

**Capítulo I:**  
**El Proyecto**

## **I.1. Presentación del proyecto**

### **I.1.1 Título del proyecto**

Mejorar el control y la administración de los vehículos de la empresa Constructora y Consultora Bristlecone S.R.L. a través de las TIC.

### **I.1.2 Carrera y/o unidad**

Ingeniería Informática.

### **I.1.3 Facultad**

Ciencias y Tecnología.

### **I.1.4 Entidades asociadas**

- Universidad Autónoma Juan Misael Saracho.
- Consultora y Constructora Bristlecone S.R.L.

### **I.1.5 Duración del proyecto**

El desarrollo del proyecto fue de 8 meses de acuerdo a lo establecido.

### **I.1.6 Área y/o línea de investigación priorizado**

Tecnologías de información y comunicación/Desarrollo de Sistemas y Software/Sistemas de Información Geográfica.

### **I.1.7 Responsabilidad del proyecto**

Carrera de Ingeniería Informática – Taller 3.

## I.1.8 Personal vinculado al proyecto

### I.1.8.1 Director del proyecto

**Tabla 1**

*Director del proyecto*

Apellido Paterno	Apellido Materno	Nombres	C.I.
Ancasi	Zegarra	Saúl	7152403
Profesión	Carrera	Facultad	
Estudiante	Ingeniería Informática	Ciencias y Tecnología	
Celular	Correo	Firma	
68709094	<a href="mailto:saulanze@gmail.com">saulanze@gmail.com</a>		

### I.1.8.2 Equipo de trabajo

**Tabla 2**

*Equipo de trabajo*

Categoría	Nombres y apellidos	Carrera y/o profesión	C.I.	Firma
Director	Saúl Ancasi Zegarra	Ingeniería Informática	7152403	
Analista	Saúl Ancasi Zegarra	Ingeniería Informática	7152403	
Programador	Saúl Ancasi Zegarra	Ingeniería Informática	7152403	
Ingeniero de software	Saúl Ancasi Zegarra	Ingeniería Informática	7152403	

### I.1.8.3 Equipo de trabajo que cooperará con el proyecto

**Tabla 3**

*Equipo de trabajo que cooperará con el proyecto*

Institución	Consultora y Constructora Bristlecone S.R.L.		
Dirección	Calle José Luis Boyan		
Teléfono	66-64608		
Nombres y apellidos	Cargo	C.I.	Firma
Saúl Ancasi Zegarra	Director del proyecto, analista, programador	7152403	
Ing. Sergio Maldonado Duchén	Gerente General		

### I.1.8.4 Actividades previstas para los integrantes del equipo de trabajo

**Tabla 4**

*Actividades previstas del equipo de trabajo*

Responsable	Actividades
Director del proyecto	<ul style="list-style-type: none"> <li>▪ Planificar y definir los objetivos.</li> <li>▪ Planificar el cronograma del proyecto.</li> <li>▪ Controlar las actividades de acuerdo al cronograma del proyecto.</li> <li>▪ Coordinar y supervisar el desarrollo y la implementación del software.</li> <li>▪ Dirigir el equipo.</li> <li>▪ Presentar el proyecto finalizado.</li> </ul>

Analista	<ul style="list-style-type: none"><li>▪ Recopilar información de la empresa a través de distintos medios.</li><li>▪ Realizar un análisis y diseño de los requerimientos de la empresa.</li><li>▪ Realizar un análisis y diseño de los requisitos del sistema.</li><li>▪ Realizar la documentación del proyecto.</li></ul>
Programador	<ul style="list-style-type: none"><li>▪ Diseñar los distintos componentes del sistema de acuerdo a los requisitos del sistema.</li><li>▪ Definir las herramientas óptimas para el desarrollo del sistema.</li><li>▪ Desarrollar el sistema de acuerdo a los requisitos.</li></ul>
Ingeniero de software	<ul style="list-style-type: none"><li>▪ Elaboración del modelo de datos.</li><li>▪ Análisis de las pruebas de funcionales.</li><li>▪ Ejecución de las pruebas funcionales.</li></ul>

## **I.2. Perfil del proyecto**

### **I.2.1 Introducción**

En los tiempos modernos en los que vivimos la tecnología significo para la humanidad un cambio tan grande que llegó a cambiar por completo nuestra manera de vivir. Los ingenieros informáticos haciendo uso de las TIC (tecnologías de la información y comunicación) tenemos como objetivo mejorar y agilizar procesos, además de dar solución a distintos problemas en distintas áreas laborales.

El sector empresarial que se dedica a la construcción de obras civiles sin duda es uno de los más demandados, por lo cual este sector como cualquier otro, con el pasar de los años va evolucionando por lo cual deben hacer uso de las distintas tecnologías actuales para administrar y controlar de una mejor manera todos sus recursos ya sean materiales o humanos.

Bristlecone S.R.L. es una empresa consultora y constructora con una trayectoria de más de 10 años, que tiene sede en la ciudad de Tarija, actualmente hace uso de distintos sistemas que facilitan el desempeño de su labor, pero existen todavía falencias en algunos aspectos que maneja la empresa, tal es el caso de la administración y el control de sus vehículos y maquinaria pesada.

Por tanto, teniendo en cuenta lo mencionado previamente, el presente proyecto pretende dar la facilidad a la empresa Consultora y Constructora Bristlecone S.R.L. para un mejor control y administración de sus vehículos, a través de un sistema informático.

## **I.2.2 Descripción del proyecto**

### **I.2.2.1 Antecedentes**

#### **I.2.2.2 Antecedentes internacionales**

Como antecedentes internacionales tomaremos las siguientes investigaciones que contemplan aspectos similares al presente proyecto.

Audistillo y Delgado (2012) que en su trabajo de investigación “Sistema de Localización monitoreo y control vehicular basado en los protocolos GPS/GSM/GPRS.” Tesis para obtener el título de Ingeniero Electrónico, concluye que: el sistema permite optimizar las rutas de los vehículos, el ahorro de combustible, reducción de costos relacionados con el mantenimiento y además minimizaría los problemas medio ambientales relacionados con la contaminación.

Uvidia y Estrada (2015) con su trabajo de investigación “Diseño e implementación de un sistema electrónico prototipo georreferenciado para monitoreo y control de puertas de buses en paradas específicas de la ciudad de Riobamba.” Tesis para obtener el título de Ingeniero en Electrónica, Telecomunicaciones y Redes concluye que: el trabajo realizado permitió el control electrónico para abrir y cerrar automáticamente los seguros de las puertas de los vehículos en sitios específicos, de dónde podemos deducir que la información GPS obtenida nos permite delimitar de cierta forma los lugares que pueden recorrer los vehículos.

## **I.3. Justificación del proyecto**

Para entrar en contexto con el proyecto a desarrollarse es necesario mencionar que sucede actualmente en la Consultora y Constructora Bristlecone S.R.L., entonces la empresa presenta un problema con la administración de sus vehículos y maquinaria pesada que se emplean en los

proyectos que ejecuta la empresa, esto debido a que existen situaciones en las que los conductores encargados de los vehículos hacen uso inadecuado de los vehículos y no cumplen con sus horas de trabajo diarias impuestas por los residentes de obra, quienes son los encargados de administrar la ejecución del proyecto.

Cada vehículo y maquinaria pesada es asignado a un trabajo en específico durante el día, en ciertas ocasiones suceden imprevistos, en estos casos se designan a los vehículos con labores de menor impacto dejar de cumplir su trabajo asignado, para dar solución a las situaciones imprevistas.

Con el fin de subsanar los problemas del uso inadecuado de los vehículos la gerencia decidió implementar “Hojas de Parte Diaria”, en dichas hojas los empleados deben registrar cada actividad realizada durante el día, incluyendo la hora, combustible, kilometraje inicial y final. El método funciona hasta cierto punto ya que es fácil para cada empleado no registrar con la verdad sus hojas de parte diaria o simplemente no realizarlas cada día y dejarlo para último momento acumulando el trabajo para los residentes de obra.

Por otro lado, las “hojas de parte diario” dan más trabajo a los contadores ya que ellos son los encargados en documentar cada parte diario de cada empleado, parte del problema está en que la mayoría de empleados carecen de buena caligrafía y ortografía, esto provoca que se tarde mucho más en la documentación del parte diario al tratar de entender lo escrito por los empleados, por otro lado, al ser esta información física es vulnerable a ser pedida.

Con el fin de dar solución a los problemas mencionados se presenta este proyecto para el control y administración de los vehículos de la empresa.

Entonces el presente proyecto planea ser implementado en la empresa Consultora y Constructora Bristlecone S.R.L. para contar con información relevante respecto a la localización

de los vehículos haciendo uso del Sistema de Posicionamiento Global (GPS), permitiendo tener respaldo de los trayectos que realizan los vehículos.

De igual manera se pretende facilitar la redacción y documentación de los partes diarios de los empleados.

### **Tecnológico**

Las empresas deben estar en constante actualización respecto a las nuevas tecnologías, la incorporación de tecnologías de la información y comunicación que permiten la automatización de procesos en las empresas permiten facilitar los procesos y realizarlos en tiempos reducidos, de esta manera se ahorra valioso tiempo que puede ser empleado en actividades de mayor relevancia.

### **Económico**

Para el uso y disponibilidad del sistema se tendrán que tomar en cuenta algunos aspectos económicos en cuanto al costo para mantener el sistema en funcionamiento, pero al hacer uso del sistema se ahorra tiempo que se cree es de alto valor para una empresa consultora y constructora que tiene que estar al tanto de los proyectos que ejecuta, por tanto, sin duda el sistema de alguna manera llegara a aumentar sus ganancias.

### **Social**

El personal de la empresa tendrá mayor facilidad para redactar y documentar los partes diarios, realizándolos de manera digital por medio del sistema.

## **I.4. Planteamiento del problema**

La empresa Consultora y Constructora Bristlecone S.R.L. no cuenta con información pertinente y respaldo referente a la ubicación de sus vehículos y el recorrido que realizan estos,

además de presentar demora en cuanto a la documentación de los partes diarios que realizan los empleados de manera manual.

**Tabla 5**

*Cuadro de involucrados*

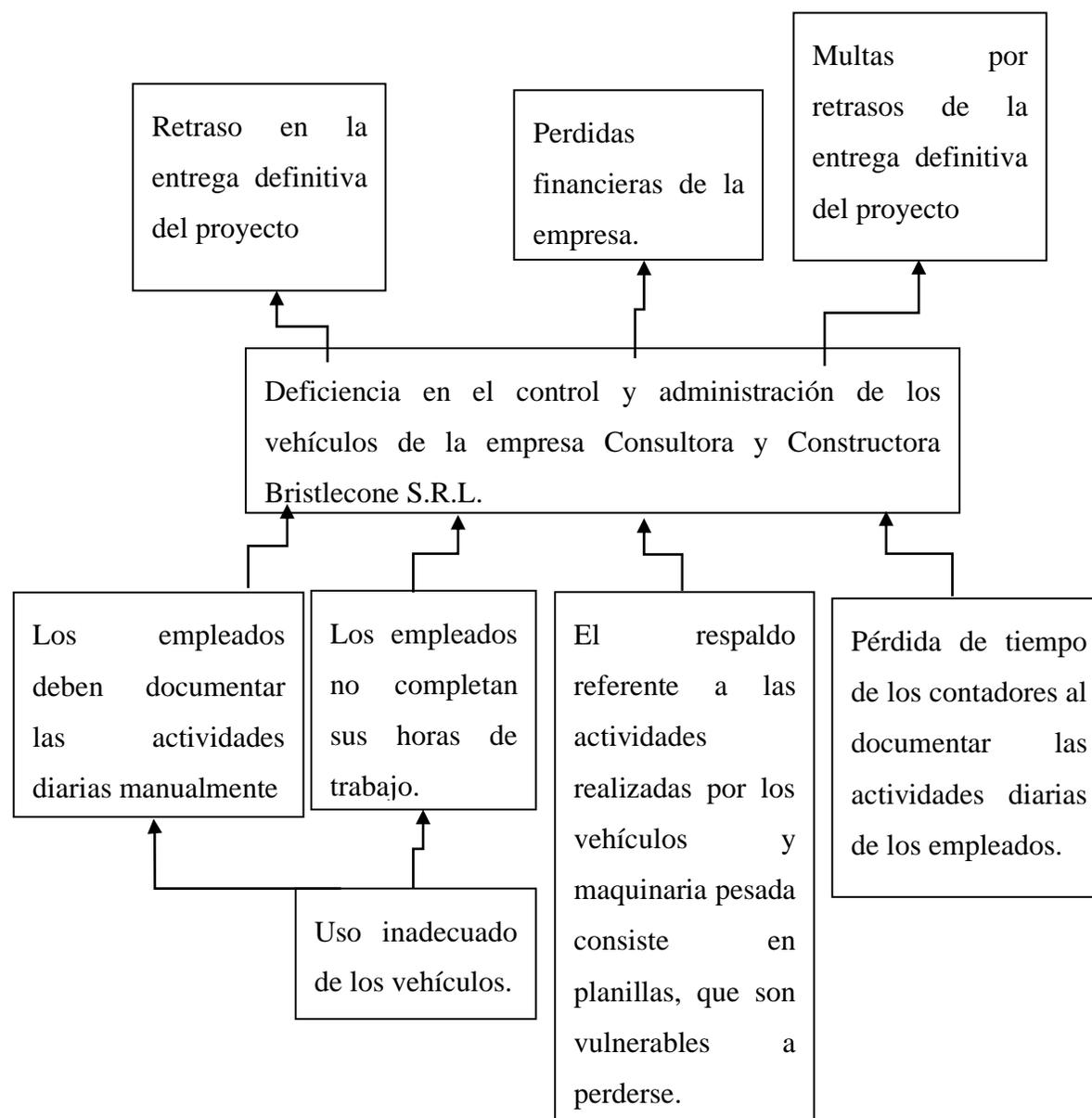
<b>Grupo</b>	<b>Intereses</b>	<b>Problemas</b>	<b>Recursos/Mandatos</b>
Gerencia	Conclusión a tiempo de los proyectos adjudicados por la empresa	Retrasos en la ejecución de los proyectos. Multas por retraso en la entrega definitiva del proyecto. Pérdidas financieras.	R: Presupuesto del proyecto. Crédito bancario. Cuenta Corriente. Caja de Ahorro. M: Administrar la empresa
Jefe de Obra (Administrador)	Administrar de manera óptima todos los recursos de la empresa para el desarrollo del proyecto a ejecutarse. Cumplir con los plazos para la entrega provisional y definitiva del proyecto. Conocer la ubicación de los vehículos y si están siendo operados de manera eficiente. Obtener información	No se cuenta con información pertinente sobre el uso de los vehículos. El respaldo referente a las actividades realizadas por los vehículos y maquinaria pesada consiste en planillas, que son vulnerables a perderse.	R: Presupuesto del proyecto. M: Administrar todo lo relacionado con el proyecto asignado

	referente a los vehículos.		
Contadores	Realizar la documentación de la parte diaria de los empleados lo antes posible.	Pérdida de tiempo de los contadores al documentar las actividades diarias de los empleados.	M: Administrar la documentación contable R: Personal capacitado
Empleados	Desempeñar actividades designadas oportunamente	Uso inadecuado de los vehículos. No completar sus horas de trabajo diarias.	M: Trabajar en sus actividades designadas R: Personal capacitado

### I.4.1 Árbol de problemas

**Figura 1**

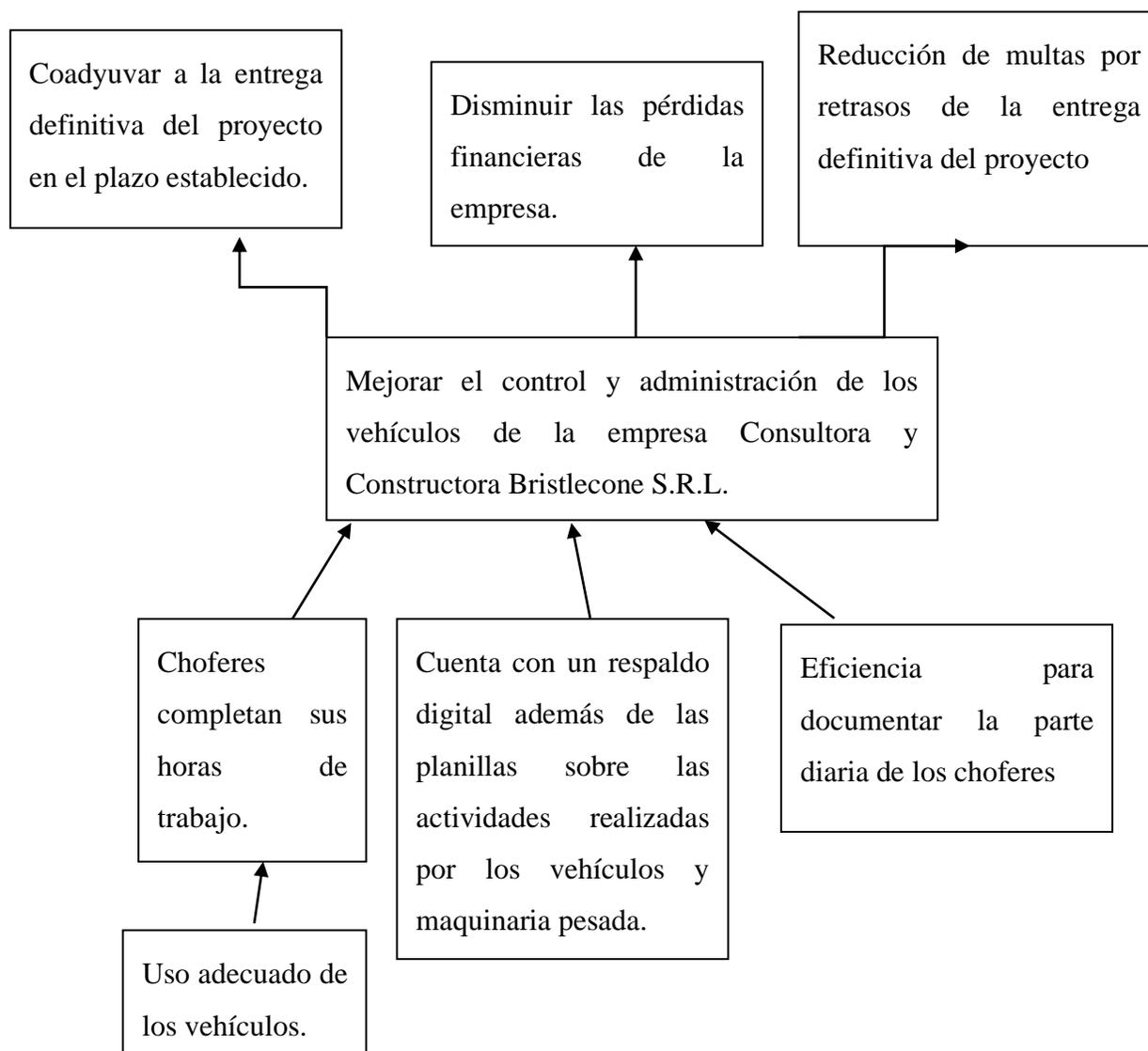
*Árbol de problemas*



## I.4.2 Árbol de objetivos

**Figura 2**

*Árbol de objetivos*



### I.4.3 Situación planteada con y sin proyecto

**Tabla 6**

*Situación planteada con y sin proyecto*

Situación sin proyecto	Situación con proyecto
<ul style="list-style-type: none"> <li>• No existen registros del trayecto recorrido por los vehículos de la empresa.</li> <li>• No es posible conocer la ubicación actual de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L.</li> <li>• Existe demora en la documentación de los partes diarios de los empleados.</li> <li>• Pérdida de información referente a los partes diarios de los empleados.</li> </ul>	<p>Existen registros del trayecto recorrido por los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L.</p> <p>Es posible conocer la ubicación actual de los vehículos de empresa Consultora y Constructora Bristlecone S.R.L.</p> <p>La documentación de los partes diarios es oportuna.</p> <p>Almacenamiento de la información en una base de datos.</p>

## I.5. Objetivos

### I.5.1 Objetivo general

Mejorar el control y administración de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L. a través de las TIC.

### I.5.2 Objetivos específicos

Desarrollar un sistema informático para administrar y controlar los vehículos.

Capacitar al personal que hará uso del sistema.

## **I.6. Metodología de desarrollo**

En el presente proyecto la metodología de desarrollo a utilizarse será la Metodología RUP (Rational Unified Process).

RUP es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. (Wikipedia)

### Ciclo de vida

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema.

### Fases

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición.

**Fase de Inicio.** Durante la fase de inicio se desarrolla una descripción del producto final, y se presenta el análisis del negocio.

**Fase de Elaboración.** Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.

**Fase de Construcción.** Durante la fase de construcción se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo.

**Fase de Transición.** La fase de transición cubre el período durante el cual el producto se convierte en la versión beta.

## I.7. Matriz del marco lógico

**Tabla 7**

*Matriz del marco lógico*

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
<p>Fin</p> <p>Contribuir a mejorar la imagen corporativa de la empresa Consultora y Constructora Bristlecone S.R.L.</p>	<p>Luego de dos años de finalizado el proyecto, la percepción de satisfacción del personal de la empresa Consultora y Constructora Bristlecone S.R.L. se ha incrementado en un 30%</p>	<p>Cuadro comparativo de encuestas realizadas en el año base y luego de dos años, respecto a la percepción de los clientes de la empresa.</p>	<p>La empresa Bristlecone S.R.L. tiene respaldo del sistema para el control y administración de sus vehículos.</p> <p>Se mantienen las condiciones tecnológicas y administrativas necesarias para el funcionamiento del sistema.</p>
<p>Propósito</p> <p>Control y administración de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L., mejorado.</p>	<p>Al finalizar el proyecto, se ha automatizado (como un criterio de mejora) al menos un 90% de los procesos inherentes a la administración de los vehículos de la empresa Consultora y</p>	<p>Cuadro comparativo referente a los procesos inherentes a la administración de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L. y los procesos automatizados,</p>	<p>Se presupuesta anualmente el sistema informático.</p> <p>Los funcionarios hacen uso frecuente del sistema.</p>

	Constructora Bristlecone S.R.L.	avalado por el docente de Taller III	
<p>Componentes</p> <p>C1.- Sistema informático para administrar los vehículos desarrollado.</p> <p>C2.- Estrategia, implementada de capacitación al personal que hará uso del sistema.</p>	<p>Al finalizar el proyecto, se ha desarrollado un sistema informático para la administración de los vehículos de la empresa de acuerdo a los requerimientos expresados bajo la norma IEE830.</p> <p>Finalizado el proyecto, se ha desarrollado un curso de capacitación a los funcionarios de la empresa.</p>	<p>Informe de entrega final del sistema informático, avalado por la gerencia de la empresa Bristlecone SRL y/o el docente de Taller III.</p> <p>Informe de la gerencia respecto a la capacitación realizada, adjuntando la lista de participantes.</p>	<p>Interés por parte del personal de la empresa en la capacitación para el uso del sistema informático</p>
<p>Actividades</p> <p>C1.- Sistema informático para la administración de los vehículos.</p> <ul style="list-style-type: none"> <li>• Inicio</li> <li>• Elaboración</li> <li>• Construcción</li> </ul>	<p>Resumen presupuesto C1</p> <p>Servicios personales 16000Bs</p> <p>Servicios no personales 1990Bs</p> <p>Materiales y suministros 4740Bs</p> <p>Activos reales 5000Bs</p>	<p>Informe presupuestario del proyecto avalado por el director.</p>	<p>Los desembolsos se realizan de acuerdo cronograma.</p>

•C2.- Capacitación del personal	Total	27.730Bs	
• Elaboración de manual de usuario	Resumen presupuesto C2		
• Capacitación del personal de la empresa Bristlecone S.R.L. en el uso del sistema	Servicios personales	150Bs	
	Servicios no personales	330Bs	
	Materiales y suministros	40Bs	
	Total	520Bs	

## **I.8. Resultados esperados**

### **Desarrollar un sistema informático para administrar los vehículos**

Se refiere a un sistema informático que permita la interpretación de los datos recibidos por los dispositivos GPS equipados en los vehículos; como ser la visualización de la ubicación geográfica y rutas recorridas por el vehículo; además de contar con información referente a las actividades realizadas por los vehículos.

### **Capacitar al personal que hará uso del sistema**

Se refiere a la capacitación del personal que hará uso del sistema “GEOBRIS”, que una vez realizado al menos un 80% se capaz de usar de manera óptima el sistema.

## **I.9. Beneficiarios**

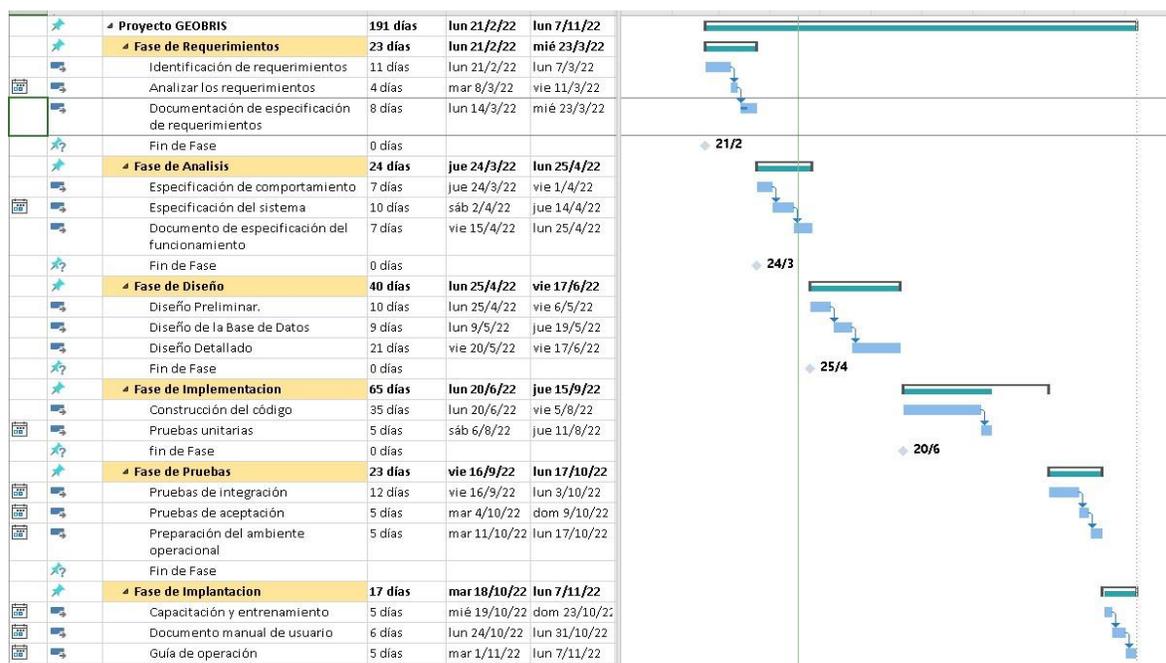
### **I.9.1 Beneficiarios Directos**

Tanto el personal administrativo como el personal trabajador de la empresa Consultora y Constructora Bristlecone S.R.L.

