**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO  
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGIA  
CARRERA DE INGENIERIA INFORMÁTICA**



**“SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DEL SINDICATO DE TRANSPORTE 27 DE DICIEMBRE”**

**PROYECTO EMAIL DE LA MATERIA DE TECNOLOGIA WEB**

**ESTUDIANTES:**

**WEIMAR GUSTAVO USTARES RIVERA**

**CARLOS ALFREDO RAMOS CARBALLO**

**AYELEN ESTEVEZ SEGOVIA**

**TUTOR:   
ING. EVANS BALCAZAR VEISAGA**

**SANTA CRUZ DE LA SIERRA –BOLIVIA  
2023**

Contenido

[1.1 Introducción 8](#_Toc141582506)

[1.2 Antecedentes y justificación 9](#_Toc141582507)

[1.2.1 Antecedentes 9](#_Toc141582508)

[1.2.2 Justificación 10](#_Toc141582509)

[1.3 Planteamiento del problema 11](#_Toc141582510)

[1.4 OBJETIVOS 11](#_Toc141582511)

[1.4.1 OBJETIVO GENERAL 11](#_Toc141582512)

[1.4.2 OBJETIVOS ESPECIFICOS 11](#_Toc141582513)

[1.5 ALCANCE 12](#_Toc141582514)

[1.5.1 Modulo Usuario. 12](#_Toc141582515)

[1.5.2 Modulo Personal. 12](#_Toc141582516)

[1.5.3 Modulo Gestión de Sindicato y rutas. 12](#_Toc141582517)

[1.5.4 Módulo de Gestión de Recursos de Transporte. 12](#_Toc141582518)

[1.5.5 Modulo reportes y estadísticas. 13](#_Toc141582519)

[1.6 METODOLOGIA 13](#_Toc141582520)

[1.7 Arquitectura del software 13](#_Toc141582521)

[2 CAPITULO II MARCO TEORICO EL SINDICATO 15](#_Toc141582522)

[2.1 Organización 15](#_Toc141582523)

[2.2 Afiliados 15](#_Toc141582524)

[2.3 CHOFERES 15](#_Toc141582525)

[2.4 Tipos de vehículos 15](#_Toc141582526)

[2.4.1 Trufi: 16](#_Toc141582527)

[2.4.2 Minivan. 16](#_Toc141582528)

[3 CAPÍTULO III HERRAMIENTA DE DESARROLLO DEL SOFTWARE 17](#_Toc141582529)

[3.1 HERRAMIENTAS DE DESARROLLO BACKEND Y BASES DE DATOS 17](#_Toc141582530)

[3.1.1 GESTOR DE BASE DE DATOS POSTGRESQL. 17](#_Toc141582531)

[3.1.2 ENTORNO DE DESARROLLO INTEGRADO JETBRAINS PHPSTORM. 17](#_Toc141582532)

[3.1.3 FRAMEWORK PHP LARAVEL. 18](#_Toc141582533)

[3.1.4 SISTEMA DE GESTIÓN DE PAQUETES PARA PHP COMPOSER. 20](#_Toc141582534)

[3.2 LENGUAJES DE PROGRAMACIÓN, HERRAMIENTAS DE DESARROLLO DE SOFTWARE. 20](#_Toc141582535)

[3.2.1 HTML. 20](#_Toc141582536)

[3.2.2 CSS. 21](#_Toc141582537)

[3.2.3 JAVASCRIPT. 21](#_Toc141582538)

[3.2.4 PHP. 22](#_Toc141582539)

[3.2.5 MATERIALIZECSS. 23](#_Toc141582540)

[3.2.6 DRAWIO.IO 23](#_Toc141582541)

[3.3 HERRAMIENTAS PARA LA GESTIÓN DE CONTROL DE VERSIONES 24](#_Toc141582542)

[3.3.1 GIT. 24](#_Toc141582543)

[3.3.2 GITHUB. 25](#_Toc141582544)

[3.4 ARQUITECTURA DE SOFTWARE 25](#_Toc141582545)

[3.4.1 MVC 25](#_Toc141582546)

[3.5 Metodología 26](#_Toc141582547)

[3.5.1 PROCESO Unificado del Desarrollo de Software (PUDS) 26](#_Toc141582548)

[3.5.2 Lenguaje unificado de modelado (UML) 28](#_Toc141582549)

[4 CAPITULO IV DISEÑO 30](#_Toc141582550)

[4.1 CAPTURA DE REQUISITOS 30](#_Toc141582553)

[4.1.1 IDENTIFICACIÓN DE ACTORES 30](#_Toc141582554)

[4.1.2 IDENTIFICACIÓN DE CASOS DE USO 31](#_Toc141582555)

[4.1.3 PRIORIZACIÓN DE LA LISTA DE REQUERIMIENTOS DE CASOS DE USO. 32](#_Toc141582556)

[4.1.4 DESCRIPCIÓN DE LOS CASOS DE USO. 33](#_Toc141582557)

[4.2 DIAGRAMA GENERAL DE CASOS DE USO 50](#_Toc141582558)

[4.3 MODELO DE DOMINIO INICIAL- DIAGRAMA DE CLASES 50](#_Toc141582559)

[5 CAPITULO V ANÁLISIS 52](#_Toc141582560)

[5.1 IDENTIFICACIÓN DE PAQUETES O MÓDULOS DEL SISTEMA 52](#_Toc141582562)

[5.2 Relación de módulos y casos de uso 52](#_Toc141582563)

[5.2.1 Relación modulo Administración de usuario. 52](#_Toc141582564)

[5.2.2 Relación modulo Personal. 53](#_Toc141582565)

[5.2.3 Relación modulo Gestión de sindicato y rutas. 53](#_Toc141582566)

[5.2.4 Relación modulo Gestión de recurso de transporte. 53](#_Toc141582567)

[5.2.5 Relación modulo reportes y estadísticas. 54](#_Toc141582568)

[5.3 REALIZACIÓN DE LOS CASOS DE USO - ANÁLISIS. 54](#_Toc141582569)

[5.3.1 Clases de Interfaz: 54](#_Toc141582570)

[5.3.2 Clases de Entidad: 54](#_Toc141582571)

[5.3.3 Clases de Control: 55](#_Toc141582572)

[5.4 ANÁLISIS DE CASOS DE USO 55](#_Toc141582573)

[5.4.1 Caso de uso Gestionar Usuario 55](#_Toc141582574)

[5.4.2 Caso de uso Gestionar Personal 56](#_Toc141582575)

[5.4.3 Caso de uso Gestionar Tipo Personal 58](#_Toc141582576)

[5.4.4 Caso de uso Gestionar Sindicato 59](#_Toc141582577)

[5.4.5 Caso de uso Gestionar Vehículo 60](#_Toc141582578)

[5.4.6 Caso de uso Gestionar Ruta 61](#_Toc141582579)

[5.4.7 Caso de uso Gestionar Mantenimiento 62](#_Toc141582580)

[5.4.8 Caso de uso Gestionar Promoción 63](#_Toc141582581)

[6 CAPITULO VI DISEÑO 64](#_Toc141582582)

[6.1 Introducción 64](#_Toc141582583)

[6.2 Diseño de la arquitectura del diagrama de paquetes 64](#_Toc141582584)

[6.2.1 Diseño de la arquitectura general del sistema 64](#_Toc141582585)

[6.3 Diseño procedimental- Diagrama de secuencias 64](#_Toc141582586)

[6.3.1 Diagrama de Secuencia: Gestionar Usuario 65](#_Toc141582587)

[6.3.2 Diagrama de Secuencia: Gestionar Personal 66](#_Toc141582588)

[6.3.3 Diagrama de Secuencia: Gestionar TipoPersonal 67](#_Toc141582589)

[6.3.4 Diagrama de Secuencia: Gestionar Sindicato 68](#_Toc141582590)

[6.3.5 Diagrama de Secuencia: Gestionar Vehículo 69](#_Toc141582591)

[6.3.6 Diagrama de Secuencia: Gestionar Ruta 70](#_Toc141582592)

[6.3.7 Diagrama de Secuencia: Gestionar Mantenimiento 71](#_Toc141582593)

[6.3.8 Diagrama de Secuencia: Gestionar Promoción 72](#_Toc141582594)

[6.4 Diseño físico de la arquitectura. 73](#_Toc141582595)

[6.4.1 Diagrama de despliegue. 73](#_Toc141582596)

[6.5 Identificar paquetes 74](#_Toc141582597)

[6.6 Diseño conceptual de la base de datos 75](#_Toc141582598)

[6.6.1 Diagrama de clases de la base de datos. 75](#_Toc141582599)

[6.7 Diseño lógico 75](#_Toc141582600)

[6.8 Diseño físico 77](#_Toc141582601)

[6.9 Creación de tablas en la base de datos. 83](#_Toc141582602)

[6.10 Diseño de interfaces gráficas para el usuario 95](#_Toc141582603)

[7 BIBLIOGRAFÍA 96](#_Toc141582604)

1. **CAPITULO I PERFIL DEL PROYECTO**

## Introducción

En la actualidad, muchos países ya cuentan con sistemas de control de una asociación de una empresa de transporte, el cual facilita el arduo registro de cada uno de los personales, vehículos, así mismo poder llevar control de los aspectos más importantes en cualquier día, a cualquier hora, además proporciona la facilidad de consulta de los datos que se desea saber en un momento preciso. El registro de todos los datos es muy importante, ya que mediante la revisión de esos datos se puede tener un control para poder tomar decisiones óptimas que ayuden de manera eficiente en el futuro.

El sindicato de transporte 27 de diciembre en general actualmente maneja sus procesos a través del método de archivo de documentos (facturas, cheques, horarios en papel etc.), pero la información allí contenida está expuesta a diferentes peligros derivados de la conservación y seguridad ya que en algún momento podrían perderse, mezclarse con otros documentos o simplemente deteriorarse por las condiciones ambientales y del medio en general. Por lo anterior con el actual desarrollo e implementación de herramientas Web, las cuales tienen la enorme facilidad de permitir actualizar los datos de forma inmediata, utilizando además de las nuevas tecnologías (Tablet, celulares, portátiles), se dará paso a un servicio tecnológico que esté al alcance de cualquier usuario que tenga los privilegios correspondientes del SINDICATO DE TRANSPORTE 27 DE DICIEMBRE.

Es el caso del “sindicato de transporte 27 DE DICIEMBRE” en la cual, opta de manera acertada el desarrollo de Sistemas de Información que les brinden la información que ella necesite y a la vez solucionar problemas internos que va en contra de la visión de la empresa, y a través del Sistema de Información que se va a plantear, se pretende obtener beneficios tanto económicos como competitivos. Lo cual hará que la Sindicato de transporte 27 DE DICIEMBRE sea más eficiente y eficaz en llevar el control de su asociación, ya que ellas estarán un paso atrás en lo que se refiere a manejo efectivo de la información y toma de decisiones con información confiable y en tiempo real.

## Antecedentes y justificación

### Antecedentes

**Historia**

El Sindicato de transporte 27 DE DICIEMBRE fue fundada por el señor Mijael Zúñiga Montaño aproximadamente a inicios del año 2020.

Desde que empezó el funcionamiento, el sindicato no cuenta con un sistema de información que sea capaz de registrar el flujo de control de su gestión de manera óptima.

El sindicato de transporte brinda su servicio a la población para que puedan viajar de manera confiable, y la visión es poder integrar un grupo de profesionales en el tema de viaje de transporte y asi mismo brindar un buen servicio los clientes, con responsabilidad social y ética de valores, que sirvan e inspiren confianza entre los afiliados.

El sindicato de transporte 27 DE DICIEMBRE, atiende de lunes a viernes de 08:00 AM. hasta 22:00 PM y los sábados y domingos de 08:00 AM a 13:00 PM.

A nivel local, no se ha encontrado documentación de un proyecto desarrollado para control de un sindicato de transporte.

Por tanto, el presente proyecto es viable y pretende satisfacer algunas de las necesidades principales en la gestión del sindicato de transporte 27 de diciembre como ser: Control del personal, vehículos, los mantenimientos, las rutas con las que cuenta, sus promociones, así como sus respectivos afiliados.

**Ubicación**

El sindicato de Transporte 27 de diciembre se encuentra ubicado en la ciudad de Villamontes con coordenadas de -21.266107277708123, -63.459996366182395, Ruta 9, Villa Montes.



**Figura 1**: Ubicación vía satélite del Sindicato de Transporte 27 de diciembre

Fuente: Google Max (2023)



**Figura 1**: Instalaciones del Sindicato de Transporte 27 de diciembre

Fuente: fotografía propia (2023)

### Justificación

Mediante el presente proyecto se pretende mejorar el registro y control de todas las funcionalidades que la asociación de sindicato “27 DE DICIEMBRE” realiza para posteriormente tener un acceso rápido a todos los datos e información que necesite obtener los usuarios que interactuaran con el sistema de información web, para un buen control de las actividades del sindicato de transporte y así mismo se pretende tener un buen registro del personal que es parte de la asociación.

## Planteamiento del problema

Actualmente el sindicato de transporte 27 DE DICIEMBRE no tiene un manejo de sus actividades de manera sistematizada, no cuenta con un sistema de información computarizada para administrar la información que generan, los registros de personal, información de tarifas, horarios, vehículos y datos relevantes lo manejan manualmente causando que se ponga en riesgo la integridad y confiabilidad del sindicato y sea vulnerable a factores de peligro como la perdida de información relevante.

Debido a estos problemas mencionados anteriormente, se vio la necesidad de desarrollar un sistema de información web que gestione y administre de manera automatizada y sistematizada la información que genera y maneja el sindicato de transporte 27 DE DICIEMBRE, con sus respectivos servicios que brinda y las actividades que realiza para el cumplimiento de su gestión, de esa forma facilitar y reducir el tiempo que se demora en llevar a cabo el registro de cada una de las funciones que cumple.

## OBJETIVOS

### OBJETIVO GENERAL

Desarrollar un Sistema de información web para el control del Sindicato de Transporte 27 de diciembre ubicada en la ciudad de Villamontes.

### OBJETIVOS ESPECIFICOS

* Realizar el levantamiento de información necesaria para hacer la recolección de datos sobre la organización y el funcionamiento del Sindicato de Transporte 27 de diciembre.
* Elaborar los requerimientos a través de la captura de requisitos, así mismo analizar las funcionalidades del sistema mediante diagramas de casos de usos y diagramas de clases.
* Elaborar el análisis correspondiente, como el empaquetado y diseño de los diferentes módulos en base a los casos de uso y requerimientos definidos para representar las funcionalidades del sistema.
* Implementar el sistema haciendo uso del lenguaje de alto nivel (PHP, Python) y gestor de base de datos (PostgreSQL).
* Realizar pruebas necesarias para verificar su correcto funcionamiento y corregir los posibles errores y comparar el nivel de los requerimientos que cumple con el usuario final.

## ALCANCE

### Modulo Usuario.

* Gestionar usuario.

Este módulo tiene funcionalidades relacionadas con la gestión de usuarios (propietario, secretaria, chofer). Incluiría la creación, actualización y eliminación de usuarios, así como la gestión de permisos y roles.

### Modulo Personal.

* Gestionar personal.
* Gestionar tipo personal

El usuario con los permiso y privilegios en el sistema podrá asignar los cargos a los diferentes personales con lo que cuenta el sindicato de transporte 27 de diciembre, así como creación, actualización de datos y eliminación y consulta de los mismos.

### Modulo Gestión de Sindicato y rutas.

* Gestionar sindicato.
* Gestionar rutas.
* Gestionar promociones

El módulo Sindicato permite gestionar las actividades relacionadas con el sindicato y sus líneas. En este módulo, se registrarán los datos del sindicato, se administrarán las líneas o secciones, y se realizará el seguimiento de la información correspondiente, así como creación, actualización, eliminación o búsquedas de líneas aliadas. Además, Tiene funcionalidades relacionadas con la gestión de promoción, permitirá llevar la administración de creación, actualización y eliminación de promociones especiales

### Módulo de Gestión de Recursos de Transporte.

* Gestionar vehículo.
* Gestionar mantenimiento.

El módulo incluye el control y manejo de información importante relacionada con la gestión de vehículos, incluiría la creación, actualización y eliminación de vehículos, así como la consulta de información de los mismos también están presentes las funcionalidades relacionadas con el mantenimiento de los vehículos, incluiría la programación, seguimiento y registro de tareas de mantenimiento de los vehículos.

### Modulo reportes y estadísticas.

* Gestionar reportes y estadísticas

Este módulo sus funcionalidades están con la generación de informes y estadísticas basados en los datos filtrados del sistema

## METODOLOGIA

La metodología de desarrollo que utilizaremos en este proyecto se basa en el Proceso Unificado de Desarrollo Software (PUDS) y se apoya en el Lenguaje de Modelado Unificado (UML), además define un conjunto de prácticas y roles y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutara durante el proyecto. Esta metodología se caracteriza:

* Por ser guiada por casos de uso.
* Es iterativo e incremental.
* Está centrado en la arquitectura.

Su objetivo es proporcionar un marco de trabajo estructurado y prácticas para el desarrollo de software eficiente y de alta calidad. Esta metodología proporciona un enfoque estructurado y flexible para el desarrollo del proyecto, permitiendo una mejor comprensión de los requisitos, una arquitectura sólida y una implementación incremental del sistema.

## Arquitectura del software

Se va a utilizar la arquitectura de software MVC, que actualmente se ocupa en su gran mayoría en proyectos web que implican hacer aplicaciones grandes con comunicación a bases de datos. Este permite dividir el desarrollo en tres componentes que se describen como: MVC (Model, View, Controller) ó Modelo, Vista y Controlador.

Para lograr el desarrollo del presente proyecto se va a utilizar:

* **Laravel** (un framework del lenguaje de programación PHP).
* **PostgreSQL** (un sistema de gestión de bases de datos relacionales).
* **HTML** (documento de etiquetas para enmaquetado web).
* **CSS** (lenguaje de estilos para documentos web).
* **JavaScript** (lenguaje de programación web).
* **MaterializeCss** (framework para diseño basado en MaterialDesign de Google).
* **Drawio.io** (aplicación para modelar diagramas UML).
* **Git** (sistema de control de versiones).
* **GitHub** (repositorio remoto para alojar proyectos con git).

# CAPITULO II MARCO TEORICO EL SINDICATO

## Organización

El Sindicato de Trufis y Minivanes “27 de Diciembre” a través de sus afiliados con vehículos propios, cuenta con las siguientes modalidades de servicio de transporte público:

* Trufis
* Minivan

Este, no puede dedicarse a ningún otro rubro más que a los mencionados anteriormente, ya que así lo definieron.

## Afiliados

Las personas que cumplan con los requisitos, obligaciones económicas y posean uno o dos vehículos de su propiedad, pueden formar parte del sindicato como afiliados del mismo.

Para que estas personas permanezcan como afiliados deben cumplir con las cuotas mensuales y otras obligaciones que son de carácter sindical. En caso de que el afiliado haya incumplido con las obligaciones en un rango de tiempo notable, automáticamente pierde su derecho de afiliado de manera indefinida.

## CHOFERES

Para ser miembro del personal de choferes del sindicato se deben cumplir con ciertos requisitos como: Ser boliviano(a) de nacimiento, tener como mínimo 21 años de edad, tener una experiencia para conducir cualquier tipo de vehículo, no ser dirigente en un partido político ni ocupar cargos públicos, ser chofer profesional y así mismo contar con su respectiva licencia de conducir.

## Tipos de vehículos

El sindicato en su sector de vehículos, disponen de dos tipos de vehículos los cuales serán definidos para mayor claridad:

### Trufi:

Es un vehículo de transporte público con capacidad para llevar en promedio hasta 10 a 15 pasajeros, aunque esto puede variar según el modelo y la marca del vehículo. los trufis ofrecen un servicio de transporte compartido, proporcionando una alternativa económica para los usuarios que necesitan desplazarse por la ciudad o entre diferentes localidades.

### **MINIVAN.**

Es un medio de transporte diseñado para el desplazamiento de personas o carga. puede ser impulsado por gasolina y gas, aquí se presentan las características más destacadas del vehículo:

Están diseñados para transportar personas, mercancías o ambos, utilizados para viajes personales, transporte de viaje público, reparto de mercancías, entre otras aplicaciones.

La capacidad del vehículo puede variar significativamente según su tipo y uso. Los vehículos suelen tener capacidad para unas 5 a 6 personas.

