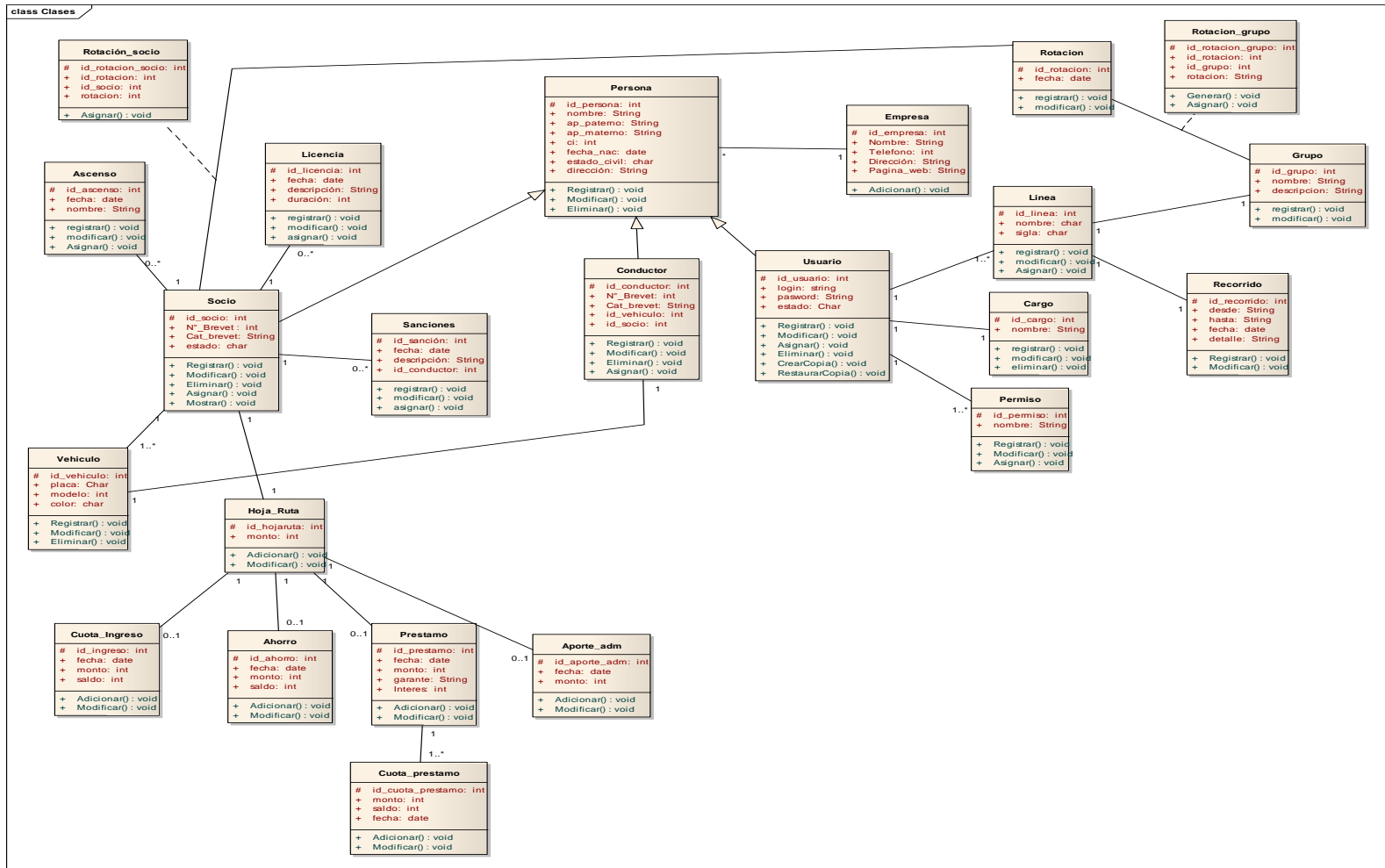


3.5.22 Diagrama de Actividades: Generar Reporte Rotaciones

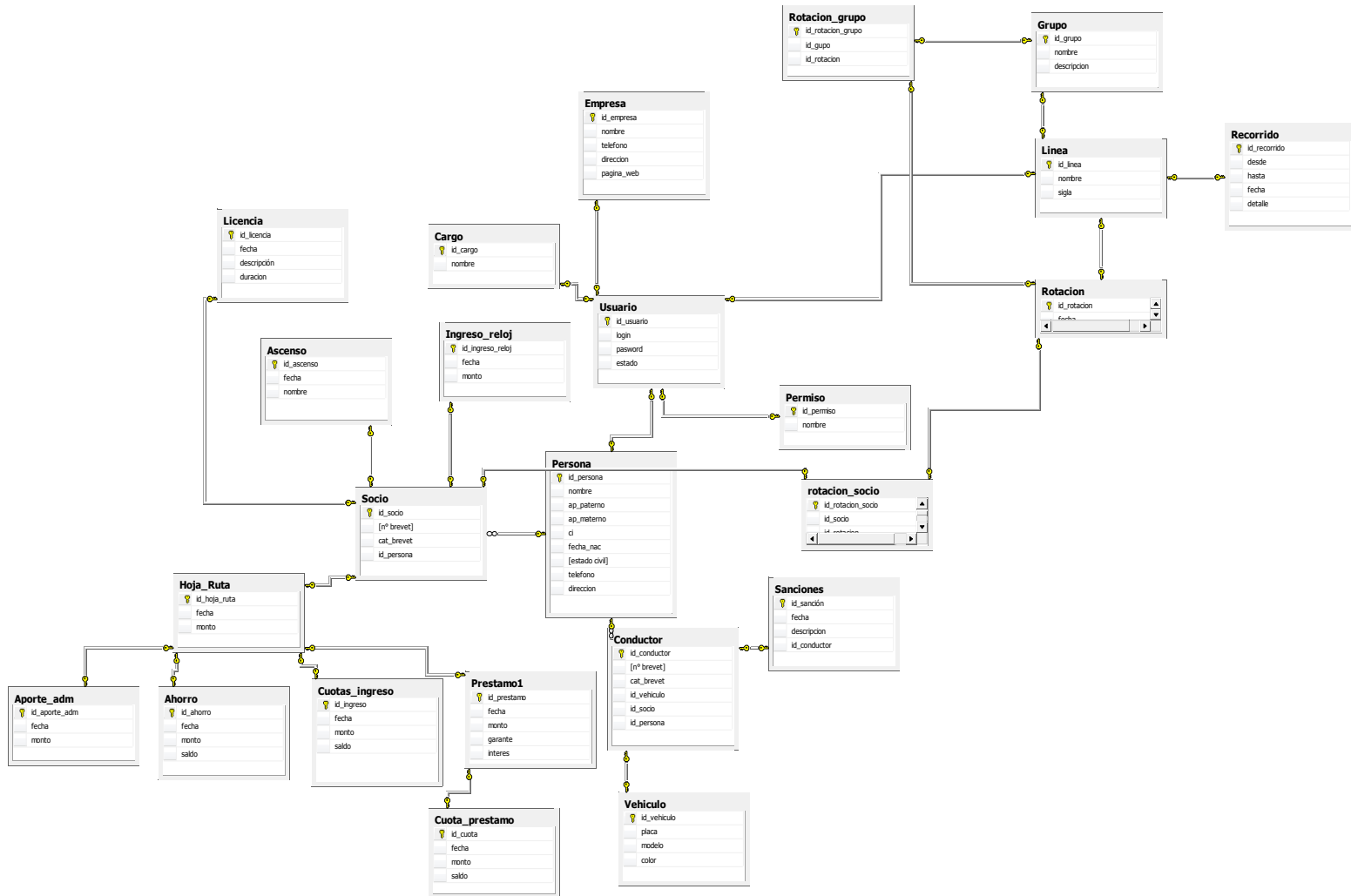


4. DESCRIPCIÓN DE LA BASE DE DATOS

4.1 Diagrama de Clases



4.2 Diseño de la Base Datos



4.3 Descripción de las tablas

Nombre de la Tabla		SOCIO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_SOCIO	Integer	YES	NO	NO
NO	N° BREVET	Integer	NO	NO	NO
NO	CAT_BREVET	Char	NO	NO	NO
NO	ESTADO	Char	NO	NO	NO
NO	ID_PERSONA	Integer	NO	NO	YES

Nombre de la Tabla		CONDUCTOR			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_CONDUCTOR	Integer	YES	NO	NO
NO	N° BREVET	Integer	NO	NO	NO
NO	CAT_BREVET	Char	NO	NO	NO
NO	ID_VEHICULO	Date	NO	NO	YES
NO	ID_SOCIO	Time	NO	NO	YES

Nombre de la Tabla		ASCENSO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_ASCENSO	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	NOMBRE	Varchar	NO	NO	NO
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		LICENCIA			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_LICENCIA	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	DESCRIPCION	Varchar	NO	NO	NO
NO	DURACION	Integer	NO	NO	NO
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		EMPRESA			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_EMPRESA	Integer	YES	NO	NO
NO	NOMBRE	Varchar	NO	NO	NO
NO	DIRECCION	Varchar	NO	NO	NO