## I.10. Cronograma

Figura 3

*Cronograma de actividades*



# **Capítulo II:**

## **Componentes**

## **II Componente 1: Sistema informático**

### **II.1. Plan de desarrollo de software**

#### **II.1.1 Introducción**

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido ofertado por el universitario Saul Ancasi Zegarra basado en la metodología de Rational Unified Process (RUP) en la que únicamente se procederá a cumplir con las tres primeras fases, las cuales marcan la metodología. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio, Elaboración y Construcción.

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

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

**Construcción.** - En esta fase se profundiza en el diseño de los componentes del sistema y de manera iterativa se van añadiendo las funcionalidades al software a medida que se construyen y prueban, permitiendo a la vez que se puedan ir incorporando cambios.

Al final de esta fase se obtiene un sistema completamente operativo y la documentación (diagrama de clases, de secuencia, modelo entidad-relación, modelo de dominio, manual de instalación, manual de usuario, manual de seguridad y otros) para entregar a los usuarios.

El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los entregables que serán generados. Este documento es a su vez uno de los artefactos de RUP.

### Elementos del RUP

- Actividades: Procesos que se han de realizar en cada etapa/iteración.
- Trabajadores: Personas involucradas en cada actividad del proyecto.
- Artefactos: Herramientas empleadas para el desarrollo del proyecto. Puede ser un documento, un modelo, un elemento del modelo. Estos artefactos (entre otros) son los siguientes:
  - Inicio
    - Documento Visión
    - Especificación de Requerimientos
  - Elaboración
    - Diagramas de caso de uso
  - Construcción
    - Documento Arquitectura que trabaja con las siguientes vistas:
      - Vista lógica
        - Diagrama de clases
        - Modelo E-R
      - Vista de implementación
        - Diagrama de secuencia
        - Diagrama de actividades

## **II.1.2 Vista general del proyecto**

### **II.1.2.1 Propósito**

Contribuir a mejorar la imagen corporativa de la empresa Consultora y Constructora Bristlecone S.R.L.

### **II.1.2.2 Alcance**

Al ser un sistema web, a este se podrá acceder desde cualquier dispositivo compatible con algún navegador.

Los usuarios tendrán que autenticarse para ingresar al sistema.

El sistema será desarrollado de acuerdo a los requerimientos de la empresa Consultora y Constructora Bristlecone S.R.L.

Para la ubicación en tiempo real se utilizará el GPS incorporado de los teléfonos inteligentes.

Cualquier empleado podrá realizar su parte diario.

### **II.1.2.3 Objetivos**

#### **II.1.2.3.1 Objetivo general**

Mejorar el control y administración de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L.

#### **II.1.2.3.2 Objetivos específicos**

Desarrollar un sistema informático para administrar y controlar los vehículos.

Capacitar al personal que hará uso del sistema.

#### **II.1.2.4 Suposiciones y restricciones**

##### **II.1.2.4.1 Suposiciones**

Los equipos y dispositivos móviles en los que se ejecute el sistema cumplirán con los requisitos mínimos para así evitar un mal funcionamiento del sistema.

Al ser el sistema desarrollado en el sistema operativo Windows 10, puede que no se ejecute de manera correcta en sistemas operativos inferiores.

El sistema permitirá el desarrollo para diversas plataformas.

##### **II.1.2.4.2 Restricciones**

El sistema a desarrollarse presenta restricciones las cuales se deben tener en cuenta tanto al momento de desarrollo del software, como cuando se implemente.

El sistema necesitará de conexión a internet.

La empresa tendrá que tener un presupuesto para mantener el sistema en funcionamiento.

El sistema deberá tener un diseño simple e intuitivo.

Los dispositivos GPS deben tener conexión constante con los servidores.

El sistema solo estará disponible para su instalación en sistemas operativos Windows.

##### **II.1.2.5 Marco Teórico**

Con el fin de cumplir con los requerimientos de la empresa Consultora y Constructora Bristlecone S.R.L. consideraremos las siguientes herramientas y tecnologías para el desarrollo del sistema.

#### **Metodología RUP**

El Proceso Racional Unificado o RUP (por sus siglas en inglés de Rational Unified Process) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software, también desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo con las necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el Rational Unified Process, que se vendiera como producto independiente.

### **Principios de desarrollo**

La Filosofía del RUP está basado en 6 principios clave que son los siguientes:

- **Adaptar el proceso**

El proceso deberá adaptarse a las necesidades del cliente ya que es muy importante interactuar con él. Las características propias del proyecto, el tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

- **Equilibrar prioridades**

Los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe poder encontrarse un equilibrio que satisfaga los deseos de todos. Gracias a este equilibrio se podrán corregir desacuerdos que surjan en el futuro. Al igual esta metodología está acorde con el Lenguaje Unificado de Modelado (UML).

- **Demostrar valor iterativamente**

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto, así como también los riesgos involucrados.

- **Colaboración entre equipos**

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes, resultados, etc.

- **Enfocarse en la calidad**

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente, también es una estrategia de desarrollo de software.

- **Elevar el nivel de abstracción**

Este principio dominante motiva el uso de conceptos reutilizables tales como patrones de diseño del software, lenguajes 4GL o esquemas por nombrar algunos. Estos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo, con UML.

## **Ciclo de vida**

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan pocas pero grandes y formales iteraciones en número variable según el proyecto. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (línea base) de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se seleccionan algunos Casos de Uso, se refinan su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

## **Fases**

Establece oportunidad y alcance.

Identifica las entidades externas o actores con las que se trata.

Identifica los casos de uso.

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

Proceso. Las etapas de esta sección son:

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

Soporte. En esta parte nos encontramos con las siguientes etapas:

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

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

- Inicio (también llamado Incepción o Concepción).
- Elaboración.
- Desarrollo (también llamado Implementación, Construcción).
- Cierre (también llamado Transición).

## **Fase de Inicio**

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores o involucrados del proyecto en el cual tenemos que, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

## **Fase de Elaboración**

En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

## **Fase de Desarrollo o Construcción**

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

## **Fase de Transición**

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

## **Visual Studio Code**

Sera nuestro editor y entorno de desarrollo principal para la construcción de nuestro sistema.

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de

sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Básicamente, Visual Studio Code (también conocido como VS Code) es un software editor de código fuente de Microsoft, con extensiones que pueden admitir prácticamente cualquier lenguaje de programación. Es muy rápido e ideal para el uso diario.

Este software se ha convertido en una de las herramientas de entorno de desarrollo más utilizadas, a pesar del peso relativamente modesto que tiene este software, pero con un sólido conjunto de completas funciones.

Disponiendo de un soporte para cientos de idiomas, el resaltado de sintaxis, la sangría automática, la combinación de corchetes, la selección de cuadros, los fragmentos y otras características de VS Code te permiten comenzar a trabajar en tu proyecto de inmediato.

### **Características de Visual Studio Code**

Las características de Visual Studio Code son las siguientes:

Admite una amplia gama de compatibilidad con lenguajes de programación, incluidos CSS, Go y Dockerfile, además de Java, JavaScript, C++ y Python.

Comparando con otros editores de texto, la interfaz de usuario de VS Code ofrece una muy buena interacción.

Los filtros de código, los depuradores y la compatibilidad con la nube y el soporte del desarrollo web son solo algunas de las extensiones que pueden agregar a VS Code, o incluso crearlas desde cero.

A continuación, analizamos sus principales características y funciones:

### **Paleta de comandos**

El teclado se puede usar para acceder a VS Code con la misma facilidad. La combinación de teclas que abre la paleta de comandos, Ctrl+Shift+P, es la más crucial para recordar. Todas las herramientas de VS Code, incluidos los atajos de teclado para las acciones más frecuentes, son accesibles desde allí.

### **Línea de comandos**

Con la ayuda de la sólida interfaz de línea de comandos de Visual Studio Code, puedes administrar cómo se inicia el editor. Puedes instalar extensiones, abrir varios tipos de archivos e incluso cambiar el idioma de visualización inicial.

### **Modificar el idioma**

Puedes usar el comando "Establecer asociación de archivos" para vincular la extensión de archivo actual con un idioma instalado si deseas mantener el nuevo modo de idioma para ese tipo de archivo.

### **Integración de Git**

La integración de Git está integrada en Visual Studio Code la cual te permite insertar, extraer y confirmar cambios en su código en un repositorio de Git remoto.

### **Atajos de teclado por defecto**

Cada comando y la combinación de teclas correspondiente se enumeran en la paleta de comandos. Puedes usar la paleta de comandos a su favor si olvida el método abreviado de teclado. También puedes descargar la hoja de trucos para los métodos abreviados de teclado de tu plataforma.

### **Vista dividida**

Puedes usar la vista dividida si eres un experto en realizar múltiples tareas y trabajar en dos archivos diferentes en el mismo proyecto a la vez, o si necesitas comparar dos archivos.

### **Barra de estado**

Desde la barra de estado, los usuarios pueden ver los errores y advertencias. Tecla de acceso directo es: Ctrl+Shift+M

Los problemas se pueden filtrar por coincidencia de texto o tipo ("Errores", "Advertencias").

A continuación, veremos algunas ventajas y desventajas de Visual Studio Code.

### **Ventajas:**

- Es gratuito. Es increíble que un software tan bien hecho, confiable y bien mantenido esté disponible para su descarga gratuita.
- Puedes completar la mayoría de las acciones necesarias mientras aún está en una aplicación gracias a la ventana de terminal integrada.
- Presenta una sólida arquitectura de complementos llena de excelentes funciones que hacen que ser un desarrollador sea divertido.
- Los atajos de teclado Comando+P y Comando+Shift+P mejorarán significativamente tu productividad.
- Las opciones de personalización son excelentes. Es sencillo personalizar VS Code según tus necesidades.
- Está desarrollado por Microsoft.
- Fácil de usar.
- Es multiplataforma.

### **Desventajas:**

- La capacidad de mostrar una vista previa de un archivo SVG no está incorporada.

- Ocasionalmente, pueden surgir mensajes de error de la aplicación con detalles insuficientes para corregirlos.

## **JavaScript**

Se hará uso de JavaScript como lenguaje primario para el desarrollo del sistema.

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.

JavaScript se diseñó con una sintaxis similar a C++ y Java, 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.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras

tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, 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 dinámicas 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 es también significativo.

### **Lenguaje Unificado de Modelado (UML)**

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema, 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.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construirlo. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software, gran variedad de metodologías de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que la programación estructurada es una forma de programar como lo

es la programación orientada a objetos (POO), la POO ha sido un complemento perfecto de UML, pero no por eso se utiliza UML solo para lenguajes orientados a objetos.

El Lenguaje Unificado de Modelado (UML) desempeña un rol importante no solo en el desarrollo de software, sino también en los sistemas que no tienen software en muchas industrias, ya que es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso. UML ayuda a mostrar errores potenciales en las estructuras de aplicaciones, el comportamiento del sistema y otros procesos empresariales.

#### **Ventajas:**

- Simplifica las complejidades.
- Mantiene abiertas las líneas de comunicación.
- Automatiza la producción de software y los procesos.
- Ayuda a resolver los problemas arquitectónicos constantes.
- Aumenta la calidad del trabajo.
- Reduce los costos y el tiempo de comercialización.

#### **Tipos de diagramas UML**

Existen dos tipos principales de diagramas UML: diagramas de estructura y diagramas de comportamiento (y dentro de esas categorías se encuentran varios otros). Estas variaciones existen para representar los numerosos tipos de escenarios y diagramas que usan los diferentes tipos de personas.

Desde clientes y gestores de proyectos hasta autores técnicos, diseñadores, analistas, codificadores y encargados de pruebas y control de calidad, cada rol utilizará un diagrama específico que se adapte a sus necesidades. Eso significa que cada disposición requiere un

enfoque y nivel de detalle diferente. El objetivo es que el UML exprese visualmente diagramas que sean fáciles de entender para todos.

### **Diagramas estructurales**

Los diagramas estructurales representan la estructura estática de un software o sistema, y también muestran diferentes niveles de abstracción e implementación. Estos se usan para ayudarlo a visualizar las diversas estructuras que componen un sistema, como una base de datos o aplicación. Muestran la jerarquía de componentes o módulos y cómo se conectan e interactúan entre sí. Estas herramientas ofrecen orientación y garantizan que todas las partes de un sistema funcionen según lo previsto en relación con todas las demás partes.

- **Diagrama de clases.** Este diagrama, el más común en el desarrollo de software, se usa para representar el diseño lógico y físico de un sistema, y muestra sus clases. Tiene un aspecto similar al del diagrama de flujo porque las clases se representan con cuadros. Este diagrama ofrece una imagen de las diferentes clases y la forma en la que se interrelacionan.
- **Diagrama de objetos.** A menudo, este diagrama se usa como una forma de comprobar la revisión de un diagrama de clases para fines de precisión. Muestra los objetos de un sistema y sus relaciones, y ofrece una mejor visión de los potenciales defectos de diseño que necesitan reparación.
- **Diagrama de componentes.** También conocido como diagrama de flujo de componentes, muestra agrupaciones lógicas de elementos y sus relaciones. En otras palabras, ofrece una vista más simplificada de un sistema complejo al desglosarlo en componentes más pequeños. Cada una de las piezas se muestra con una caja

rectangular, que tiene su nombre escrito dentro. Los conectores definen la relación/las dependencias entre los diferentes componentes.

- **Diagrama de estructura compuesta.** Este lo utilizan rara vez las personas externas al campo de desarrollo de software. Aunque es similar a un diagrama de clases, adopta un enfoque más profundo, que describe la estructura interna de múltiples clases y muestra las interacciones entre ellas. Salvo que usted sea desarrollador, la vista de nivel superior probablemente le entregará información suficiente.
- **Diagrama de despliegue.** Este diagrama muestra los componentes de hardware (nodos) y software (artefactos) y sus relaciones. Ofrece una representación visual exacta del lugar donde se implementa cada componente de software.
- **Diagrama de paquetes.** Este se utiliza para representar las dependencias entre los paquetes que componen un modelo. Su objetivo principal es mostrar la relación entre los diversos componentes grandes que forman un sistema complejo.
- **Diagrama de perfiles.** Este es más similar a un lenguaje que a un diagrama. Un diagrama de perfil ayuda a crear nuevas propiedades y semántica para los diagramas UML al definir estereotipos personalizados, valores marcados y restricciones. Estos perfiles le permiten personalizar un metamodelo de UML para diferentes plataformas.

### **Diagramas de comportamiento**

El enfoque aquí está en los aspectos dinámicos del sistema de software o proceso. En estos diagramas se muestra la funcionalidad de un sistema y se enfatiza lo que debe ocurrir en el sistema que se está modelando.

- **Diagrama de actividades.** Este representa un proceso paso a paso con un inicio y final claros. Es un conjunto de actividades que deben realizarse para lograr un objetivo. Muestra cómo cada actividad conduce a la siguiente y cómo todas estas se conectan. Además del desarrollo de software, estas se pueden utilizar en casi cualquier entorno empresarial. También se denominan asignación o modelado de proceso empresarial.
- **Diagrama de casos de uso.** Este describe lo que un sistema hace las cosas, pero no la forma en que las hace. Un caso de uso es un conjunto de eventos que ocurren cuando un “actor” usa un sistema para completar un proceso. Un actor se define como cualquier persona o cualquier cosa que interactúa con el sistema (persona, organización o aplicación) desde fuera del sistema. Por lo tanto, un diagrama de casos de uso describe visualmente ese conjunto de secuencias y representa los requisitos funcionales del sistema.
- **Diagrama de descripción general de interacción.** Este diagrama, a menudo complejo, es similar al diagrama de actividad, ya que ambos muestran una secuencia paso a paso de las actividades. Sin embargo, un diagrama de descripción general de interacción es un diagrama de actividad que se compone de diferentes diagramas de interacción. Usan la misma composición que un diagrama de actividad e incorpora elementos como la interacción, el uso de la interacción, restricción de tiempo y restricción de la duración.
- **Diagrama de tiempos.** Cuando el tiempo ocupa un lugar central, se usa este diagrama de UML. También conocido como un diagrama de secuencia o eventos, no muestra la forma en que los objetos interactúan o cambian entre sí. Funcionalmente, muestra cómo los objetos y actores se desempeñan en una línea de tiempo. El enfoque aquí está en la duración de los eventos y los cambios que se producen en función de las restricciones de duración.
- **Diagrama de máquina de estados.** También denominado gráfico de estados, este diagrama se aplica cuando el comportamiento de un objeto es complejo y el detalle es esencial. Ayuda a describir el comportamiento de un objeto (o a veces de un operador) y la forma en que cambia según los eventos internos y externos.
- **Diagrama de secuencia.** Popular más allá de la comunidad de diseño, este diagrama visualmente atractivo es bueno para mostrar todo tipo de procesos empresariales.

Simplemente revela la estructura de un sistema, mostrando la secuencia de mensajes e interacciones entre actores y objetos cronológicamente. Los diagramas de secuencia muestran iteraciones y ramificaciones simples. Es favorable al realizar múltiples tareas.

- **Diagrama de comunicación.** Un diagrama de comunicación o colaboración es similar a un diagrama de secuencia. Sin embargo, enfatiza la comunicación entre objetos. Muestra la organización de los objetos que participan en una interacción y presenta iteraciones y ramificaciones más complejas.

### **Enterprise Architect**

Herramienta que nos facilitará realizar diagramas UML con el fin de visualizar, especificar, construir y documentar el sistema.

Sparx Systems Enterprise Architect es una herramienta de diseño y modelado visual basada en OMG UML. La plataforma admite: el diseño y la construcción de sistemas de software; modelado de procesos comerciales; y modelado de dominios basados en la industria.

El modelado de sistemas mediante UML proporciona una base para modelar todos los aspectos de la arquitectura organizacional, junto con la capacidad de proporcionar una base para diseñar e implementar nuevos sistemas o cambiar los sistemas existentes. Los aspectos que puede cubrir este tipo de modelado van desde el diseño de arquitecturas organizacionales o de sistemas, la reingeniería de procesos comerciales, el análisis comercial y las arquitecturas orientadas a servicios y el modelado web, hasta el diseño y reinsertión de aplicaciones y bases de datos. - Ingeniería y desarrollo de sistemas embebidos. Junto con el modelado de sistemas, Enterprise Architect cubre los aspectos centrales del ciclo de vida del desarrollo de aplicaciones, desde la gestión de requisitos hasta las fases de diseño, construcción, prueba y mantenimiento, con soporte para la trazabilidad, la gestión de proyectos y el control de cambios de estos procesos, así como

instalaciones para el desarrollo basado en modelos del código de la aplicación utilizando una plataforma interna de desarrollo integrado.

La base de usuarios va desde programadores y analistas comerciales hasta arquitectos empresariales, en organizaciones que van desde pequeñas empresas de desarrollo, corporaciones multinacionales y organizaciones gubernamentales hasta organismos internacionales de estándares de la industria. Sparx Systems lanzó inicialmente Enterprise Architect en 2000. Originalmente diseñado como una herramienta de modelado UML para modelar UML 1.1, el producto ha evolucionado para incluir otras especificaciones OMG UML 1.3, 2.0, 2.1, 2.3, 2.4. 1 y 2.5.

### **PgAdmin**

Herramienta que nos servirá para la gestión de la base de datos de nuestro sistema.

PgAdmin 4 es una herramienta de gestión de código abierto para Postgres. PgAdmin está diseñado para monitorear y administrar múltiples servidores de bases de datos PostgreSQL y EDB Advanced Server, tanto locales como remotos, a través de una sola interfaz gráfica que permite la fácil creación y administración de objetos de bases de datos, así como una serie de otras herramientas para administrar sus bases de datos.

### **PostgreSQL**

PostgreSQL, también llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que

trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales.

Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

## **Pgpool**

Nos permitirá hacer peticiones desde del backend a la base de datos.

Pgpool-II es un middleware que funciona entre servidores PostgreSQL y un cliente de base de datos PostgreSQL. Se distribuye bajo una licencia similar a BSD y MIT. Proporciona las siguientes características.

### **Agrupación de conexiones**

Pgpool-II guarda las conexiones a los servidores PostgreSQL y las reutiliza cada vez que ingresa una nueva conexión con las mismas propiedades (es decir, nombre de usuario, base de datos, versión del protocolo). Reduce la sobrecarga de conexión y mejora el rendimiento general del sistema.

### **Replicación**

Pgpool-II puede administrar múltiples servidores PostgreSQL. El uso de la función de replicación permite crear una copia de seguridad en tiempo real en 2 o más discos físicos, de modo que el servicio pueda continuar sin detener los servidores en caso de falla del disco.

### **Balanceo de carga**

Si se replica una base de datos, ejecutar una consulta SELECT en cualquier servidor devolverá el mismo resultado. Pgpool-II aprovecha la función de replicación para reducir la carga en cada servidor PostgreSQL mediante la distribución de consultas SELECT entre varios servidores, lo que mejora el rendimiento general del sistema. En el mejor de los casos, el rendimiento mejora proporcionalmente a la cantidad de servidores PostgreSQL. El equilibrio de carga funciona mejor en una situación en la que muchos usuarios ejecutan muchas consultas al mismo tiempo.

### **Limitación de conexiones excedentes**

Hay un límite en el número máximo de conexiones simultáneas con PostgreSQL y las conexiones se rechazan después de esta cantidad de conexiones. Sin embargo, establecer el número máximo de conexiones aumenta el consumo de recursos y afecta el rendimiento del sistema. Pgpool-II también tiene un límite en la cantidad máxima de conexiones, pero las conexiones adicionales se pondrán en cola en lugar de devolver un error de inmediato.

### **Perro guardián**

Watchdog puede coordinar múltiples Pgpool-II, crear un sistema de clúster robusto y evitar el punto único de falla o el cerebro dividido. Watchdog puede realizar una verificación de vida contra otros nodos Pgpool-II, para detectar una falla de Pgpool-II. Si el Pgpool-II activo deja de funcionar, el Pgpool-II en espera puede promoverse a activo y tomar el control de la IP virtual.

### **Memoria caché de consultas**

Memoria caché de consultas permite guardar un par de sentencias SELECT y su resultado. Si ingresa un SELECT idéntico, Pgpool-II devuelve el valor del caché. Dado que no se requiere análisis de SQL ni acceso a PostgreSQL, el uso en la memoria caché es extremadamente rápido. Por otro lado, puede ser más lento que la ruta normal en algunos casos, porque agrega cierta sobrecarga de almacenamiento de datos de caché.

Pgpool-II llama al protocolo backend y frontend de PostgreSQL y transmite una conexión entre ellos. Por lo tanto, una aplicación de base de datos (frontend) piensa que Pgpool-II es el servidor PostgreSQL real, y el servidor (backend) ve a Pgpool-II como uno de sus clientes. Debido a que Pgpool-II es transparente tanto para el servidor como para el cliente, se puede usar una aplicación de base de datos existente con Pgpool-II casi sin cambiar sus fuentes.

## **NodeJs**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de C, EventMachine de Ruby, vibe.d de D y Java EE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS. Node.js incluye un entorno REPL para depuración interactiva.

## **Aspectos técnicos**

### **Concurrencia**

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto. Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback. Un inconveniente de este enfoque de único hilo de ejecución es que Node.js requiere de módulos adicionales como cluster para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta.

## V8

V8 es el entorno de ejecución para JavaScript creado para Google Chrome. Es software libre desde 2008, está escrito en C++ y compila el código fuente JavaScript en código de máquina en lugar de interpretarlo en tiempo real.

Node.js contiene libuv para manejar eventos asíncronos. Libuv es una capa de abstracción de funcionalidades de redes y sistemas de archivo en sistemas Windows y sistemas basados en POSIX como Linux, Mac OS X y Unix.

El cuerpo de operaciones de base de Node.js está escrito en JavaScript con métodos de soporte escritos en C++.

## Módulos

Node.js incorpora varios "módulos básicos" compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona y otros módulos fundamentales, como por ejemplo Path, FileSystem, Buffer, Timers y el de propósito más general Stream. Es posible utilizar módulos desarrollados por terceros, ya sea como archivos "node" precompilados, o como archivos en Javascript plano. Los módulos Javascript se implementan siguiendo la especificación CommonJS para módulos, utilizando una variable de exportación para dar a estos scripts acceso a funciones y variables implementadas por los módulos.

Los módulos de terceros pueden extender node.js o añadir un nivel de abstracción, implementando varias utilidades middleware para utilizar en aplicaciones web, como por ejemplo los frameworks connect y express. Pese a que los módulos pueden instalarse como archivos simples, normalmente se instalan utilizando el Node Package Manager que nos facilitará la

compilación, instalación y actualización de módulos, así como la gestión de las dependencias. Además, los módulos que no se instalen en el directorio por defecto de módulos de Node necesitarán la utilización de una ruta relativa para poder encontrarlos. El wiki Node.js proporciona una lista de varios de los módulos de terceros disponibles.

### **Desarrollo homogéneo entre cliente y servidor**

Node.js puede ser combinado con una base de datos documental (por ejemplo, MongoDB o CouchDB) y bases de datos relacionales como MySQL, PostgreSQL, entre otros, lo que permite desarrollar en un entorno de desarrollo JavaScript unificado. Con la adaptación de los patrones para desarrollo del lado del servidor tales como MVC y sus variantes MVP, MVVM, etc. Node.js facilita la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor.

### **Bucle de eventos**

Node.js se registra con el sistema operativo y cada vez que un cliente establece una conexión se ejecuta un callback. Dentro del entorno de ejecución de Node.js, cada conexión recibe una pequeña asignación de espacio de memoria dinámico, sin tener que crear un hilo de ejecución. A diferencia de otros servidores dirigidos por eventos, el bucle de gestión de eventos de Node.js no es llamado explícitamente, sino que se activa al final de cada ejecución de una función callback. El bucle de gestión de eventos se termina cuando ya no quedan eventos por atender.

### **Nodemon**

Nodemon es un programa de utilidad que puede monitorear cualquier cambio en su fuente de directorio y reiniciar automáticamente su servidor o aplicaciones. Desarrollado por REMY, esta herramienta de programación es genial para facilitar la recarga del servidor y no tener que hacerlo

manualmente nunca más. Sólo tienes que sustituir el código por defecto por el de esta herramienta para que el proceso se automatice.

Nodemon es una herramienta de monitorización para node.js y cualquier aplicación basada en él. Reinicia automáticamente la aplicación node cuando se detectan cambios en los archivos. Lo bueno de esta ligera herramienta es que en realidad no necesita ningún cambio adicional en tu código o incluso en la forma en que trabajas en tu código. Esto se debe a que es sólo un reemplazo del comando "node".

## **Express**

Express es un marco de aplicación web de Node.js mínimo y flexible que proporciona un conjunto sólido de funciones para aplicaciones web y móviles.

Con varios métodos de utilidad HTTP y middleware a su disposición, crear una API robusta que es rápido y fácil.

Express proporciona una capa delgada de características fundamentales de aplicaciones web, sin oscurecer las características de Node.js.

## **ReactJs**

ReactJs será el framework que nos facilitará el desarrollo del frontend del sistema.

React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. React sólo maneja la

interfaz de usuario en una aplicación; React es la Vista en un contexto en el que se use el patrón MVC (Modelo-Vista-Controlador) o MVVM (Modelo-vista-modelo de vista). También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (que no forman parte de la interfaz de usuario) de una aplicación web.

Según el servicio de análisis JavaScript, Libscore, React actualmente está siendo utilizado en las páginas principales de Imgur, Bleacher Informe, Feedly, Airbnb, SeatGeek, HelloSign, entre otras.