# CAPÍTULO III HERRAMIENTA DE DESARROLLO DEL SOFTWARE

## **HERRAMIENTAS DE DESARROLLO BACKEND Y BASES DE DATOS**

### **GESTOR DE BASE DE DATOS POSTGRESQL**.

PostgreSQL es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Las características Según (PostgreSQL, 2008) :

* Corre en casi todos los principales sistemas operativos : Linux, Unix, BSDs, Mac OS,Beos, Windows, etc. (34)
* Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
* Comunidades muy activas, varias comunidades en castellano.
* Bajo Costo de Propiedad Total (TCO) y rápido Retorno de la Inversión Inicial (ROI)
* Altamente adaptable a las necesidades del cliente.

### **ENTORNO DE DESARROLLO INTEGRADO JETBRAINS PHPSTORM**.

Es un software de propiedad de la empresa JetBrains, que está optimizado para desarrollar aplicaciones web en lenguajes como: PHP, JavaScript, CSS y HTML.

Lo describe a detalle (PHPStorm, 2020) de la siguiente manera:

Es un IDE (entorno de desarrollo integrado) escrito en Java, comercial multiplataforma con análisis de código “refactoring” para PHP, HTML, JavaScript, creado por la empresa JetBrains con sede en la República Checa, compatible con los sistemas operativos Linux, Solaris, MacOS y Windows, su licencia es de suscripción de pago con una versión para estudiantes gratuita por un año, renovable.

Características:

* Sistema de control de versiones incluida.
* Auto-completado de namespace, clases y objetos y nombres duplicados.
* Soporte de estilo de codificación PSR1 / PSR2, Drupal, Symfony, Zend.
* Conexión de variables con base datos PostgreSQL, MySQL, MariaDB.
* Xdebug herramienta parar testing con PHPUnit.
* Soporte de pestañas en el terminal incluido.

### **FRAMEWORK PHP LARAVEL**.

Laravel es un popular framework PHP. En estos momentos es el framework más utilizado y de mayor progresión en la comunidad PHP y en general también es el más popular de los frameworks backend para desarrollo de proyectos en la web.

Laravel tiene un enfoque bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web y su mantenimiento. Como todo framework nos aporta una base de código para el desarrollo de proyectos. Esa base ofrece principalmente dos ventajas:

Una arquitectura depurada, que nos anima a seguir para crear proyectos con unos estándares de calidad elevados.Una serie de bibliotecas de utilidad diversa que nos ayudarán en el desarrollo de áreas habituales dentro de las aplicaciones.

Algunas características mencionas por (startechup, s.f.) son:

* Arquitectura MVC: Laravel adopta el patrón arquitectónico MVC (Modelo-Vista-Controlador), que aporta estructura y escalabilidad al desarrollo de aplicaciones web. Con MVC, los desarrolladores pueden crear bases de código bien organizadas y mantenibles para aplicaciones de cualquier tamaño.Siguiendo el enfoque MVC, Laravel permite un desarrollo eficiente. Varios desarrolladores pueden trabajar simultáneamente en diferentes componentes (vistas y controladores), lo que agiliza el proceso de desarrollo. El framework también soporta múltiples vistas para un modelo y asegura que la duplicación de código se minimiza mediante la separación de la lógica de negocio de la lógica de presentación.
* Autenticación y autorización: Implementar la autenticación y autorización de usuarios en Laravel es increíblemente fácil. El framework proporciona un andamiaje de autenticación integrado, funcionalidad de restablecimiento de contraseñas y compatibilidad perfecta con permisos basados en roles y control de acceso. Con Laravel, puedes configurar la autenticación y autorización para tu aplicación web con sólo unos pocos comandos Artisan. El framework ofrece soluciones fiables y robustas listas para usar, ahorrándote tiempo y esfuerzo en la implementación de estas características de seguridad cruciales.
* Interfaz de línea de comandos (CLI) de Artisan: Artisan CLI de Laravel es una herramienta integrada de línea de comandos diseñada para manejar tareas de programación repetitivas que de otro modo requerirían intervención manual.Con la ayuda de Artisan CLI de Laravel, el desarrollo web se convierte en un juego de niños. Esta potente herramienta de línea de comandos ofrece características como la generación de código y la gestión eficiente de bases de datos, todo a sólo unos comandos de distancia. En lugar de estancarse escribiendo código repetitivo o configurando manualmente una base de datos, los desarrolladores pueden centrarse en crear la lógica y la funcionalidad de la aplicación.
* Mapeador Objeto-Relacional (ORM) Eloquent: El framework Laravel incluye Eloquent, un impresionante ORM que simplifica la gestión de bases de datos. Eloquent ofrece una forma sencilla y directa de interactuar con una base de datos. En Laravel, cada tabla de base de datos tiene un modelo asociado, lo que permite a los desarrolladores interactuar con la tabla fácilmente. Con Eloquent, no solo puedes recuperar registros de una tabla, sino que también puedes insertar, actualizar y eliminar registros utilizando un modelo de Eloquent.
* Motor de plantillas Blade: Blade, el motor de plantillas PHP proporcionado por Laravel, es una valiosa herramienta para crear vistas dinámicas y reutilizables. Ofrece una serie de características, incluyendo la herencia de plantillas, estructuras de control, y la perfecta integración de código PHP dentro de HTML. Blade juega un papel crucial en Laravel y ayuda a mantener una clara separación entre la lógica de negocio y las plantillas HTML. A diferencia de otros motores de plantillas, Blade destaca por permitir el uso de código PHP plano junto con sus características únicas.
* Bibliotecas orientadas a objetos: La popularidad de Laravel se debe en parte a su inclusión de librerías orientadas a objetos y preinstaladas que no se encuentran en otros frameworks PHP. Su biblioteca de autenticación, por ejemplo, ofrece funciones como el restablecimiento de contraseñas, la supervisión activa de usuarios, el hashing Bcrypt y la protección CSRF. El framework también es modular y sigue los principios de PHP, lo que facilita a los desarrolladores la creación de aplicaciones modulares y con capacidad de respuesta.

### **SISTEMA DE GESTIÓN DE PAQUETES PARA PHP COMPOSER**.

Es un administrador de instalación de dependencias parar Laravel y proyectos en PHP 5,3,2 o superior, compatible con los sistemas operativos Linux, MacOS y Windows. Desarrollado por Nils Adermann y Jordi Boggiano, licencia por el MIT, de libre uso (Composer, 2021).

## **LENGUAJES DE PROGRAMACIÓN, HERRAMIENTAS DE DESARROLLO DE SOFTWARE**.

### HTML.

Es importante aclarar que este lenguaje no es de programación, sino que es de marcado, es decir su principal función es maquetar a través de etiquetas una página web, con éste solo se crea la estructura en el que posteriormente se podrá añadir diseños y estilos con otra herramienta específico para ese fin.

(HTML, 2023)Lo define:

HTML, siglas en inglés de HyperText Markup Language (‘lenguaje de marcado de hipertexto’), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.HTML es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas. Este lenguaje nos ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpre­tar tanto por humanos como por máquinas.

### **CSS**.

Según (CSS, 2023) es definido de la siguiente manera:

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.2 Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS). CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes.4 Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

### JAVASCRIPT.

El autor (JavaScript, 2023) lo describe como:

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,2 basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas3 y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio de 2015.4

JavaScript se diseñó con una sintaxis similar a C++ y Java,56 aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes. Su relación es puramente comercial, tras la compra del creador de Java (Sun Microsystems) de Netscape Navigator (creador de LiveScript) y el cambio de nombre del lenguaje de programación.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). Javascript es el único lenguaje de programación que entienden de forma nativa los navegadores.

### PHP.

La definición de (PHP, 2023) es la siguiente:

PHP es un lenguaje de programación interpretado2 del lado del servidor y de uso general que se adapta especialmente al desarrollo web.3 Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994.4 En la actualidad, la implementación de referencia de PHP es producida por The PHP Group.5 PHP originalmente significaba Personal Home Page (Página personal), pero ahora significa el inicialismon 1 recursivo PHP: Hypertext Preprocessor.7

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta. Por otra parte, PHP puede utilizarse para muchas tareas de programación fuera del contexto de la web, como aplicaciones gráficas autónomas8 y el control de drones.9 También se puede interpretar y ejecutar un código PHP cualquiera a través de una interfaz de línea de comandos (CLI).

El intérprete estándar de PHP, impulsado por Motor Zend, es un software libre publicado bajo Licencia PHP. PHP ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi todos los sistemas operativos y plataformas, de forma gratuita.

### MATERIALIZECSS.

Es un framework que tiene como objetivo diseñar de forma eficaz y rápida las páginas web, además de que los diseños sean modernos y responsivos para todos los dispositivos.

El sitio oficial (MaterializeCSS, 2023) lo describe:

Material Design es un lenguaje de diseño que combina los principios clásicos del diseño exitoso junto con la innovación y la tecnología. El objetivo de Google es desarrollar un sistema de diseño que permita una experiencia de usuario unificada en todos sus productos en cualquier plataforma.

### DRAWIO.IO

Draw.io es un software utilizado para diseñar diagramas de forma gratuita y offline, aunque también tiene una versión completamente funcional en el navegador web, y además, facilita la integración con múltiples plataformas y programas. Esta herramienta permite realizar cualquier tipo de diagrama de flujo, diagramas de procesos, organigramas, así como diagramas de red, UML, mapas conceptuales y otros elementos necesario para realizar un diseño.

Las características según (keepcoding.io, 2023) se describe como:

* Esta aplicación web puede funcionar en un gran número de navegadores. Está desarrollada principalmente en el lenguaje de programación JavaScript y cuenta con la Apache License en su versión 2.0.
* Draw.io se caracteriza por ser un programa de fácil funcionamiento, debido a que facilita el uso de herramientas básicas y sencillas como el sistema drag and drop, también conocido como arrastrar y soltar, y que se utiliza para llevar formas, imágenes, flechas y otros elementos de manera rápida. También facilita la creación de colecciones de diagramas, así como imágenes personalizadas que puedes utilizar en los diagramas diseñados.
* La arquitectura de esta herramienta permite el diseño de diagramas de tipo profesional, administrativo y académico y está disponible en línea, lo que le permite al usuario almacenar sus diagramas y trabajos en la nube de Google Drive, OneDrive o en el archivo local.
* Esta aplicación de creación de diagramas también está disponible para la descarga en su versión offline, que permite el trabajo sin conexión, a través de su descarga previa para el uso en plataformas como Windows, Mac Os, Linux y el entorno GNU, Chrome OS, entre otros.

## **HERRAMIENTAS PARA LA GESTIÓN DE CONTROL DE VERSIONES**

### GIT.

El sitio oficial (atlassian, 2023) lo describe:

Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados).Git, que presenta una arquitectura distribuida, es un ejemplo de DVCS (sistema de control de versiones distribuido, por sus siglas en inglés). En lugar de tener un único espacio para todo el historial de versiones del software, como sucede de manera habitual en los sistemas de control de versiones antaño populares, como CVS o Subversion (también conocido como SVN), en Git, la copia de trabajo del código de cada desarrollador es también un repositorio que puede albergar el historial completo de todos los cambios.

Además de contar con una arquitectura distribuida, Git se ha diseñado teniendo en cuenta el rendimiento, la seguridad y la flexibilidad.

### GITHUB.

Conocido también como la red social de los programadores, es una plataforma que gestiona repositorios de Git, que pueden estar con visibilidad pública (gratis) o privada (de pago), lo cual, siendo pública, muchos otros desarrolladores lo pueden ver y aportar si así lo desean, esto es un gran factor de apoyo para el desarrollo Open Source.

El sitio oficial (GitHub, 2023) lo describe:

Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

## ARQUITECTURA DE SOFTWARE

### MVC

Es un modelo de arquitectura de software que actualmente se ocupa en su gran mayoría en proyectos web que implican hacer aplicaciones grandes con comunicación a bases de datos. Este permite dividir el desarrollo en tres componentes que se describen como: MVC (Model, View, Controller) ó Modelo, Vista y Controlador.

Según (Wikipedia M. , 2020) se define como:

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

## Metodología

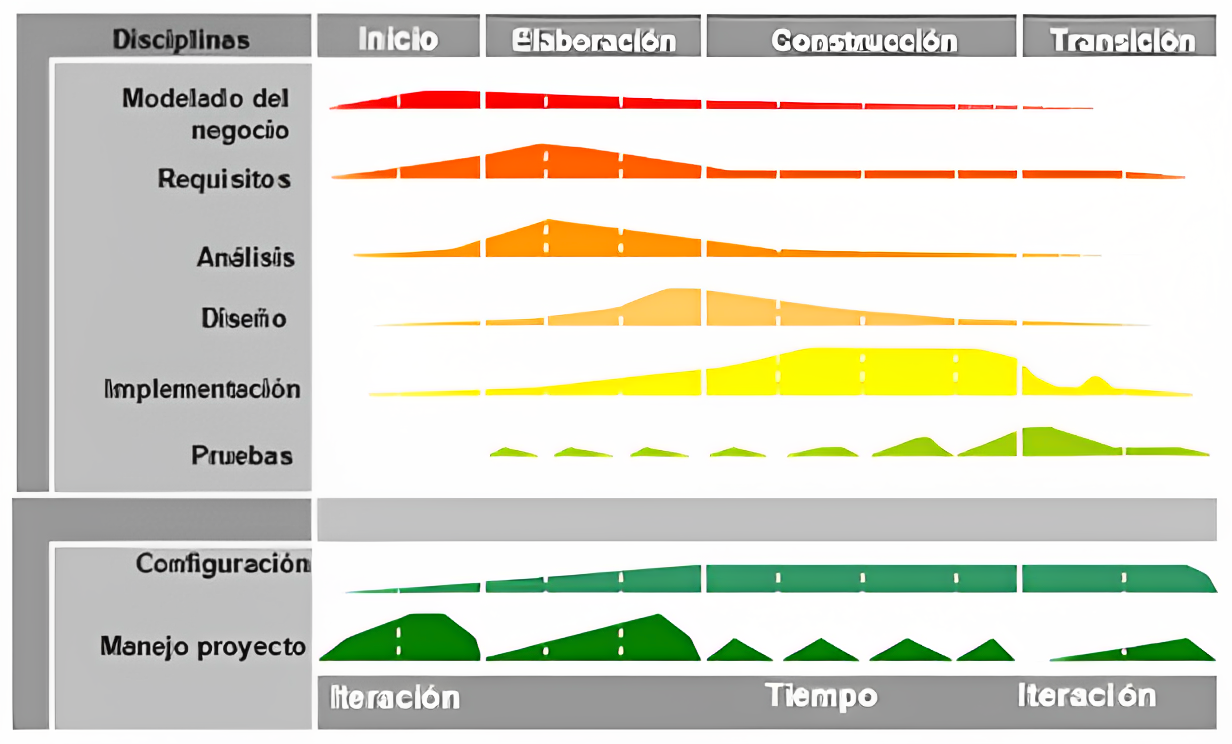
### PROCESO Unificado del Desarrollo de Software (PUDS)

El proceso Unificado de Desarrollo de Softare según (Wikipedia, 2023) dice:

El Proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational (RUP) o simplemente UP.El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

El nombre Proceso Unificado se usa para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes. También permite evitar problemas legales ya que Proceso Unificado de Rational o RUP son marcas registradas por IBM (desde su compra de Rational Software Corporation en 2003). El primer libro sobre el tema se denominó, en su versión española, El Proceso Unificado de Desarrollo de Software (ISBN 84-7829-036-2) y fue publicado en 1999 por Ivar Jacobson, Grady Booch y James Rumbaugh, conocidos también por ser los desarrolladores del UML, el Lenguaje Unificado de Modelado. Desde entonces los autores que publican libros sobre el tema y que no están afiliados a Rational utilizan el término Proceso Unificado, mientras que los autores que pertenecen a Rational favorecen el nombre de Proceso Unificado de Rational

Ilustración . *Ciclo de Vida de la metodología PUDS.*



*Fuente: Wikipedia.*

Características

Esta metodología se caracteriza según (Wikipedia, 2023) :

* Iterativo e Incremental: El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio puede incluir varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.
* Dirigido por los casos de uso: En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. El proceso dirigido por casos de uso es el rup. Nota: en UP se está Dirigido por requisitos y riesgos de acuerdo con el Libro UML 2 de ARLOW, Jim que menciona el tema.
* Centrado en la arquitectura: El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.
* Enfocado en los riesgos: El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

**Sus Fases son:**

* Fase de Inicio: Es la fase más pequeña del proyecto e, idealmente, debe realizarse también en un periodo de tiempo pequeño (una única iteración).
* Fase de Elaboración: Durante esta fase se capturan la mayoría de los requisitos del sistema. Los objetivos principales de esta fase serán la identificación de riesgos y establecer y validar la arquitectura del sistema.
* Fase de construcción: Es la fase más larga de proyecto. El sistema es construido en base a lo especificado en la fase de elaboración. Las características del sistema se implementan en una serie de iteraciones cortas y limitadas en el tiempo.
* Fase de transición: En esta fase el sistema es desplegado para los usuarios finales. La retroalimentación recibida permite incorporar refinamientos al sistema en las sucesivas iteraciones. Esta iteración también cubre el entrenamiento de los usuarios para la utilización del sistema.

### Lenguaje unificado de modelado (UML)

Este es un complemento del proceso unificado de desarrollo, que ayuda en realizar un modelo visual de los requerimientos y a través de ellos comprender mejor el camino para desarrollo del software.

Según (Wikipedia C. d., 2021) es definido UML, como:

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Existen dos clases principales de tipos de diagramas. Algunos de ellos son según (Wikipedia C. d., 2021) ::

* Diagramas Estructurales
* Diagrama de clases
* Diagrama de paquetes.
* Diagramas De comportamiento
* Diagrama de casos de uso.
* Diagrama de secuencia.

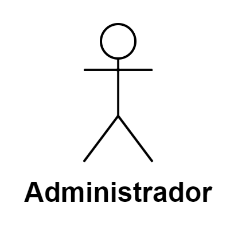
# CAPITULO IV DISEÑO

## CAPTURA DE REQUISITOS

La captura de requisitos es el primer paso a cumplir con el objetivo general descrito anteriormente. Siguiendo la metodología descrita en el Proceso Unificado de Desarrollo del software se procederá a la identificación y descripción de requisitos, actores y casos de uso, los cuales proporcionarán la funcionalidad del sistema y los actores que interactúan con el mismo.

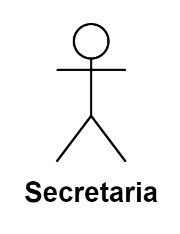
### IDENTIFICACIÓN DE ACTORES

Administrador



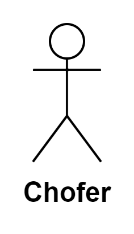
Este actor es el que más interactúa con el sistema, se le asigna la mayoría de las funcionalidades en el sistema, encargado de gestionar al personal del sindicato y organizar sus respectivos horarios de trabajo, cuenta con la capacidad para poder administrar y llevar control de la información que maneja el sistema.

Secretaria



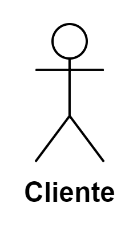
Este actor interactúa con el sistema, es el responsable de gestionar información en cuanto a los vehículos, personal, mantenimientos, promociones tarifas y las líneas pertenecientes al sindicato de transporte 27 de diciembre, podrá tener las funcionalidades de gestionar a excepción de la acción eliminar, esta acción estará restringida para este actor.

Chofer



Persona que cumple un rol dentro del sindicato de transporte, cuenta con un salario y sus respectivas obligaciones, se le asignará funciones a este actor, pero con restricciones en las acciones que muestra el sistema.

Cliente



Este actor interactúa con el sistema haciendo consultas sencillas, no permitiendo que tenga permisos para funcionalidades relevantes del sistema, es decir tendrá restricciones y limitaciones al momento de utilizarlo.

### IDENTIFICACIÓN DE CASOS DE USO

CU1. Gestión de usuarios (Administrador, secretaria, Chofer, cliente).

CU2. Gestionar Vehículos.

CU3.Gestionar rutas.

CU4. Gestionar personal.

CU5.Gestionar tipo\_personal.

CU6. Gestionar Mantenimiento.

CU7. Gestionar sindicato.

CU8. Gestionar promociones.

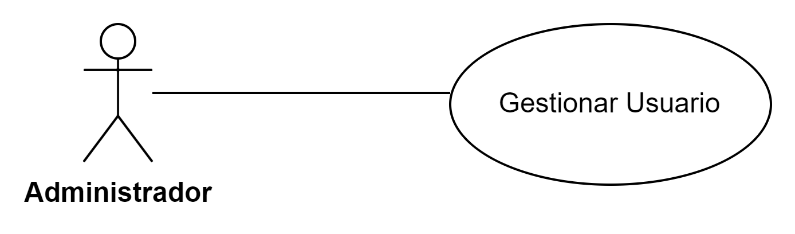
CU9. Gestionar Reportes y Estadísticas.