NO	TELEFONO	Varchar	NO	NO	NO
NO	PAGINA_WEB	Varchar	NO	NO	NO

Nombre de la Tabla		PERSONA			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_PERSONA	Integer	YES	NO	NO
NO	NOMBRE	Varchar	NO	NO	NO
NO	AP_PATERNO	Varchar	NO	NO	NO
NO	AP_MATERNO	Varchar	NO	NO	NO
NO	CI	Varchar	NO	NO	NO
NO	FECHA_NAC	Datetime	NO	NO	NO
NO	ESTADO_CIVIL	Varchar	NO	NO	NO
NO	DIRECCION	Varchar	NO	NO	NO
NO	TELEFONO	Varchar	NO	NO	NO

Nombre de la Tabla		ROTACION_SOCIO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_ROTACION_SOCIO	Integer	YES	NO	NO
NO	ID_ROTACION	Integer	NO	NO	YES
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		USUARIO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_USUARIO	Integer	YES	NO	NO
NO	LOGIN	Varchar	NO	NO	NO
NO	PASSWORD	Varchar	NO	NO	NO
NO	ESTADO	Varchar	NO	NO	NO
NO	ID_PERSONA	Integer	NO	NO	YES

Nombre de la Tabla		ROTACION			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_ROTACION	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO

Nombre de la Tabla	PERMISO				
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_PERMISO	Integer	YES	NO	NO
NO	ID_USUARIO	Integer	NO	NO	YES
NO	NOMBRE	Varchar	NO	NO	NO

Nombre de la Tabla	CARGO				
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_CARGO	Integer	YES	NO	NO
NO	NOMBRE	Varchar	NO	NO	NO
NO	ID_USUARIO	Integer	NO	NO	YES