## **HTML**

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web.

Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).

Hipertexto hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la World Wide Web.

HTML utiliza marcas para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Las marcas HTML incluyen "elementos" especiales como <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output> (en-US), <progress>, <video>, <ul>, <ol>, <li> y muchos otros.

Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla. Por ejemplo, la etiqueta <title> se puede escribir como <Title>, <TITLE> o de cualquier otra forma.

## CSS

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. 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. 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.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en

voz (mediante un navegador de voz o un lector de pantalla), y dispositivos táctiles basados en el sistema Braille. También se puede mostrar una página web de manera diferente dependiendo del tamaño de la pantalla o tipo de dispositivo. Los lectores pueden especificar una hoja de estilos diferente, como una hoja de estilos CSS guardado en su computadora, para sobrescribir la hoja de estilos del diseñador.

La especificación CSS describe un esquema prioritario para determinar qué reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas son aplicadas con un sistema llamado de cascada, de modo que las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C). El MIME type `text/css` está registrado para su uso por CSS descrito en el RFC 2318. El W3C proporciona una herramienta de validación de CSS gratuita para los documentos CSS.

## **Material UI**

Es un framework que dispone de componentes que nos permitirá el construir la interfaz de usuario.

Material UI es una biblioteca de componentes React de código abierto que implementa Material Design de Google.

Material UI y MUI Base cuentan con muchos de los mismos componentes de UI, pero MUI Base viene sin estilos predeterminados o soluciones de estilo.

Material UI es completo porque viene empaquetado con estilos predeterminados y está optimizado para trabajar con Emotion (o componentes con estilo).

MUI Base, por el contrario, podría considerarse la contraparte "esquelética" o "sin cabeza" de Material UI; de hecho, las versiones futuras de Material UI utilizarán componentes y ganchos de MUI Base para su estructura fundamental.

Incluye una colección integral de componentes preconstruidos que están listos para usar en producción desde el primer momento.

## **Google Maps Plataform**

La API de Maps JavaScript te permite personalizar los mapas con tus propias imágenes y contenido para mostrarlos en páginas web y dispositivos móviles. La API de Maps JavaScript cuenta con cuatro tipos de mapas básicos (mapa de rutas, satélite, híbrido y terreno), los cuales puedes modificar mediante capas y diseños, controles y eventos, y varios servicios y bibliotecas.

### **Tipos de Mapas**

La API usa un objeto MapType para contener información sobre estos mapas. Un MapType es una interfaz que define la visualización y el uso de los mosaicos de mapas y la traducción de los sistemas de coordenadas, desde coordenadas en pantalla hasta coordenadas mundiales (en el mapa). Cada MapType debe contener algunos métodos para controlar la recuperación y liberación de mosaicos, y las propiedades que definen su comportamiento visual.

El funcionamiento interno de los tipos de mapas en la API de Maps JavaScript es un tema avanzado. La mayoría de los desarrolladores pueden usar los tipos de mapas básicos que se indican a continuación. Sin embargo, también puedes modificar la presentación de los tipos de mapas existentes mediante los mapas con diseños o definir tus propios mosaicos mediante los tipos de mapas personalizados. Cuando proporciones tipos de mapas personalizados, deberás comprender cómo modificar el registro de tipos de mapas.

## **Tipos de mapas básicos**

Hay cuatro tipos de mapas disponibles en la API de Maps JavaScript. Además de los mosaicos de mapas de rutas "pintados", la API de Maps JavaScript también admite otros tipos de mapas.

Los siguientes tipos de mapas están disponibles en la API de Maps JavaScript:

- `roadmap` muestra la vista del mapa de ruta predeterminado. Este es el tipo de mapa predeterminado.
- `satellite` muestra imágenes satelitales de Google Earth.
- `hybrid` muestra una combinación de vistas normales y satelitales.
- `terrain` muestra un mapa físico basado en la información del terreno.

## **Coordenadas de mapas y mosaicos**

La API de Maps JavaScript usa los siguientes sistemas de coordenadas:

- Valores de latitud y longitud que hacen referencia a un punto del planeta de manera inequívoca. (Google utiliza el estándar World Geodetic System WGS84).
- Coordenadas mundiales que hacen referencia a un punto del mapa de manera inequívoca.
- Coordenadas de píxeles, las cuales hacen referencia a píxeles específicos en el mapa con un nivel de zoom específico.
- Coordenadas del mosaico, las cuales hacen referencia a un mosaico específico en el mapa con un nivel de zoom específico.

## **Coordenadas mundiales**

Cada vez que la API necesite traducir una ubicación del mundo en una ubicación en un mapa, primero traducirá los valores de latitud y longitud a una coordenada mundial. La API usa la proyección de Mercator para realizar esta traducción.

Para aportar practicidad al cálculo de coordenadas de píxeles (consulta la imagen de abajo), se aplica la suposición de que un mapa con nivel 0 de zoom es un mosaico de tamaño básico. Luego se definen coordenadas mundiales relacionadas con las coordenadas de píxeles en el nivel 0 de

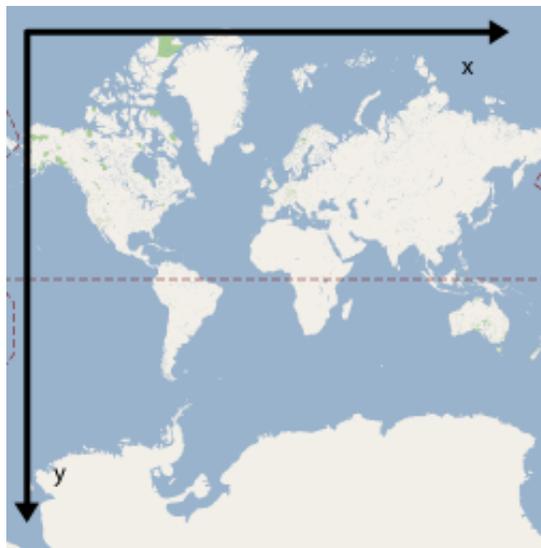
zoom y se usa la proyección para convertir latitudes y longitudes en posiciones de píxeles en este mosaico básico. Estas coordenadas mundiales son valores de puntos flotantes medidos desde el origen de la proyección del mapa hasta la ubicación específica. Ten en cuenta que este es un valor de punto flotante. Puede tener una precisión mucho mayor que la resolución actual de la imagen del mapa que se muestra. En otras palabras, las coordenadas mundiales son independientes del nivel de zoom actual.

Las coordenadas mundiales de Google Maps se miden desde el origen de la proyección de Mercator (la esquina noroeste del mapa con una longitud de 180 grados y una latitud de 85 grados, aproximadamente) y aumentan en la dirección x hacia el este (derecha) y en la dirección y hacia el sur (abajo). Como el mosaico básico de Google Maps de Mercator es de 256 x 256 píxeles, el espacio de coordenadas mundiales que se puede usar es {0-256}, {0-256}.

Ten en cuenta que las proyecciones de Mercator tienen un ancho limitado en sentido longitudinal, pero un alto ilimitado en sentido latitudinal. Las imágenes del Mapa base se “recortan” usando la proyección de Mercator a aproximadamente +/- 85° para que la forma del mapa resultante sea cuadrada, lo cual facilita la lógica de selección de mosaicos. Ten en cuenta que una proyección puede generar coordenadas mundiales fuera del espacio de coordenadas utilizable del Mapa base si la representación se realiza muy cerca de los polos, por ejemplo.

## Figura 4

*Mapa geográfico base de Google Maps*



### Coordenadas de píxeles

Las coordenadas de píxeles hacen referencia a píxeles específicos del mapa con un nivel de zoom específico, mientras que las coordenadas mundiales reflejan ubicaciones absolutas en una proyección determinada. Estas coordenadas de píxeles se calculan usando la siguiente fórmula:

$$\text{pixelCoordinate} = \text{worldCoordinate} * 2^{\text{zoomLevel}}$$

A partir de la ecuación anterior, ten en cuenta que cada nivel de zoom que aumenta es el doble tanto en la dirección x como y. Por lo tanto, cada nivel de zoom superior da como resultado una resolución cuatro veces mayor que el nivel anterior. Por ejemplo, en el nivel de zoom 1, el mapa consta de 4 mosaicos de 256 x 256 píxeles. Esto da como resultado un área de 512 x 512 píxeles. En el nivel de zoom 19, se puede hacer referencia a cada píxel de x y de y en el mapa mediante un valor entre 0 y  $256 * 2^{19}$ .

Debido a que las coordenadas mundiales se basan en el tamaño de los mosaicos del mapa, el efecto del valor entero de las coordenadas de un píxel es identificar el píxel exacto en dicha

ubicación con el nivel de zoom actual. Ten en cuenta que en el nivel de zoom 0, las coordenadas de píxeles son iguales a las coordenadas mundiales.

Ahora tenemos una manera de indicar con precisión cada ubicación en el mapa, en cada nivel de zoom. La API de Maps JavaScript construye un viewport en función del centro del nivel de zoom del mapa (como un objeto `LatLng`) y el tamaño del elemento del DOM que lo contiene, y convierte este cuadro delimitador en coordenadas de píxeles. Luego, la API determina de manera lógica todos los mosaicos del mapa que caben dentro de los límites de píxeles determinados. Se hace referencia a cada uno de estos mosaicos del mapa mediante las coordenadas del mosaico, lo cual simplifica enormemente la visualización de imágenes de los mapas.

### **Coordenadas del mosaico**

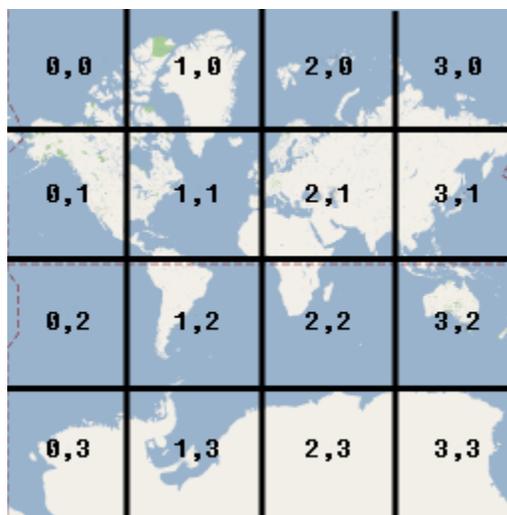
La API no puede cargar todas las imágenes de mapa a la vez para los niveles de zoom más altos. Lo que hace, en cambio, es dividir las imágenes de cada nivel de zoom en un conjunto de mosaicos del mapa, organizados de manera lógica en un orden que la aplicación comprenda. Cuando en un mapa se realiza un desplazamiento hacia una nueva ubicación, o un cambio en el nivel de zoom, la API determina los mosaicos necesarios mediante coordenadas de píxeles y convierte esos valores en un conjunto de mosaicos que se deben recuperar. Estas coordenadas de mosaicos se asignan usando un esquema que permite determinar fácilmente, desde la perspectiva lógica, el mosaico que contiene las imágenes de un punto determinado.

La numeración de los mosaicos de Google Maps se origina en el mismo punto que la de los píxeles. En la implementación de la proyección de Mercator en Google, el mosaico de origen siempre se encuentra en la esquina noroeste del mapa. Los valores de `x` aumentan de oeste a este y los de `y` aumentan de norte a sur. Los mosaicos se indexan utilizando las coordenadas `x,y` a

partir de ese origen. Por ejemplo, en el nivel de zoom 2, cuando la Tierra se divide en 16 mosaicos, se puede hacer referencia a cada uno de estos mosaicos mediante un par x,y único:

### Figura 5

*Ejemplo coordenadas de mosaico*



### Firestore

Firestore es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014.

Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización. (Firestore, 2022)

Los desarrolladores tendrán una serie de ventajas al usar esta plataforma:

Sincronizar fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.

Usa un conjunto de herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como IOS, Android, aplicaciones web, Unity y C++.

Usa la infraestructura de Google y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.

Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

Firebase dota a sus usuarios de una gran documentación para crear aplicaciones usando esta plataforma. Aparte de esto, ofrece soporte gratuito mediante correo electrónico para todos sus usuarios, y además sus desarrolladores participan activamente en plataformas como Github y StackOverflow, así como poseen un canal de Youtube explicando el funcionamiento de varias de sus herramientas.

## **Servicios**

### **Analytics**

#### **Firestore Analytics**

Firestore Analytics es una aplicación gratuita que proporciona una visión profunda sobre el uso de la aplicación por parte de los usuarios.

### **Desarrollo**

Firestore permite la creación de mejores aplicaciones, minimizando el tiempo de optimización y desarrollo mediante diferentes funciones, entre las que destacan la detección de errores y el testeado, lo cual supone poder dar un salto de calidad a la aplicación. Poder almacenar toda la

información en la nube y configurarla de manera distribuida, son las características más destacadas de Firebase.

### **Firestore Cloud Messaging**

Antiguamente conocido como Google Cloud Messaging (GCM), Firestore Cloud Messaging (FCM) es una plataforma para mensajes y notificaciones para Android, iOS, y aplicaciones web que actualmente puede ser usada de forma gratuita.

### **Firestore Auth**

Firestore Auth es un servicio que puede autenticar los usuarios utilizando únicamente código del lado del cliente. Incluye la autenticación mediante proveedores de inicio de sesión como Facebook, GitHub, Twitter, Google, Yahoo y Microsoft; así como los métodos clásicos de inicio de sesión mediante correo electrónico y contraseña. Además, incluye un sistema de administración del usuario por el cual los desarrolladores pueden habilitar la autenticación de usuarios con correo electrónico y contraseña que se almacenarán en Firestore.

Este servicio busca facilitar la creación de sistemas de autenticación, a la vez que mejora la incorporación, acceso y seguridad para los usuarios. Gracias a esto, el cliente no tiene que preocuparse por desarrollar métodos de autenticación clásicos, ya que Firestore le aporta de manera sencilla, eficaz y segura métodos para gestionar sus usuarios.

También aporta muchas funcionalidades extra, como la recuperación y verificación de cuentas, tanto por correo electrónico como por SMS, y cuotas de registro para los usuarios, todo esto gestionado mediante los servidores de la plataforma.

### **Realtime Database**

Firestore proporciona una base de datos en tiempo real, back-end y organizada en forma de árbol JSON. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firestore. La compañía habilita integración con aplicaciones Android, iOS, JavaScript, Java, Objective-C, Swift y Node.js. La base de datos es también accesible a través de una REST API e integración para varios sistemas de Javascript como AngularJS, React, Ember.js y Backbone.js. La REST API utiliza el protocolo SSE (del inglés Server-Sent Events), el cual es una API para crear conexiones de HTTP para recibir notificaciones push de un servidor.

La sincronización en tiempo real de esta base de datos permite que los usuarios accedan a la información de sus datos desde cualquier dispositivo en tiempo real, compartiendo una instancia de Realtime Database, y cada vez que un usuario realice una modificación en esta, se almacena dicha información en la nube y se notifica simultáneamente al resto de dispositivos.

Una funcionalidad interesante de esta base de datos, es que si un usuario realiza cambios y pierde a la vez su conexión a Internet, el SDK de la plataforma usa una caché local en el dispositivo donde guarda estos cambios; y una vez que vuelva a tener conexión, automáticamente se sincronizan los datos locales.

### **Firestore Storage**

Firestore Storage proporciona cargas y descargas seguras de archivos para aplicaciones Firestore, sin importar la calidad de la red. El desarrollador lo puede utilizar para almacenar imágenes, audio, vídeo, o cualquier otro contenido generado por el usuario. Firestore Storage se basa en el almacenamiento de Google Cloud Storage.

## **Firestore Cloud Firestore**

Cloud Firestore es un servicio de almacenamiento de datos derivado de Google Cloud Platform, adaptado a la plataforma de Firebase. Al igual que Realtime Database, es una base de datos NoSQL, aunque presenta diversas diferencias. Se organiza en forma de documentos agrupados en colecciones, y en ellos se pueden incluir tanto campos de diversos tipos (cadenas de texto, números, puntos geográficos, referencias a la propia base de datos, arrays, booleanos, marcas de tiempo, e incluso objetos propios) como otras subcolecciones. Entre sus limitaciones más destacadas encontramos la de no soportar las búsquedas de texto tipo "LIKE", eso es, buscar por subcadenas del texto almacenado, y la de no poder filtrar las búsquedas con condiciones que impliquen más de un campo, si no es por búsquedas por el texto exacto.

### **II.1.2.6 Entregables del proyecto**

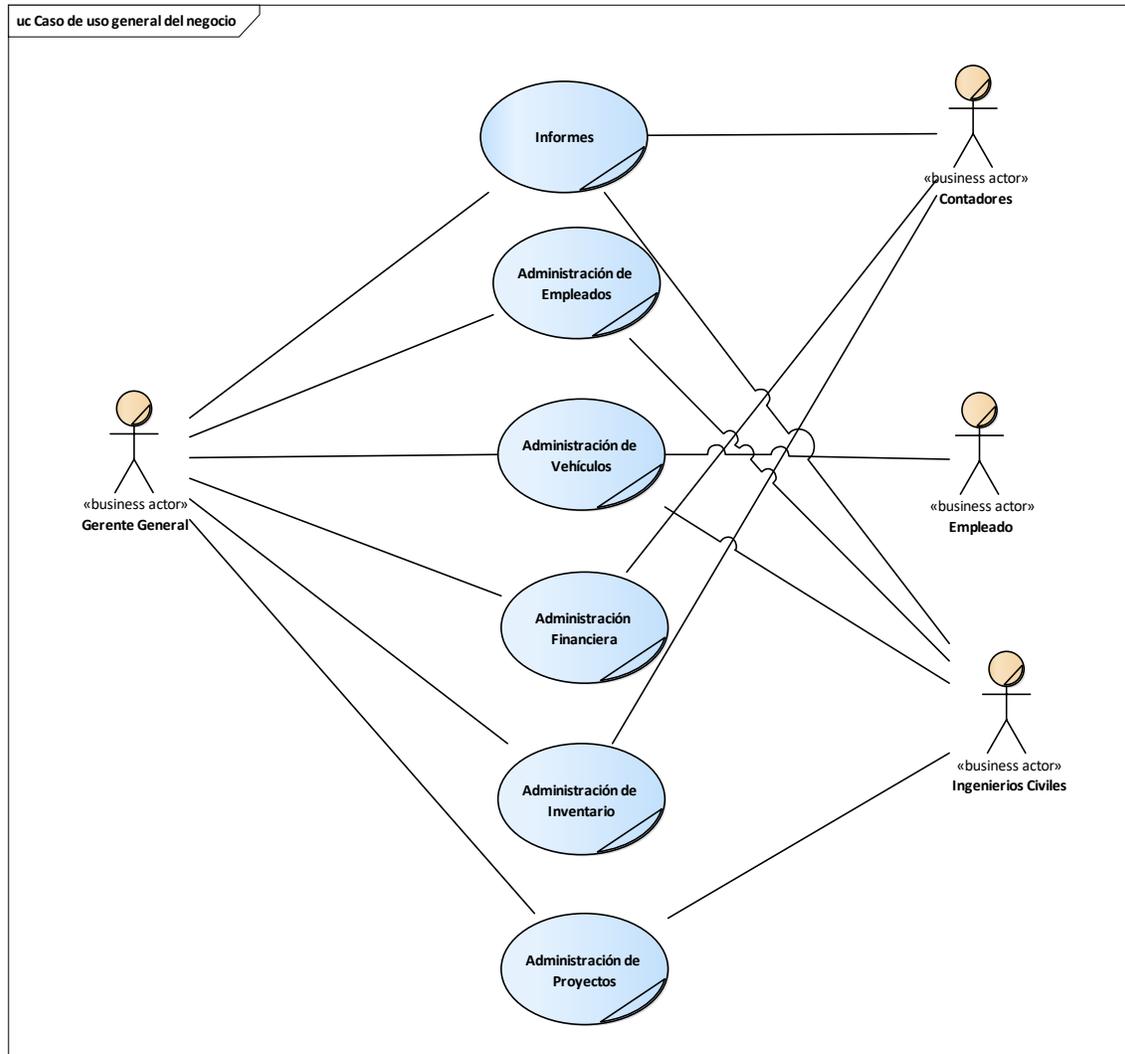
A continuación, se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

- Plan de desarrollo de software
- Modelo de casos de uso del negocio
- Modelo de casos de uso
- Especificación de los casos de uso
- Diagrama de actividades
- Diagrama de secuencia
- Modelo de datos
- Prototipo interfaces de usuario
- Manual de usuario

### **II.1.2.7 Modelo de casos de uso del negocio**

Los casos de uso de negocio nos permiten tener contexto de los procesos administrativos que realiza la empresa, en base a este caso de uso se podrán modelar los diseños para los casos de uso del sistema teniendo en cuenta de plasmar de la mejor manera el sistema y que esté bien adaptado para los usuarios, con el fin de no comprometer o alterar demasiado el modo de trabajo del personal de la empresa.

A continuación, se muestra el caso de uso general del negocio de la empresa Consultora y Constructora Bristlecone S.R.L.

**Figura 6***Caso de uso general del negocio***II.1.2.8 Modelo de casos de uso**

El modelo de casos de uso nos permite representar por medio de actores, relaciones y casos de uso el comportamiento que tendrá el sistema y las funcionalidades a las que podrán acceder los actores.

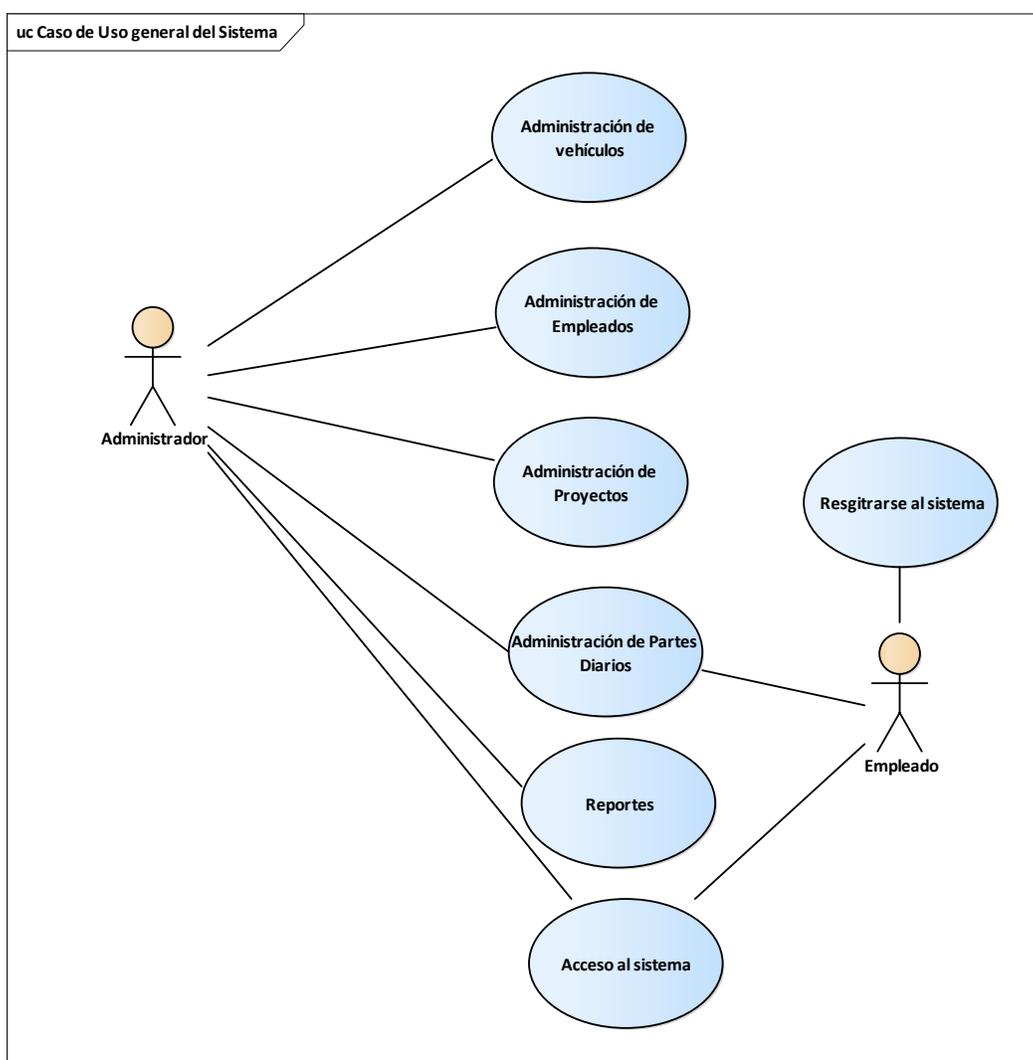
Teniendo como referencia los casos de uso del negocio, podemos representar los casos de uso que nos servirán para el desarrollo del sistema.

### II.1.2.8.1 Diagrama de casos de uso general

A continuación, se muestran el caso de uso general del sistema GEOBRIS.

#### Figura 7

*Caso de uso general del sistema*



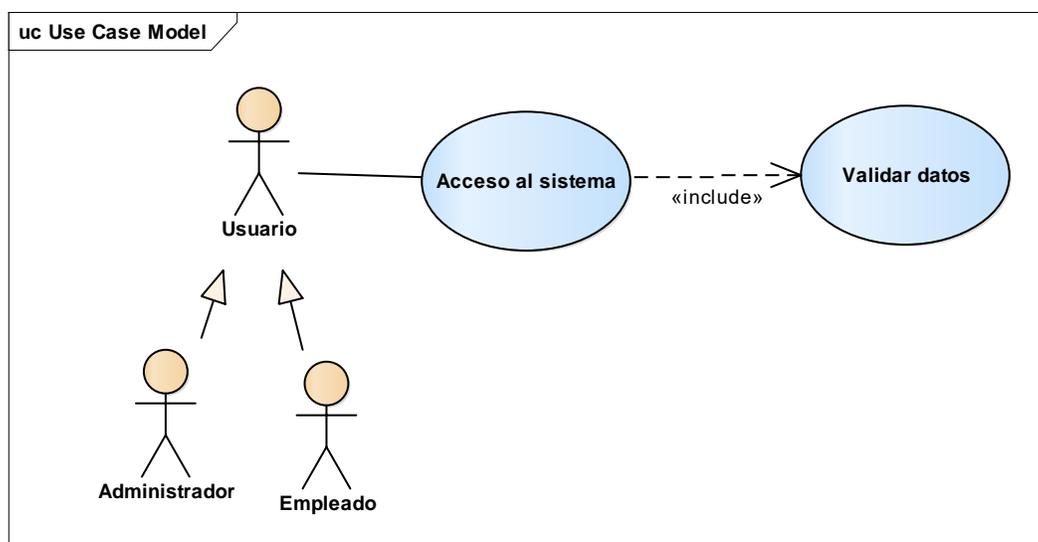
## II.1.2.9 Especificación de casos de uso

Con la especificación de casos de uso se realiza un desglose y una descripción detallada de cada caso de uso del sistema.