### PRIORIZACIÓN DE LA LISTA DE REQUERIMIENTOS DE CASOS DE USO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nro. | Nombre | Estado | Prioridad | Riesgo |
| R1 | Gestionar Usuarios. | Aprobado | Alto | Critico |
| R2 | Gestionar Vehículos. | Aprobado | Alto | Crítico |
| R3 | Gestionar Rutas. | Aprobado | Alto | Normal |
| R4 | Gestionar Personal. | Aprobado | Alto | Crítico |
| R5 | Gestionar Tipo Personal | Aprobado | Alto | Normal |
| R6 | Gestionar Mantenimiento. | Aprobado | Alto | Normal |
| R7 | Gestionar Sindicatos. | Aprobado | Alto | Normal |
| R8 | Gestionar Promociones. | Aprobado | Medio | Normal |
| R9 | Gestionar Reportes y Estadísticas. | Aprobado | Medio | Normal |

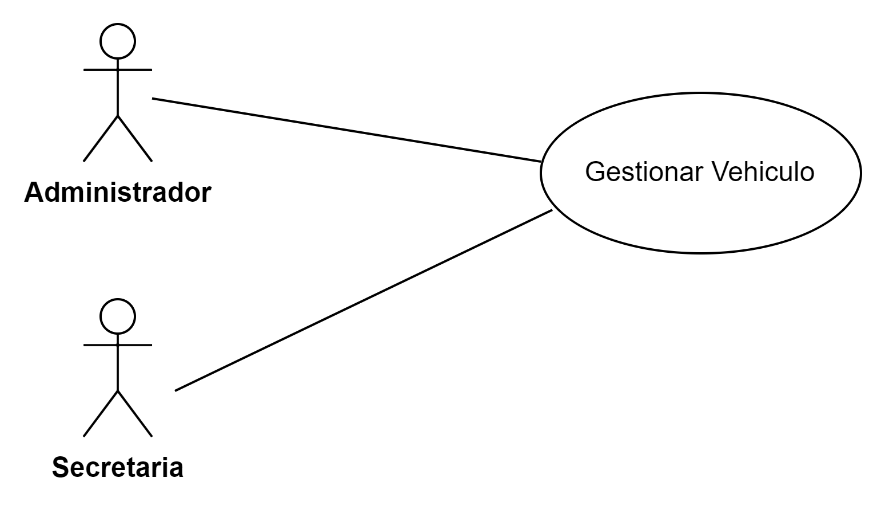
### DESCRIPCIÓN DE LOS CASOS DE USO.

CU1. Gestionar usuario



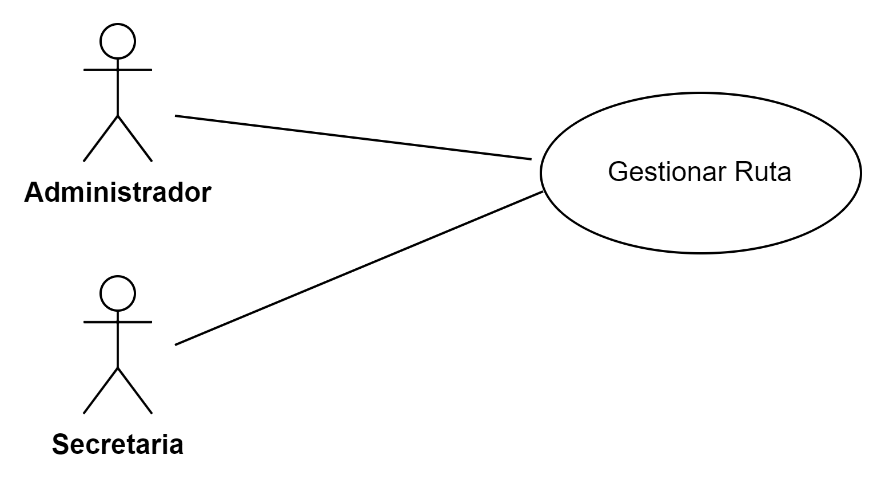
|  |  |
| --- | --- |
| **Gestionar Usuario** | |
| **ID Caso de Uso** | CU-1 |
| **Actor** | Administrador |
| **Descripción** | Encargado de gestionar las cuentas de los usuarios que interactuarán con el sistema web para el funcionamiento del sistema web para el sindicato de transporte 27 de diciembre. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar, datos de los usuarios correspondiente. |
| **Precondiciones** |  |
| **postcondiciones** | Todos los registros y cambios hecho por este actor han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Registrar**   Acción del actor   * Rellena datos de los usuarios a crear en los campos correspondientes. * Guardar cuenta de usuario   Respuesta del sistema   * Validar datos si no hay errores. * Guarda cuenta de usuario. * **Actualizar**   Acción del actor   * Selecciona al usuario a modificar   Respuesta del sistema   * Verifica si la cuenta de usuario está registrada * Si esta registrado permite modificar los datos de la cuenta de usuario buscada.   Acción del actor   * Modifica los datos. * Guardar los cambios.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Buscar la cuenta de usuario a eliminar.   Respuesta del sistema   * Verifica si la cuenta esta creada o registrada. * Si esta creada permite eliminar permanentemente la cuenta del usuario.   Acción del actor   * Eliminar cuenta de usuario.   Respuesta del sistema   * Elimina cuenta de usuario y actualiza la base de datos. |

CU2. Gestionar vehículo.



|  |  |
| --- | --- |
| **Gestionar Vehículos.** | |
| **ID Caso de Uso** | CU-2 |
| **Actor** | Administrador.  Secretaria. |
| **Descripción** | Este actor será encargado de gestionar información de vehículos en el sindicato de transporte. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador.  Insertar/Actualizar/Listar datos, para actor: Secretaria. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El usuario pulsa sobre el botón de registrar vehículo.   Respuesta del sistema   * El sistema muestra campos de texto para introducir los datos del registro.   Acción del actor.   * Introduce toda la información necesaria para el registro del vehículo nuevo. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Buscar el vehículo a modificar por placa o modelo.   Respuesta del sistema   * Verifica si el vehículo está registrado * Si esta registrado permite modificar los datos.   Acción del actor   * Modifica los datos del vehículo. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Buscar al vehículo a eliminar.   Respuesta del sistema   * Verifica si el vehículo está registrado * Si esta registrado permite eliminar permanentemente al vehículo buscado con toda su información existente.   Acción del actor   * Apreta el botón Eliminar   Respuesta del sistema   * Elimina información de vehículo y actualiza la base de datos. |

CU3. Gestionar rutas.



|  |  |
| --- | --- |
| **Gestionar rutas** | |
| **ID Caso de Uso** | CU-3 |
| **Actor** | Administrador.  Secretaria. |
| **Descripción** | Gestiona las rutas de viaje que dispone el sindicato de transporte 27 de diciembre. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador.  Insertar/Actualizar/Listar datos, para actor: Secretaria. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El usuario pulsa sobre el botón de registrar.   Respuesta del sistema   * El sistema muestra campos de texto para introducir los datos necesarios para el registro.   Acción del actor.   * Introduce toda la información necesaria para el registro de la ruta nueva. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Modificar**   Acción del actor   * Buscar ruta a modificar.   Respuesta del sistema   * Verifica si la ruta está registrada * Si está registrada permite modificar los datos.   Acción del actor   * Modifica los datos de la ruta. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Buscar ruta a eliminar.   Respuesta del sistema   * Verifica si la ruta está registrada * Si esta registrada permite eliminar permanentemente a la ruta con toda su información existente.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina ruta y actualiza la base de datos. |

CU4.Gestionar personal.



|  |  |
| --- | --- |
| **Gestionar Personal** | |
| **ID Caso de Uso** | CU-4 |
| **Actor** | Administrador. |
| **Descripción** | Gestiona al personal que trabaje para el Sindicato de transporte 27 de diciembre, ya sean antiguo o un nuevo personal. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **postcondiciones** | Todos los registros y cambios hecho por el usuario han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El actor pulsa sobre el botón de registrar personal.   Respuesta del sistema   * El sistema muestra campos de texto para introducir los datos del nuevo personal para el registro correcto.   Acción del actor.   * Introduce toda la información necesaria para el registro. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Buscar al personal a modificar por su nombre o CI.   Respuesta del sistema   * Verifica si el personal está registrado * Si esta registrado permite modificar los datos del personal buscado.   Acción del actor   * Modifica los datos del personal. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Buscar al personal a eliminar.   Respuesta del sistema   * Verifica si el personal está registrado * Si esta registrado permite eliminar permanentemente al personal buscado con toda su información existente.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina personal y actualiza la base de datos. |

CU5.Gestionar Tipo Personal



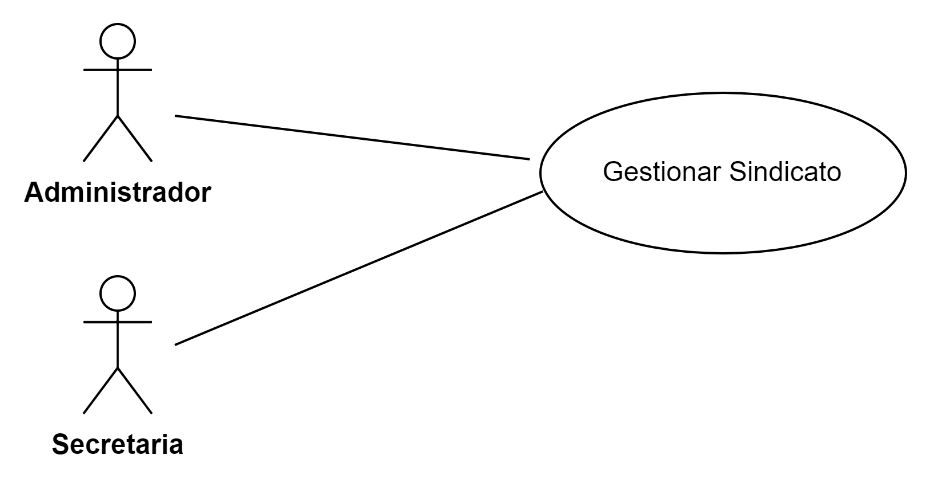
|  |  |
| --- | --- |
| **Gestionar tipo Personal** | |
| **ID Caso de Uso** | CU-5 |
| **Actor** | Administrador. |
| **Descripción** | Gestiona al tipo de personal que trabaje para el Sindicato de transporte 27 de diciembre, ya sean antiguo o un nuevo personal. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Postcondiciones** | Todos los registros y cambios hecho por el usuario han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El actor pulsa sobre el botón de registrar.   Respuesta del sistema   * El sistema muestra campos de texto para introducir los datos para el registro correcto del tipo\_personal.   Acción del actor.   * Introduce toda la información necesaria para el registro. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Buscar al TipoPersonal a modificar por su cargo.   Respuesta del sistema   * Verifica si existe. * Si existe permite modificar.   Acción del actor   * Modifica los datos de TipoPersonal. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Buscar el TipoPersonal a eliminar.   Respuesta del sistema   * Verifica si existe. * Si existe permite eliminar.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina y actualiza la base de datos. |

CU6.Gestionar Mantenimiento.



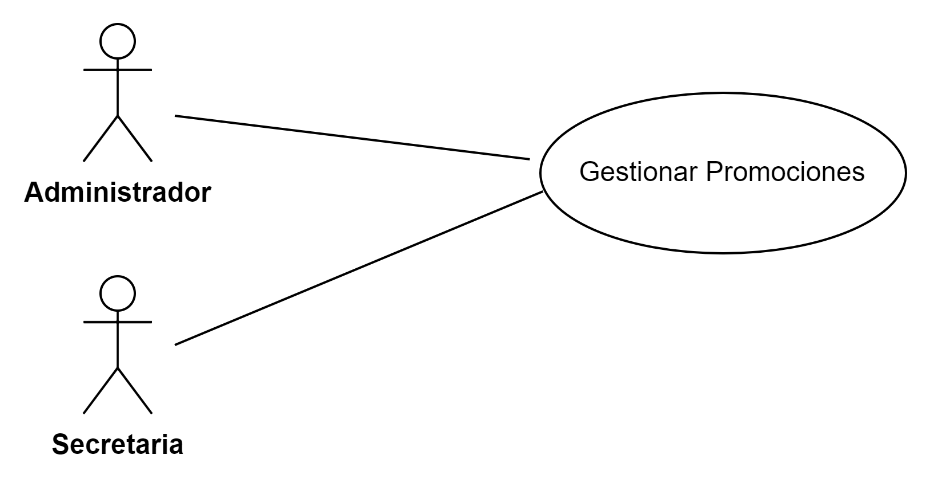
|  |  |
| --- | --- |
| **Gestionar Mantenimiento** | |
| **ID Caso de Uso** | CU-6 |
| **Actor** | Administrador.  Secretaria. |
| **Descripción** | Gestiona información acerca de los mantenimientos que se realizan de los vehículos del Sindicato de transporte 27 de diciembre. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador.  Insertar/Actualizar/Listar datos, para actor: Secretaria. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Postcondiciones** | Todos los registros y cambios hecho por el usuario han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El actor llama al vehículo donde se realizó el mantenimiento.   Respuesta del sistema   * El sistema muestra campos de texto para introducir los datos necesarios para el registro de la información del mantenimiento.   Acción del actor.   * Introduce toda la información necesaria para el registro. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Busca el vehículo donde se realizó y registró el mantenimiento. * Buscar el mantenimiento a modificar.   Respuesta del sistema   * Verifica si está registrado * Si esta registrado permite modificar la información del mantenimiento buscado.   Acción del actor   * Modifica los datos del mantenimiento registrado. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Buscar información de mantenimiento a eliminar.   Respuesta del sistema   * Verifica si el mantenimiento está registrado * Si esta registrado permite eliminar permanentemente al mantenimiento buscado con toda su información existente.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina y actualiza la base de datos. |

CU7. Gestionar sindicato.



|  |  |
| --- | --- |
| **Gestionar sindicato** | |
| **ID Caso de Uso** | CU-7 |
| **Actor** | Administrador.  Secretaria. |
| **Descripción** | Gestiona información del sindicato, como las líneas aliadas que trabajan en él. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador.  Insertar/Actualizar/Listar datos, para actor: Secretaria. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Postcondiciones** | Todos los registros y cambios hecho por el usuario han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El actor pulsa sobre el botón de registrar.   Respuesta del sistema   * El sistema muestra campos de texto a rellenar para el registro correcto de la nueva línea aliada.   Acción del actor.   * Introduce toda la información necesaria para el registro. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Buscar línea a modificar por su nombre.   Respuesta del sistema   * Verifica si existe. * Si existe permite modificar la información.   Acción del actor   * Modifica los datos de la línea buscada. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Busca la línea a eliminar.   Respuesta del sistema   * Verifica si existe. * Si existe permite eliminar.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina y actualiza la base de datos. |

CU8. Gestionar promociones.



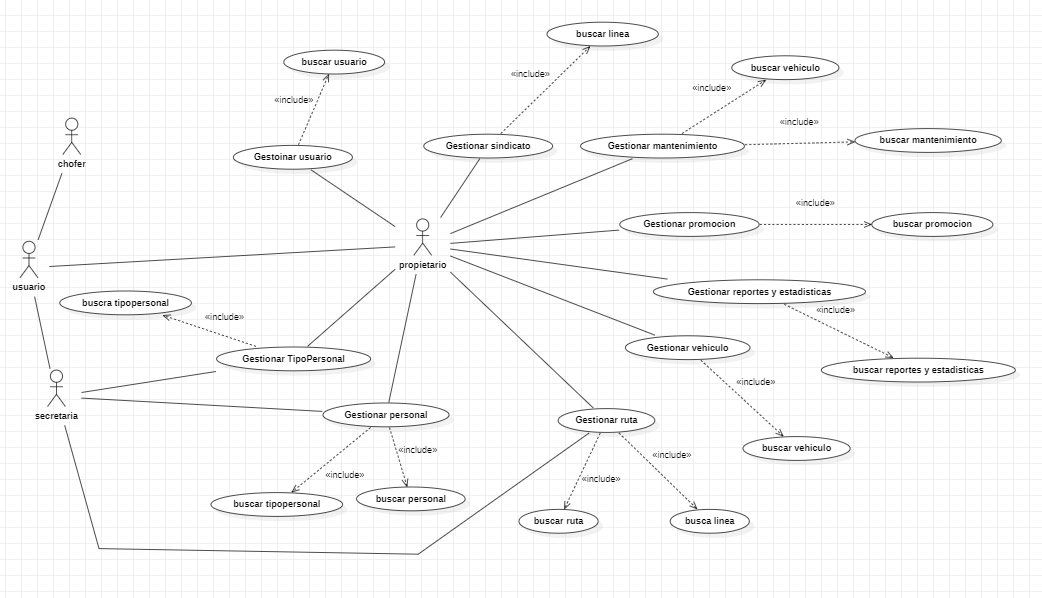
|  |  |
| --- | --- |
| **Gestionar promociones** | |
| **ID Caso de Uso** | CU-8 |
| **Actor** | Administrador.  Secretaria. |
| **Descripción** | Gestiona información de las promociones que ofrece el sindicato de transporte 27 de diciembre. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos, para actor: Administrador.  Insertar/Actualizar/Listar datos, para actor: Secretaria. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Postcondiciones** | Todos los registros y cambios hecho por el usuario han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Registrar**   Acción del actor   * El actor pulsa sobre el botón de registrar.   Respuesta del sistema   * El sistema muestra campos de texto a rellenar para el registro correcto de la nueva promoción.   Acción del actor.   * Introduce toda la información necesaria para el registro. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Buscar promoción a modificar.   Respuesta del sistema   * Verifica si existe. * Si existe la promoción permite modificar la información.   Acción del actor   * Modifica los datos de la promoción buscada. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Busca la promoción a eliminar.   Respuesta del sistema   * Verifica si existe. * Si existe permite eliminar la promoción.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina y actualiza la base de datos. |

CU9. Gestionar Reportes y Estadísticas.