Nombre de la Tabla	GRUPO				
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_GRUPO	Integer	YES	NO	NO
NO	NOMBRE	Varchar	NO	NO	NO
NO	DESCRIPCION	Varchar	NO	NO	NO

Nombre de la Tabla		ROTACION_GRUPO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_ROTACION_GRUPO	Integer	YES	NO	NO
NO	ID_GRUPO	Integer	NO	NO	YES
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		LINEA			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_LINEA	Integer	YES	NO	NO
NO	NOMBRE	Varchar	NO	NO	NO
NO	SIGLA	Varchar	NO	NO	NO
NO	ID_USUARIO	Integer	NO	NO	YES

Nombre de la Tabla		RECORRIDO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_RECORRIDO	Integer	YES	NO	NO
NO	DESDE	Varchar	NO	NO	NO
NO	HASTA	Varchar	NO	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	DETALLE	Varchar	NO	NO	NO
NO	ID_LINEA	Integer	NO	NO	YES

Nombre de la Tabla		SANCIONES			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_SANCIONES	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	DESCRIPCION	Varchar	NO	NO	NO
NO	ID_CONDUCTOR	Integer	NO	NO	YES

Nombre de la Tabla		VEHICULO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_VEHICULO	Integer	YES	NO	NO
NO	PLACA	Varchar	NO	NO	NO
NO	MODELO	Integer	NO	NO	NO
NO	COLOR	Varchar	NO	NO	NO
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		INGRESO_RELOJ			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_INGRESO_RELOJ	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		HOJA_RUTA			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_HOJA_RUTA	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	ID_SOCIO	Integer	NO	NO	YES

Nombre de la Tabla		CUOTAS_INGRESO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_INGRESO	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	SALDO	Integer	NO	NO	NO
NO	ID_HOJA_RUTA	Integer	NO	NO	YES

Nombre de la Tabla		AHORRO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_AHORRO	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	SALDO	Integer	NO	NO	NO
NO	ID_HOJA_RUTA	Integer	NO	NO	YES

Nombre de la Tabla		APORTE_ADM			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_APORTE_ADM	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	ID_HOJA_RUTA	Integer	NO	NO	YES

Nombre de la Tabla		PRESTAMO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_PRESTAMO	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	GARANTE	Varchar	NO	NO	NO
NO	INTERES	Integer	NO	NO	NO
NO	ID_HOJA_RUTA	Integer	NO	NO	YES

Nombre de la Tabla		CUOTA_PRESTAMO			
Llave Primaria	Campos	Tipo de Dato	No Nulo	Único	Llave Foránea
YES	ID_CUOTA_PRES TAMO	Integer	YES	NO	NO
NO	FECHA	Datetime	NO	NO	NO
NO	MONTO	Integer	NO	NO	NO
NO	SALDO	Integer	NO	NO	NO
NO	ID_HOJA_RUTA	Integer	NO	NO	YES

4.4 Creación de la Base de Datos

```
USE[Cooperativa]
GO
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[Rotacion_grupo](
    [id_rotacion_grupo][int]NOTNULL,
    [id_gupo][int]NULL,
    [id_rotacion][int]NULL,
CONSTRAINT[PK_Rotacion_grupo]PRIMARYKEYCLUSTERED
(
    [id_rotacion_grupo]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Sanciones]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Sanciones](
    [id_sanción][int]NOTNULL,
    [fecha][date]NULL,
    [descripcion][varchar](50)NULL,
    [id_conductor][int]NULL,
CONSTRAINT[PK_Sancion]PRIMARYKEYCLUSTERED
(
    [id_sanción]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Persona]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
```

```

GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Persona](
    [id_persona][int]NOTNULL,
    [nombre][varchar](50)NOTNULL,
    [ap_paterno][varchar](50)NULL,
    [ap_materno][varchar](50)NULL,
    [ci][int]NULL,
    [fecha_nac][date]NULL,
    [estado_civil][char](10)NULL,
    [telefono][int]NULL,
    [direccion][varchar](50)NULL,
CONSTRAINT[PK_Persona]PRIMARYKEYCLUSTERED
(
    [id_persona]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Empresa]   Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Empresa](
    [id_empresa][int]NOTNULL,
    [nombre][varchar](50)NULL,
    [telefono][int]NULL,
    [direccion][varchar](50)NULL,
    [pagina_web][varchar](50)NULL,
CONSTRAINT[PK_Empresa]PRIMARYKEYCLUSTERED
(
    [id_empresa]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO

```

```

/***** Object: Table [dbo].[Ascenso]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Ascenso](
    [id_ascenso][int]NOTNULL,
    [fecha][date]NULL,
    [nombre][varchar](50)NULL,
CONSTRAINT[PK_Ascenso]PRIMARYKEYCLUSTERED
(
    [id_ascenso]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Licencia]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Licencia](
    [id_licencia][int]NOTNULL,
    [fecha][date]NOTNULL,
    [descripción][varchar](50)NULL,
    [duracion][int]NOTNULL,
CONSTRAINT[PK_Licencia]PRIMARYKEYCLUSTERED
(
    [id_licencia]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Usuario]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO

```

```

SETQUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Usuario](
    [id_usuario][int] NOT NULL,
    [login][varchar](50) NOT NULL,
    [password][varchar](50) NOT NULL,
    [estado][char](10) NOT NULL,
CONSTRAINT [PK_Usuario] PRIMARY KEY CLUSTERED
(
    [id_usuario] ASC
) WITH
(PAD_INDEX=OFF, STATISTICS_NORECOMPUTE=OFF, IGNORE_DUP_KEY=OFF, ALLOW_ROW_LOCKS=ON, ALLOW_PAGE_LOCKS=ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Socio]  Script Date: 07/07/2013 14:19:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Socio](
    [id_socio][int] NOT NULL,
    [n° brevet][int] NOT NULL,
    [cat_brevet][varchar](50) NOT NULL,
    [id_persona][int] NULL,
CONSTRAINT [PK_Socio] PRIMARY KEY CLUSTERED
(
    [id_socio] ASC
) WITH
(PAD_INDEX=OFF, STATISTICS_NORECOMPUTE=OFF, IGNORE_DUP_KEY=OFF, ALLOW_ROW_LOCKS=ON, ALLOW_PAGE_LOCKS=ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Conductor]  Script Date: 07/07/2013 14:19:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Conductor](
    [id_conductor][int]NOTNULL,
    [n° brevet][int]NOTNULL,
    [cat_brevet][varchar](50)NOTNULL,
    [id_vehiculo][int]NOTNULL,
    [id_socio][int]NOTNULL,
    [id_persona][int]NOTNULL,
CONSTRAINT[PK_Conductor]PRIMARYKEYCLUSTERED
(
    [id_conductor]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Vehiculo]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Vehiculo](
    [id_vehiculo][int]NOTNULL,
    [placa][char](10)NULL,
    [modelo][int]NULL,
    [color][char](10)NULL,
CONSTRAINT[PK_Vehiculo]PRIMARYKEYCLUSTERED
(
    [id_vehiculo]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Cargo]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO

```

```

SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Cargo](
    [id_cargo][int]NOTNULL,
    [nombre][varchar](50)NULL,
CONSTRAINT[PK_Cargo]PRIMARYKEYCLUSTERED
(
    [id_cargo]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Ingreso_reloj]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[Ingreso_reloj](
    [id_ingreso_reloj][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
CONSTRAINT[PK_Ingreso_reloj]PRIMARYKEYCLUSTERED
(
    [id_ingreso_reloj]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Hoja_Ruta]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[Hoja_Ruta](
    [id_hoja_ruta][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
CONSTRAINT[PK_Hoja_Ruta]PRIMARYKEYCLUSTERED
(
    [id_hoja_ruta]ASC

```

```

)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Permiso]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Permiso](
    [id_permiso][int]NOTNULL,
    [nombre][varchar](50)NULL,
CONSTRAINT[PK_Permiso]PRIMARYKEYCLUSTERED
(
    [id_permiso]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Linea]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Linea](
    [id_linea][int]NOTNULL,
    [nombre][varchar](50)NULL,
    [sigla][char](10)NULL,
CONSTRAINT[PK_Linea]PRIMARYKEYCLUSTERED
(
    [id_linea]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO

```

```

/***** Object: Table [dbo].[rotacion_socio]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[rotacion_socio](
    [id_rotacion_socio][int]NOTNULL,
    [id_socio][int]NULL,
    [id_rotacion][int]NULL,
CONSTRAINT[PK_rotacion_socio]PRIMARYKEYCLUSTERED
(
    [id_rotacion_socio]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Rotacion]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[Rotacion](
    [id_rotacion][int]NOTNULL,
    [fecha][date]NOTNULL,
CONSTRAINT[PK_Rotacion]PRIMARYKEYCLUSTERED
(
    [id_rotacion]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Recorrido]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Recorrido](
    [id_recorrido][int]NOTNULL,
    [desde][varchar](50)NULL,
    [hasta][varchar](50)NULL,
    [fecha][date]NULL,
    [detalle][varchar](50)NULL,

```

```

CONSTRAINT[PK_Recorrido]PRIMARYKEYCLUSTERED
(
    [id_recorrido]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Prestamo1]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Prestamo1](
    [id_prestamo][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
    [garante][varchar](50)NULL,
    [interes][int]NULL,
CONSTRAINT[PK_Prestamo1]PRIMARYKEYCLUSTERED
(
    [id_prestamo]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Grupo]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATETABLE[dbo].[Grupo](
    [id_grupo][int]NOTNULL,
    [nombre][varchar](50)NULL,
    [descripcion][varchar](50)NULL,
CONSTRAINT[PK_Grupo]PRIMARYKEYCLUSTERED
(

```

```

        [id_grupo]ASC
    )WITH
    (PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
SETANSI_PADDINGOFF
GO
/***** Object: Table [dbo].[Aporte_adm]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[Aporte_adm](
    [id_aporte_adm][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
CONSTRAINT[PK_Aporte_adm]PRIMARYKEYCLUSTERED
(
    [id_aporte_adm]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Ahorro]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
CREATETABLE[dbo].[Ahorro](
    [id_ahorro][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
    [saldo][int]NULL,
CONSTRAINT[PK_Ahorro]PRIMARYKEYCLUSTERED
(
    [id_ahorro]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Cuotas_ingreso]  Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO

```

```

SETQUOTED_IDENTIFIER ON
GO
CREATETABLE[dbo].[Cuotas_ingreso](
    [id_ingreso][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
    [saldo][int]NULL,
CONSTRAINT[PK_Cuotas_ingreso]PRIMARYKEYCLUSTERED
(
    [id_ingreso]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: Table [dbo].[Cuota_prestamo]   Script Date: 07/07/2013 14:19:08 *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIER ON
GO
CREATETABLE[dbo].[Cuota_prestamo](
    [id_cuota][int]NOTNULL,
    [fecha][date]NULL,
    [monto][int]NULL,
    [saldo][int]NULL,
CONSTRAINT[PK_Cuota_prestamo]PRIMARYKEYCLUSTERED
(
    [id_cuota]ASC
)WITH
(PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LOCKS=ON,ALLOW_PAGE_LOCKS=ON)ON[PRIMARY]
)ON[PRIMARY]
GO
/***** Object: ForeignKey [FK_Usuario_Empresa]   Script Date: 07/07/2013 14:19:08 *****/
ALTERTABLE[dbo].[Usuario]WITHCHECKADDCONSTRAINT[FK_Usuario_Empresa]FOREIGNKEY([id_usuario])
REFERENCES[dbo].[Empresa]([id_empresa])
GO
ALTERTABLE[dbo].[Usuario]CHECKCONSTRAINT[FK_Usuario_Empresa]
GO
/***** Object: ForeignKey [FK_Usuario_Persona]   Script Date: 07/07/2013 14:19:08 *****/
ALTERTABLE[dbo].[Usuario]WITHCHECKADDCONSTRAINT[FK_Usuario_Persona]FOREIGNKEY([id_usuario])
REFERENCES[dbo].[Persona]([id_persona])

```

```

GO
ALTER TABLE [dbo].[Usuario] CHECK CONSTRAINT [FK_Usuario_Persona]
GO
/***** Object: ForeignKey [FK_Socio_Ascenso]  Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Socio] WITH CHECK ADD CONSTRAINT [FK_Socio_Ascenso] FOREIGN
KEY([id_socio])
REFERENCES [dbo].[Ascenso]([id_ascenso])
GO
ALTER TABLE [dbo].[Socio] CHECK CONSTRAINT [FK_Socio_Ascenso]
GO
/***** Object: ForeignKey [FK_Socio_Licencia]  Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Socio] WITH CHECK ADD CONSTRAINT [FK_Socio_Licencia] FOREIGN
KEY([id_socio])
REFERENCES [dbo].[Licencia]([id_licencia])
GO
ALTER TABLE [dbo].[Socio] CHECK CONSTRAINT [FK_Socio_Licencia]
GO
/***** Object: ForeignKey [FK_Socio_Persona]  Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Socio] WITH CHECK ADD CONSTRAINT [FK_Socio_Persona] FOREIGN
KEY([id_persona])
REFERENCES [dbo].[Persona]([id_persona])
GO
ALTER TABLE [dbo].[Socio] CHECK CONSTRAINT [FK_Socio_Persona]
GO
/***** Object: ForeignKey [FK_Conductor_Persona]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Conductor] WITH CHECK ADD CONSTRAINT [FK_Conductor_Persona]
FOREIGN KEY([id_persona])
REFERENCES [dbo].[Persona]([id_persona])
GO
ALTER TABLE [dbo].[Conductor] CHECK CONSTRAINT [FK_Conductor_Persona]
GO
/***** Object: ForeignKey [FK_Conductor_Sanciones]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Conductor] WITH CHECK ADD CONSTRAINT [FK_Conductor_Sancione
s] FOREIGN KEY([id_conductor])
REFERENCES [dbo].[Sanciones]([id_sanción])
GO
ALTER TABLE [dbo].[Conductor] CHECK CONSTRAINT [FK_Conductor_Sanciones]
GO
/***** Object: ForeignKey [FK_Vehiculo_Conductor]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Vehiculo] WITH CHECK ADD CONSTRAINT [FK_Vehiculo_Conductor]
FOREIGN KEY([id_vehiculo])
REFERENCES [dbo].[Conductor]([id_conductor])
GO

```

```

ALTER TABLE [dbo].[Vehiculo] CHECK CONSTRAINT [FK_Vehiculo_Conductor]
GO
/***** Object: ForeignKey [FK_Cargo_Usuario] Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Cargo] WITH CHECK ADD CONSTRAINT [FK_Cargo_Usuario] FOREIGN
KEY([id_cargo])
REFERENCES [dbo].[Usuario]([id_usuario])
GO
ALTER TABLE [dbo].[Cargo] CHECK CONSTRAINT [FK_Cargo_Usuario]
GO
/***** Object: ForeignKey [FK_Ingreso_reloj_Usuario] Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Ingreso_reloj] WITH CHECK ADD CONSTRAINT [FK_Ingreso_reloj_Usu
ario] FOREIGN KEY([id_ingreso_reloj])
REFERENCES [dbo].[Usuario]([id_usuario])
GO
ALTER TABLE [dbo].[Ingreso_reloj] CHECK CONSTRAINT [FK_Ingreso_reloj_Usuario]
GO
/***** Object: ForeignKey [FK_Hoja_Ruta_Socio] Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Hoja_Ruta] WITH CHECK ADD CONSTRAINT [FK_Hoja_Ruta_Socio] F
OREIGN KEY([id_hoja_ruta])
REFERENCES [dbo].[Socio]([id_socio])
GO
ALTER TABLE [dbo].[Hoja_Ruta] CHECK CONSTRAINT [FK_Hoja_Ruta_Socio]
GO
/***** Object: ForeignKey [FK_Permiso_Usuario] Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Permiso] WITH CHECK ADD CONSTRAINT [FK_Permiso_Usuario] FOR
EIGN KEY([id_permiso])
REFERENCES [dbo].[Usuario]([id_usuario])
GO
ALTER TABLE [dbo].[Permiso] CHECK CONSTRAINT [FK_Permiso_Usuario]
GO
/***** Object: ForeignKey [FK_Linea_Usuario] Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Linea] WITH CHECK ADD CONSTRAINT [FK_Linea_Usuario] FOREIGN
KEY([id_linea])
REFERENCES [dbo].[Usuario]([id_usuario])
GO
ALTER TABLE [dbo].[Linea] CHECK CONSTRAINT [FK_Linea_Usuario]
GO
/***** Object: ForeignKey [FK_rotacion_socio_Socio] Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[rotacion_socio] WITH CHECK ADD CONSTRAINT [FK_rotacion_socio_S
ocio] FOREIGN KEY([id_rotacion_socio])
REFERENCES [dbo].[Socio]([id_socio])
GO