### II.1.2.9.1.1 Diagrama de caso de uso Acceso al sistema

**Figura 8**

*Diagrama de caso de uso Acceso al sistema*



**Tabla 8**

*Tabla descripción de caso de uso Acceso al sistema*

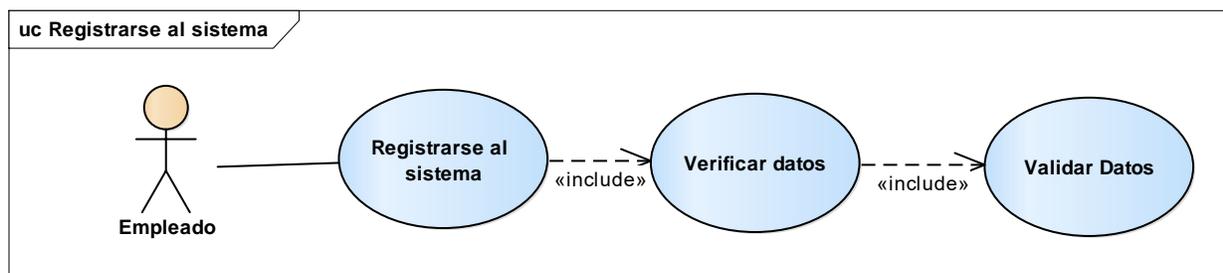
Caso de Uso #1	Acceso al sistema
<b>ACTOR</b>	Administrador, empleado
<b>PRECONDICIONES</b>	El actor deberá haber ingresado a la pantalla de Inicio de Sesión
<b>PROPÓSITO</b>	Limitar el acceso al sistema para que solo el personal de la empresa pueda acceder.

CURSO	Acción del Actor	Respuesta del Sistema
NORMAL	1.a El actor ingresa las credenciales requeridas	
	2 El actor presiona la opción Iniciar Sesión	3. El sistema verifica los datos ingresados si son correctos podrá acceder al sistema
		4 El sistema muestra diferentes funciones dependiendo el cargo que ocupe el actor
CURSOS ALTERNATIVOS		
	1.b El actor no ingresa las credenciales requeridas, el sistema muestra un mensaje de error	

### II.1.2.9.1.2 Diagrama de caso de uso Registrarse al sistema

**Figura 9**

*Diagrama de caso de uso Registrarse al sistema*



**Tabla 9**

*Tabla descripción de casos de uso Registrarse al sistema*

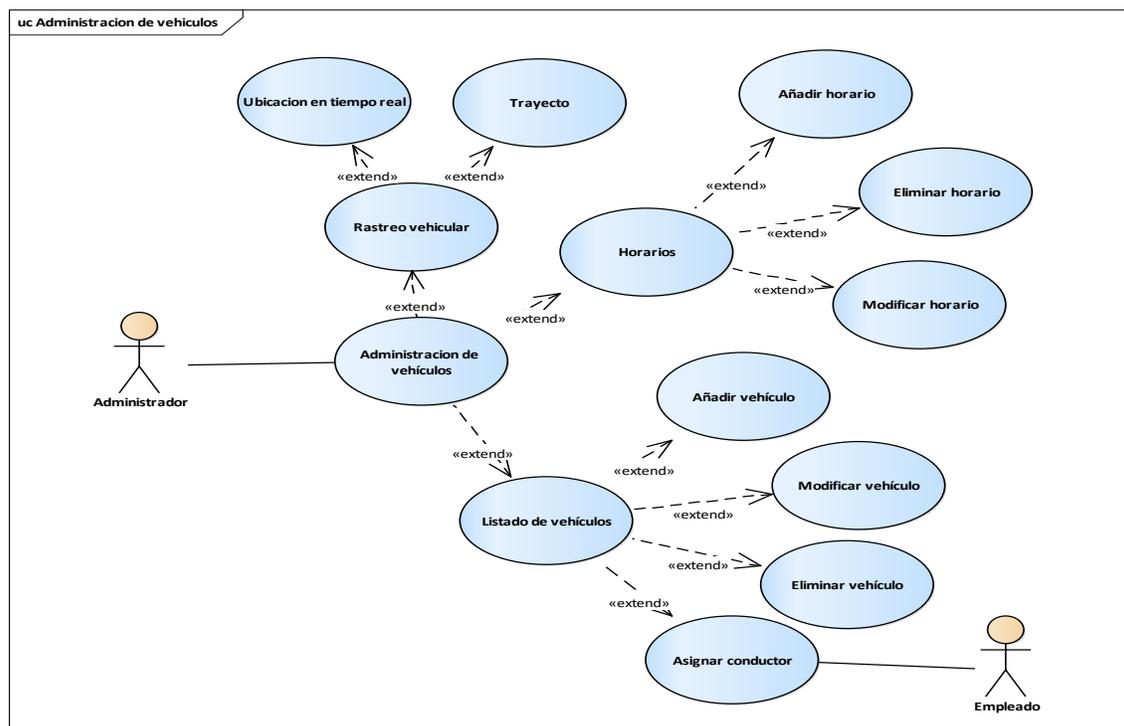
Caso de Uso #2	Registrarse al sistema
ACTOR	Empleado

<b>PRECONDICIONES</b>	El actor debe haber ingresado a la pantalla Registrarse	
<b>PROPÓSITO</b>	Permitir a los empleados de la empresa registrarse en el sistema	
<b>CURSO NORMAL</b>	<b>Accion del Actor</b>	<b>Respuesta del Sistema</b>
	1. El actor llena los datos del formulario requeridos	
	2 El actor selecciona la opción Registrarse	3. El sistema verifica y valida los datos. Si están correctos el actor es registrado y accede al sistema
<b>CURSOS ALTERNATIVOS</b>		
1.B. El actor no llena todos los datos requeridos, el sistema muestra un mensaje de error		

### II.1.2.9.1.3 Diagrama de caso de uso Administración de vehículos

**Figura 10**

*Diagrama de caso de uso Administración de vehículos*



**Tabla 10**

*Tabla descripción de casos de uso administración de vehículos*

<b>Caso de Uso #3</b>	<b>Administración de vehículos</b>	
<b>ACTOR</b>	Administrador, empleado	
<b>PRECONDICIONES</b>	El sistema debe haberse iniciado correctamente, y los dispositivos GPS deben estar funcionando adecuadamente. El actor debe haber ingresado al apartado de Administración de Vehículos	
<b>PROPÓSITO</b>	Permitir de cierta manera el control y administración de los vehículos pertenecientes a la empresa, mediante acciones de listado, adición, modificación y eliminación de registros. Dando paso al manejo de los datos técnicos y de geolocalización de cada vehículo	
<b>CURSO NORMAL</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>Listado Vehículos</b>	
	1. El actor selecciona la opción vehículos	2. El sistema despliega el listado de los vehículos y mostrará opciones de asignar conductor, adición, modificación y eliminación de datos de vehículos
	<b>Asignar Conductor</b>	
	1. El actor selecciona la opción Asignar conductor, para cualquier vehículo	2. El sistema despliega el listado de los conductores disponibles para ser asignados
	3 El actor selecciona cualquier conductor y selecciona la opción guardar	4. El sistema guarda los datos
		5. El sistema despliega y actualiza el listado de vehículos
<b>Añadir Vehículo</b>		

1. En el apartado de Listado de Vehículos el actor selecciona añadir vehículo	2. El sistema muestra un formulario para la adición de un nuevo registro de vehículo
3. El actor ingresa todos los datos permitidos por el sistema y selecciona la opción de guardar	4.AV.a. El sistema verifica que los datos ingresados sean válidos para la creación de un registro de vehículo.
	5. El sistema guarda los datos y se añade un nuevo vehículo
	6 El sistema despliega y actualiza el listado de vehículos
<b>Modificar Vehículo</b>	
1. El actor selecciona la opción modificar Vehículo, de un registro específico	2. El sistema despliega un formulario para la modificación del registro seleccionado
3. El actor ingresa los datos a modificarse y selecciona la opción de guardar	4MVa. El sistema verifica que los datos modificados sean válidos.
	5 El sistema guarda los datos del registro modificado
	6. El sistema despliega y actualiza el listado de vehículos
<b>Eliminar Vehículo</b>	
1. El actor selecciona la opción eliminar vehículo	2. El sistema muestra un mensaje de confirmación para proceder con la eliminación del registro
3.EV.a. El actor confirmar la eliminación	4.EV.a Se realiza la eliminación del registro
	5. El sistema despliega y actualiza el listado de vehículos
<b>Rastreo de Vehículo</b>	
1. El actor selecciona la opción de rastreo vehicular	2.El sistema muestra datos de los vehículos junto con opciones para ver el trayecto, y la ubicación del vehículo los vehículos
<b>Ubicación Geográfica</b>	

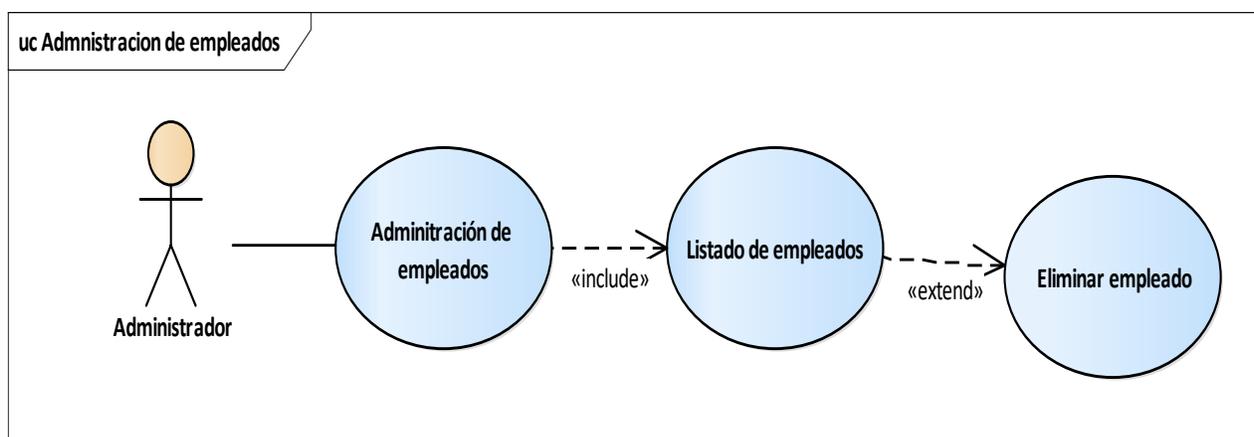
1. El actor selecciona la opción de Ubicación Geográfica	2. El sistema despliega una interfaz con un mapa geográfico con un marcador que señala la ubicación actual del vehículo
<b>Trayecto</b>	
1 El actor selecciona la opción de Trayecto	2 El sistema despliega una interfaz con un mapa con el trayecto recorrido por el vehículo
<b>Horarios</b>	
<b>Listar Horario</b>	
1. El actor selecciona la opción Horarios.	2. El sistema mostrará un listado de los horarios existentes. De igual manera se visualizarán opciones de adición, modificación y eliminación de datos.
<b>Añadir Horario</b>	
1. En el apartado de horario el actor selecciona añadir horario	2. El sistema muestra un formulario para la adición de un nuevo registro de un horario
3. El actor ingresa todos los datos permitidos por el sistema y selecciona la opción de guardar	4 AH.a. El sistema verifica que los datos ingresados sean válidos para la creación de un horario.
	5 El sistema guarda los datos ingresados por el actor
	6. El sistema despliega y actualiza el listado de horario
<b>Modificar Horario</b>	
1. En el apartado de horario el actor selecciona modificar horario	2. El sistema muestra un formulario para la modificación de un horario
3. El actor modifica todos los datos permitidos por el sistema y selecciona la opción de guardar	4 MH.a. El sistema verifica que los datos ingresados sean válidos para la modificación del horario
	5 El sistema guarda los datos modificados por el actor

		6. El sistema despliega y actualiza el listado de horario
<b>Eliminar Horario</b>		
	1. El actor selecciona la opción eliminar	2. El sistema muestra un mensaje de confirmación para proceder con la eliminación del horario
	3.EM.a. El actor confirmar la eliminación	4. Se realiza la eliminación del horario
		5. El sistema despliega y actualiza el listado de horario
<b>CURSOS ALTERNATIVOS</b>		
<b>Añadir Vehículo</b>		
4AV.b. Si los datos llegasen a ser inválidos el sistema no procede con el guardado de estos datos muestra un mensaje de error y vuelve al punto 2		
<b>Modificar Vehículo</b>		
4MV.b. Si los datos a modificarse son inválidos el sistema no procede con la acción de guardar, muestra un mensaje de error y vuelve al punto 2		
<b>Eliminar Vehículo</b>		
3EV.b. El actor no confirma la eliminación del registro y no se procede con la eliminación del registro procede con el punto 5		

#### II.1.2.9.1.4 Diagrama de caso de uso Administración de empleados

**Figura 11**

*Diagrama de caso de uso administración de empleados*



**Tabla 11**

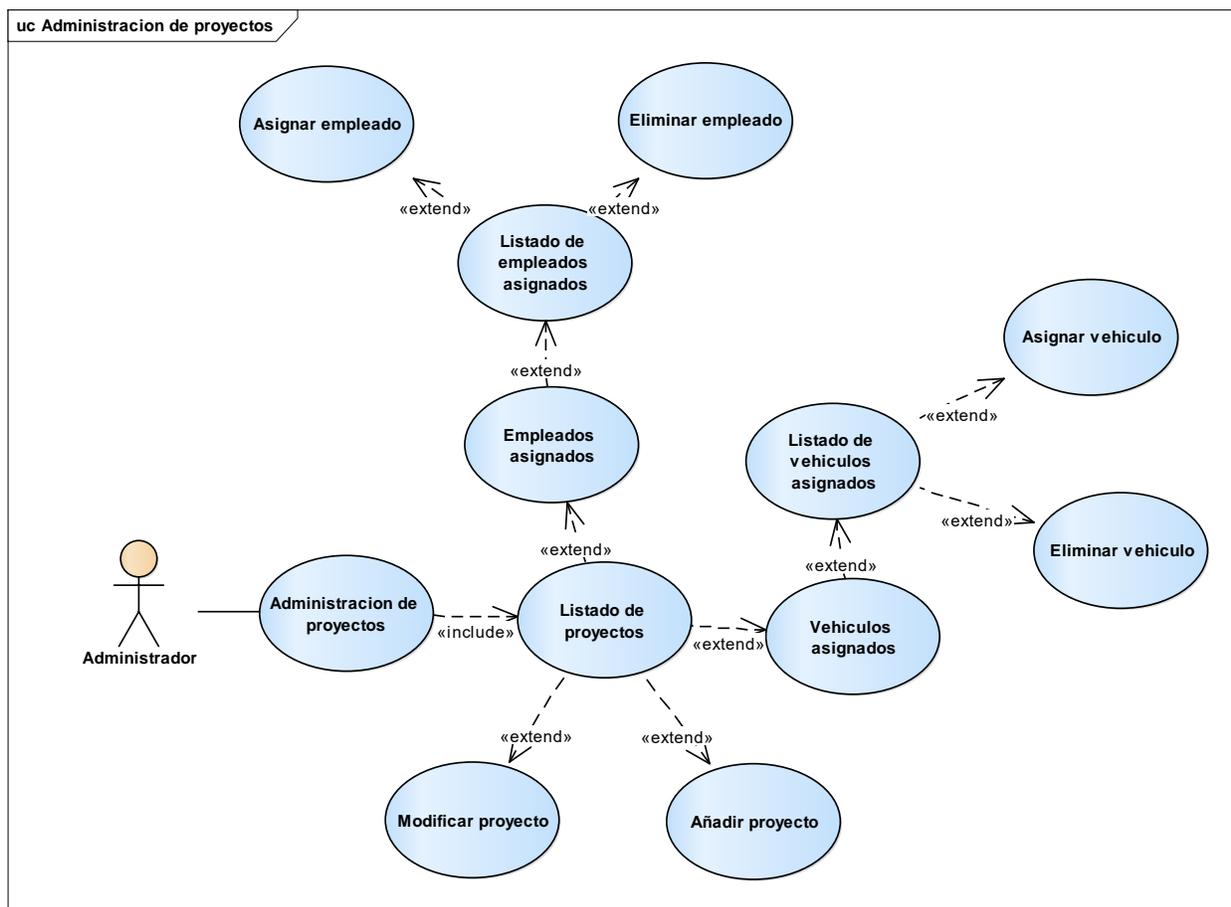
*Tabla descripción de caso de uso administración de empleados*

<b>Caso de Uso #4</b>	<b>Administración de Empleados</b>	
<b>ACTOR</b>	Administrador	
<b>PRECONDICIONES</b>	El actor debe encontrarse en el apartado de Administración de Empleados	
<b>PROPÓSITO</b>	Dar la facilidad al actor para administrar a los empleados de la empresa, permitiéndole visualizar un listado de los empleados	
<b>CURSO NORMAL</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>Listado de Empleados</b>	
	1. El actor ingresa al apartado de Administración de Empleados	2. El sistema despliega el listado de los empleados y mostrará opciones de adición, modificación y eliminación de registros de empleados
	<b>Eliminar Empleado</b>	
	1 El actor selecciona la opción eliminar empleado	2. El sistema muestra un mensaje de confirmación para proceder con la eliminación del registro
	3a. El actor confirmar la eliminación	4EE.a. Se realiza la eliminación del registro
	5. El sistema despliega y actualiza la lista de empleados	
<b>CURSOS ALTERNATIVOS</b>		
<b>Eliminar Empleado</b>		
3EE.b. El actor no confirma la eliminación del registro y no se procede con la eliminación del registro procede con el punto 5		

### II.1.2.9.1.5 Diagrama de caso de uso Administración de proyectos

**Figura 12**

*Diagrama de caso de uso administración de proyectos*



**Tabla 12**

*Tabla descripción de caso de uso administración de proyectos*

Caso de Uso #5	Administración de Proyectos
<b>ACTOR</b>	Administrador
<b>PRECONDICIONES</b>	El Actor debe iniciar sesión

<b>PROPÓSITO</b>	Administrar los proyectos, mediante las operaciones CRUD	
<b>CURSO NORMAL</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>Listado Proyectos</b>	
	1. El actor ingresa al apartado de Administración de Proyectos	2. El sistema despliega el listado de los Proyectos y mostrará opciones de adición, modificación, suspensión y finalización de proyectos
	<b>Añadir Proyecto</b>	
	1. El actor selecciona la opción de adición de registros	2. El sistema muestra un formulario para la adición de un nuevo registro de un proyecto
	3. El actor ingresa todos los datos permitidos por el sistema y selecciona la opción de guardar	4AP.a. El sistema verifica que los datos ingresados sean válidos para la creación de un registro
		5. El sistema guarda los datos ingresados por el actor
		6. El sistema despliega y actualiza la lista de proyectos
	<b>Modificar Proyecto</b>	
	1. El actor selecciona la opción modificar empleado, de un registro específico	2 El sistema despliega un formulario para la modificación del registro seleccionado
	3. El actor ingresa los datos a modificarse	4MP.a. El sistema verifica que los datos modificados sean válidos. Una vez validados son guardados
		5. El sistema guarda los datos ingresados por el actor
		6. El sistema despliega y actualiza la lista de proyectos
	<b>Empleados Asignados</b>	

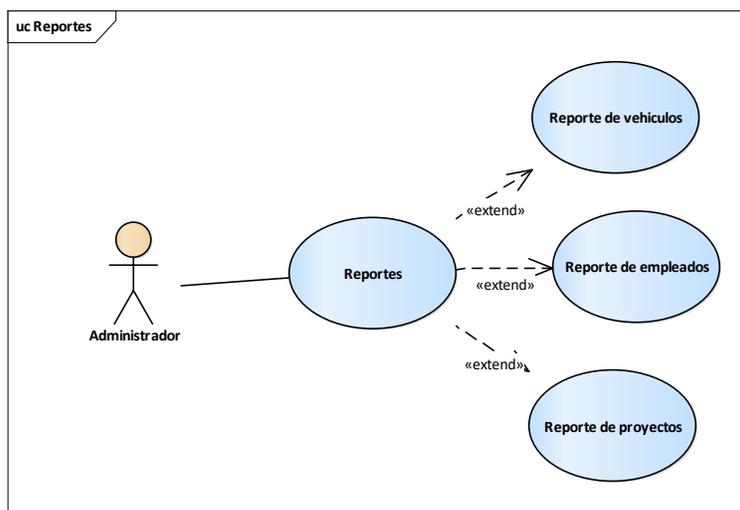
1 El actor selecciona la opción listado de empleados asignados	2. El sistema despliega el listado de los empleados designados al proyecto
<b>Asignar empleado</b>	
1 El actor selecciona la opción asignar empleado	2. El sistema despliega un formulario con los empleados disponibles para ser asignados
3 El actor selecciona el empleado a ser asignado y selecciona guardar	4. El sistema guarda los datos
	5 El sistema actualiza el listado de los empleados asignados
<b>Eliminar empleado</b>	
1 El actor selecciona la opción eliminar empleado	2. El sistema muestra un mensaje de confirmación para proceder con la eliminación del registro
3a. El actor confirmar la eliminación	4. Se realiza la eliminación del registro
	5. El sistema despliega y actualiza la lista de empleados asignados
<b>Vehículos Asignados</b>	
1 El actor selecciona la opción listado de empleados asignados	2. El sistema despliega el listado de los empleados designados al proyecto
<b>Asignar vehículo</b>	
1 El actor selecciona la opción asignar vehículo	2. El sistema despliega un formulario con los vehículos disponibles para ser asignados
3 El actor selecciona el vehículo a ser asignado y selecciona guardar	4. El sistema guarda los datos

		5 El sistema actualiza el listado de los vehículos asignados
<b>Eliminar vehículo</b>		
1 El actor selecciona la opción eliminar vehículo		2. El sistema muestra un mensaje de confirmación para proceder con la eliminación del registro
3a. El actor confirmar la eliminación		4. Se realiza la eliminación del registro
		5. El sistema despliega y actualiza la lista de vehículos asignados
<b>CURSOS ALTERNATIVOS</b>		
<b>Añadir Proyecto</b>		
4AP.b. Si los datos llegasen a ser inválidos el sistema no procede con el guardado de estos datos muestra un mensaje de error y vuelve al punto 2		
<b>Modificar Proyecto</b>		
4MP.b. Si los datos a modificarse son inválidos el sistema no procede con la acción de guardar, muestra un mensaje de error y vuelve al punto 2		

### II.1.2.9.1.6 Diagrama de caso de uso Reportes

#### Figura 13

*Diagrama de caso de uso reportes*



**Tabla 13**

*Tabla descripción de caso de uso reportes*

<b>Caso de Uso #6</b>	<b>Reportes</b>	
<b>ACTOR</b>	Usuario	
<b>PRECONDICIONES</b>	El Actor debe iniciar sesión	
<b>PROPÓSITO</b>	Brindar reportes de acuerdo a la información recopilada del sistema	
<b>CURSO NORMAL</b>	<b>Accion del Actor</b>	<b>Respuesta del Sistema</b>
	<b>Reportes</b>	
	1. El actor ingresa al apartado de reportes	2. El sistema despliega los reportes disponibles.
	<b>Reporte de vehículos</b>	
	1. El actor selecciona la opción de reporte de vehículos	2. El sistema muestra opciones de filtrado de información para generar reportes
	3. El actor selecciona los filtros que desea para generar reportes	4.V.a El sistema carga los datos solicitados y genera el reporte
		5. El sistema muestra el reporte y permite la descarga del documento
	<b>Reporte Empleados</b>	
	1. El actor selecciona la opción de reporte de empleados	2. El sistema muestra opciones de filtrado de información para generar reportes
	3. El actor selecciona los filtros que desea para generar reportes	4 E.a. El sistema carga los datos solicitados y genera el reporte

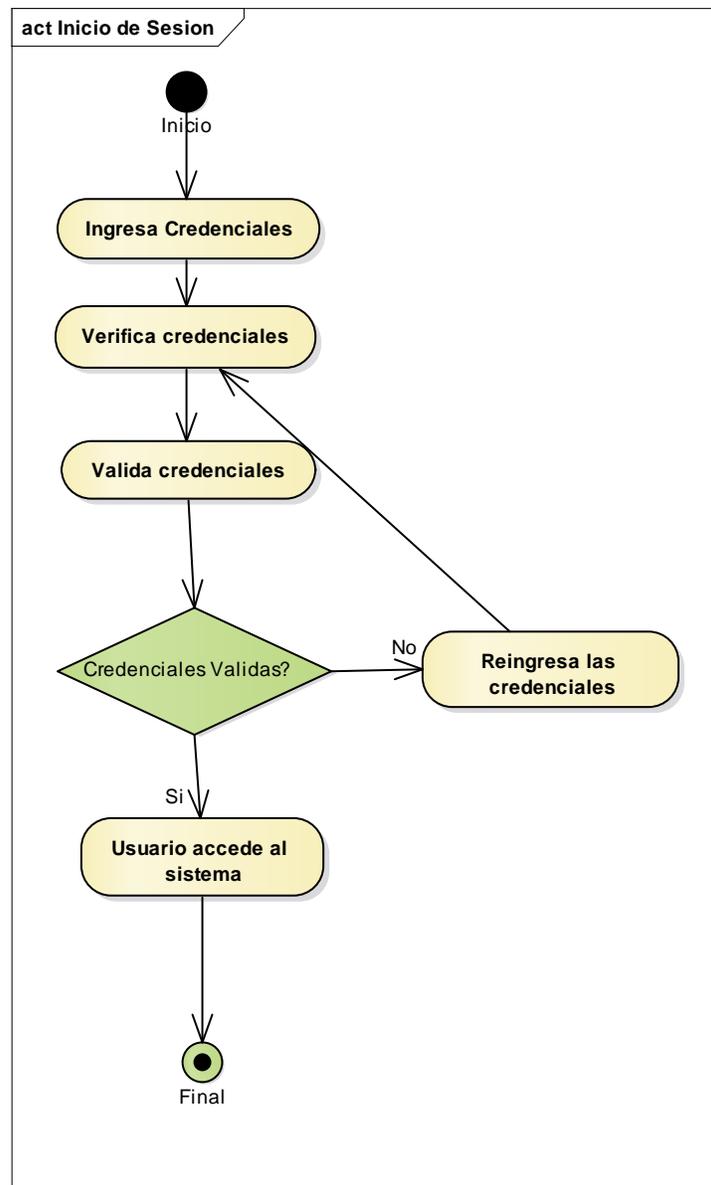
		5. El sistema muestra el reporte y permite la descarga del documento
<b>Reporte proyectos</b>		
	1. El actor selecciona la opción de reporte de proyectos	2. El sistema muestra opciones de filtrado de información para generar reportes
	3. El actor selecciona los filtros que desea para generar reportes	4 P.a. El sistema carga los datos solicitados y genera el reporte
		5. El sistema muestra el reporte y permite la descarga del documento
<b>CURSOS ALTERNATIVOS</b>		
<b>Reporte de vehículos</b>		
4V.b. Si existe algún error al cargar los datos o al generar el reporte, se mostrará un mensaje de error		
<b>Reporte de empleados</b>		
4E.b. Si existe algún error al cargar los datos o al generar el reporte, se mostrará un mensaje de error		
<b>Reporte de proyectos</b>		
4P.b. Si existe algún error al cargar los datos o al generar el reporte, se mostrará un mensaje de error		