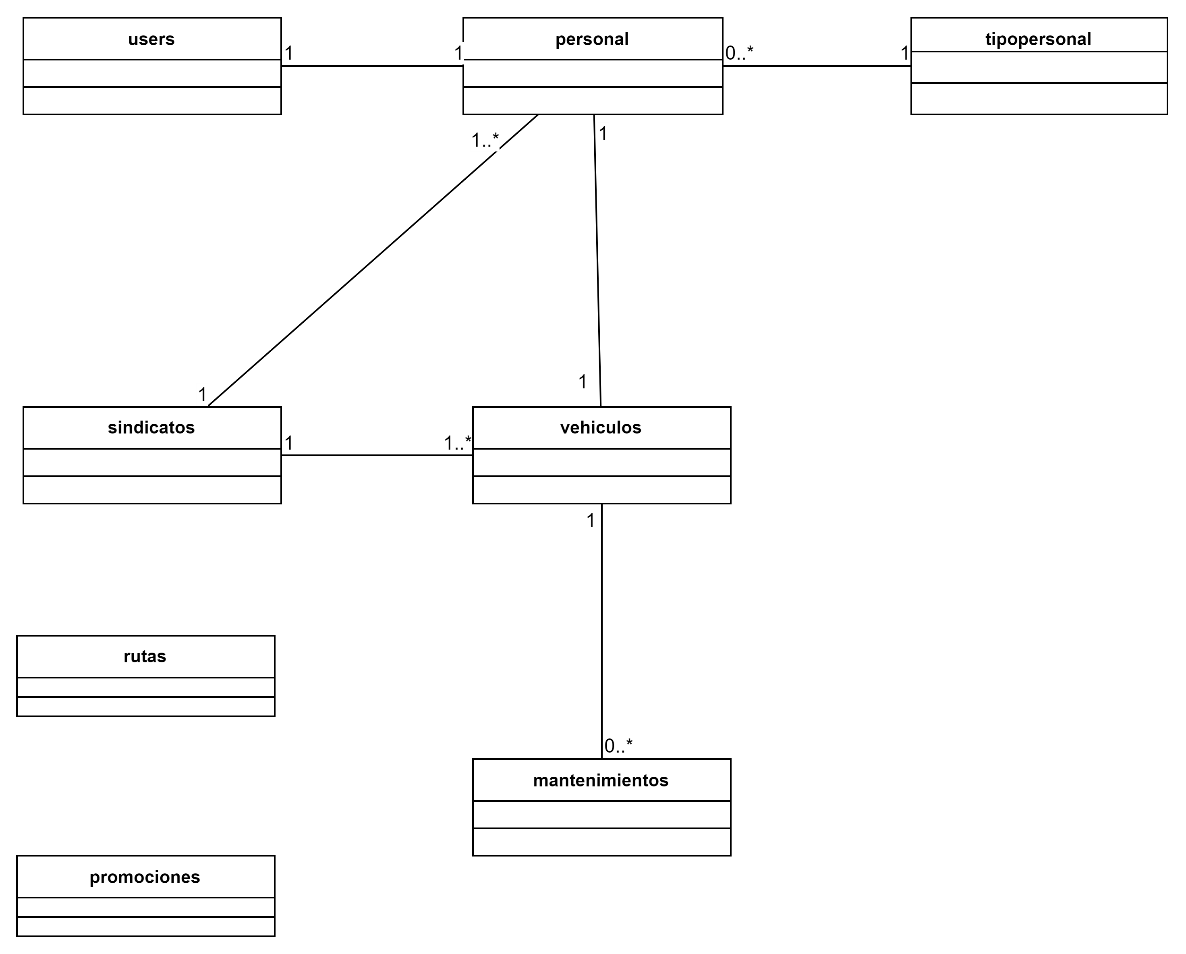


|  |  |
| --- | --- |
| **Gestionar reportes y estadísticas.** | |
| **ID Caso de Uso** | CU-9 |
| **Actor** | Administrador. |
| **Descripción** | Gestiona información relevante de las actividades y funciones que realiza el sindicato de transporte 27 de diciembre. |
| **Propósito** | Insertar/Actualizar/Eliminar/Listar datos del reporte.  Permite visualizar estadísticas con información relevante y útil. |
| **Precondiciones** | Haberse logueado con su cuenta de usuario para obtener los privilegios correspondientes a esa cuenta. |
| **Postcondiciones** | Todos los registros y cambios hecho por el usuario han sido almacenados y actualizados en la base de datos del sistema web. |
| **Flujo normal** | * **Reporte**   Acción del actor   * El actor pulsa sobre el botón de registrar.   Respuesta del sistema   * El sistema muestra campos de texto a rellenar para el registro correcto del nuevo reporte.   Acción del actor.   * Introduce toda la información necesaria para el registro. * Pulsa el botón guardar.   Respuesta del sistema   * Validar los datos introducidos. * Guarda en la base de datos. * **Actualizar**   Acción del actor   * Buscar reporte a modificar.   Respuesta del sistema   * Verifica si existe. * Si existe el reporte permite modificar la información.   Acción del actor   * Modifica los datos del reporte buscado. * Apretar el botón de Guardar.   Respuesta del sistema   * Valida los datos introducidos * Almacena los cambios y los guarda. * **Eliminar**   Acción del actor   * Busca el reporte a eliminar.   Respuesta del sistema   * Verifica si existe. * Si existe permite eliminar el reporte.   Acción del actor   * Apreta el botón Eliminar.   Respuesta del sistema   * Elimina y actualiza la base de datos. |

## DIAGRAMA GENERAL DE CASOS DE USO



## MODELO DE DOMINIO INICIAL- DIAGRAMA DE CLASES

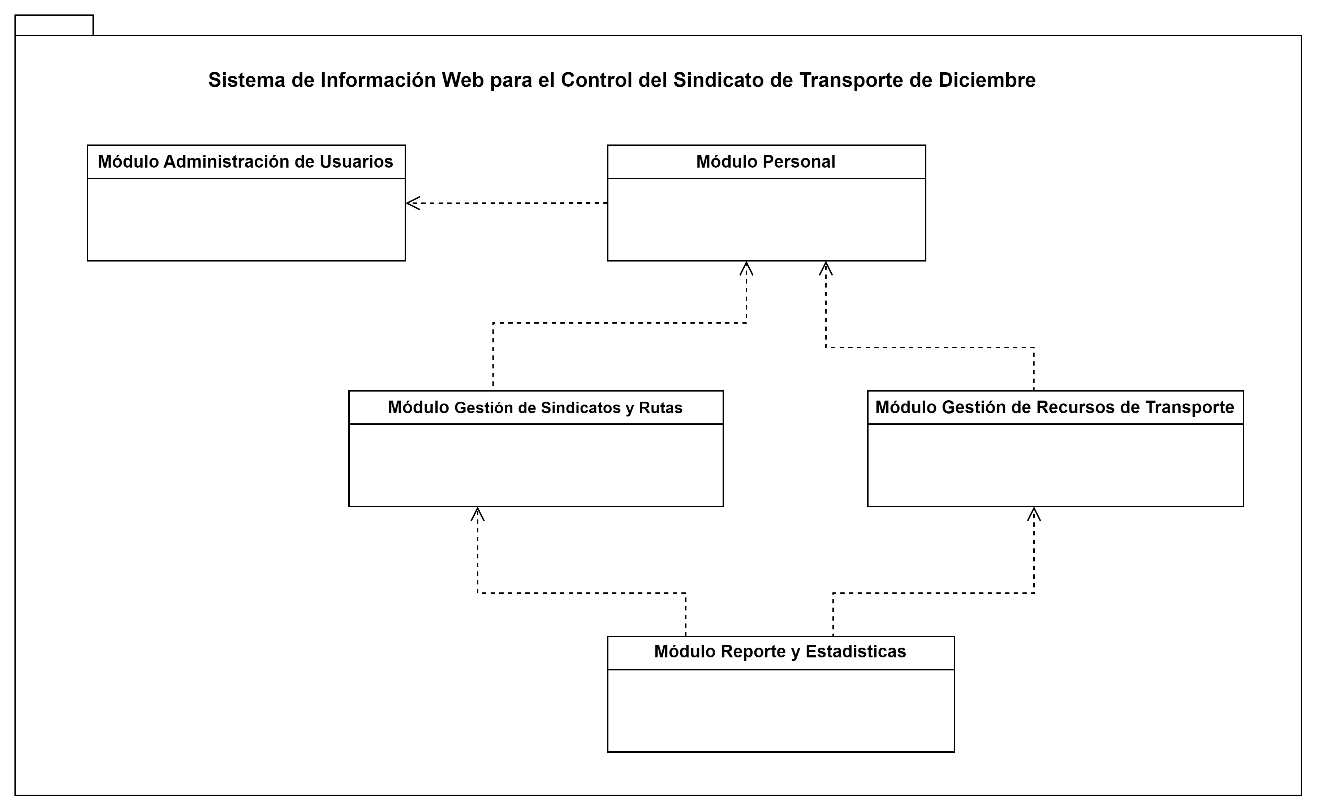


# CAPITULO V ANÁLISIS

## IDENTIFICACIÓN DE PAQUETES O MÓDULOS DEL SISTEMA

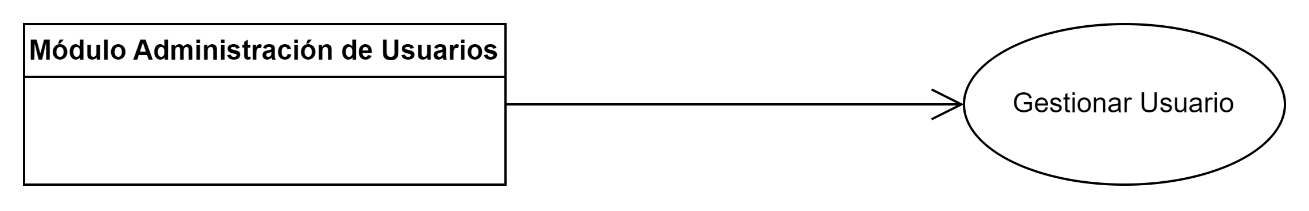
Durante el análisis del sistema se identificaron 5 paquetes, que serán tomados como subsistemas para su mejor comprensión y presentación.

Paquetes representados como subsistemas que forman parte del sistema web

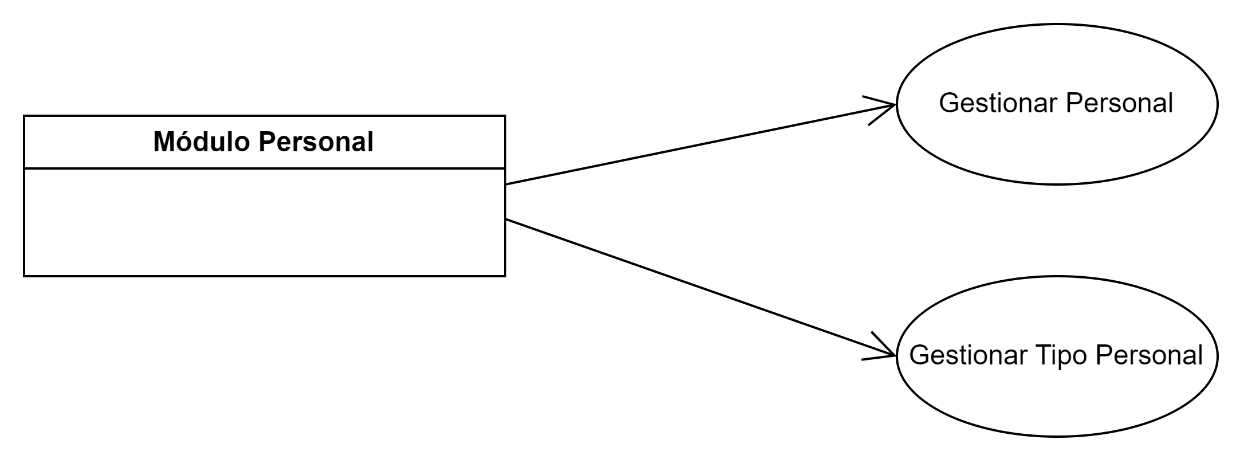


## Relación de módulos y casos de uso

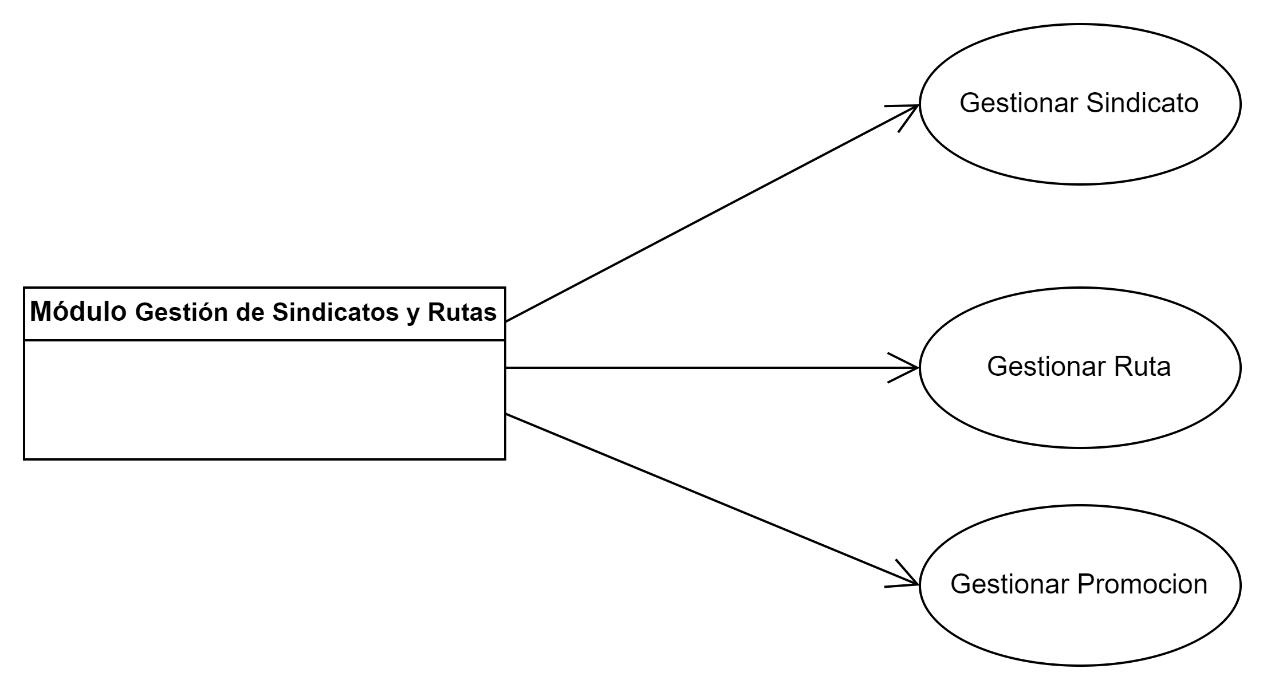
### Relación modulo Administración de usuario.



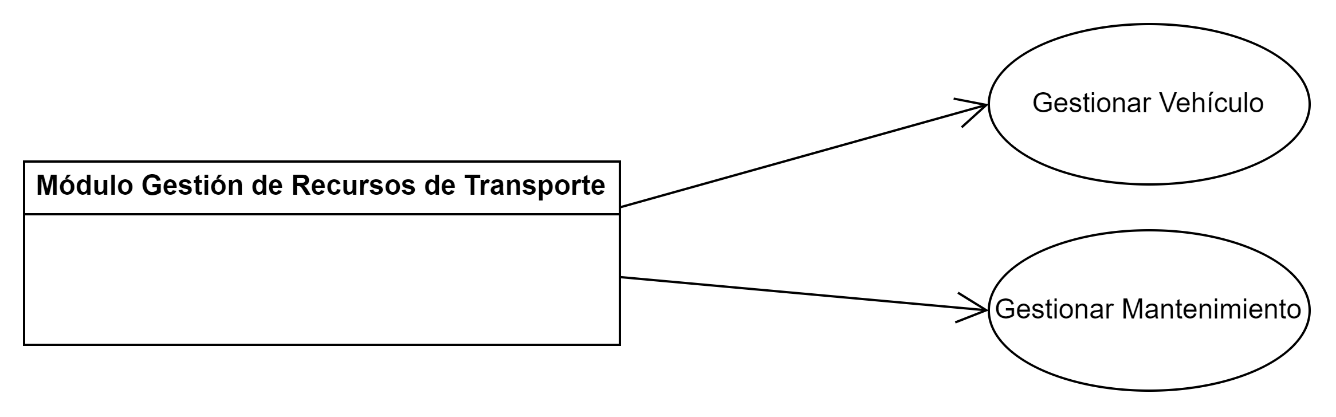
### Relación modulo Personal.



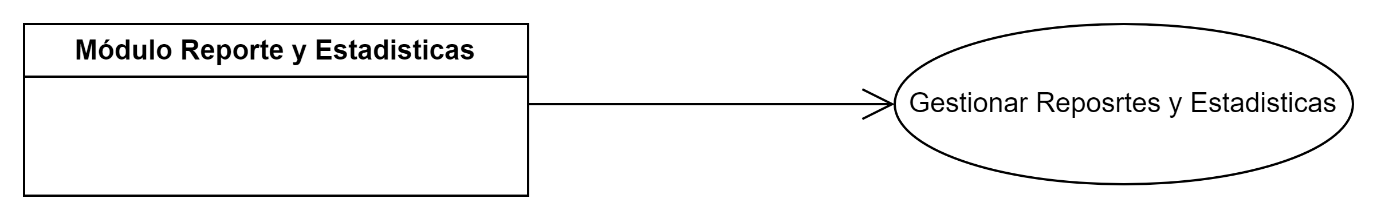
### Relación modulo Gestión de sindicato y rutas.



### Relación modulo Gestión de recurso de transporte.



### Relación modulo reportes y estadísticas.



## REALIZACIÓN DE LOS CASOS DE USO - ANÁLISIS.

La realización de casos de uso describe cómo se lleva a cabo cada caso de uso en función a sus clases. Para esto se utilizan los siguientes artefactos:

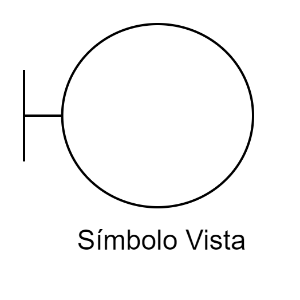
Diagrama de colaboración. Los diagramas de colaboración muestran la forma en que los objetos colaboran entre sí. Muestra los objetos junto con los mensajes que se envían entre ellos, estos diagramas destacan el contexto y organización general de los objetos que interactúan.

Flujo de Sucesos. Se realizará una explicación textual de cómo se ejecuta un Caso de Uso.

A continuación, se definen las Clases de Análisis para modelar estos diagramas, cada uno de ellos va a representar una parte del MVC (Modelo, Vista, Controlador):

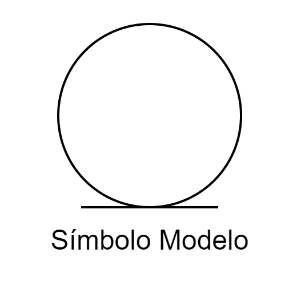
### Clases de Interfaz:

Se utilizará para modelar la interacción entre el sistema y el actor. (Vista)



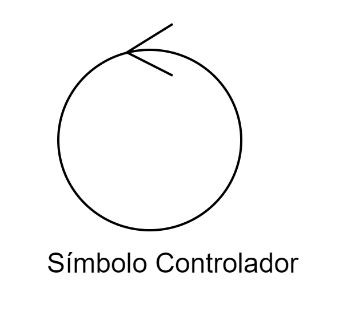
### Clases de Entidad:

Se utilizará para modelar información persistente. (Modelo)



### Clases de Control:

Representa coordinación, secuencia y transacción. (Controlador)



## ANÁLISIS DE CASOS DE USO

Se analizan los casos de para determinar las iteraciones entre objetos y los mensajes de acciones que existen entre ellos. Los beneficios de este análisis son:

Identificar las clases del diseño y/o los subsistemas cuyas instancias son necesarias para llevar a cabo el flujo de sucesos del caso de uso.