```

```

ALTER TABLE [dbo].[rotacion_socio] CHECK CONSTRAINT [FK_rotacion_socio_Socio]
GO
/***** Object: ForeignKey [FK_Rotacion_Linea]  Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Rotacion] WITH CHECK ADD CONSTRAINT [FK_Rotacion_Linea] FOREIGN KEY ([id_rotacion])
REFERENCES [dbo].[Linea] ([id_linea])
GO
ALTER TABLE [dbo].[Rotacion] CHECK CONSTRAINT [FK_Rotacion_Linea]
GO
/***** Object: ForeignKey [FK_Rotacion_Rotacion_grupo]  Script Date: 07/07/2013
14:19:08 *****/
ALTER TABLE [dbo].[Rotacion] WITH CHECK ADD CONSTRAINT [FK_Rotacion_Rotacion_grupo] FOREIGN KEY ([id_rotacion])
REFERENCES [dbo].[Rotacion_grupo] ([id_rotacion_grupo])
GO
ALTER TABLE [dbo].[Rotacion] CHECK CONSTRAINT [FK_Rotacion_Rotacion_grupo]
GO
/***** Object: ForeignKey [FK_Rotacion_rotacion_socio]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Rotacion] WITH CHECK ADD CONSTRAINT [FK_Rotacion_rotacion_socio] FOREIGN KEY ([id_rotacion])
REFERENCES [dbo].[rotacion_socio] ([id_rotacion_socio])
GO
ALTER TABLE [dbo].[Rotacion] CHECK CONSTRAINT [FK_Rotacion_rotacion_socio]
GO
/***** Object: ForeignKey [FK_Recorrido_Linea]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Recorrido] WITH CHECK ADD CONSTRAINT [FK_Recorrido_Linea] FOREIGN KEY ([id_recorrido])
REFERENCES [dbo].[Linea] ([id_linea])
GO
ALTER TABLE [dbo].[Recorrido] CHECK CONSTRAINT [FK_Recorrido_Linea]
GO
/***** Object: ForeignKey [FK_Prestamo1_Hoja_Ruta]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Prestamo1] WITH CHECK ADD CONSTRAINT [FK_Prestamo1_Hoja_Ruta] FOREIGN KEY ([id_prestamo])
REFERENCES [dbo].[Hoja_Ruta] ([id_hoja_ruta])
GO
ALTER TABLE [dbo].[Prestamo1] CHECK CONSTRAINT [FK_Prestamo1_Hoja_Ruta]
GO
/***** Object: ForeignKey [FK_Grupo_Linea]  Script Date: 07/07/2013 14:19:08 *****/
ALTER TABLE [dbo].[Grupo] WITH CHECK ADD CONSTRAINT [FK_Grupo_Linea] FOREIGN KEY ([id_grupo])
REFERENCES [dbo].[Linea] ([id_linea])
GO