## II.1.2.10 Diagramas de actividades

### II.1.2.10.1 Diagrama de actividades Acceso al sistema

Figura 14

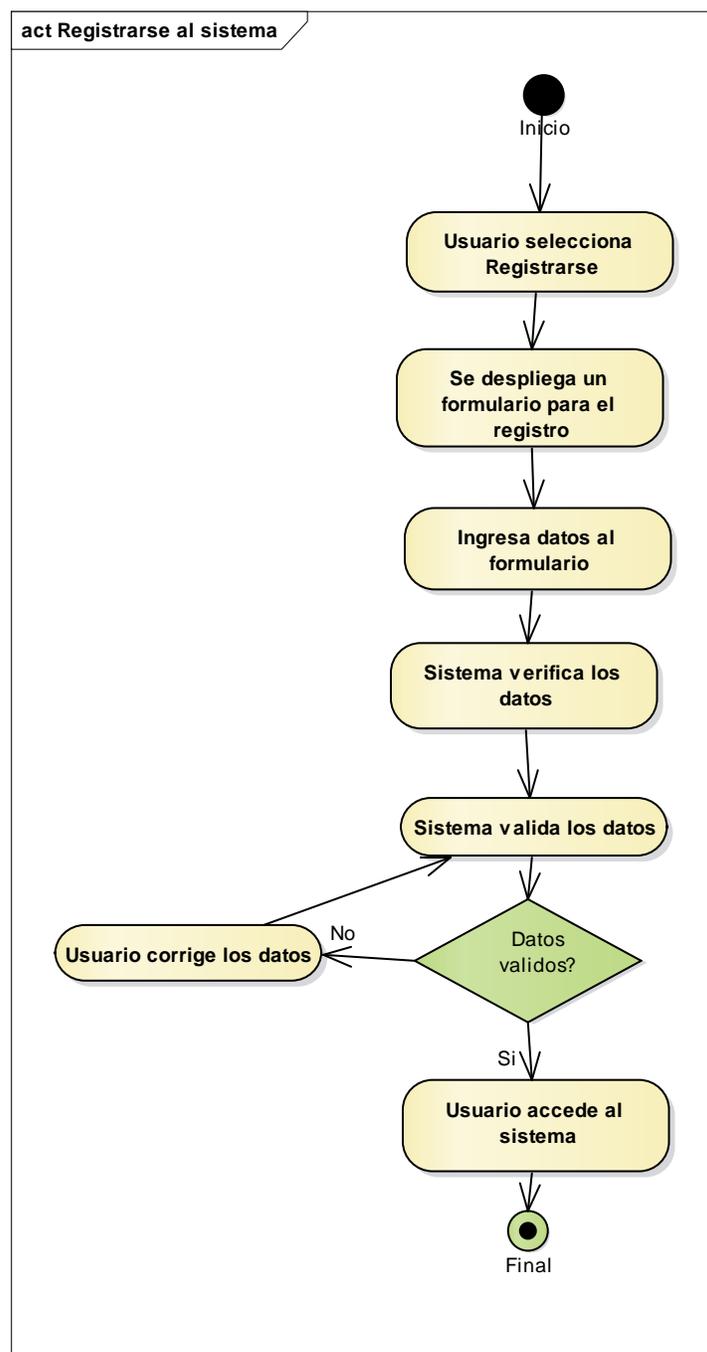
*Diagrama de actividades Acceso al sistema*



## II.1.2.10.2 Diagrama de actividades Registrarse al sistema

Figura 15

Diagrama de actividades Registrarse al sistema



### II.1.2.10.3 Diagrama de actividades Administración de vehículos

Figura 16

Diagrama de actividades Administración de vehículos

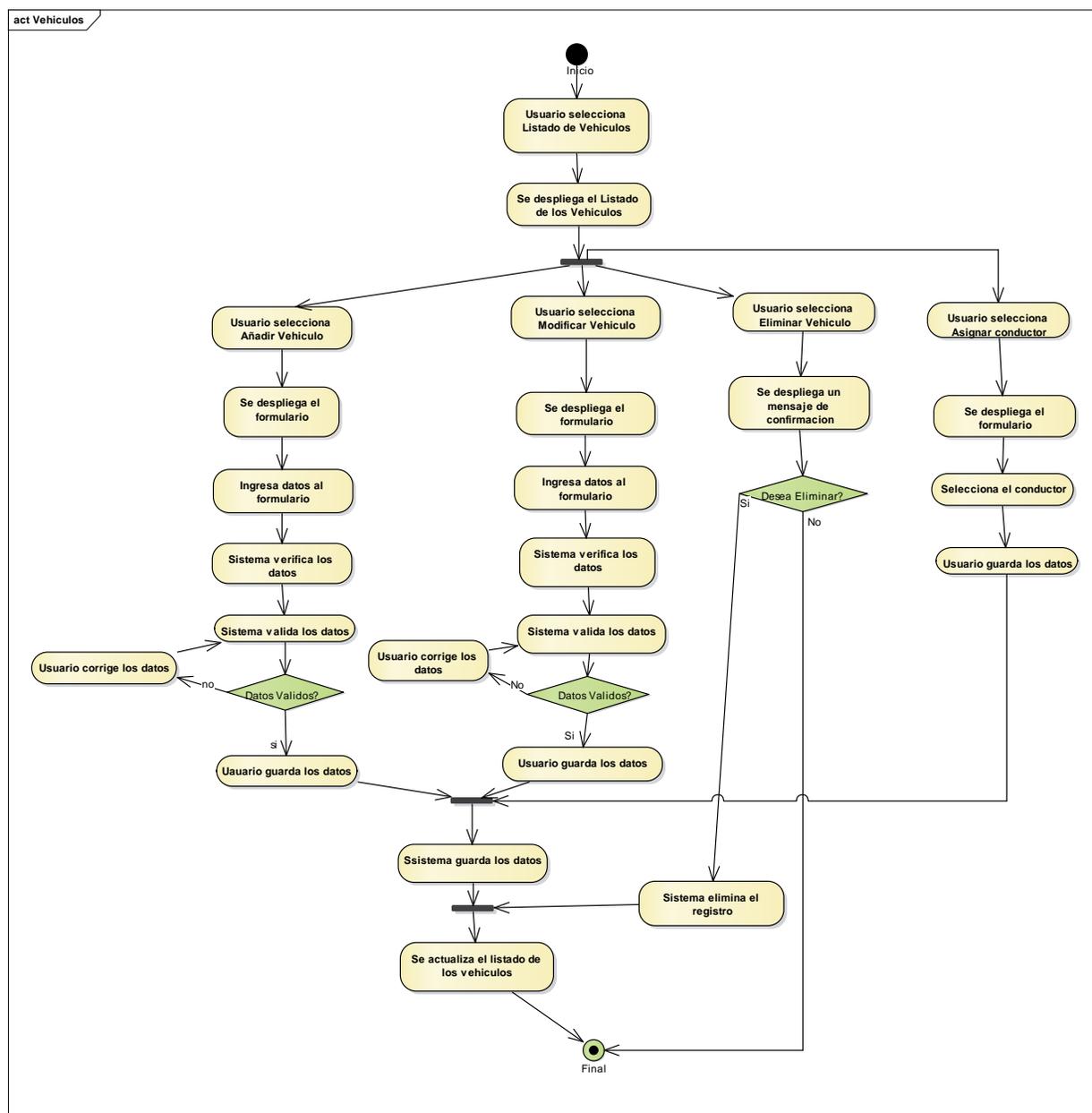
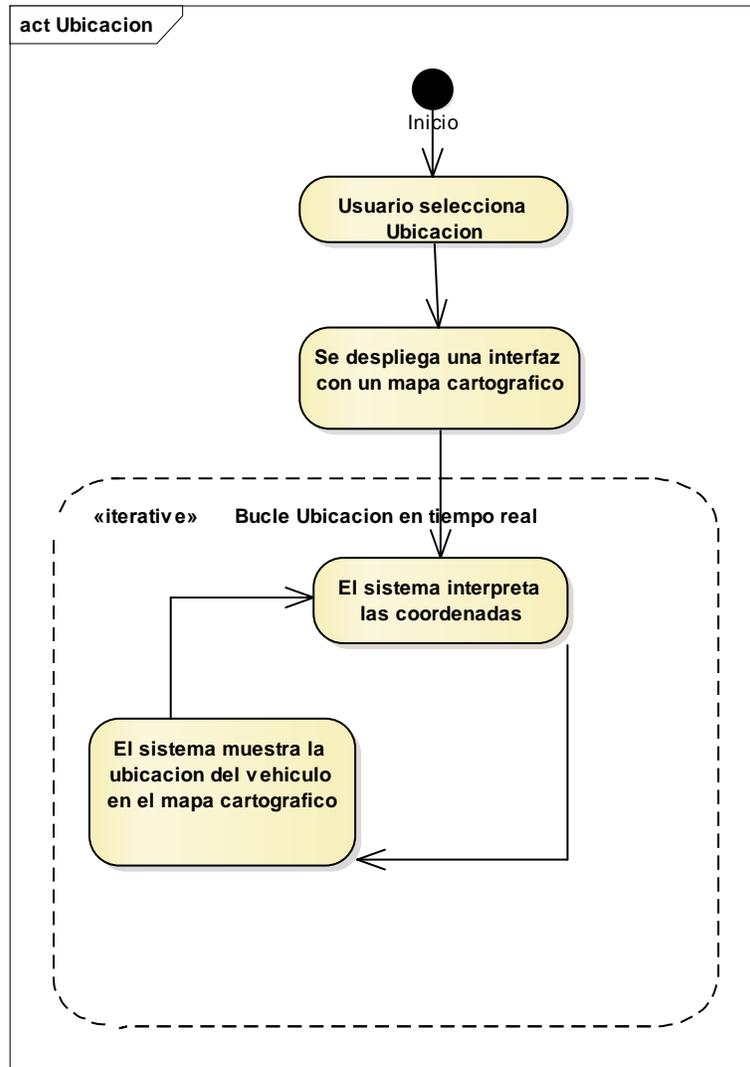


Figura 17

*Diagrama de actividades Ubicación*

**Figura 18**

*Diagrama de actividades Trayecto*

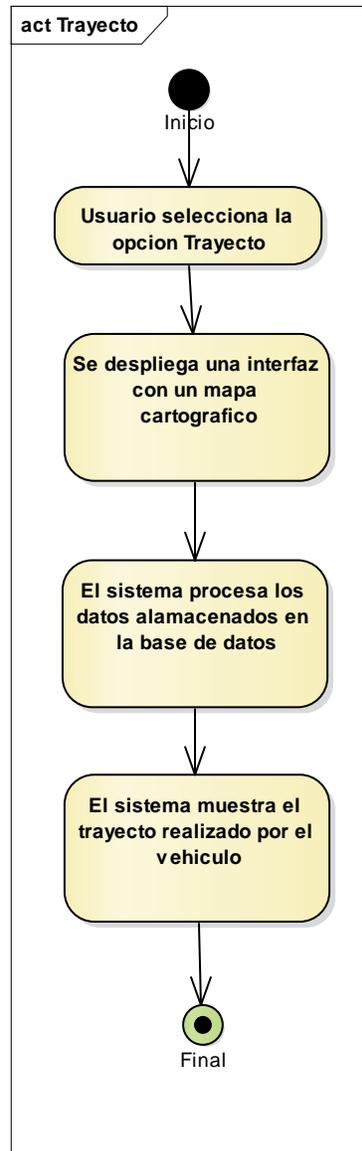
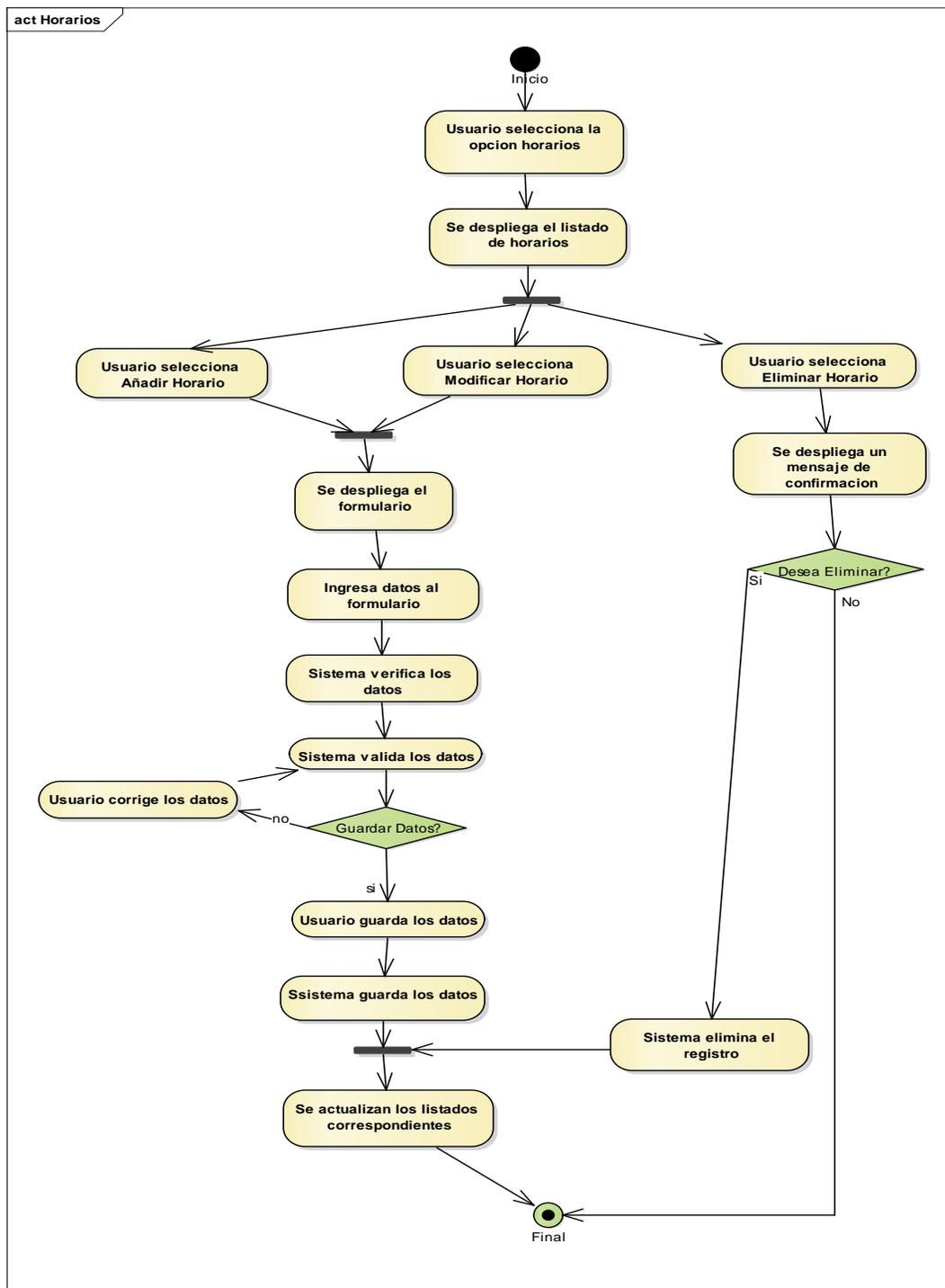


Figura 19

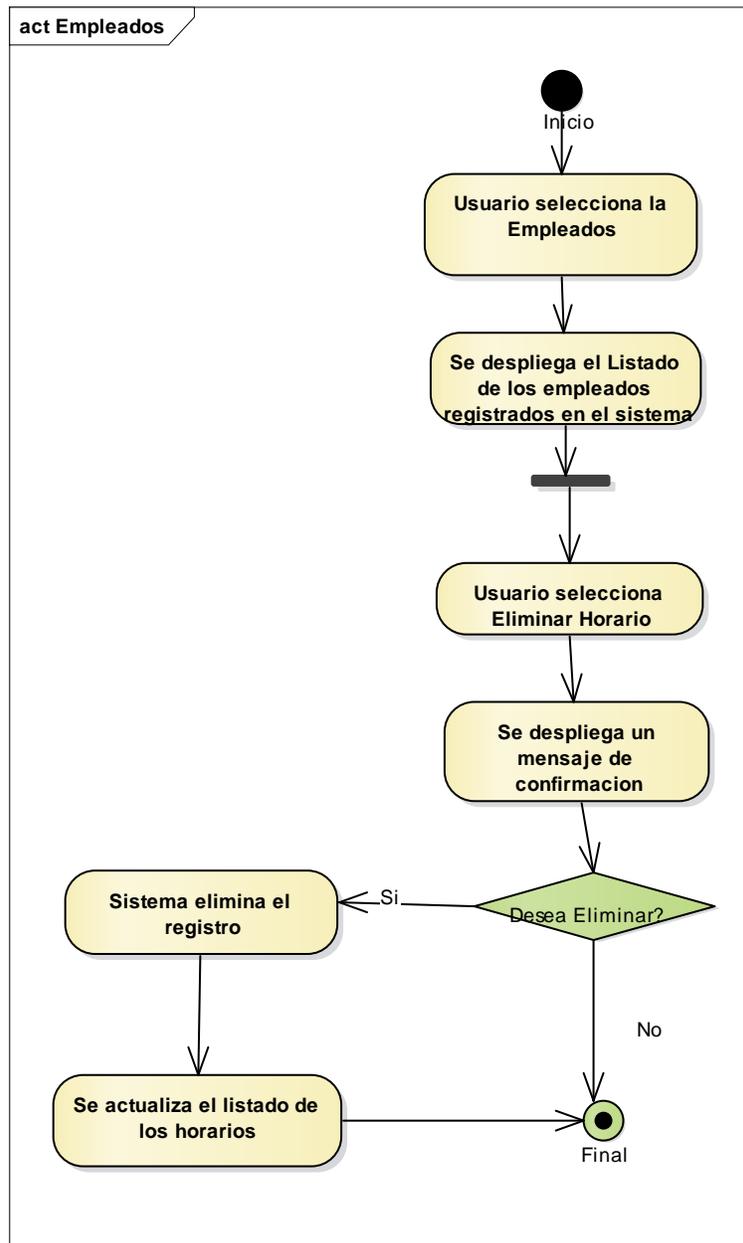
Diagrama de actividades Horario



#### II.1.2.10.4 Diagrama de actividades Administración de empleados

Figura 20

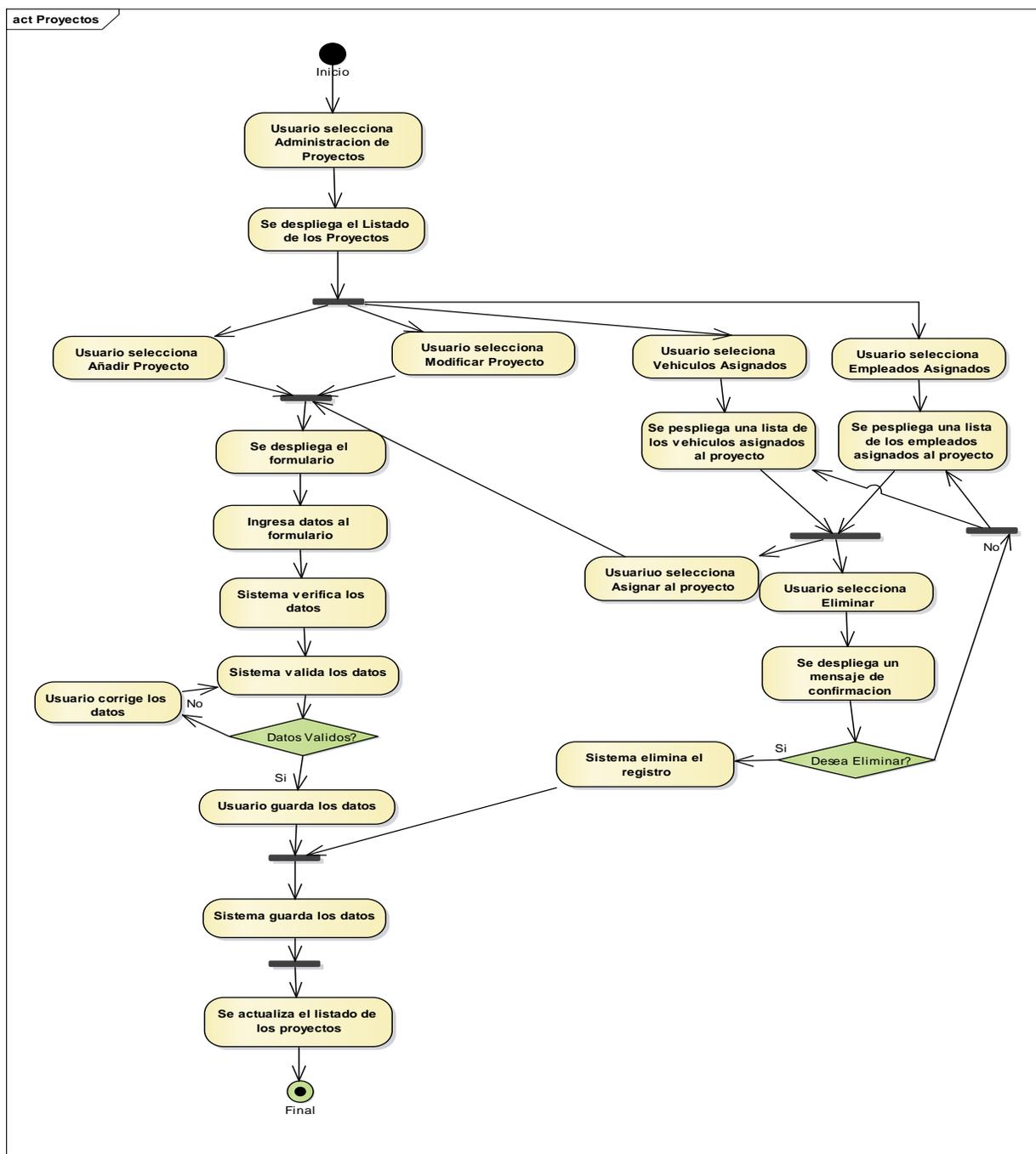
Diagrama de actividades Administración de empleados



## II.1.2.10.5 Diagrama de actividades Administración de proyectos

Figura 21

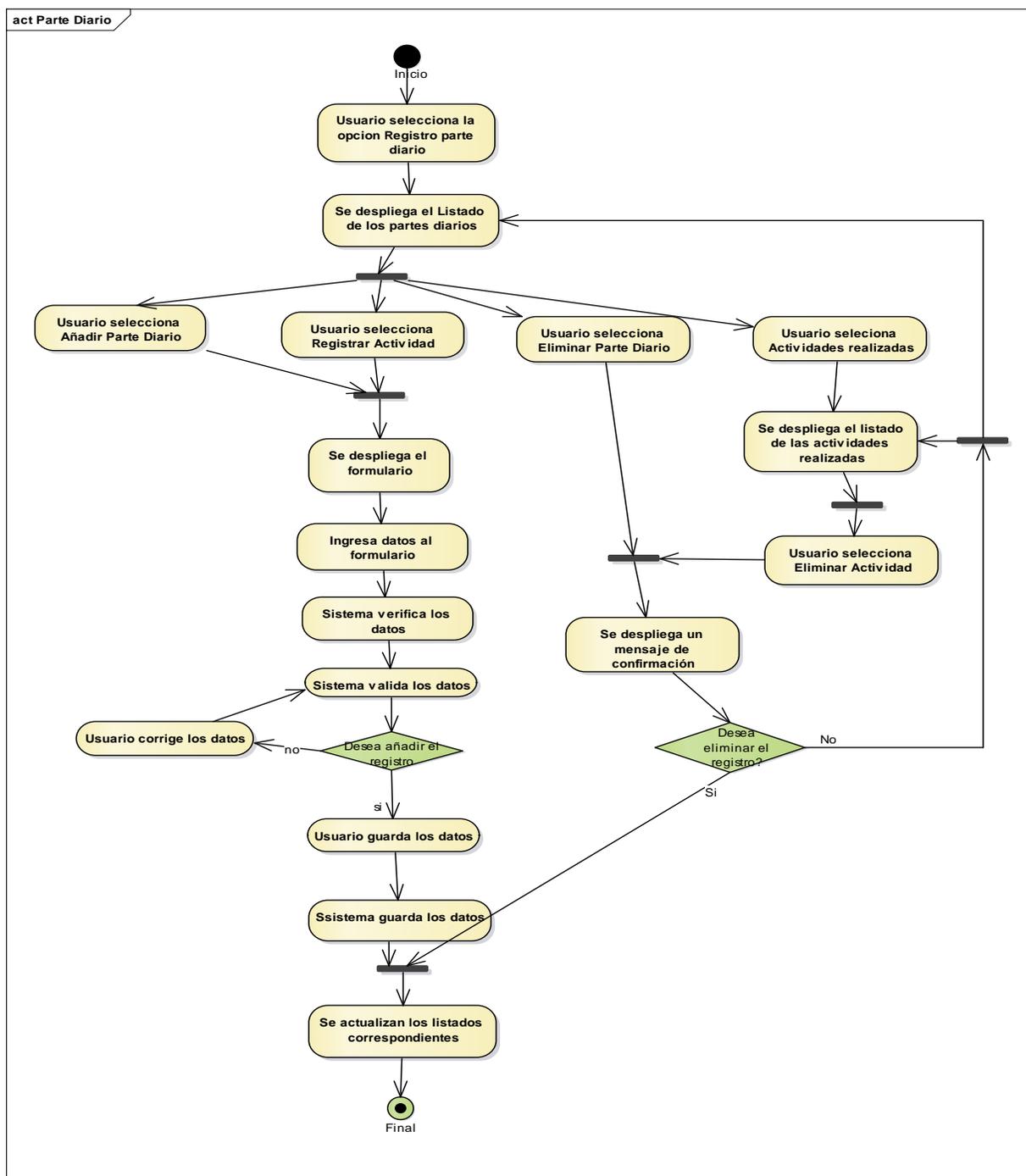
Diagrama de actividades Administración proyectos