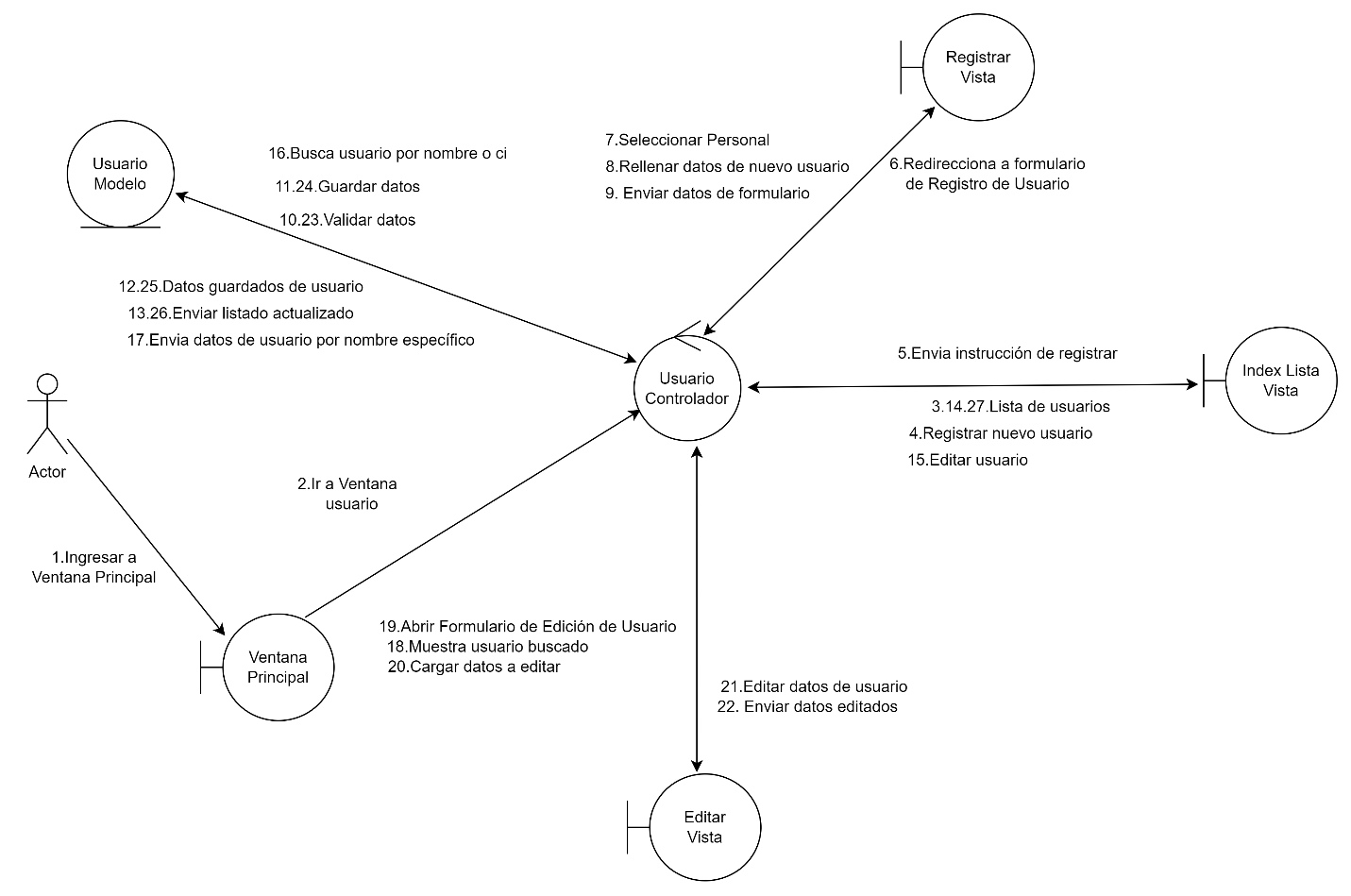
Distribuir el comportamiento del caso de uso entre los objetivos del diseño que interactúan y/o entre los subsistemas participantes

Definir los requisitos sobre las operaciones de las clases del diseño y sobre los subsistemas y sus interfaces

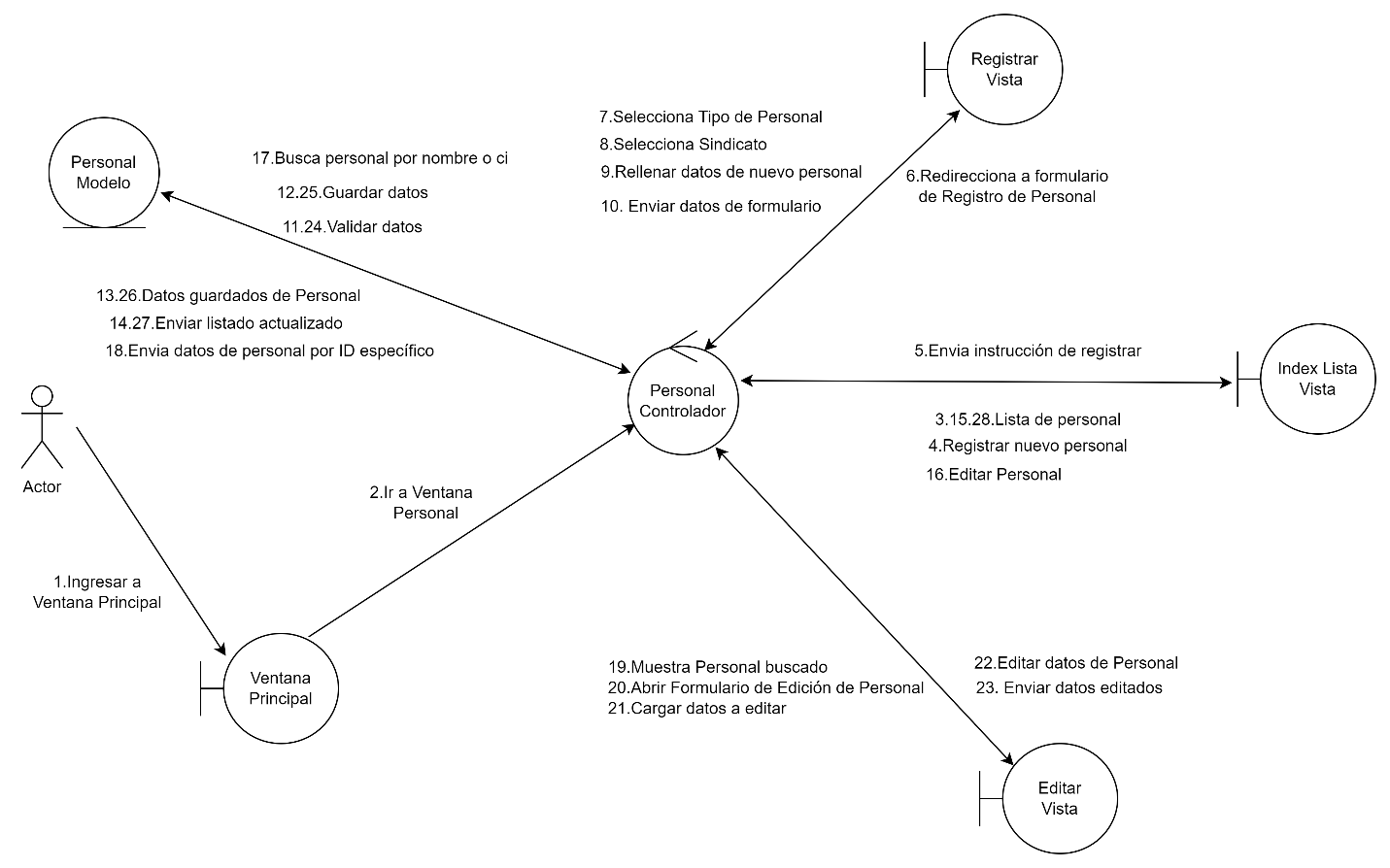
Capturar los requisitos de implementación de los casos de uso

A continuación, especificamos cada caso de uso a través de diagramas de colaboración.