```

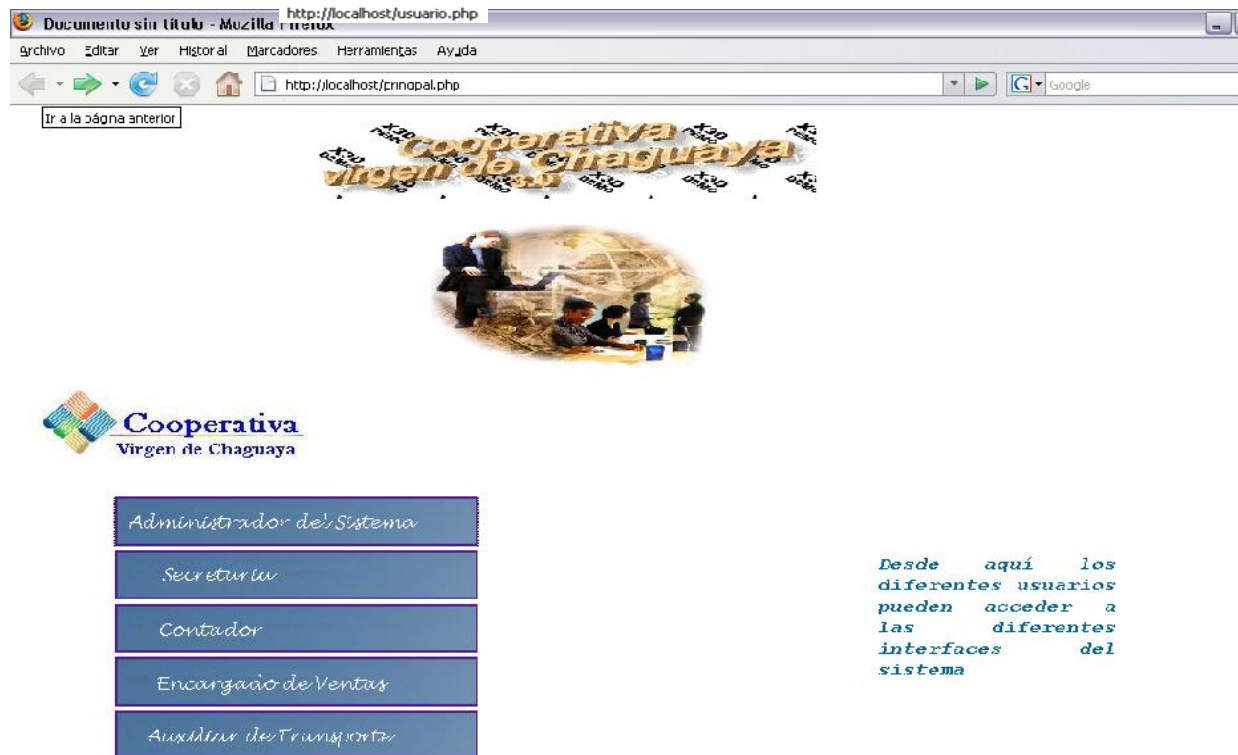
```

ALTER TABLE [dbo].[Grupo] CHECK CONSTRAINT [FK_Grupo_Linea]
GO
/***** Object: ForeignKey [FK_Grupo_Rotacion_grupo]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Grupo] WITH CHECK ADD CONSTRAINT [FK_Grupo_Rotacion_grupo]
FOREIGN KEY ([id_grupo])
REFERENCES [dbo].[Rotacion_grupo] ([id_rotacion_grupo])
GO
ALTER TABLE [dbo].[Grupo] CHECK CONSTRAINT [FK_Grupo_Rotacion_grupo]
GO
/***** Object: ForeignKey [FK_Aporte_adm_Hoja_Ruta]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Aporte_adm] WITH CHECK ADD CONSTRAINT [FK_Aporte_adm_Hoja
_Ruta] FOREIGN KEY ([id_aporte_adm])
REFERENCES [dbo].[Hoja_Ruta] ([id_hoja_ruta])
GO
ALTER TABLE [dbo].[Aporte_adm] CHECK CONSTRAINT [FK_Aporte_adm_Hoja_Ruta]
GO
/***** Object: ForeignKey [FK_Ahorro_Hoja_Ruta]  Script Date: 07/07/2013 14:19:08
*****/
ALTER TABLE [dbo].[Ahorro] WITH CHECK ADD CONSTRAINT [FK_Ahorro_Hoja_Ruta] FO
REIGN KEY ([id_ahorro])
REFERENCES [dbo].[Hoja_Ruta] ([id_hoja_ruta])
GO
ALTER TABLE [dbo].[Ahorro] CHECK CONSTRAINT [FK_Ahorro_Hoja_Ruta]
GO
/***** Object: ForeignKey [FK_Cuotas_ingreso_Hoja_Ruta]  Script Date: 07/07/2013
14:19:08 *****/
ALTER TABLE [dbo].[Cuotas_ingreso] WITH CHECK ADD CONSTRAINT [FK_Cuotas_ingreso
_Hoja_Ruta] FOREIGN KEY ([id_ingreso])
REFERENCES [dbo].[Hoja_Ruta] ([id_hoja_ruta])
GO
ALTER TABLE [dbo].[Cuotas_ingreso] CHECK CONSTRAINT [FK_Cuotas_ingreso_Hoja_Ruta]
GO
/***** Object: ForeignKey [FK_Cuota_prestamo_Prestamo1]  Script Date: 07/07/2013
14:19:08 *****/
ALTER TABLE [dbo].[Cuota_prestamo] WITH CHECK ADD CONSTRAINT [FK_Cuota_prestam
o_Prestamo1] FOREIGN KEY ([id_cuota])
REFERENCES [dbo].[Prestamo1] ([id_prestamo])
GO
ALTER TABLE [dbo].[Cuota_prestamo] CHECK CONSTRAINT [FK_Cuota_prestamo_Prestamo
1]
GO

```

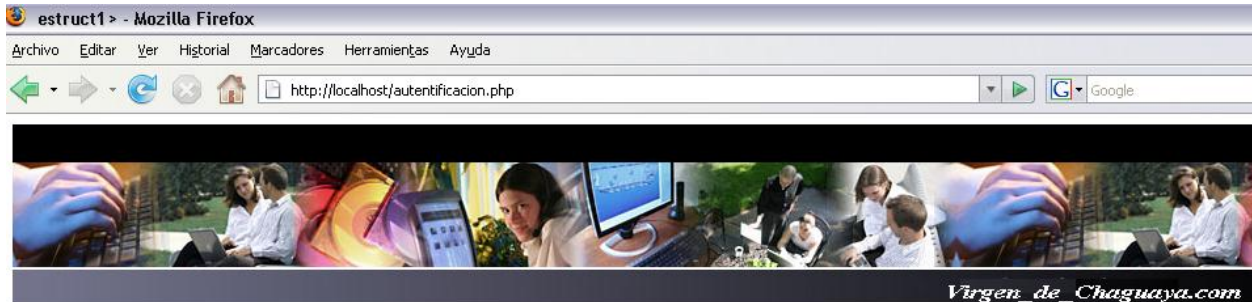
4.5 Diseño de Interfaces

Pantalla n°1: Pantalla Principal



Pantalla donde se muestra los diferentes módulos para poder navegar dependiendo el tipo de usuario.