## II.1.2.10.6 Diagrama de actividades Administración de parte diario

Figura 22

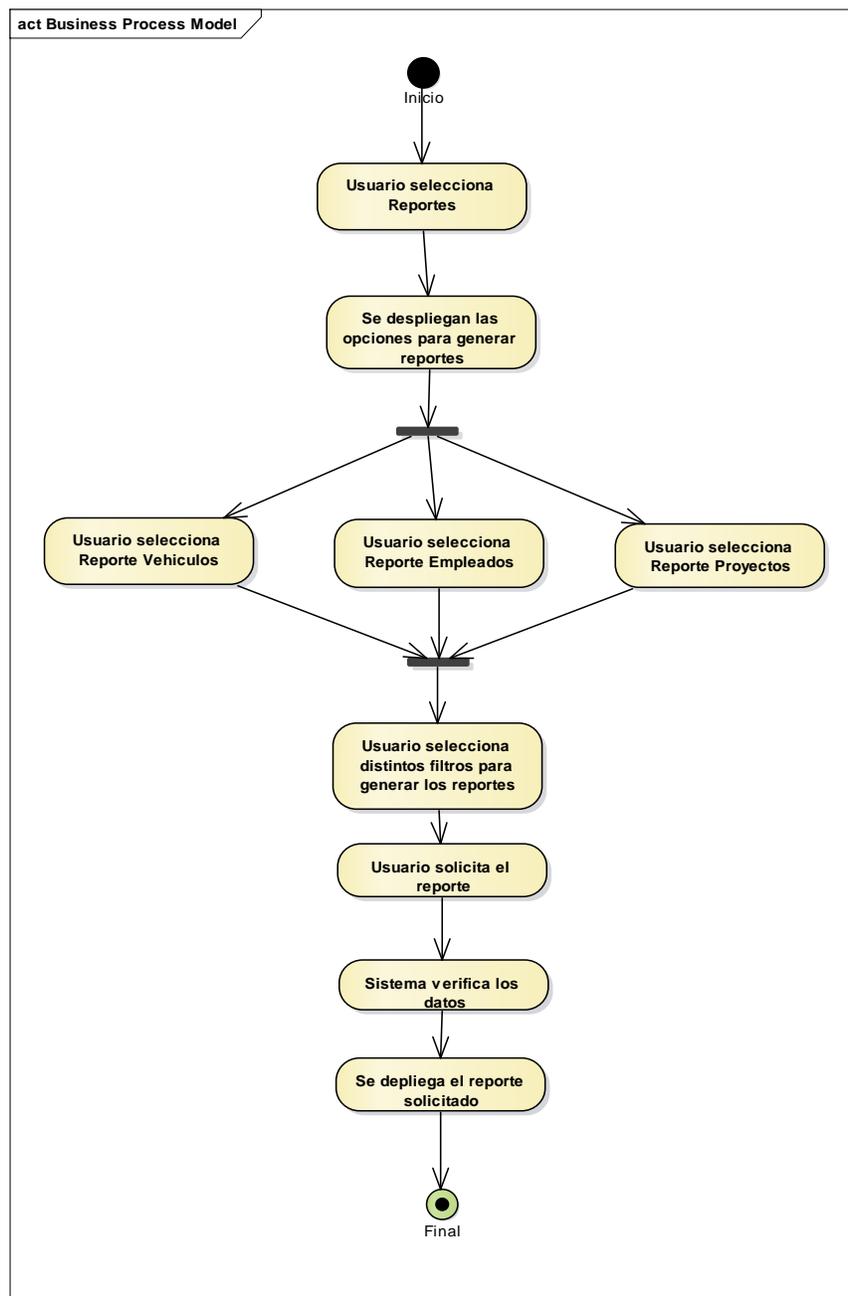
Diagrama de actividades Administración de parte diario



### II.1.2.10.7 Diagrama de actividades Reportes

Figura 23

Diagrama de actividades Reportes

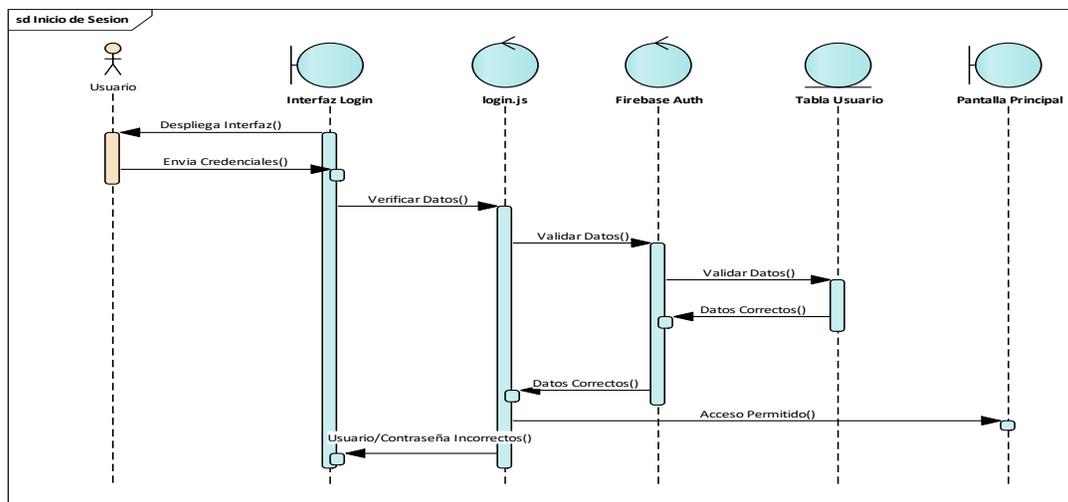


## II.1.2.11 Diagramas de secuencias

### II.1.2.11.1 Diagramas de secuencia Acceso al sistema

Figura 24

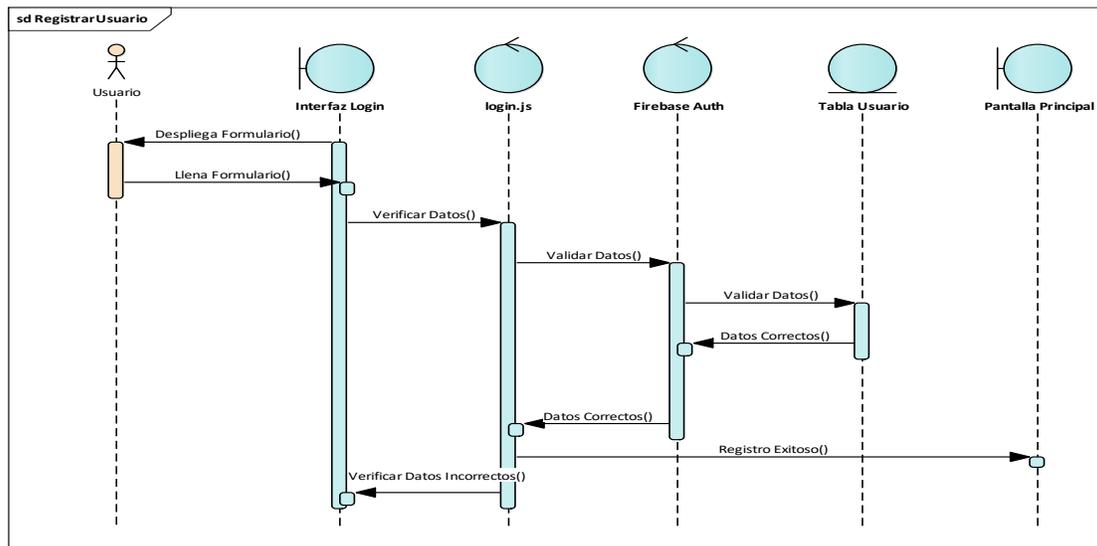
Diagrama de secuencia Accesos al sistema



### II.1.2.11.2 Diagramas de secuencia Registrarse al sistema

Figura 25

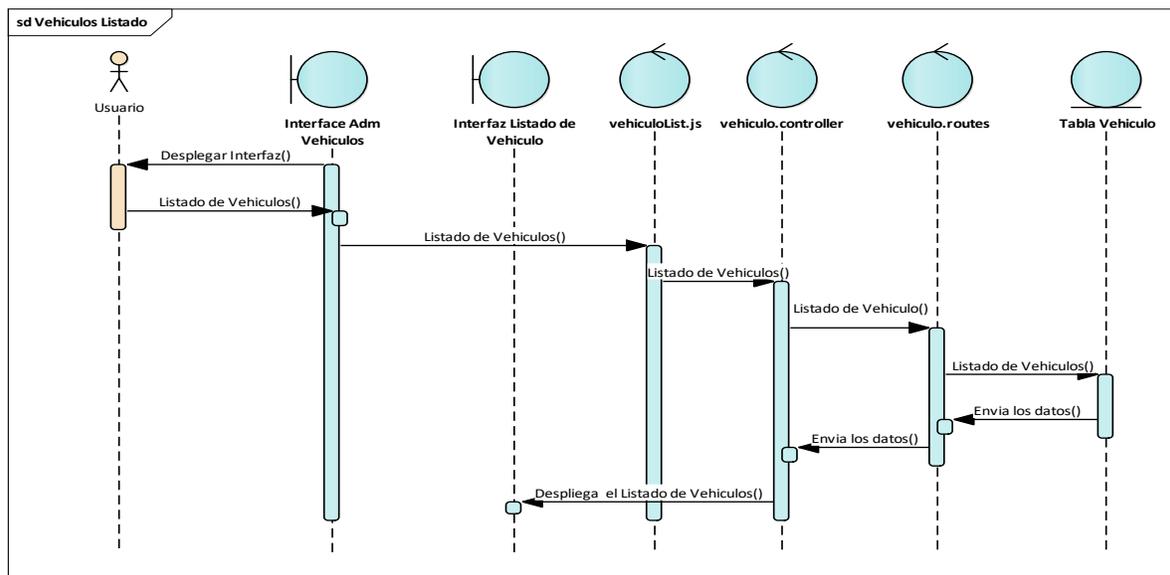
Diagrama de secuencia Registrarse al sistema