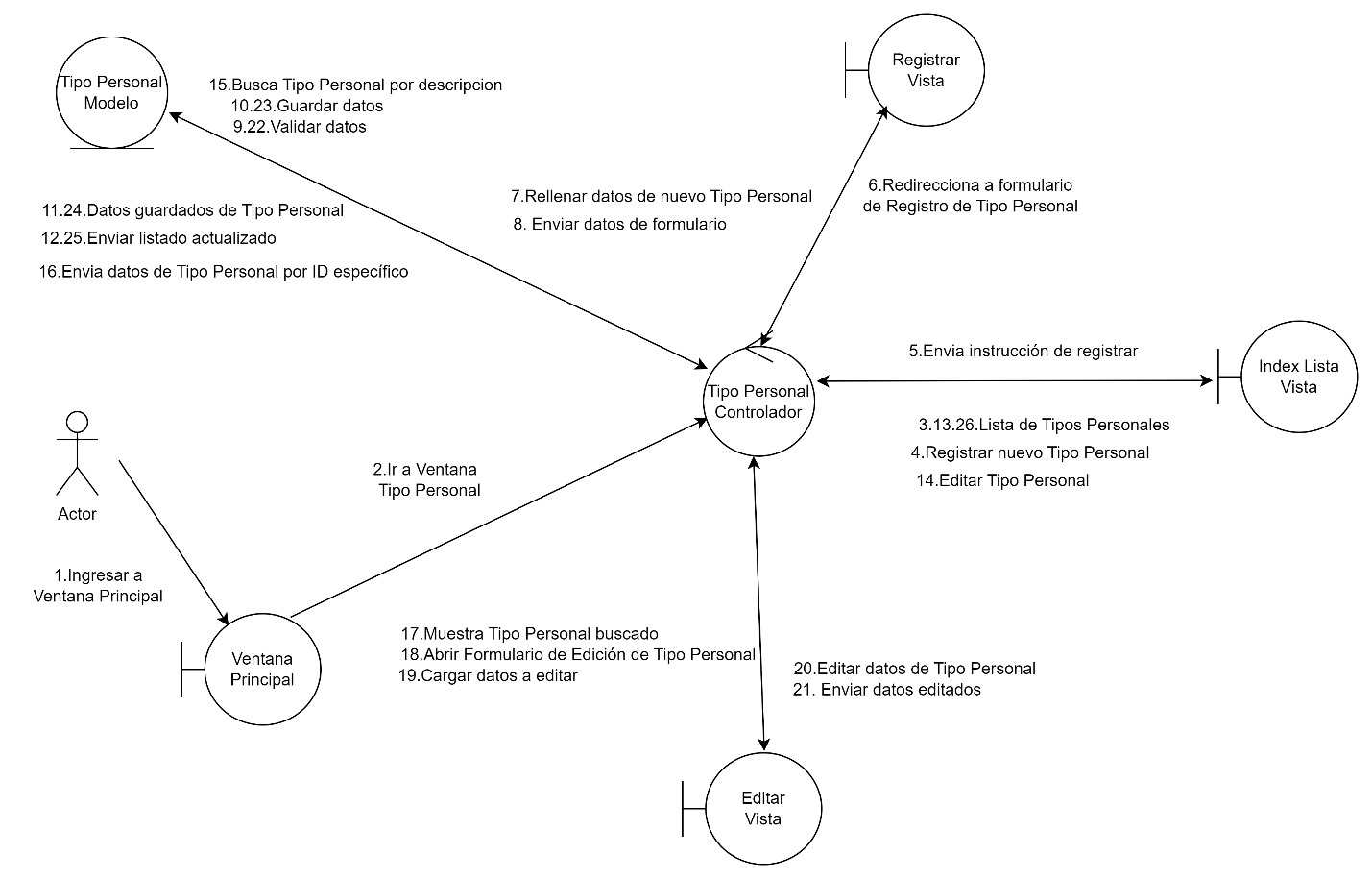
### Caso de uso Gestionar Usuario

****

### Caso de uso Gestionar Personal

****

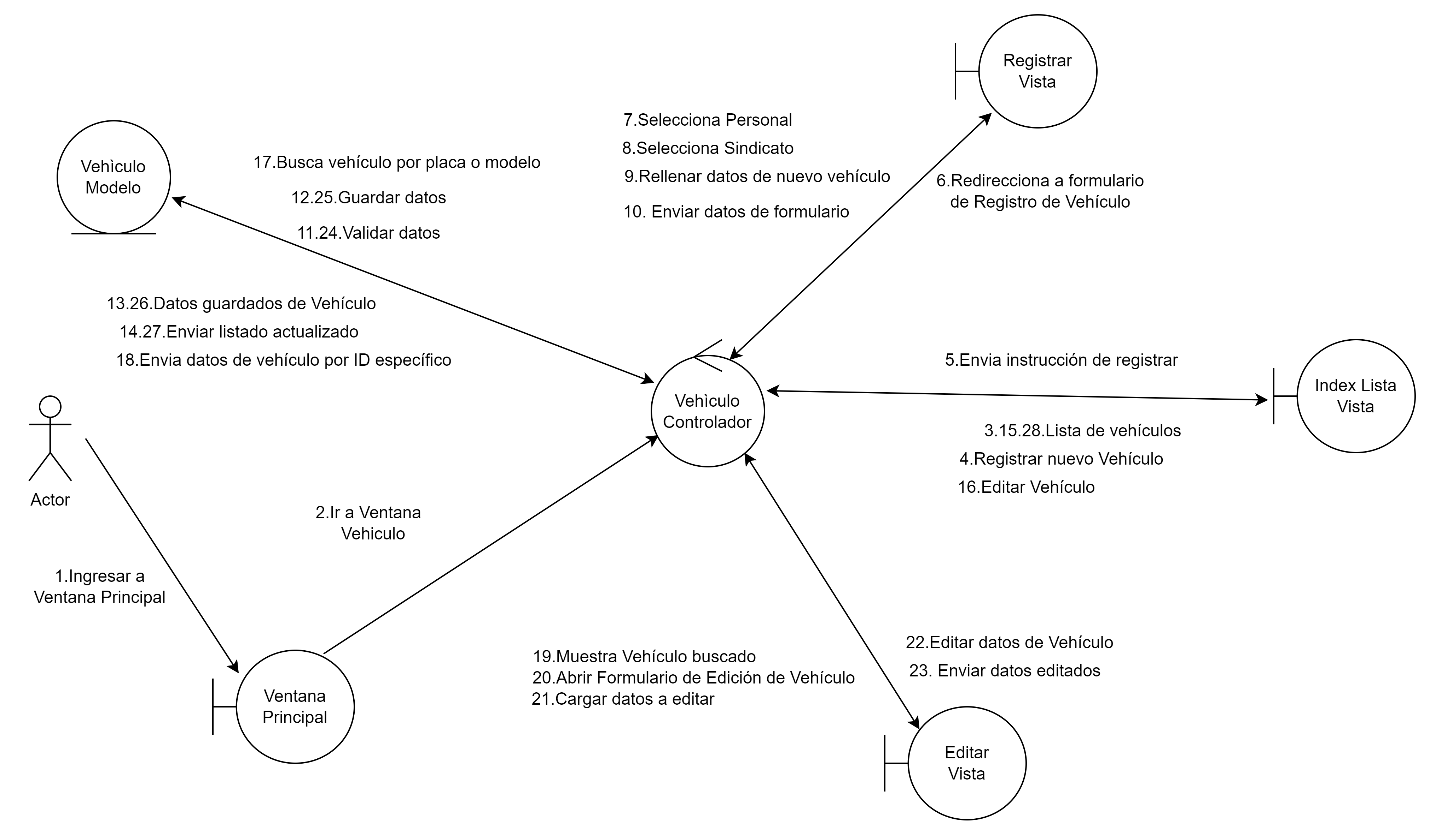
### Caso de uso Gestionar Tipo Personal

****

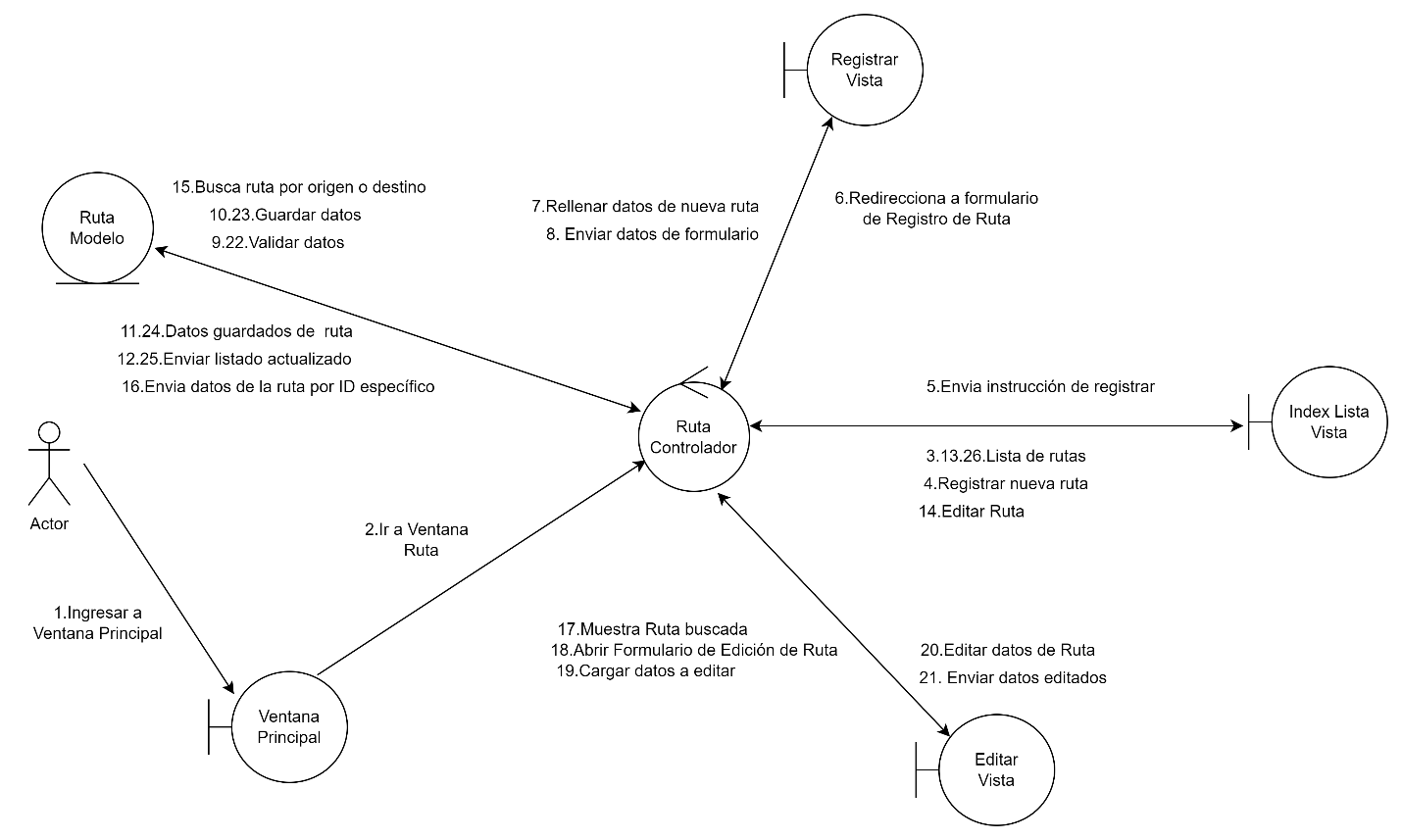
### Caso de uso Gestionar Sindicato

****

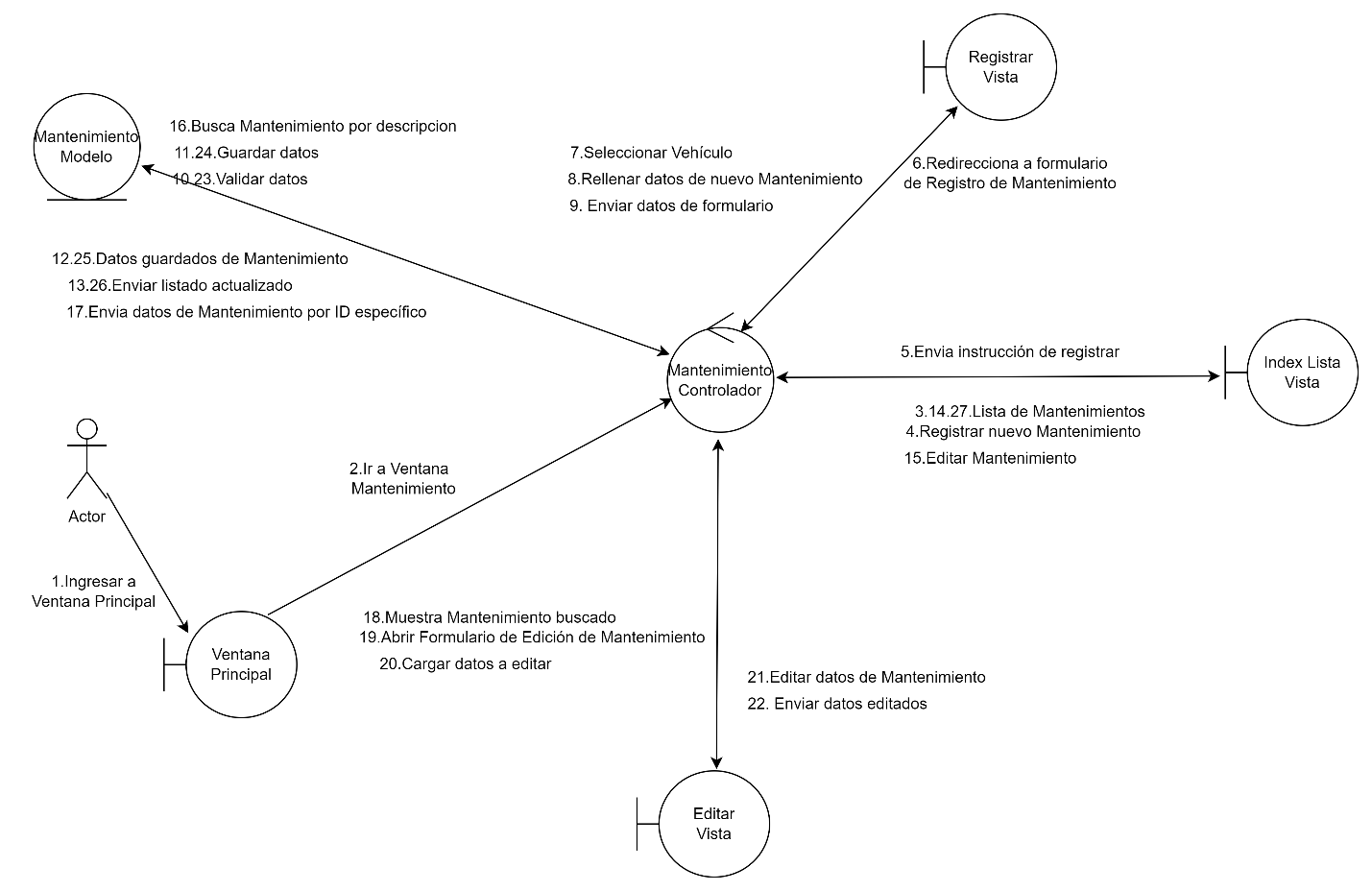
### Caso de uso Gestionar Vehículo

****

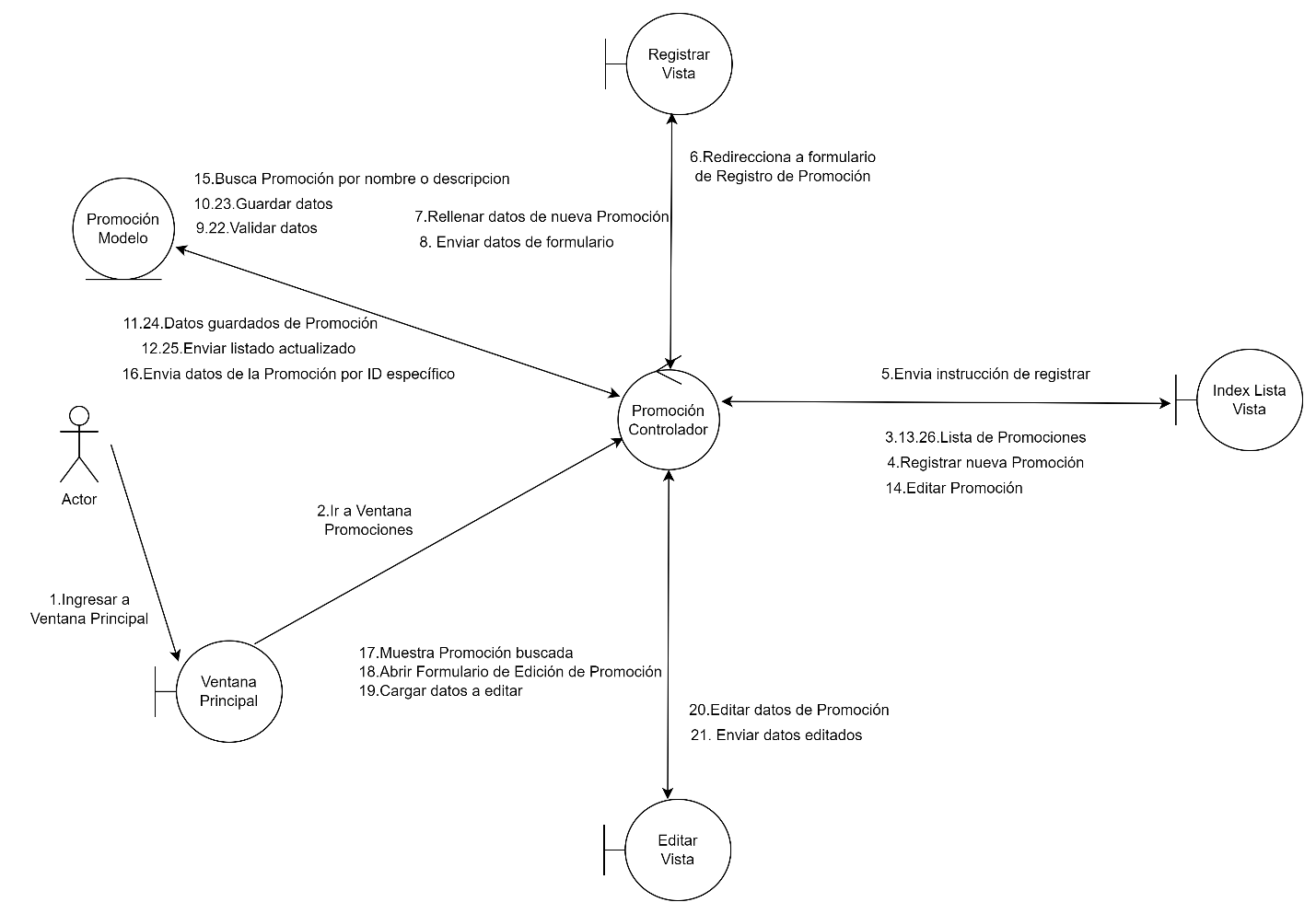
### Caso de uso Gestionar Ruta

****

### Caso de uso Gestionar Mantenimiento

****

### Caso de uso Gestionar Promoción

****

# CAPITULO VI DISEÑO

## Introducción

En el capítulo se presenta artefactos obtenidos durante el diseño del sistema web, los cuales son una entrada fundamental y un punto de partida para las actividades del análisis del sistema.

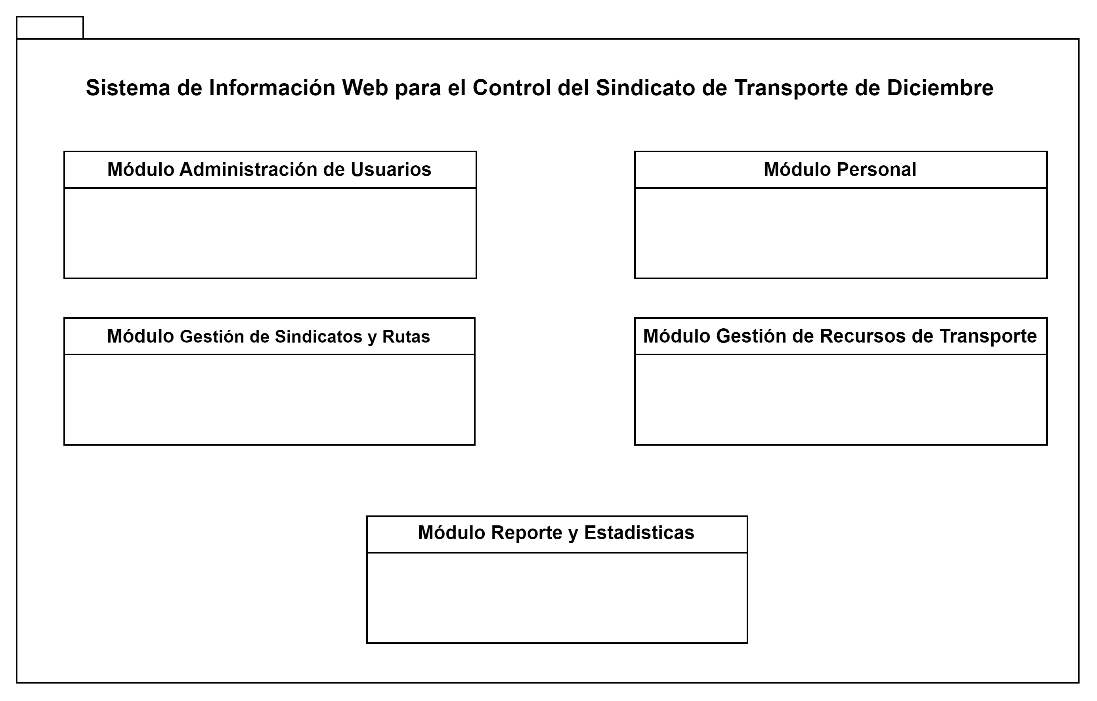
En el diseño se modela la forma del sistema que soportará todos los requisitos obtenidos, se basa fundamentalmente en los resultados proporcionados durante la etapa del análisis, observaciones del cliente y los usuarios.

Como resultado primordial del diseño, debemos obtener la estructura del sistema impuesta por el modelo del análisis, y que es la base para el esquema de implementación.

## Diseño de la arquitectura del diagrama de paquetes

### Diseño de la arquitectura general del sistema

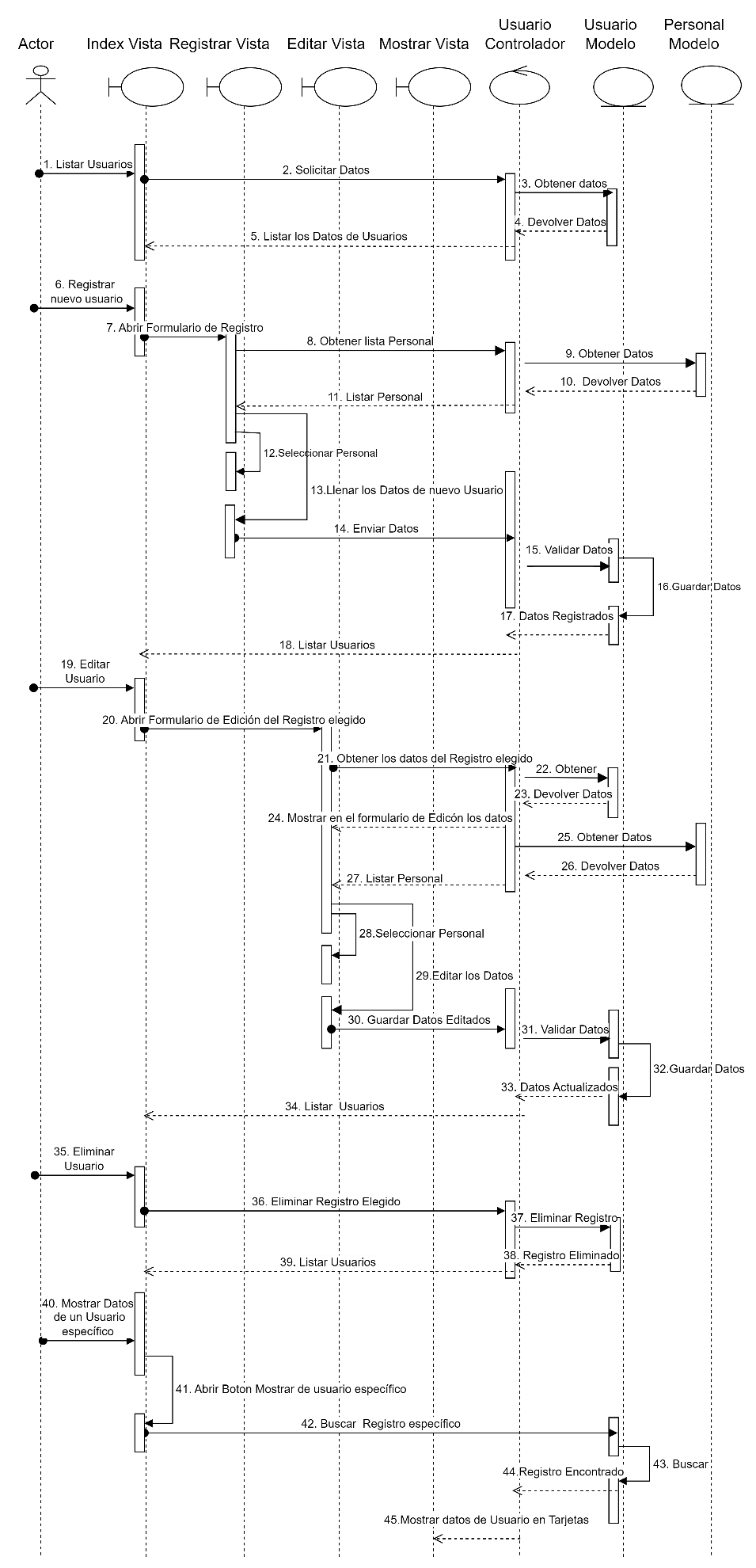
La arquitectura del sistema muestra el grado de cohesión y acoplamiento que tiene el sistema.



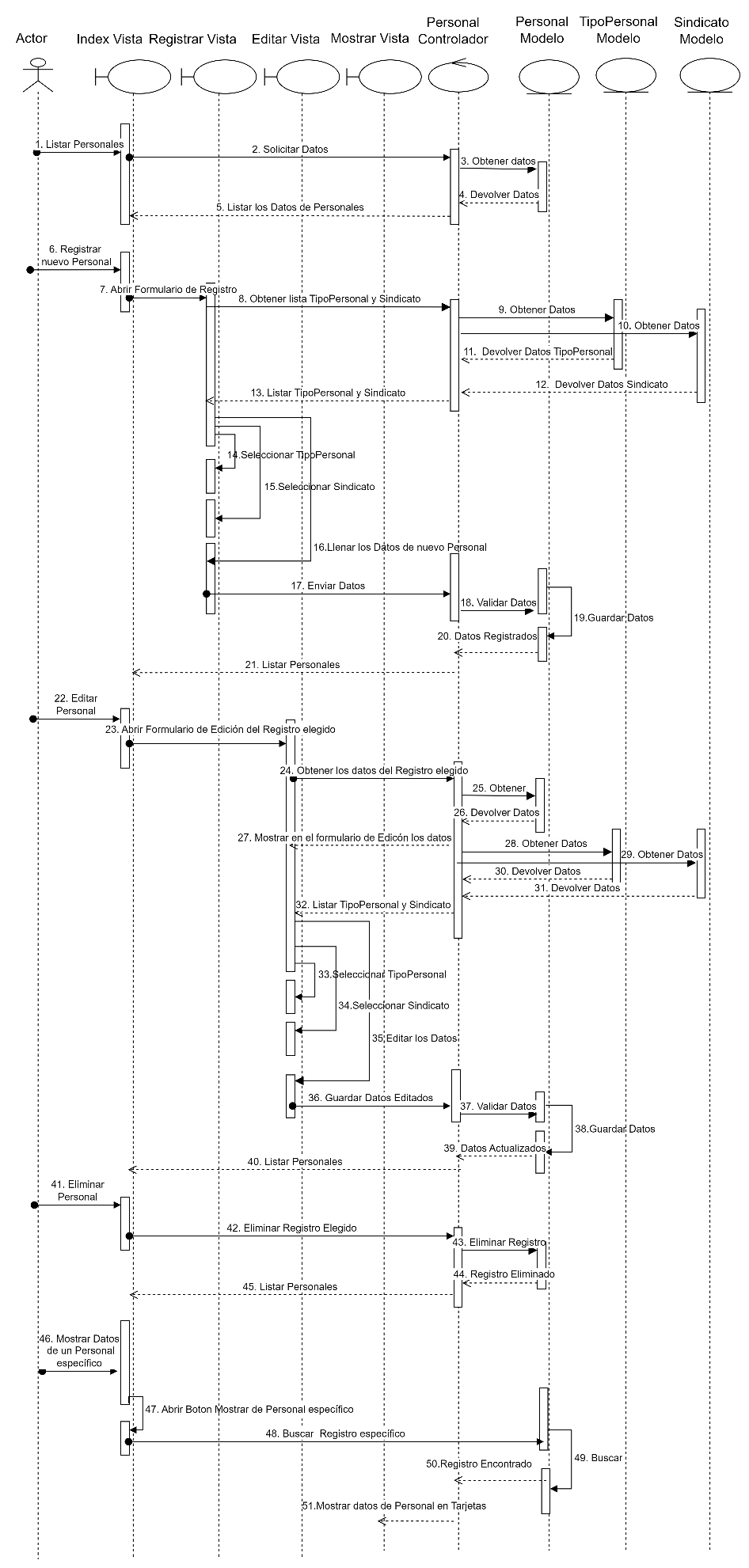
## Diseño procedimental- Diagrama de secuencias

Aquí se muestra el diseño de la realización de los casos de uso más relevantes del sistema. La finalidad de los siguientes es mostrar la interacción entre los objetos.

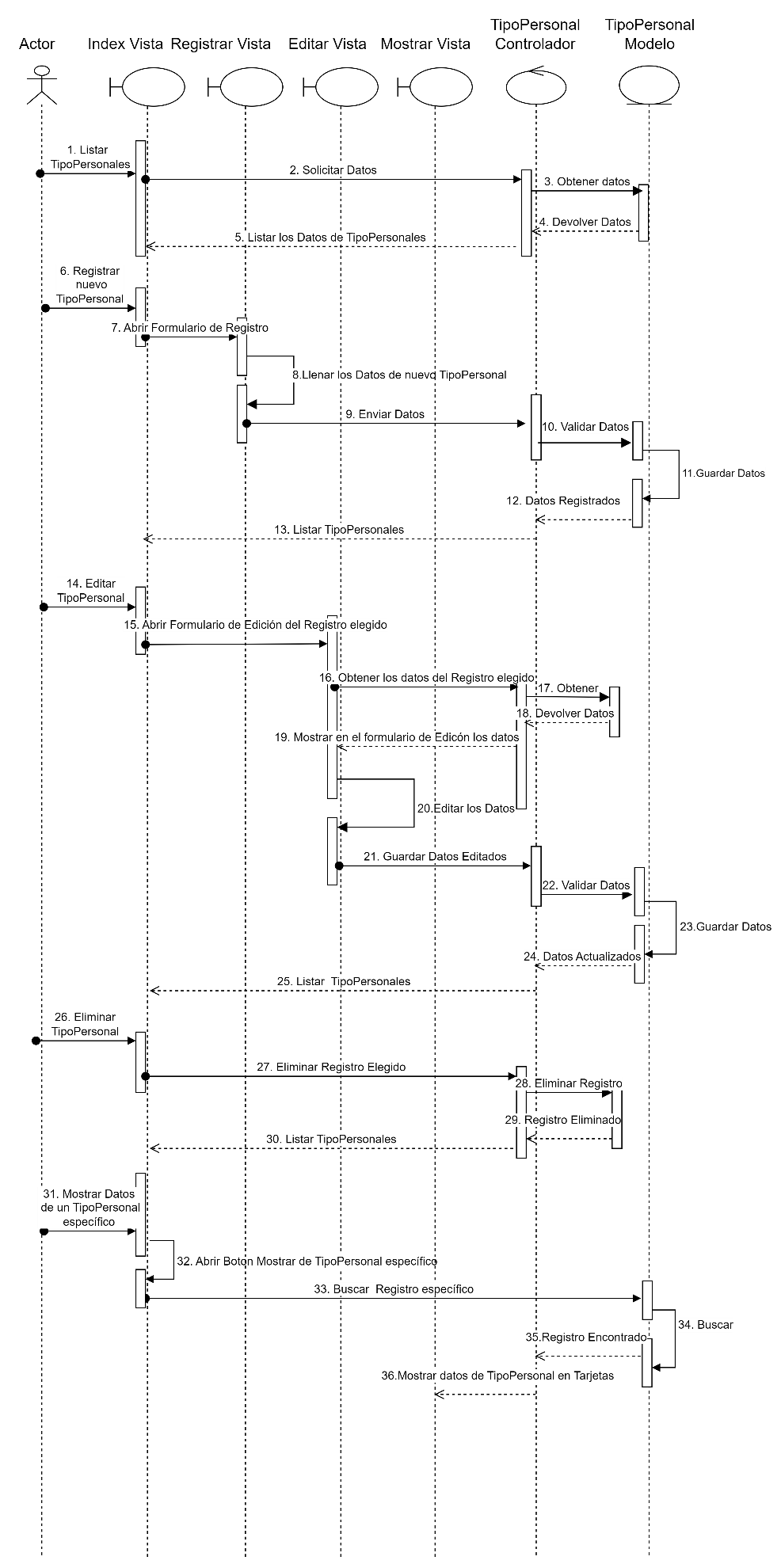
### Diagrama de Secuencia: Gestionar Usuario