Pantalla n°2: Login y Password



Autenticacion de Usuario



Login	<input type="text"/>
Password:	<input type="password"/>
Ingresar	Limpiar

Pantalla donde se muestra el formulario para la autenticación del usuario para verificar su existencia.

Pantalla n° 3: Registrar Usuario

estruct1 > - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
http://localhost/usuario.php
Google

Virgen de Chaguaya.com

Cooperativa Virgen de Chaguaya

Registra Datos del Usuario




Ci:	<input type="text"/>
Nombre:	<input type="text"/>
A_Paterno:	<input type="text"/>
A_Materno:	<input type="text"/>
Fecha_Nac:	<input type="text"/>
Estado Civil:	<input type="text"/>

Pantalla que permite registrar un usuario introduciendo sus datos personales.

Pantalla n°4: Registrar Socio

estruct1 > - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

file:///C:/Users/hry/AppData/Local/Temp/Rar5EX73.877/ejemplo/pagina de prueba.html

Virgen de Chaguaya.com

Datos de Socio

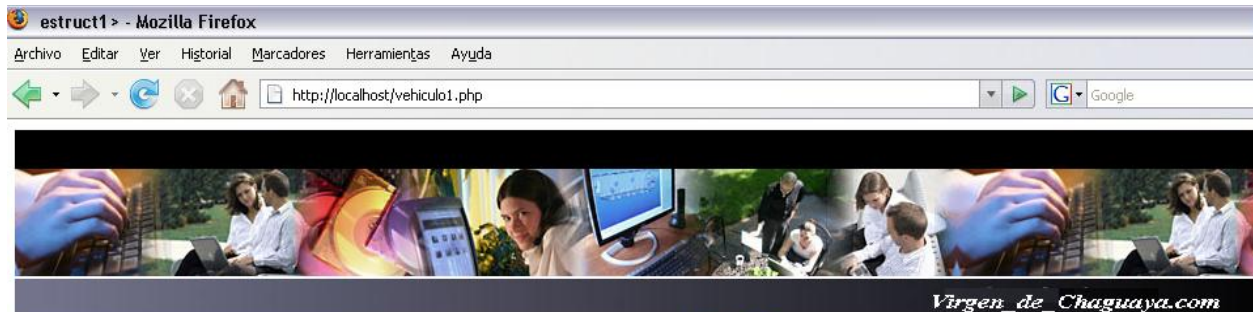


Ci:	<input type="text"/>
Nombre:	<input type="text"/>
A_Paterno:	<input type="text"/>
A_Materno:	<input type="text"/>
Fecha_Reg:	<input type="text"/>

Registrar Limpiar

Pantalla que permite registrar un socio introduciendo sus datos personales.

Pantalla n°5: Registrar Vehículo



Ingresar Datos del Vehículo



Ci_Socio:	<input type="text"/>
Placa:	<input type="text"/>
Marca:	<input type="text"/>
Modelo:	<input type="text"/>
Color:	<input type="text"/>
Num_Plazas:	<input type="text"/>

Pantalla que permite ingresar los datos del vehículo que pertenece algún determinado socio.

Pantalla n°6: Ingresar Hoja de Ruta



Datos de Hoja de Ruta



Nombre:	Antonio_Valdez
Cedula:	<input type="text"/>
Linea:	Nro-1
Fecha:	2007/12/11
Hoja_de_Ruta-Bs:	<input type="text"/>
Ahorros-Bs:	<input type="text"/>
Préstamos_por_Cob-Bs:	<input type="text"/>
Cuota_Ingreso_por_Cob-Bs:	<input type="text"/>
TOTAL-Bs:	<input type="text"/>
<input type="button" value="Calcular_Total"/> <input type="button" value="Registrar"/> <input type="button" value="Limpiar"/>	

Pantalla donde se registran los diferentes ingresos de dinero por cada hoja de ruta (préstamos, cuotas de ingresos, etc.)

Pantalla n°7: Registrar Nueva Línea



estruct1 > - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost/linea.php

Google

Virgen_de_Chaguaya.com

Registrar Nueva Línea de Servicio



Cod Línea:

Nombre Línea:

Grupo: A-1

Fecha creación:

Pantalla donde se registra las líneas que están asociados a la Cooperativa “Virgen de Chaguaya”

Pantalla n°8: Reporte de Socios



Reporte de socios Registrados

Fecha del Día

[Imprimir](#)

Cédula	Nombre	Apellido Paterno	Apellido Materno	Fecha de registro
3236564	Antonio	Valdez	Valdez	2013-06-26
656532	Carla	Almazan	Almazan	2013-06-26
5056176	Gualberto	Reyes	Solar	2013-06-26
6565	Juan Daniel	Cortez	Martinez	2013-04-06
5056178	Juan Manuel	Sandoval	Ribera	2013-04-06
333333	Maribel	Flores	Arteaga	2013-04-06
22222	Mariela	Gaité	Flores	2013-04-06
11111	Nancy	Vera	Rozas	2013-04-06
3526145	Pedro Antonio	Flores	Flores	2013-04-06
444444	Sandra	Armella	Farfan	2013-04-06
5007483	jhaclyn	navarro	bleichner	2013-04-06

Pantalla donde se muestra el detalle de los socios que están afiliados a la Cooperativa “Virgen de Chaguaya”