### II.1.2.11.3 Diagramas de secuencia Administración de vehículos

**Figura 26**

*Diagrama de secuencia Listado de vehículos*



**Figura 27**

*Diagrama de secuencia Añadir vehículo*

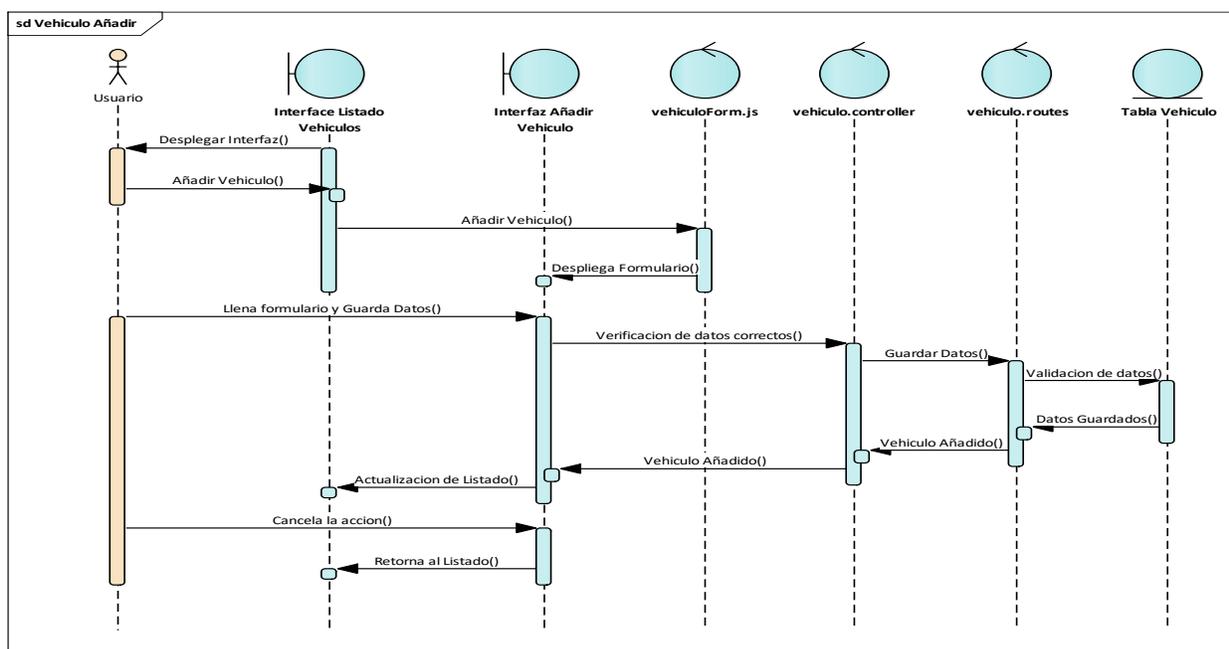


Figura 28

Diagrama de secuencia Modificar vehículo

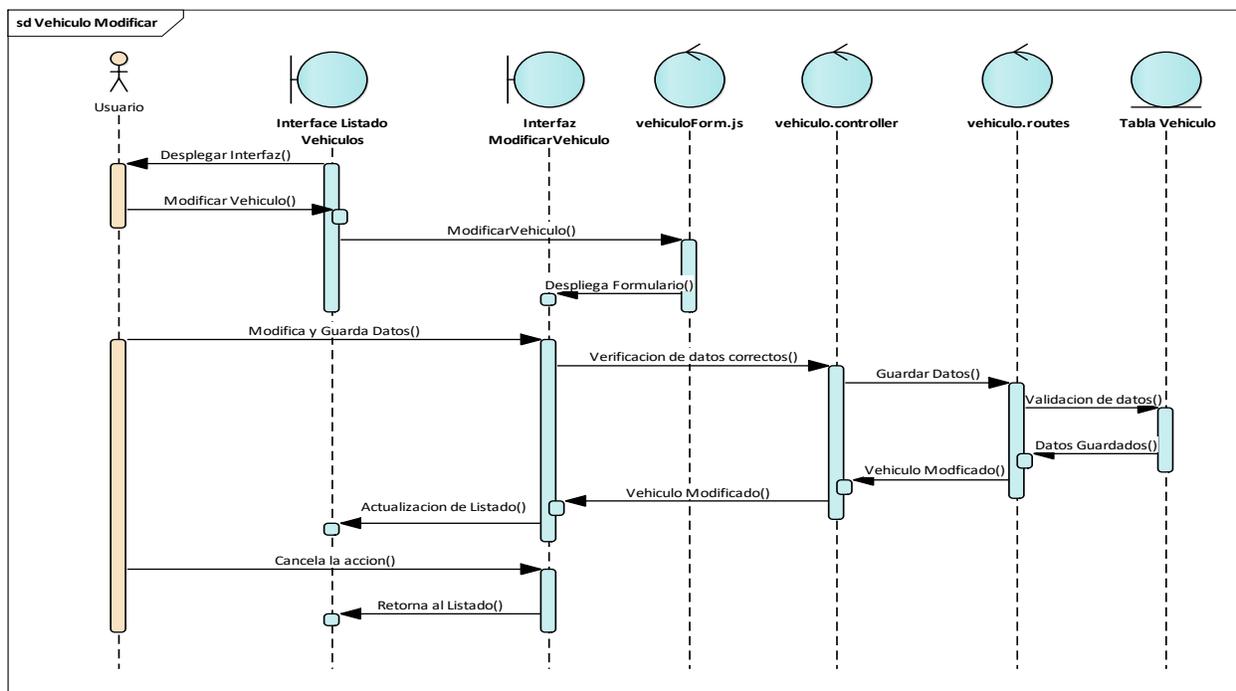


Figura 29

Diagrama de secuencia Eliminar vehículo

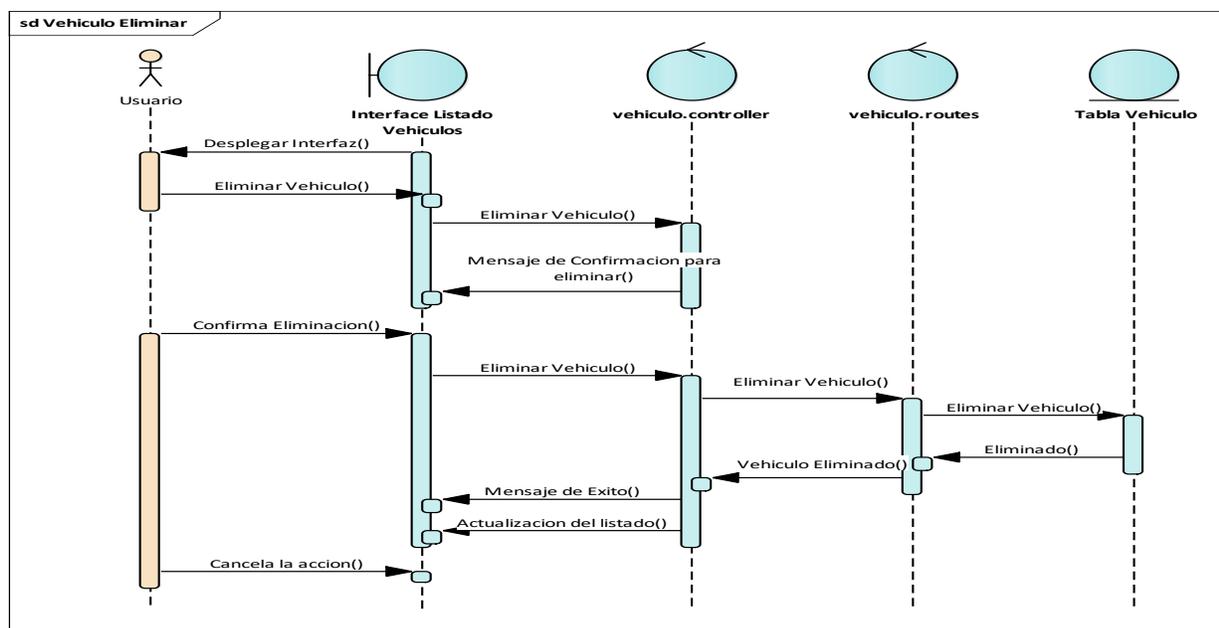


Figura 30

Diagrama de secuencia Asignar conductor

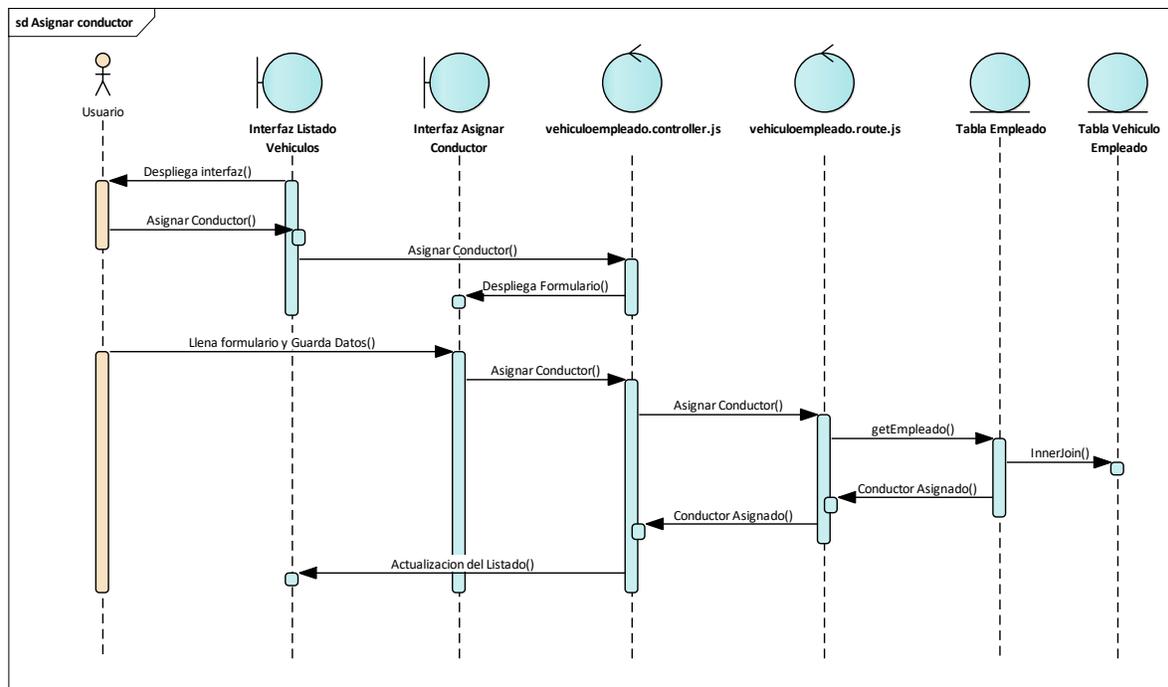


Figura 31

Diagrama de secuencia Ubicación

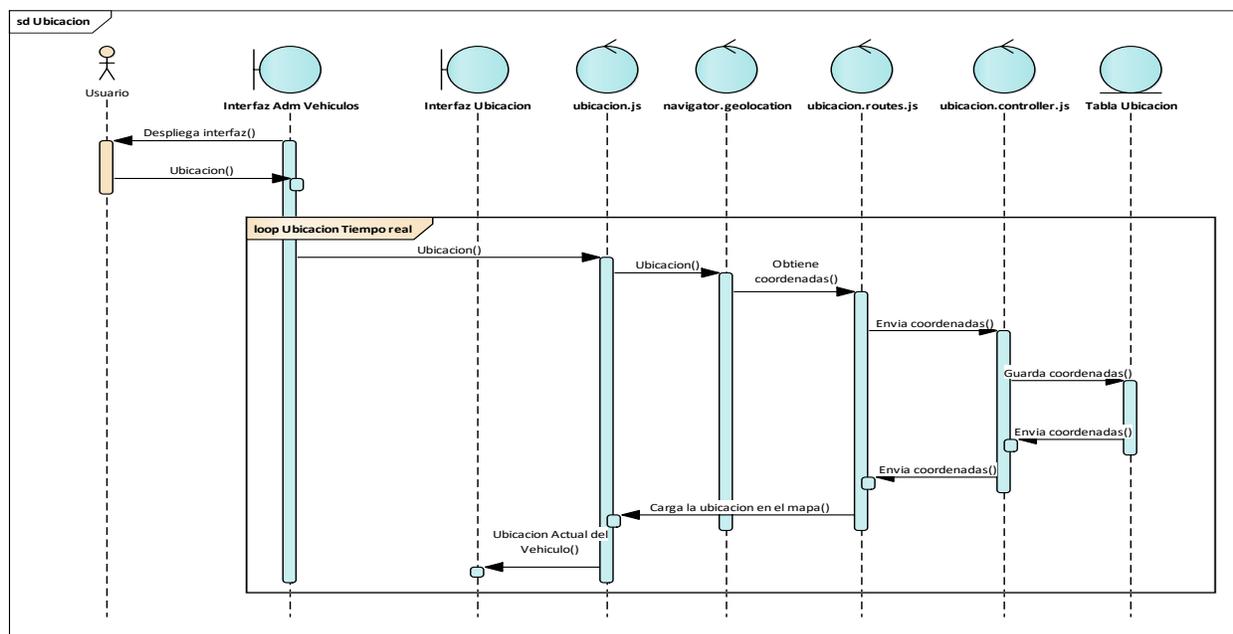


Figura 32

Diagrama de secuencia Trayecto

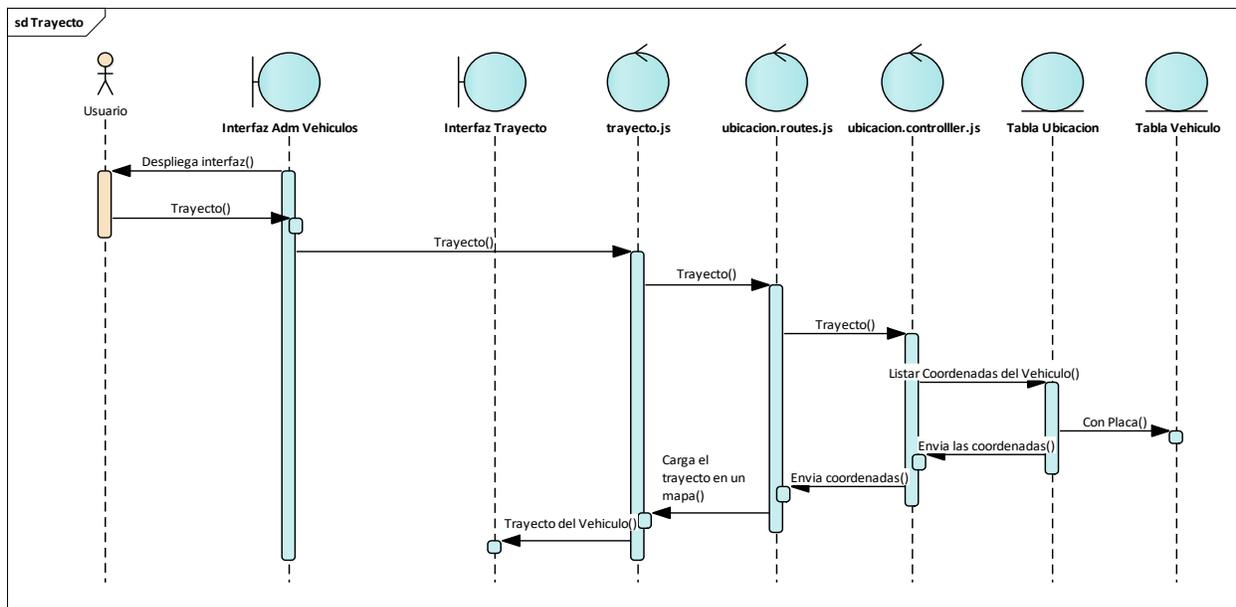


Figura 33

Diagrama de secuencia Listado horarios

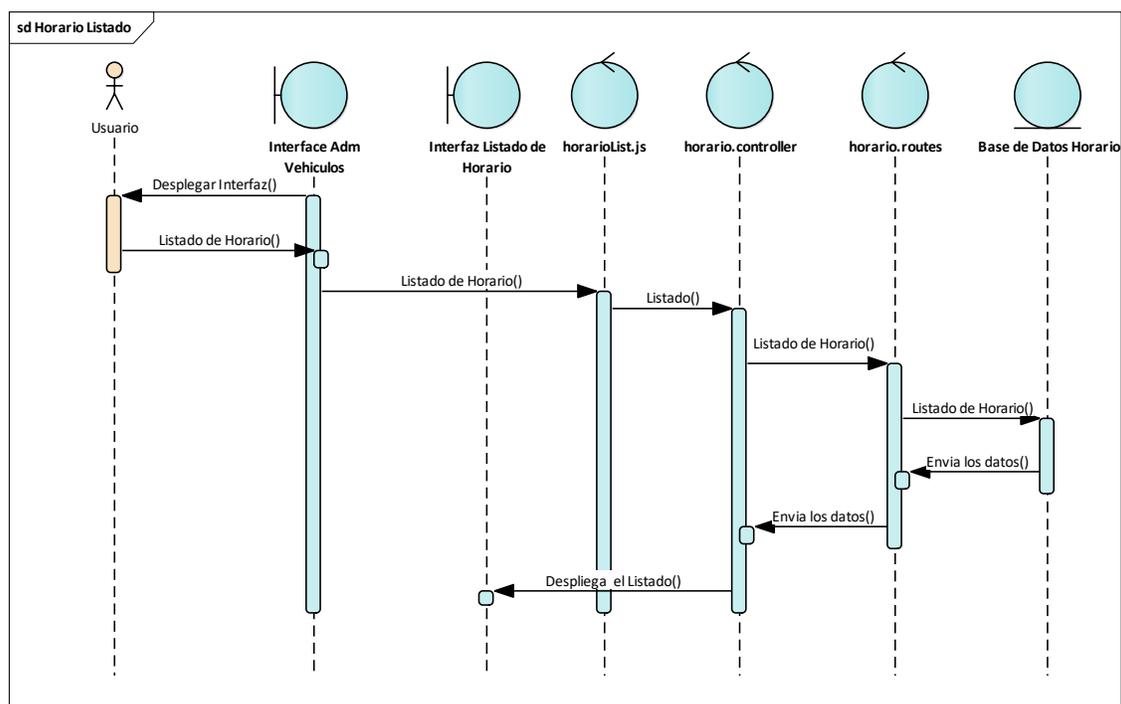


Figura 34

Diagrama de secuencia Añadir horario

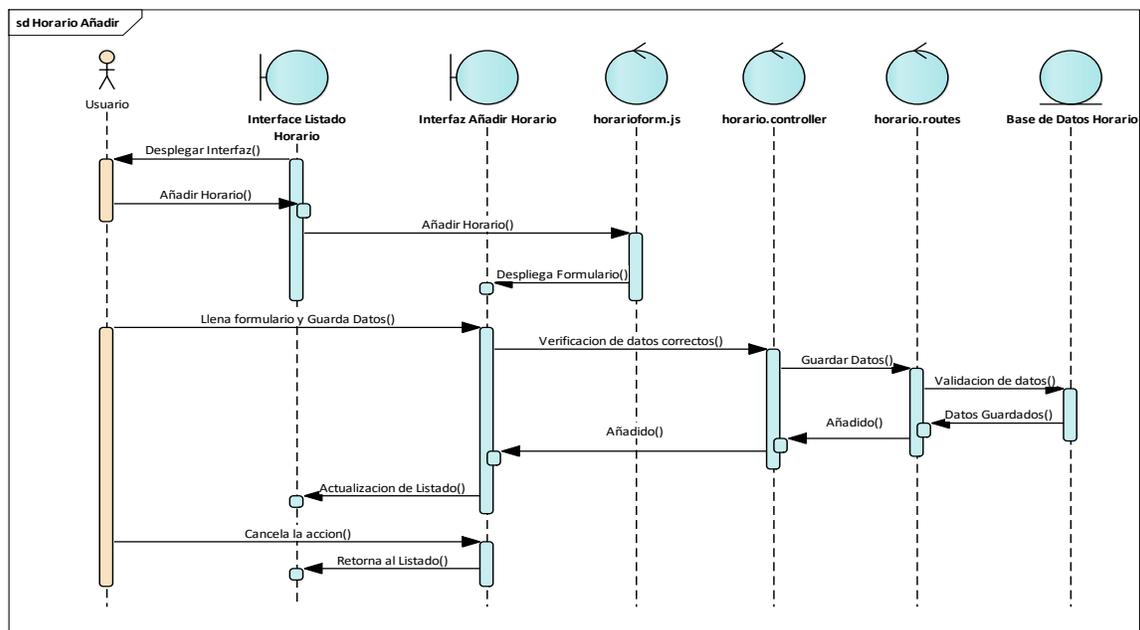


Figura 35

Diagrama de secuencia Modificar horario

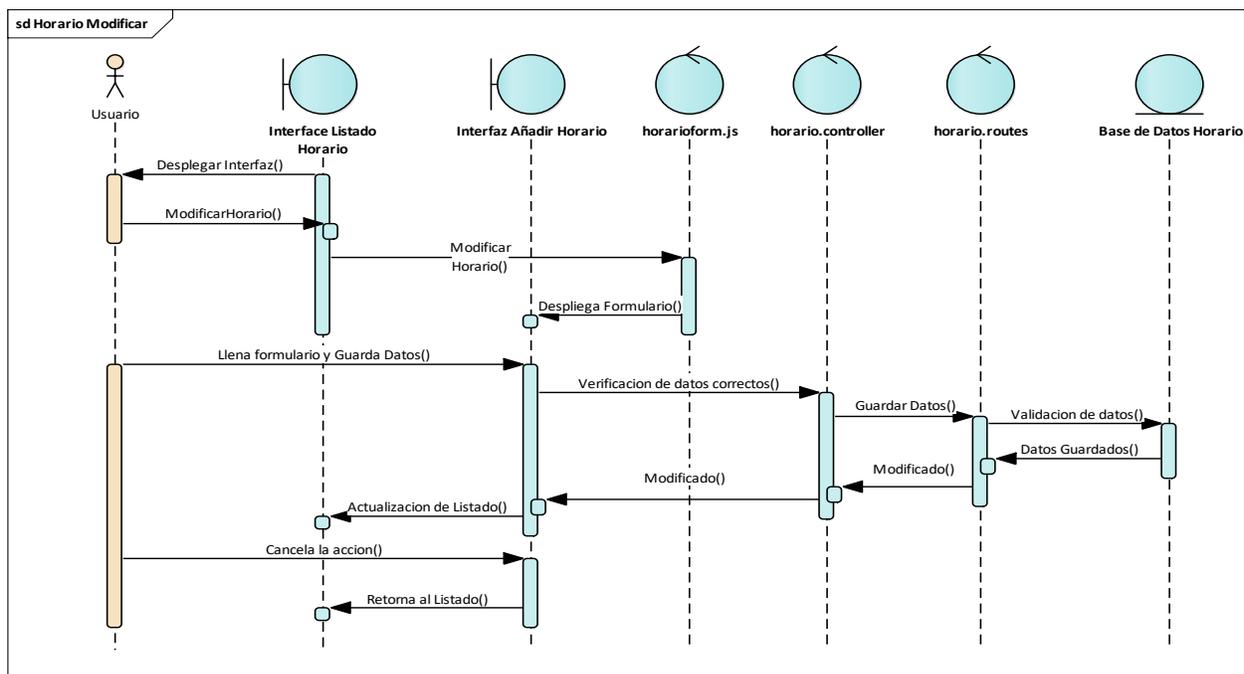
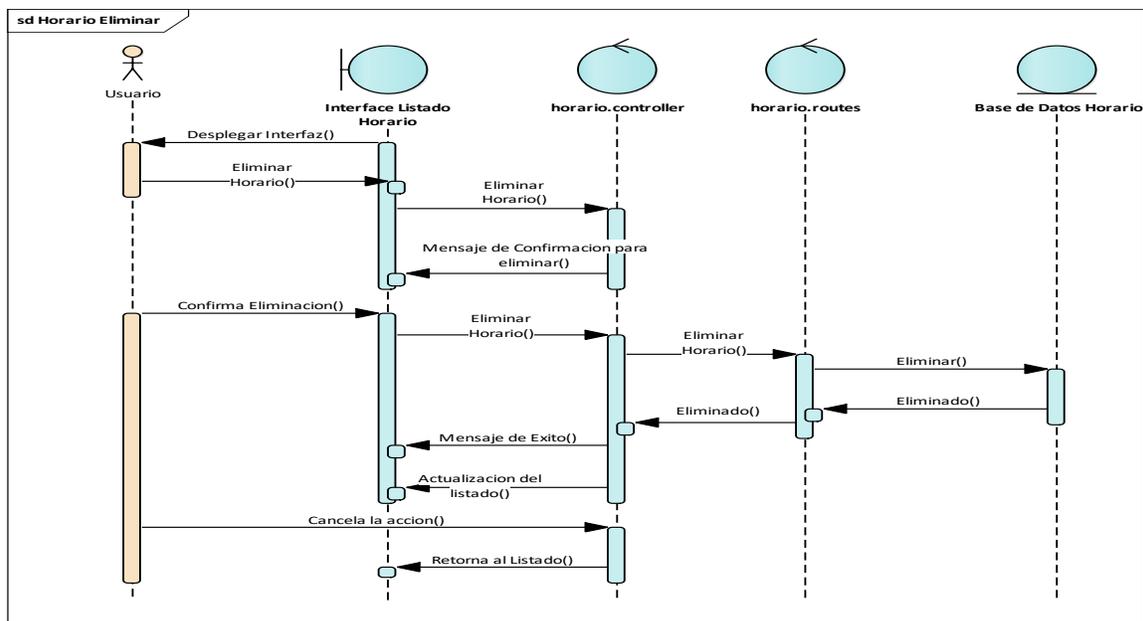


Figura 36

Diagrama de secuencia Eliminar horario



#### II.1.2.11.4 Diagramas de secuencia Administración de empleados

Figura 37

Diagrama de secuencia Listado empleados

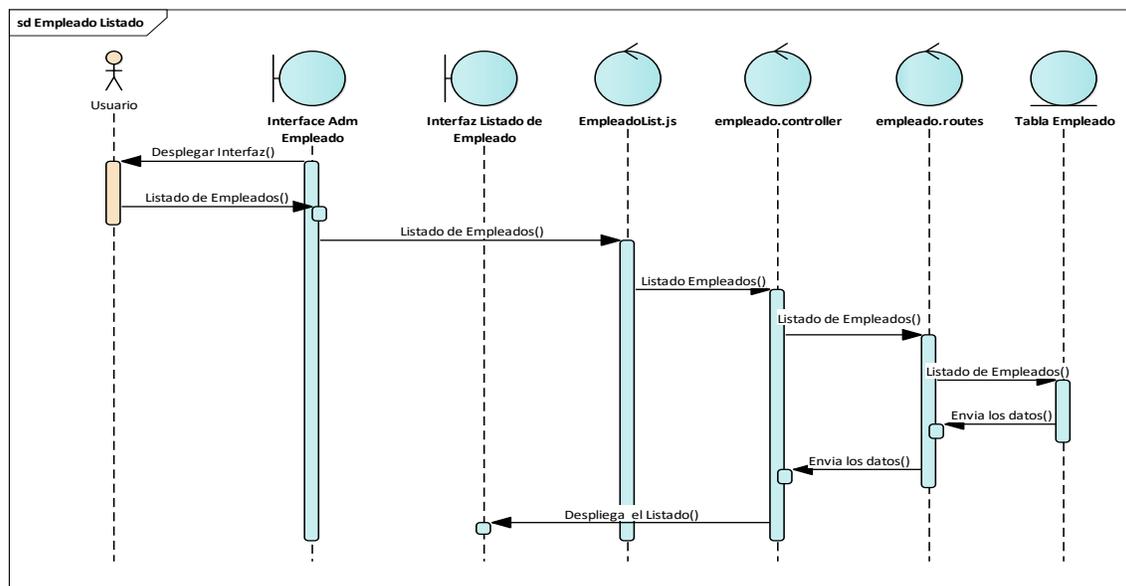
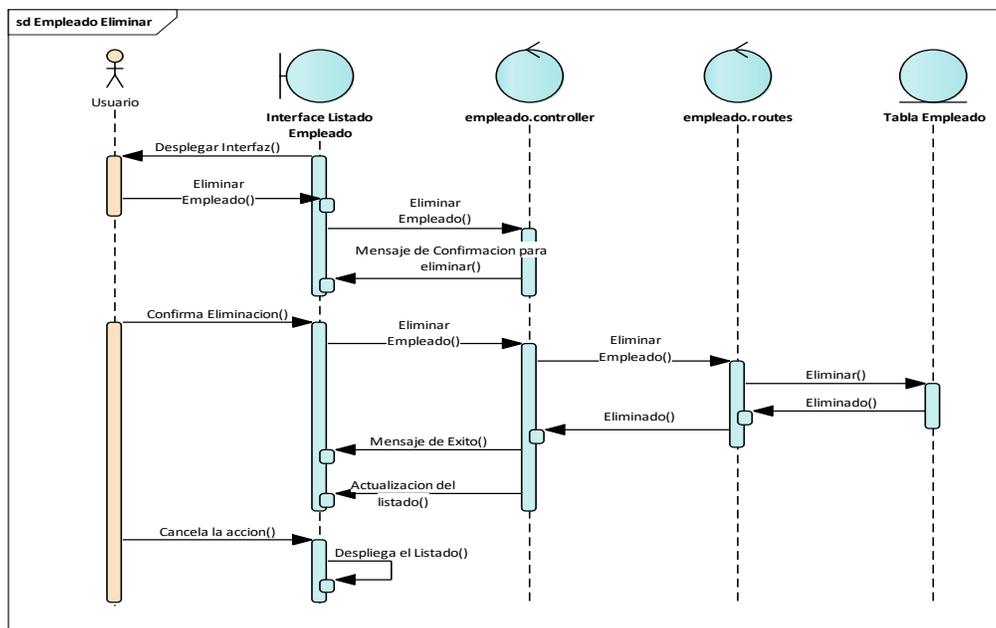


Figura 38

Diagrama de secuencia Eliminar empleado



### II.1.2.11.5 Diagramas de secuencia Administración de proyectos

Figura 39

Diagrama de secuencia Listado de proyectos

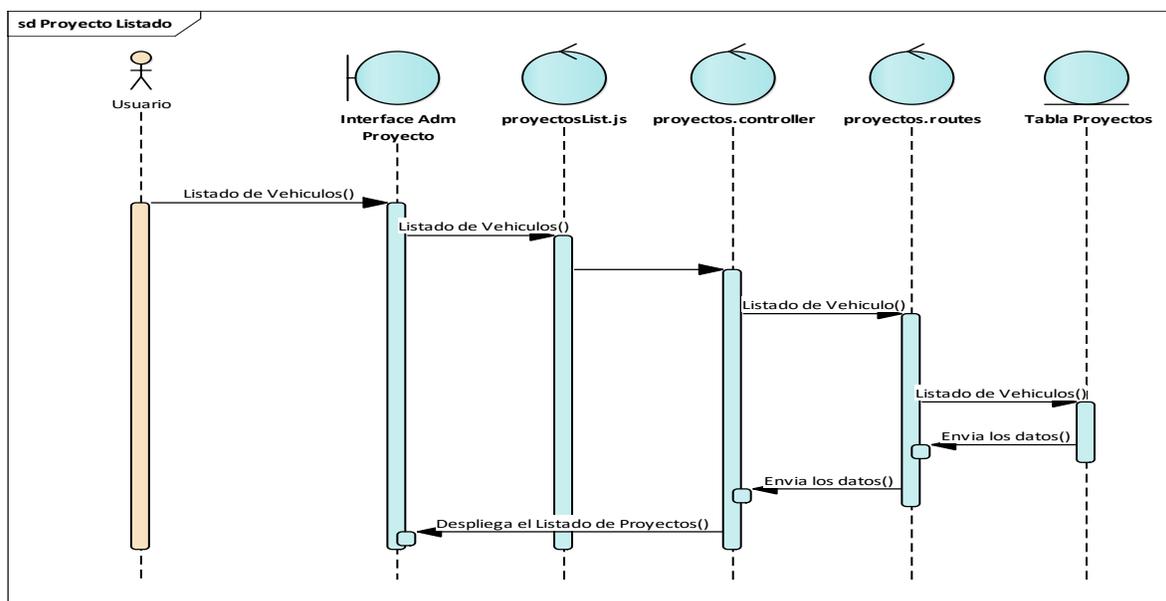


Figura 40

Diagrama de secuencia Añadir proyecto

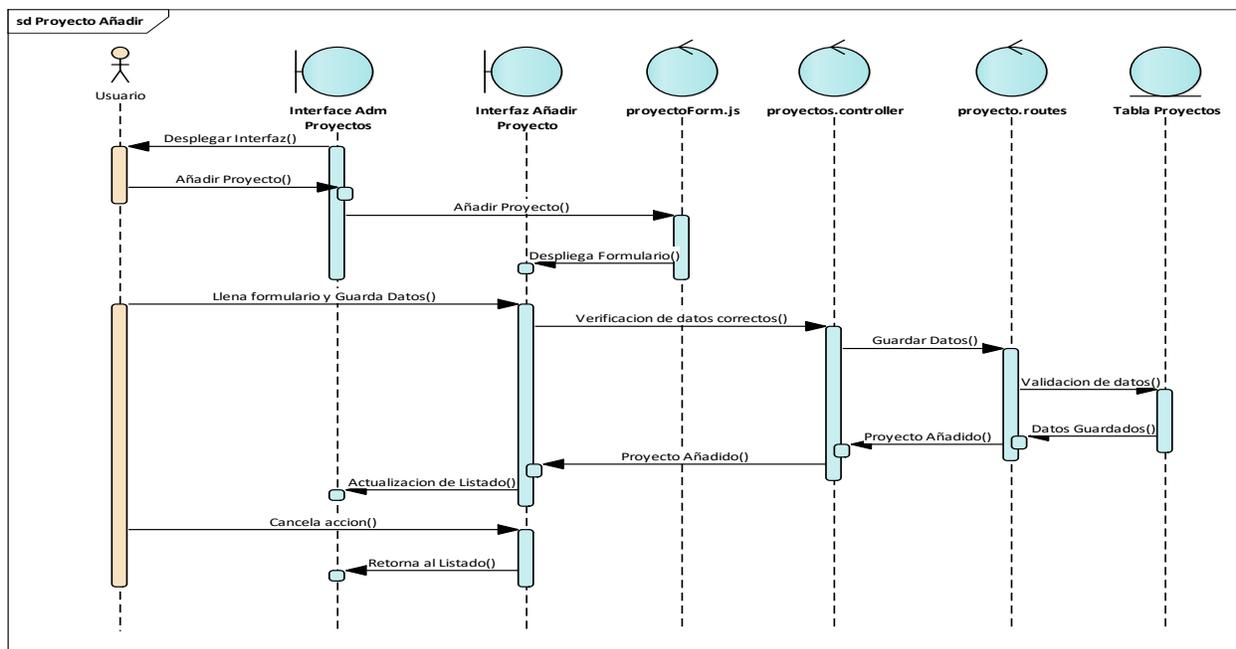


Figura 41

Diagrama de secuencia Modificar proyecto

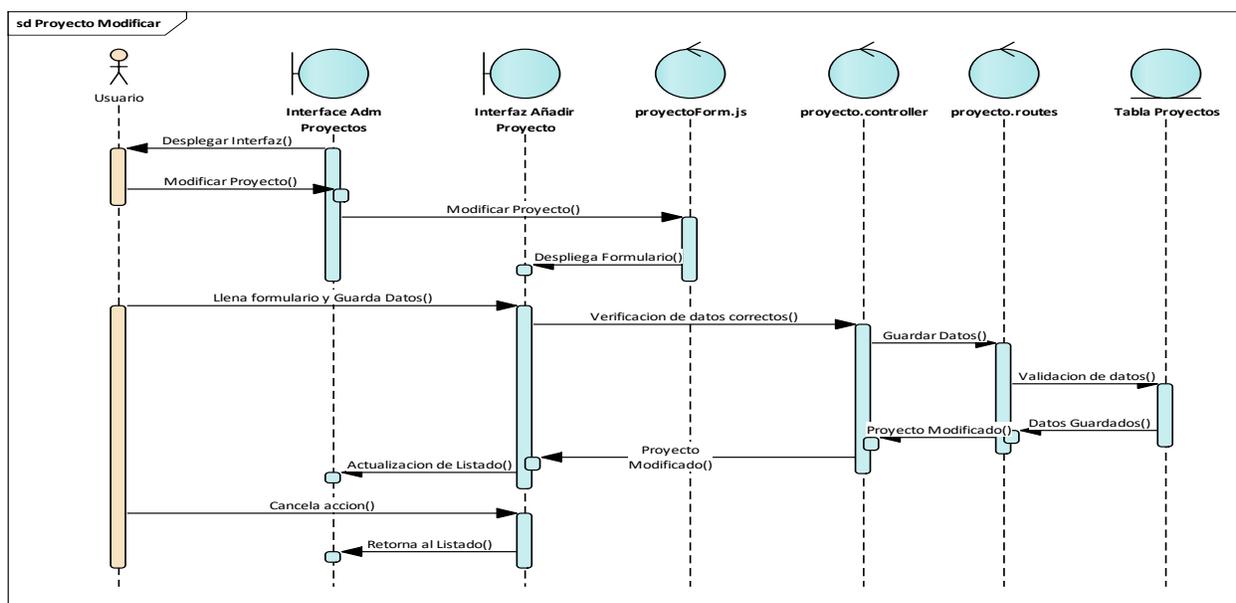


Figura 42

Diagrama de secuencia Vehículos asignados

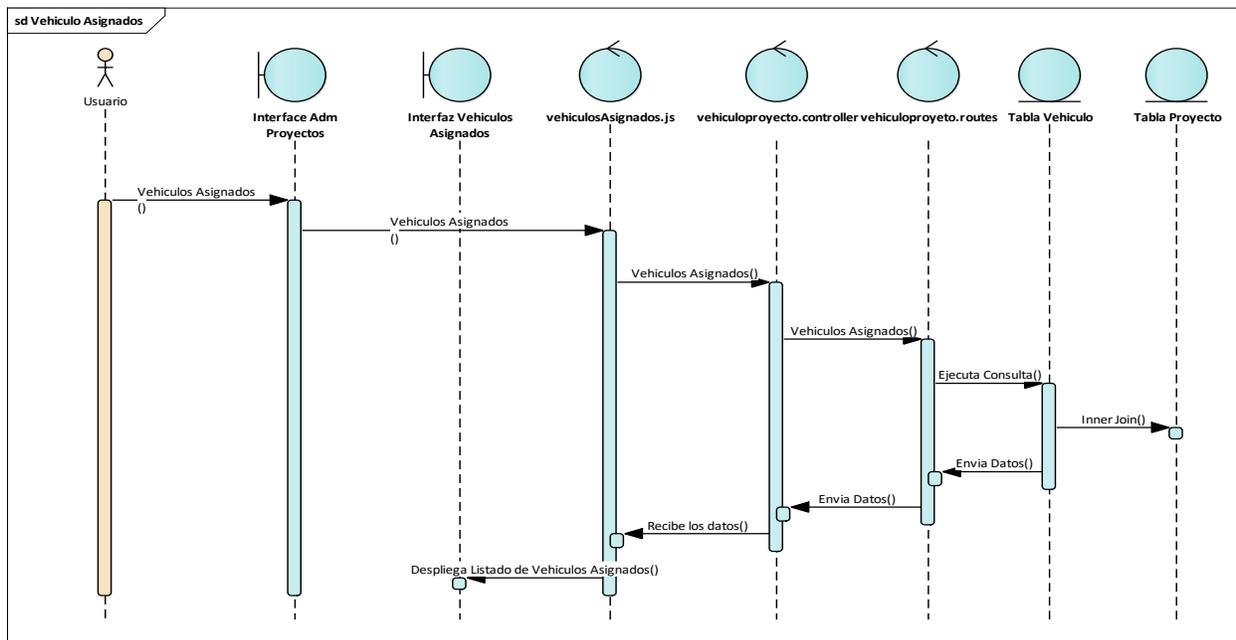
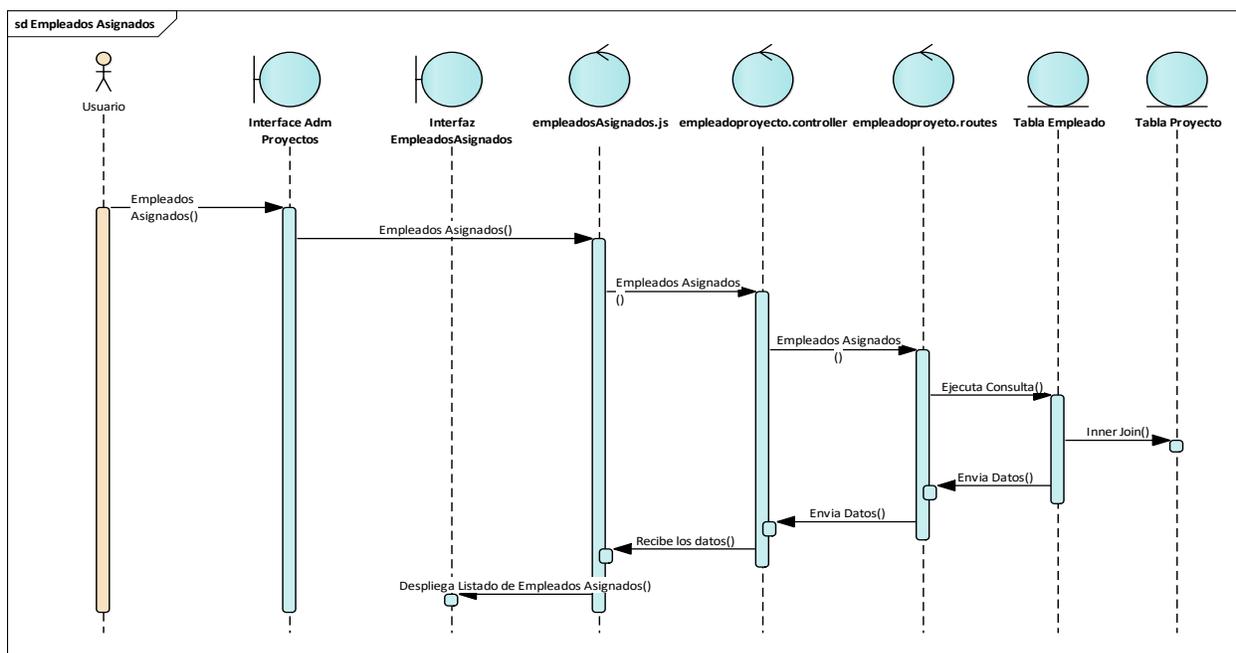