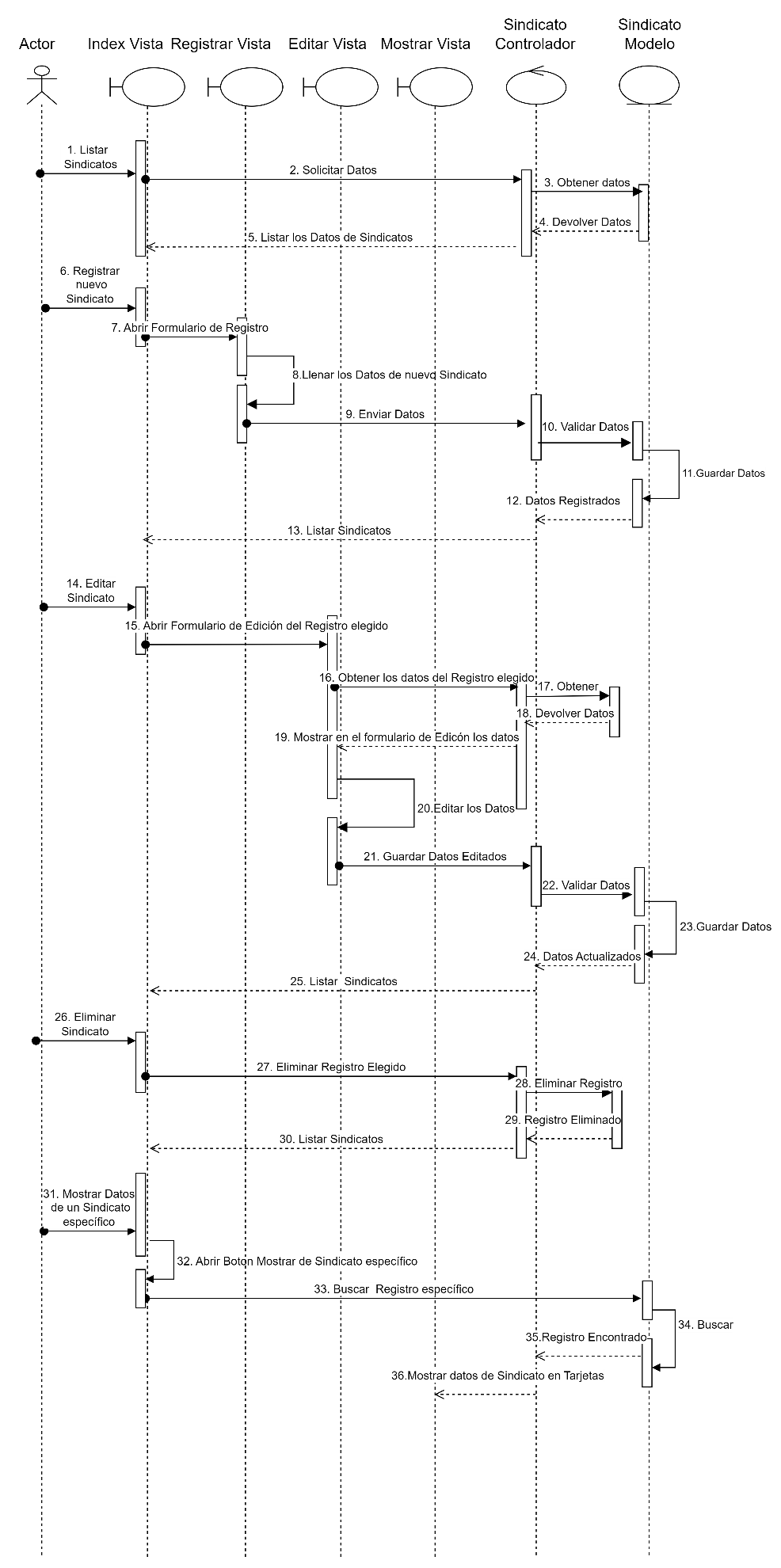
### Diagrama de Secuencia: Gestionar Personal



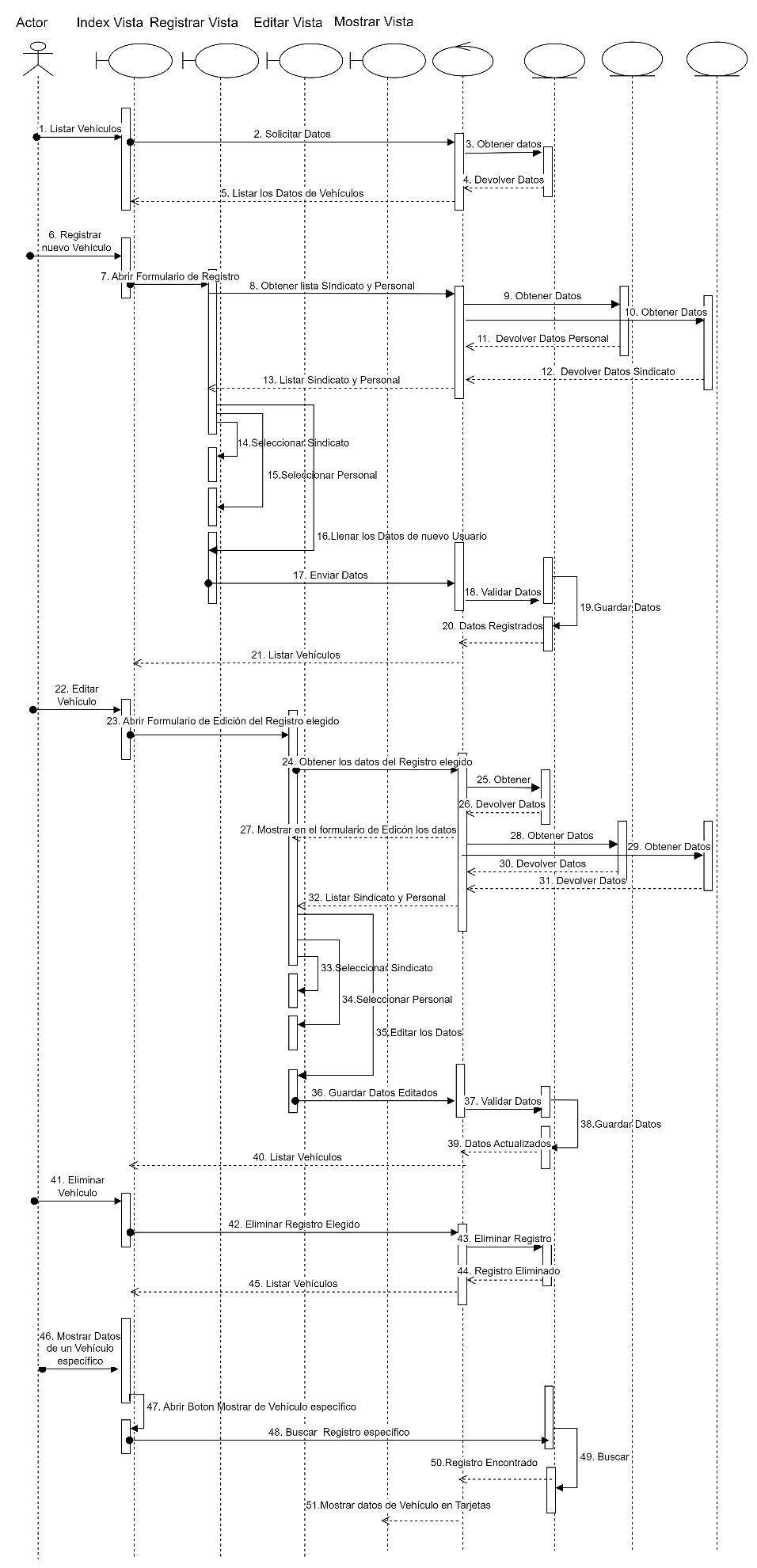
### Diagrama de Secuencia: Gestionar TipoPersonal



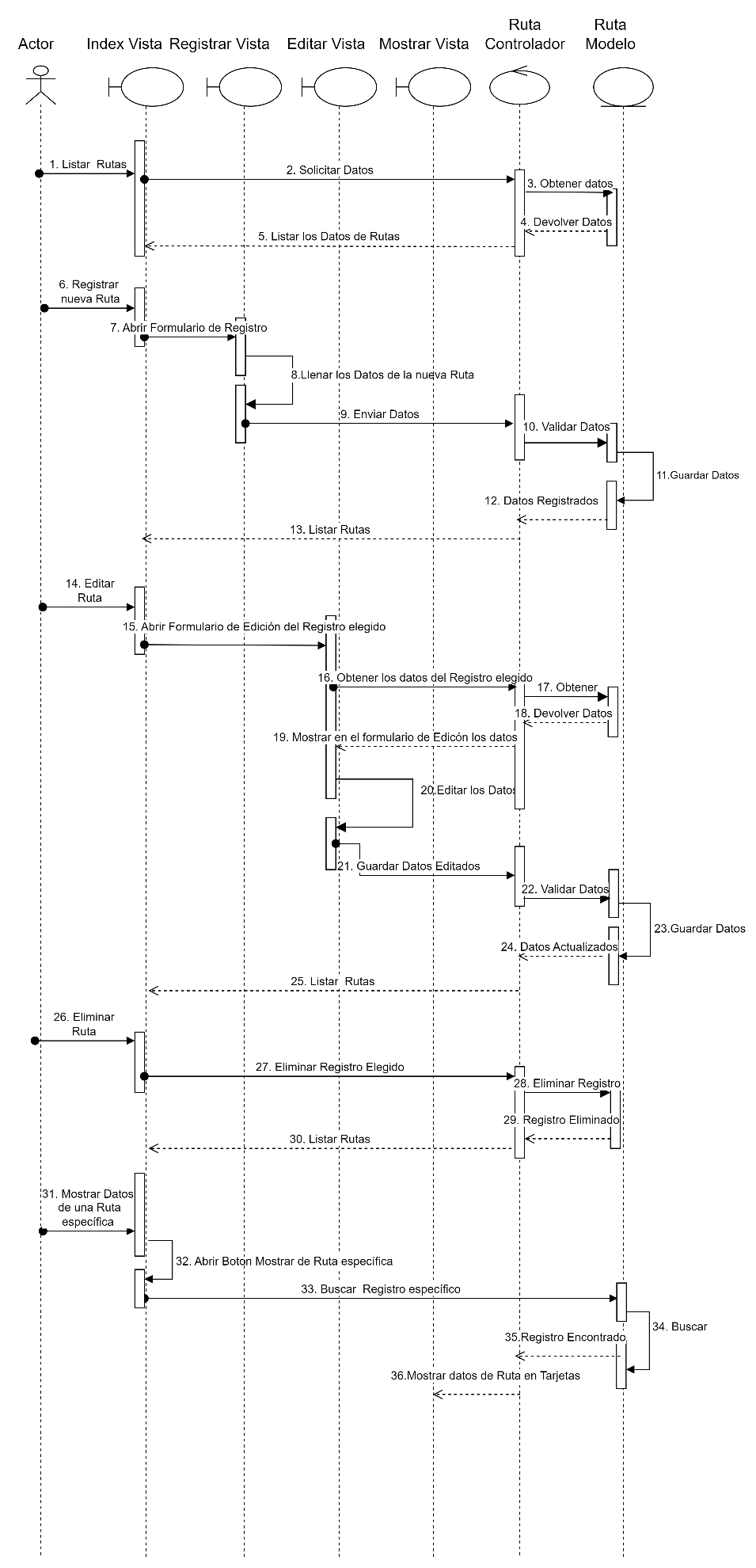
### Diagrama de Secuencia: Gestionar Sindicato



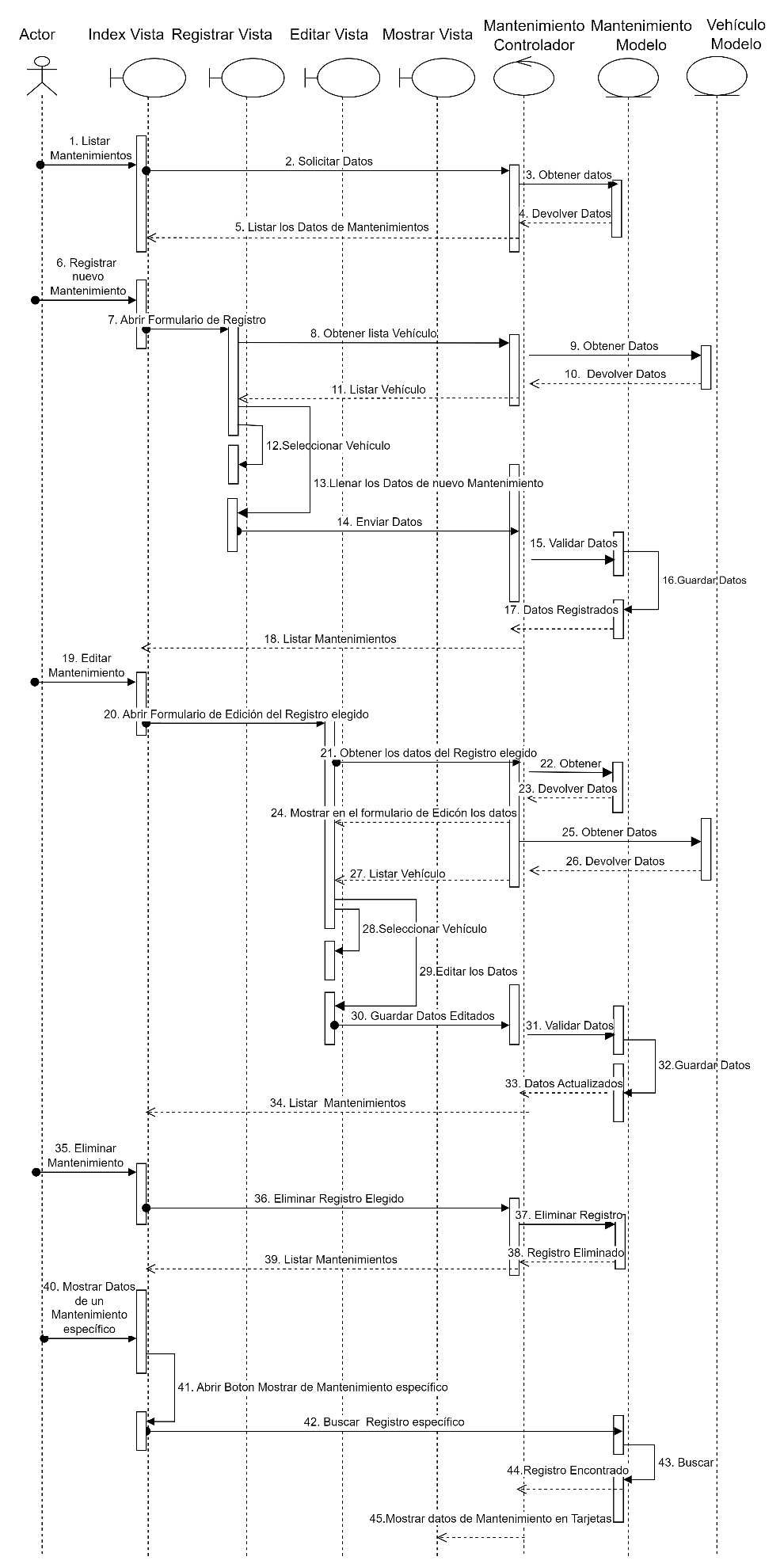
### Diagrama de Secuencia: Gestionar Vehículo



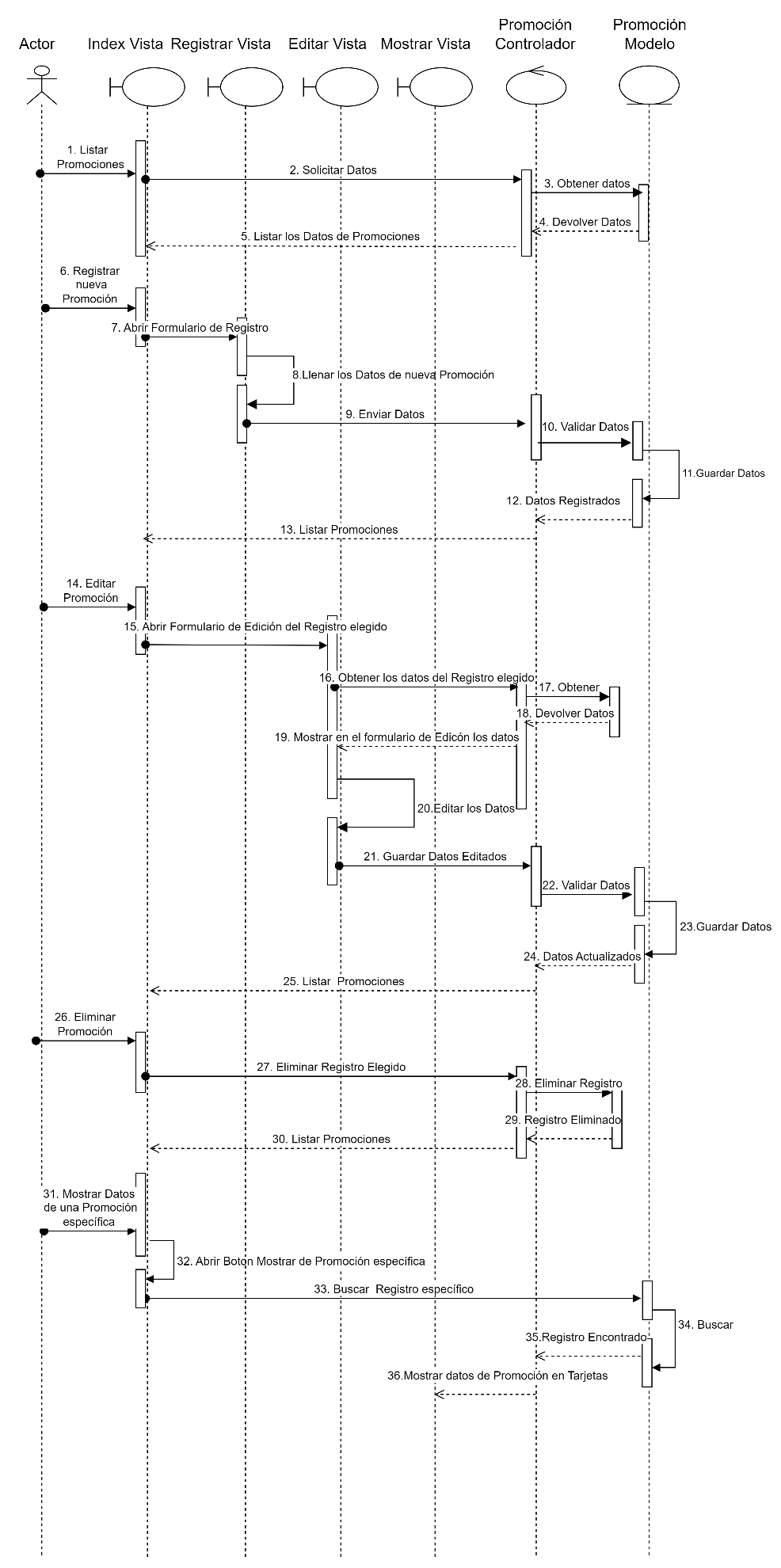
### Diagrama de Secuencia: Gestionar Ruta



### Diagrama de Secuencia: Gestionar Mantenimiento



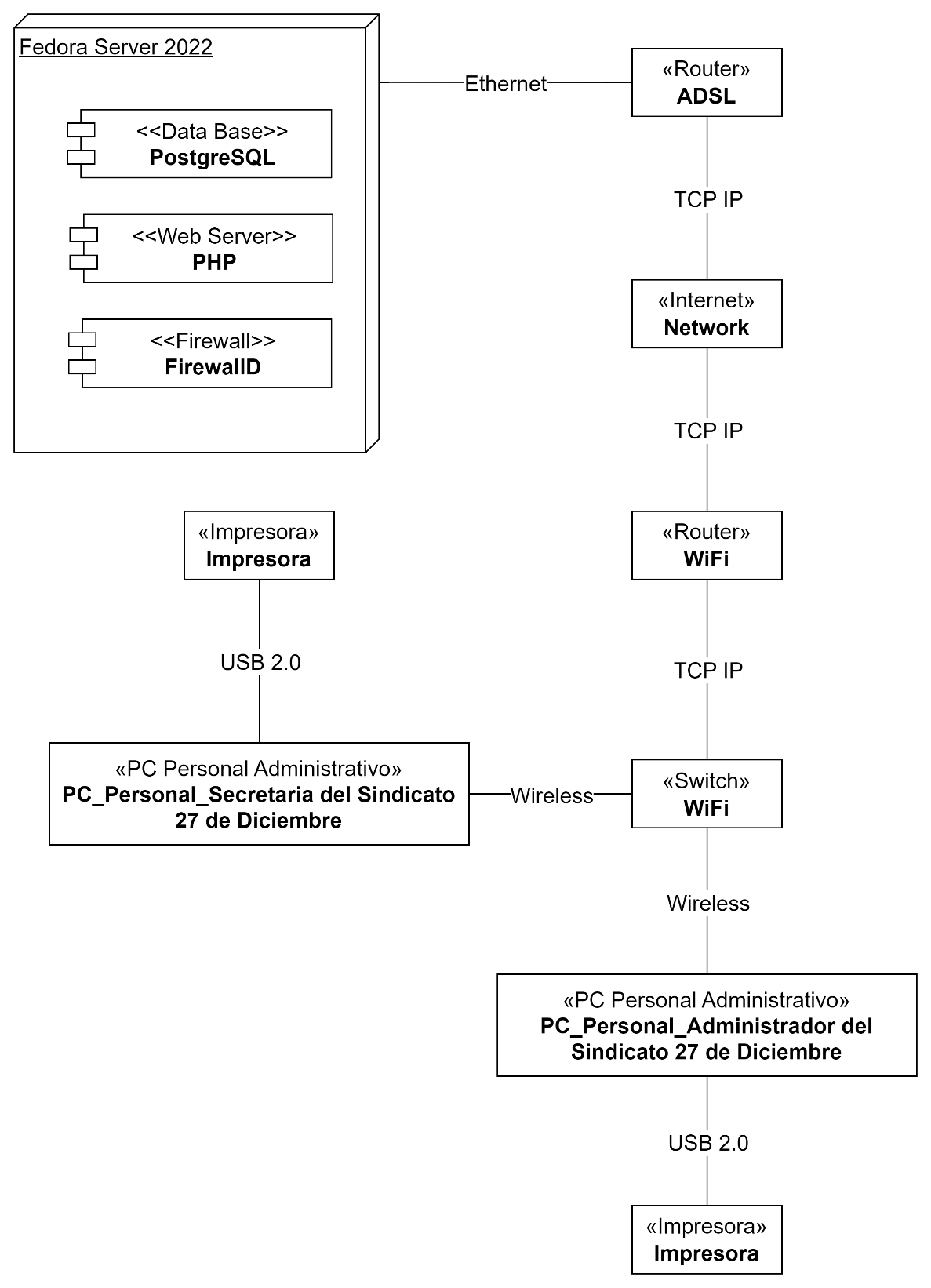
### Diagrama de Secuencia: Gestionar Promoción



## Diseño físico de la arquitectura.

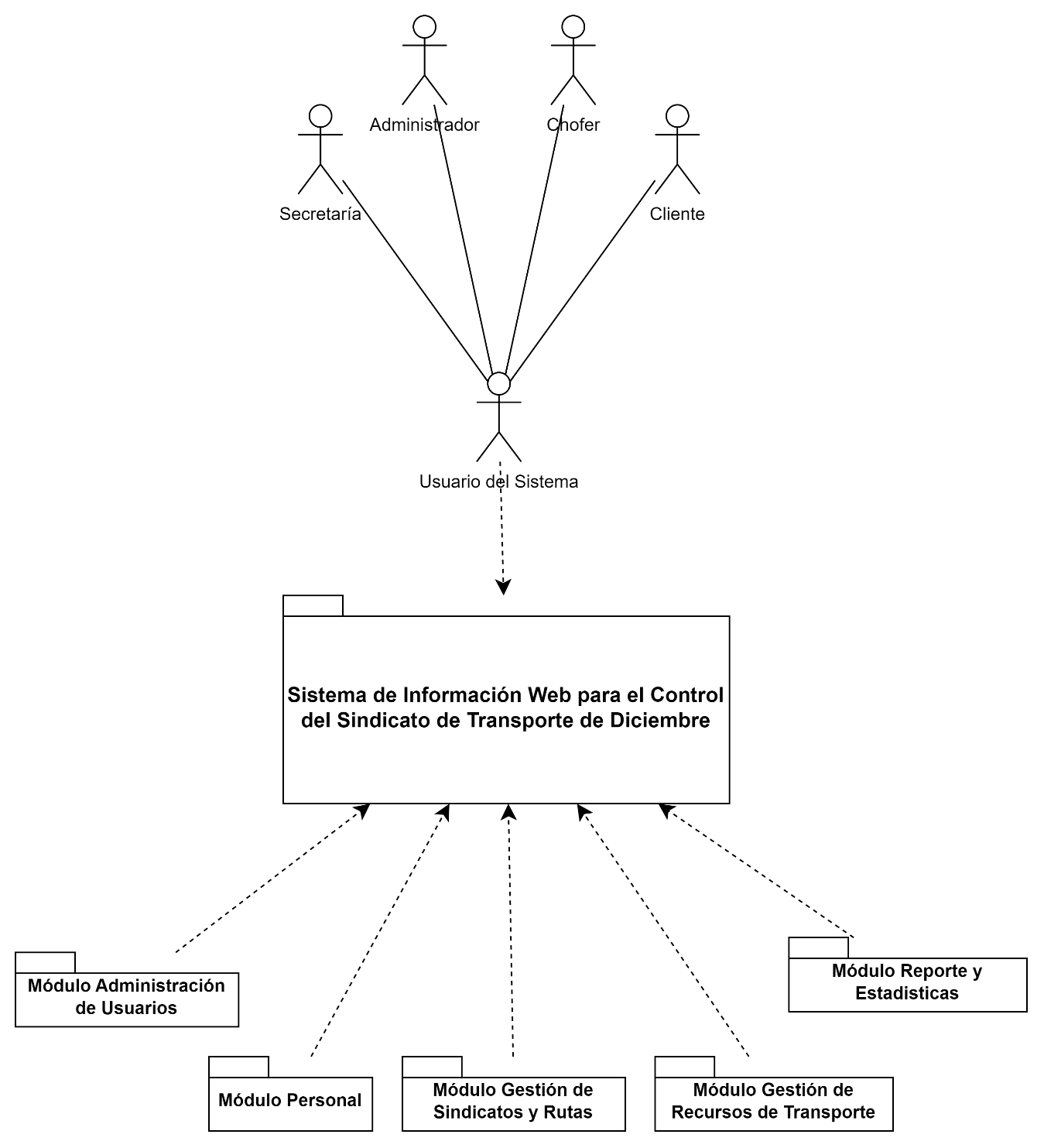
### Diagrama de despliegue.

**Ilustración 86.** Diagrama de despliegue del sistema web.



## Identificar paquetes

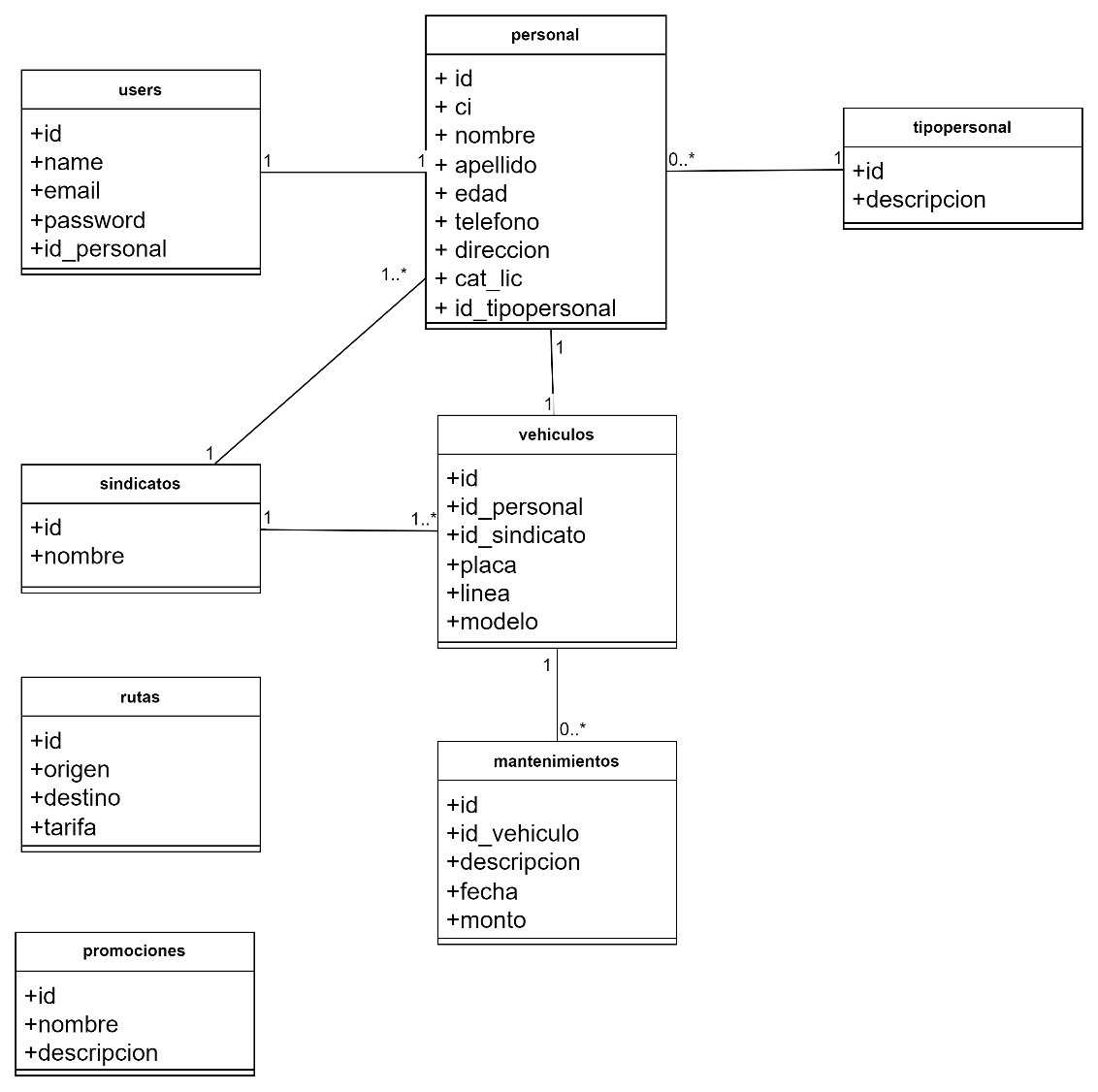
**Ilustración 87**. Sistema de Información Web para el Control del Sindicato de Transporte de Diciembre

 .

## Diseño conceptual de la base de datos

### Diagrama de clases de la base de datos.

**Ilustración 88.** Diagrama de clases final, realizado en Drawio.io



## Diseño lógico

Tabla N: Mapeo de la tabla: User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | | | | |
| id | name | email | password | id\_personal |
| pk |  |  |  | fk |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Personal** | | | | | | | | | |
| id | ci | nombre | apellido | edad | telefono | direccion | cat\_lic | id\_tipopersonal | id\_sindicato |
| pk |  |  |  |  |  |  |  | fk | fk |

Tabla N: Mapeo de la tabla: Personal

Tabla N: Mapeo de la tabla: TipoPersonal

|  |  |
| --- | --- |
| **TipoPersonal** | |
| id | descripcion |
| pk |  |

Tabla N: Mapeo de la tabla: Sindicato

|  |  |
| --- | --- |
| **Sindicato** | |
| id | nombre |
| pk |  |

Tabla N: Mapeo de la tabla: Vehículo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Vehículo** | | | | | | |
| id | linea | placa | marca | modelo | id\_personal | id\_sindicato |
| pk |  |  |  |  | fk | fk |

Tabla N: Mapeo de la tabla: Ruta

|  |  |  |  |
| --- | --- | --- | --- |
| **Ruta** | | | |
| id | origen | destino | tarifa |
| pk |  |  |  |

Tabla N: Mapeo de la tabla: Mantenimiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mantenimiento** | | | | | |
| id | descripcion | fecha | monto | edad | id\_vehiculo |
| pk |  |  |  |  | fk |

Tabla N: Mapeo de la tabla: Promoción

|  |  |  |
| --- | --- | --- |
| **Promoción** | | |
| id | nombre | descripcion |
| pk |  |  |

## Diseño físico

Tabla N: Diseño físico. Tabla User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| Id | bigint |  | Si | No | No |
| name | Character varying | 255 | No | No | No |
| email | Character varying | 255 | No | No | No |
| email\_verified\_at | timestamp without time zone |  | No | No | Si |
| password | Character varying | 255 | No | No | No |
| remember\_token | Character varying | 100 | No | No | Si |
| id\_personal | bigint |  | No | Si | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Personal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| ci | Character varying | 255 | No | No | No |
| nombre | Character varying | 255 | No | No | No |
| apellido | Character varying | 255 | No | No | No |
| edad | Character varying | 255 | No | No | No |
| telefono | Character varying | 255 | No | No | No |
| direccion | Character varying | 255 | No | No | No |
| cat\_lic | Character | 255 | No | No | Si |
| id\_tipopersonal | bigint |  | No | Si | No |
| id\_sindicato | bigint |  | No | Si | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Tipo Personal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| descripcion | Character varying | 255 | No | No | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Sindicato

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| nombre | Character varying | 255 | No | No | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Vehículo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| linea | Character varying | 255 | No | No | No |
| placa | Character varying | 255 | No | No | No |
| marca | Character varying | 255 | No | No | No |
| modelo | Character varying | 255 | No | No | No |
| id\_personal | bigint |  | No | Si | No |
| id\_sindicato | bigint |  | No | Si | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Ruta

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| origen | Character varying | 255 | No | No | No |
| destino | Character varying | 255 | No | No | No |
| tarifa | Character varying | 255 | No | No | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Mantenimiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| descripcion | Character varying | 255 | No | No | No |
| fecha | Character varying | 255 | No | No | No |
| monto | Character varying | 255 | No | No | No |
| id\_vehiculo | bigint |  | No | Si | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

Tabla N: Diseño físico. Tabla Promocion

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Amplitud | Llave Primaria(PK) | Llave Foránea(FK) | Nulos |
| id | bigint |  | Si | No | No |
| nombre | Character varying | 255 | No | No | No |
| descripcion | Character varying | 255 | No | No | No |
| created\_at | timestamp without time zone |  | No | No | Si |
| updated\_at | timestamp without time zone |  | No | No | Si |

## Creación de tablas en la base de datos.

El desarrollo de software con el Framework de Laravel, trae muchas ventajas y facilidades para desarrollar lo más óptimo posible, una de esas herramientas que trae consigo es la llamada Migrations. Esto hace que sea más rápido el proceso de crear las tablas, haciendo que solo indiquemos los atributos de una tabla y cada tabla se representa con una clase dedicado para ello.

Una vez culminado en crear todas las clases para cada tabla, se puede ejecutar todo con un comando: php artisan migrate

Asimismo, tiene las opciones para refrescar y resetear todas las tablas de manera muy fácil:

php artisan migrate:refresh

php artisan migrate:reset

**Migraciones Laravel.**

Nombre del archivo Clase para crear la tabla User:

2014\_10\_12\_000000\_create\_users\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration

{

public function up()

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->string('email')->unique();

$table->timestamp('email\_verified\_at')->nullable();

$table->string('password');

$table->rememberToken();

$table->unsignedBigInteger('id\_personal');

$table->foreign('id\_personal')->references('id')->on ('personal')->cascadeOnDelete()->cascadeOnUpdate();

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('users');

}

}

Nombre del archivo Clase para crear la tabla Personal:

2014\_07\_17\_202925\_create\_personal\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreatePersonalTable extends Migration

{

public function up()

{

Schema::create('personal', function (Blueprint $table) {

$table->id();

$table->string('ci');

$table->string('nombre');

$table->string('apellido');

$table->string('edad');

$table->string('telefono');

$table->string('direccion');

$table->char('cat\_lic')->nullable();

$table->unsignedBigInteger('id\_tipopersonal');

$table->unsignedBigInteger('id\_sindicato');

$table->foreign('id\_tipopersonal')->references('id')->on('tipopersonal')->cascadeOnDelete()->cascadeOnUpdate();

$table->foreign('id\_sindicato')->references('id')->on('sindicatos')->cascadeOnDelete()->cascadeOnUpdate();

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('personal');

}

}

Nombre del archivo Clase para crear la tabla Tipo Personal:

2014\_07\_17\_202844\_create\_tipopersonal\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateTipoPersonalTable extends Migration

{

public function up()

{

Schema::create('tipopersonal', function (Blueprint $table) {

$table->id();

$table->string('descripcion');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('tipopersonal');

}

}

Nombre del archivo Clase para crear la tabla Sindicato:

2014\_06\_25\_230757\_create\_sindicatos\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateSindicatosTable extends Migration

{

public function up()

{

Schema::create('sindicatos', function (Blueprint $table) {

$table->id();

$table->string('nombre');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('sindicatos');

}

}

Nombre del archivo Clase para crear la tabla Vehículo:

2023\_06\_18\_042034\_create\_vehiculos\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateVehiculosTable extends Migration

{

public function up()

{

Schema::create('vehiculos', function (Blueprint $table) {

$table->id();

$table->integer('linea');

$table->string('placa');

$table->string('marca');

$table->string('modelo');

$table->unsignedBigInteger('id\_personal');

$table->unsignedBigInteger('id\_sindicato');

$table->foreign('id\_personal')->references('id')->on('personal');

$table->foreign('id\_sindicato')->references('id')->on('sindicatos');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('vehiculos');

}

}

Nombre del archivo Clase para crear la tabla Ruta:

2023\_06\_25\_230922\_create\_rutas\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateRutasTable extends Migration

{

public function up()

{

Schema::create('rutas', function (Blueprint $table) {

$table->id();

$table->string('origen');

$table->string('destino');

$table->string('tarifa');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('rutas');

}

}

Nombre del archivo Clase para crear la tabla Mantenimiento:

2023\_06\_25\_220819\_create\_mantenimientos\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateMantenimientosTable extends Migration

{

public function up()

{

Schema::create('mantenimientos', function (Blueprint $table) {

$table->id();

$table->string('descripcion');

$table->string('fecha');

$table->string('monto');

$table->unsignedBigInteger('id\_vehiculo');

$table->foreign('id\_vehiculo')->references('id')->on('vehiculos');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('mantenimientos');

}

}

Nombre del archivo Clase para crear la tabla Promociones:

2023\_06\_25\_230849\_create\_promociones\_table.php

Código fuente:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreatePromocionesTable extends Migration

{

public function up()

{

Schema::create('promociones', function (Blueprint $table) {

$table->id();

$table->string('nombre');

$table->string('descripcion');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('promociones');

}

}

## Diseño de interfaces gráficas para el usuario

### Interfaces de interación para Usuario Administrador.

Se encarga de la mayoría funcionalidades del sistema, así también como crear cuentas de usuario y asignar roles de acceso,.

Ilustración 92. Formulario de inicio de sesión (Administrador)

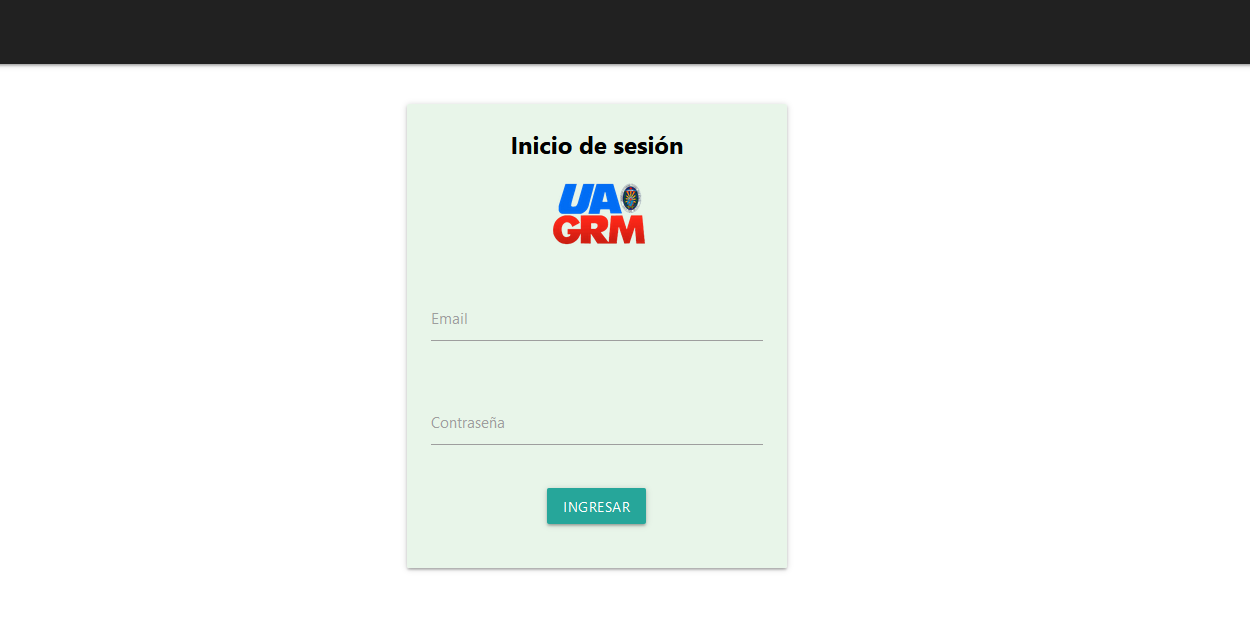


Ilustración 93. Módulos de acceso y gestión para un Administrador.

# BIBLIOGRAFÍA

<https://iessanvicente.com/colaboraciones/postgreSQL.pdf>

<https://desarrolloweb.com/home/laravel>

<https://www.startechup.com/es/blog/what-is-laravel/>

<https://es.wikipedia.org/wiki/PHP>

<https://es.wikipedia.org/wiki/HTML>

<https://es.wikipedia.org/wiki/CSS>

<https://es.wikipedia.org/wiki/JavaScript>

<https://materializecss.com/about.html>

<https://keepcoding.io/blog/que-es-drawio/#Caracteristicas_de_Drawio>

<https://www.atlassian.com/es/git/tutorials/what-is-git>

<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>

<https://es.wikipedia.org/wiki/Proceso_unificado>

<http://informatica-iutll.blogspot.com/2013/03/proceso-unificado-de-desarrollo.html>