Figura 43

Diagrama de secuencia Empleados asignados



### II.1.2.11.6 Diagramas de secuencia Administración de parte diario

Figura 44

Diagrama de secuencia Listado parte diario

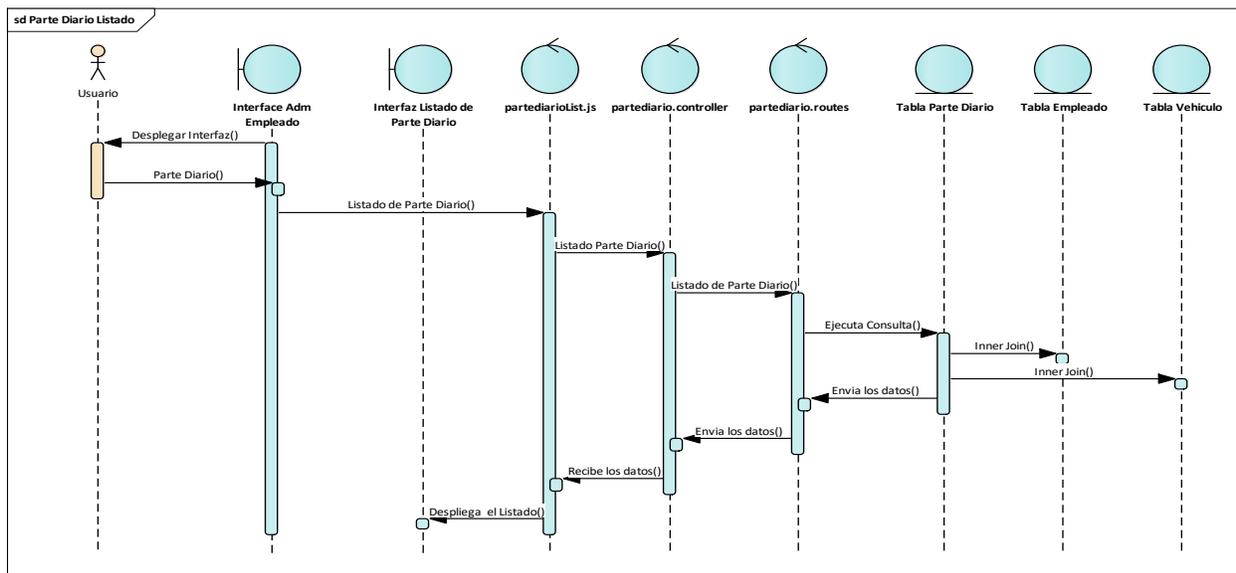


Figura 45

Diagrama de secuencia Añadir parte diario

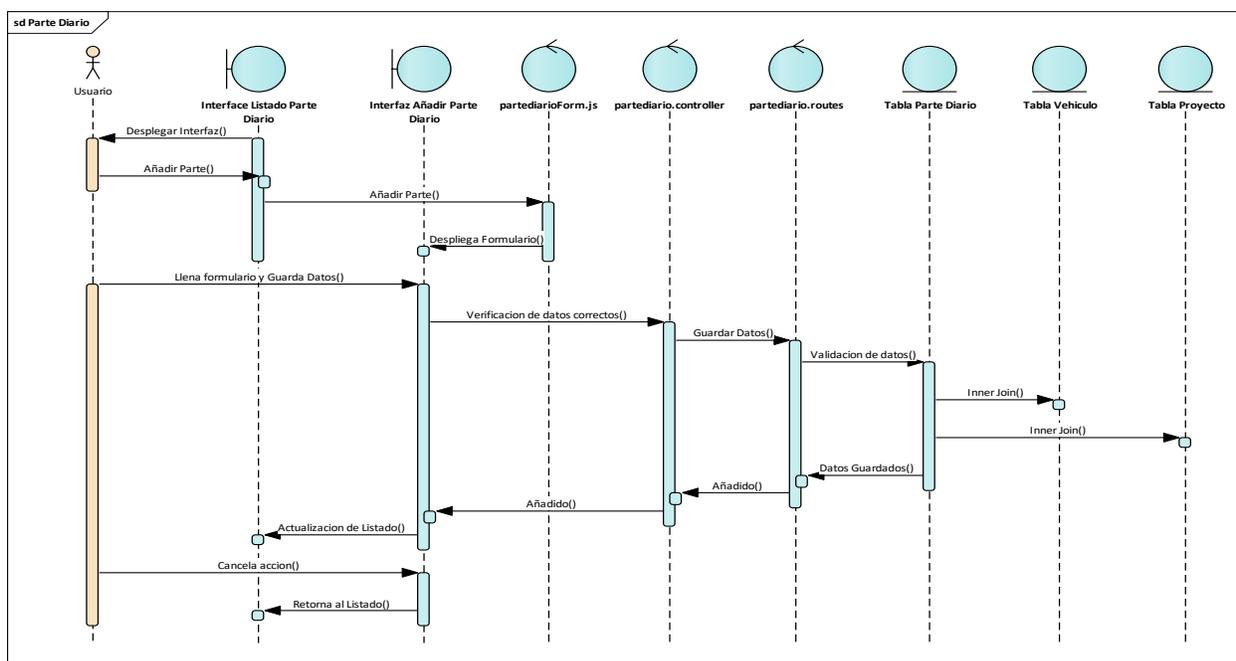


Figura 46

Diagrama de secuencia Eliminar parte diario

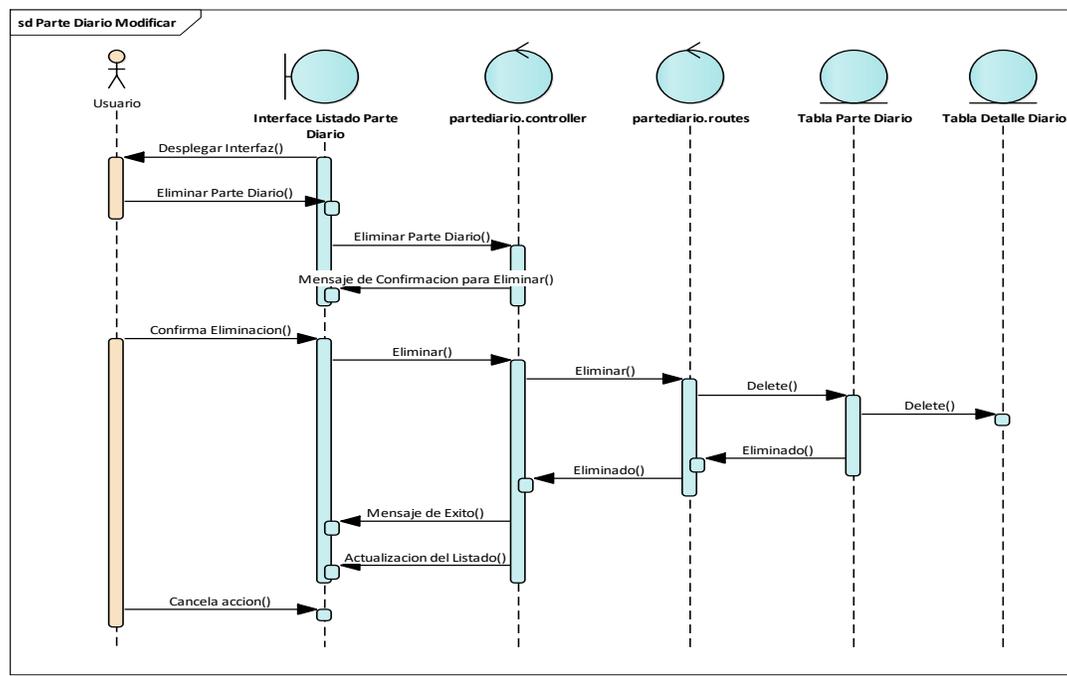


Figura 47

Diagrama de secuencia Listado de Detalle Parte Diario

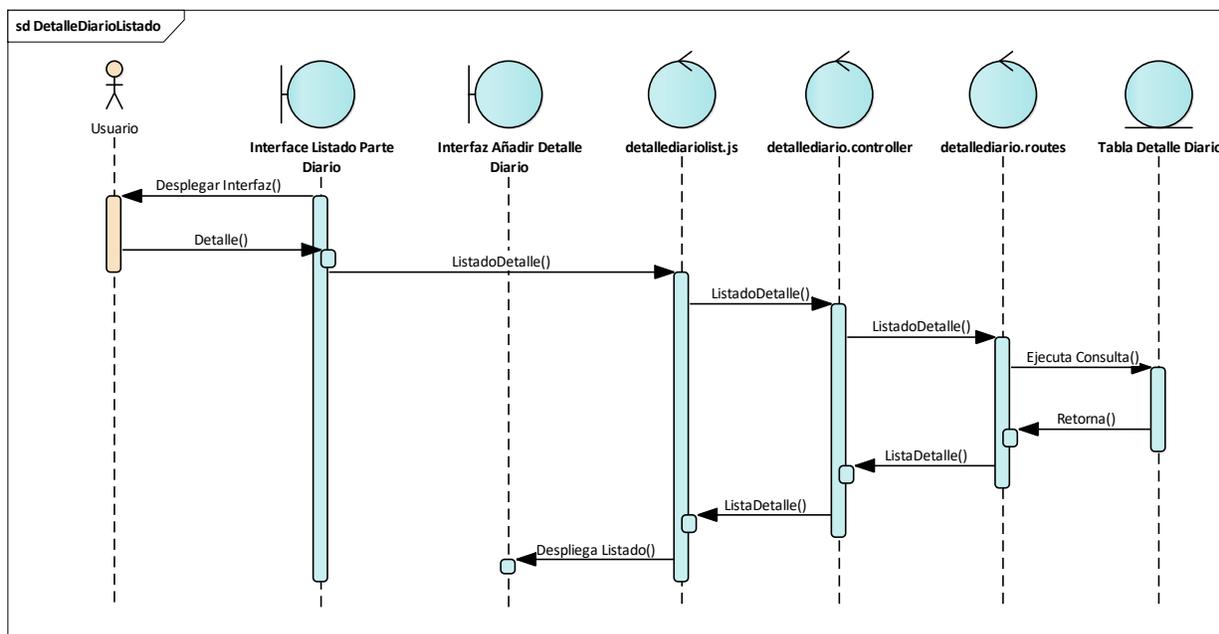


Figura 48

Diagrama de secuencia Añadir Detalle Parte Diario

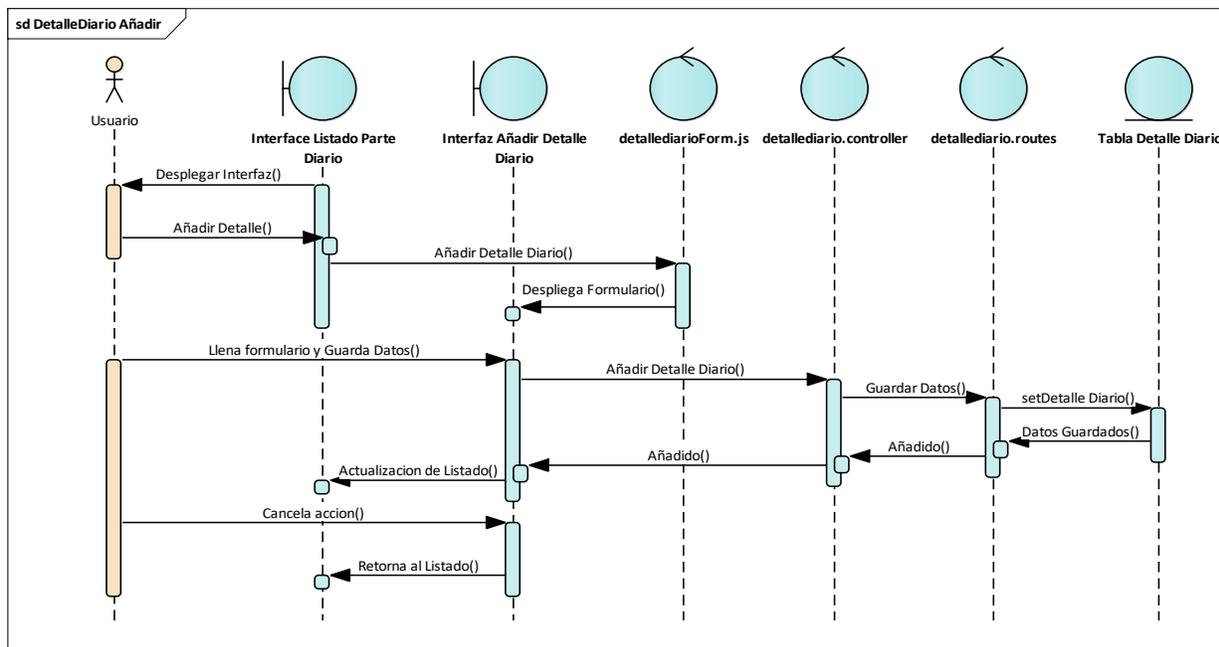
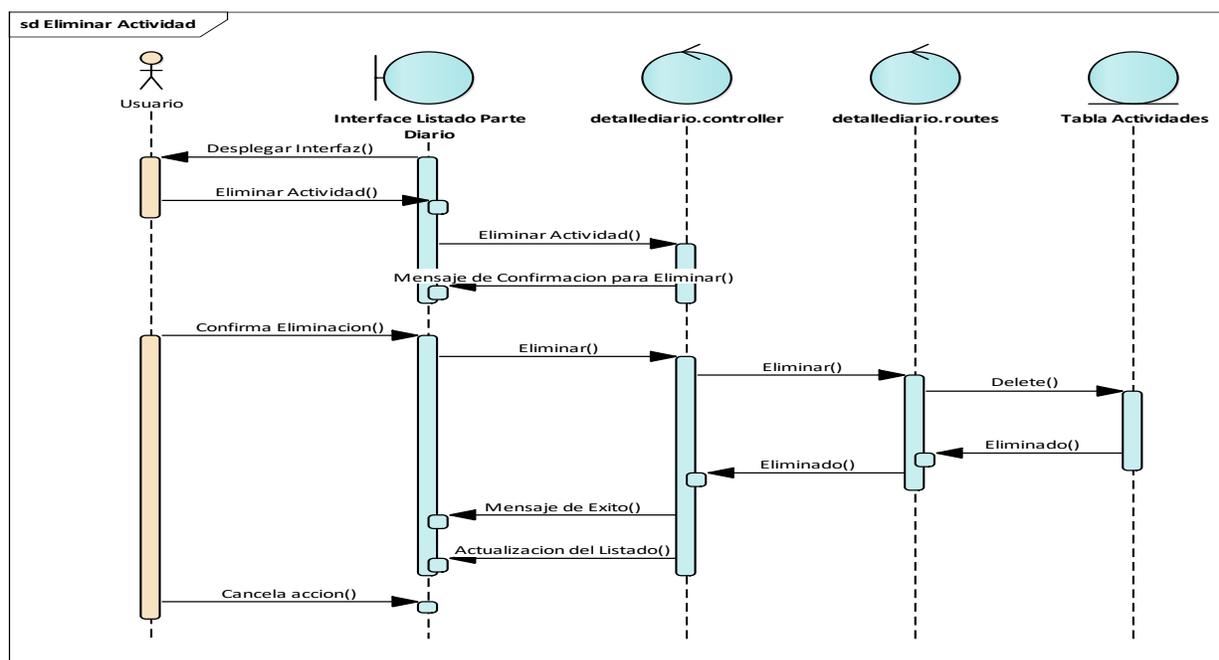


Figura 49

Diagrama de secuencia Eliminar Detalle Parte Diario



### II.1.2.11.7 Diagramas de secuencia Reportes

Figura 50

Diagrama de secuencia Reporte de Conductores

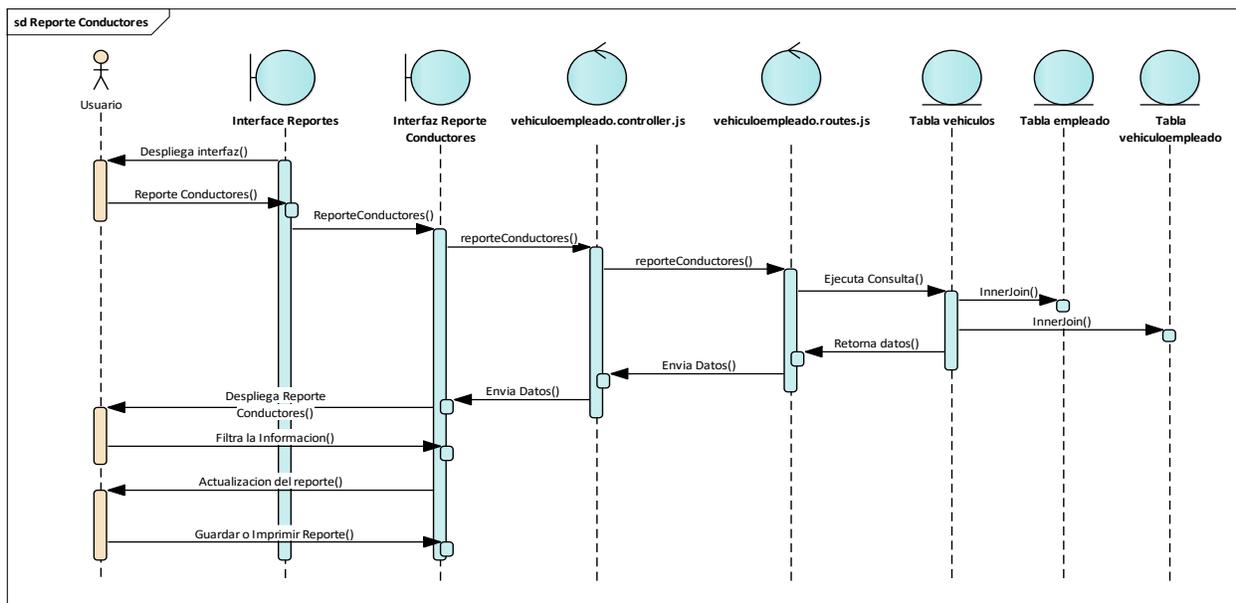


Figura 51

Diagrama de secuencia Reporte de Horarios

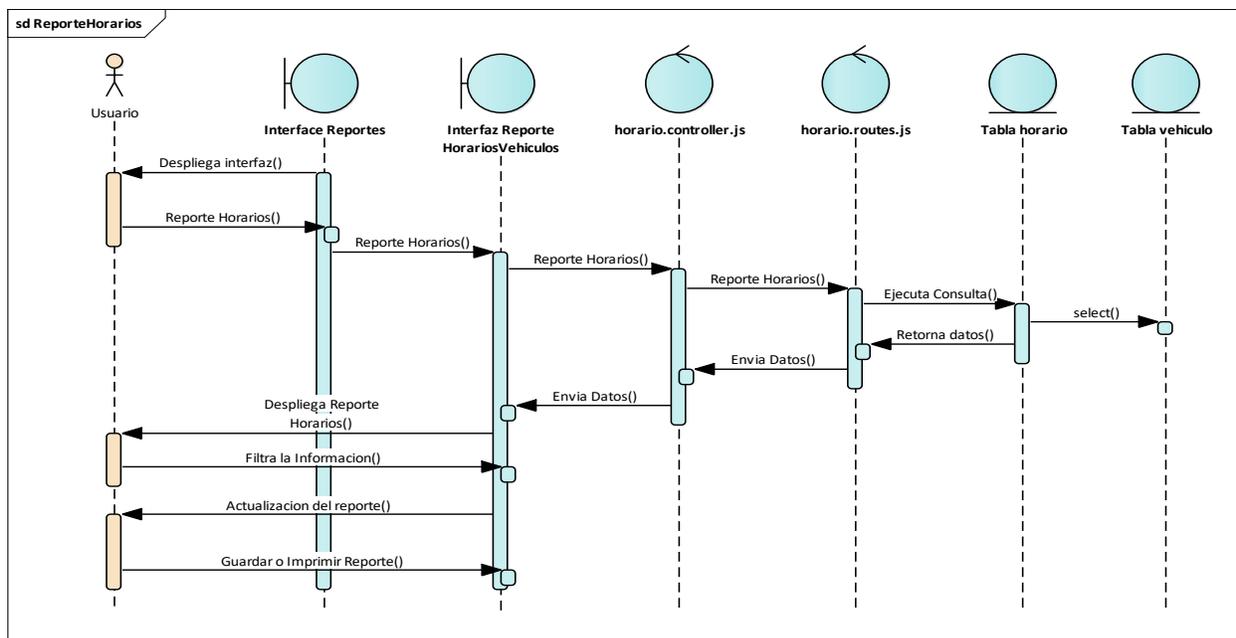


Figura 52

Diagrama de secuencia Reporte de Proyectos Vehículos

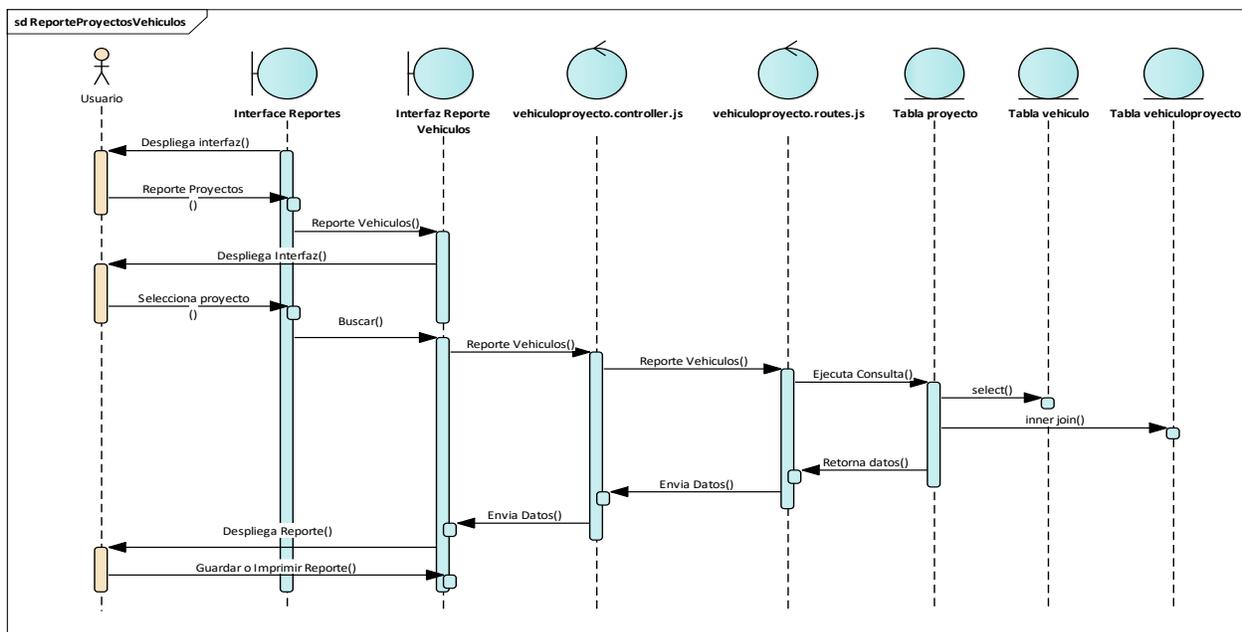
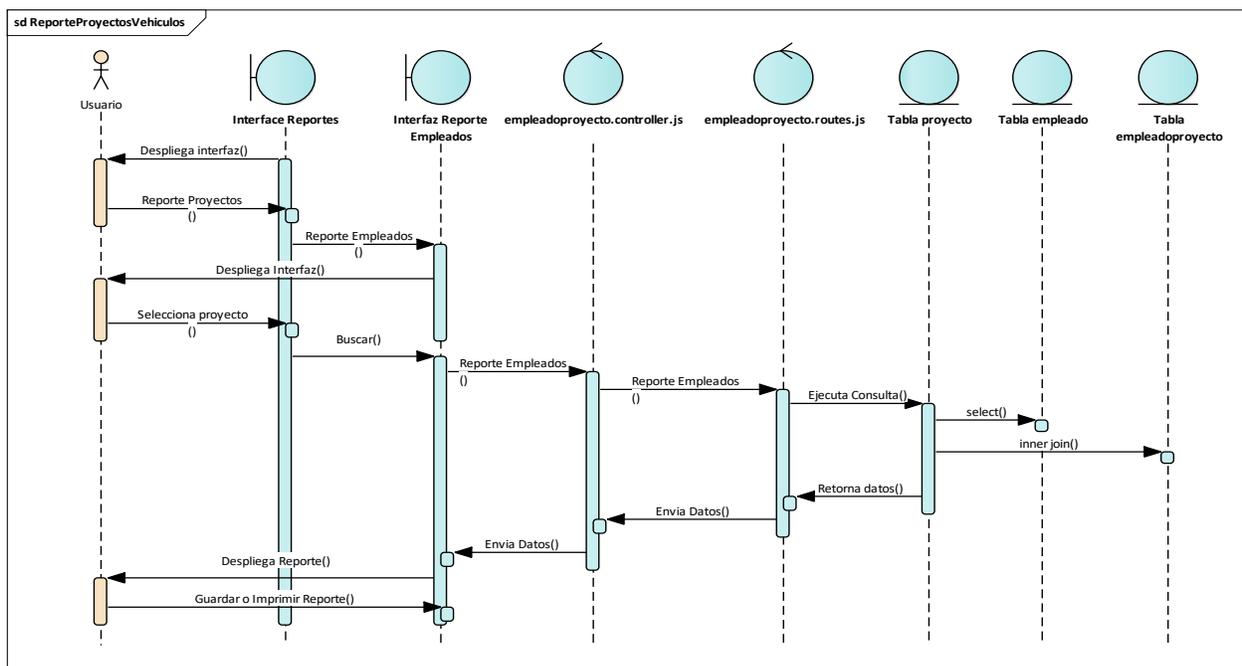


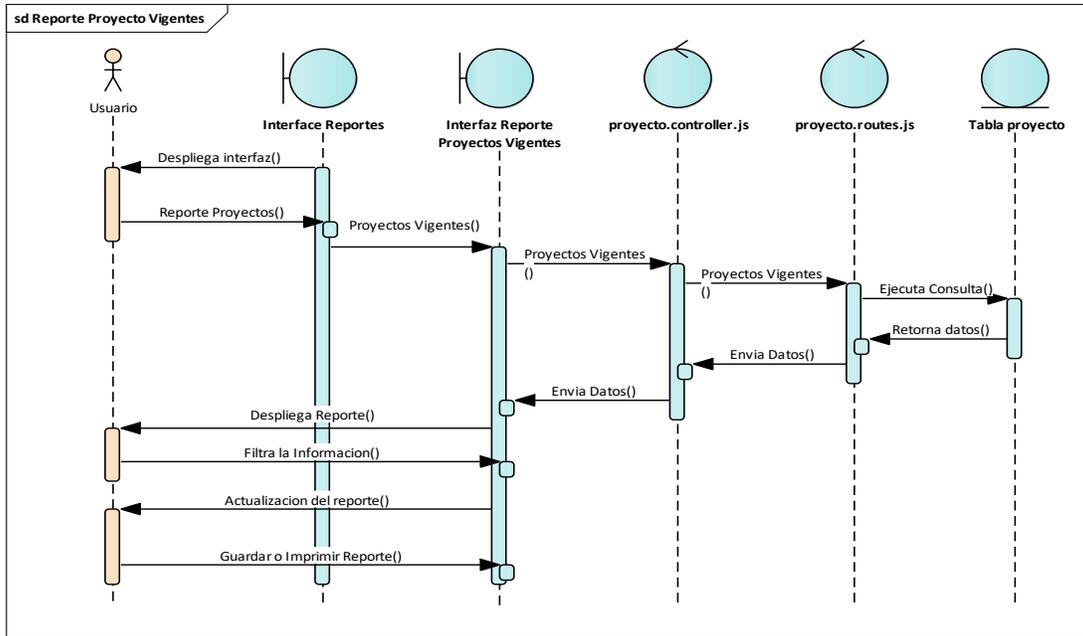
Figura 53

Diagramas de secuencia Reporte de Proyectos Empleados



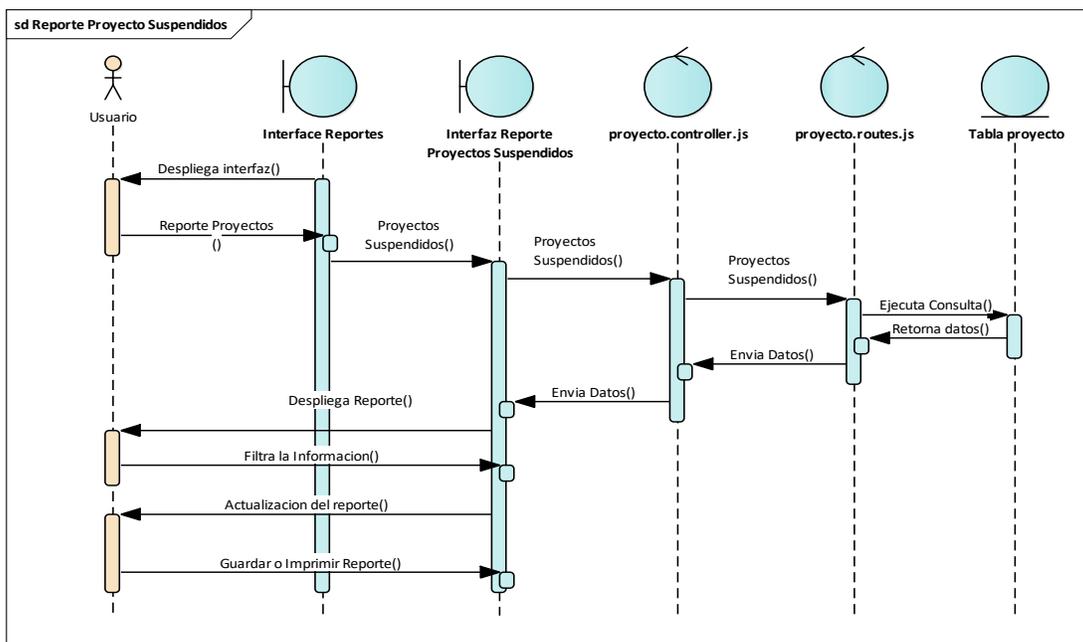
**Figura 54**

*Diagramas de secuencia Reporte de Proyectos Vigentes*



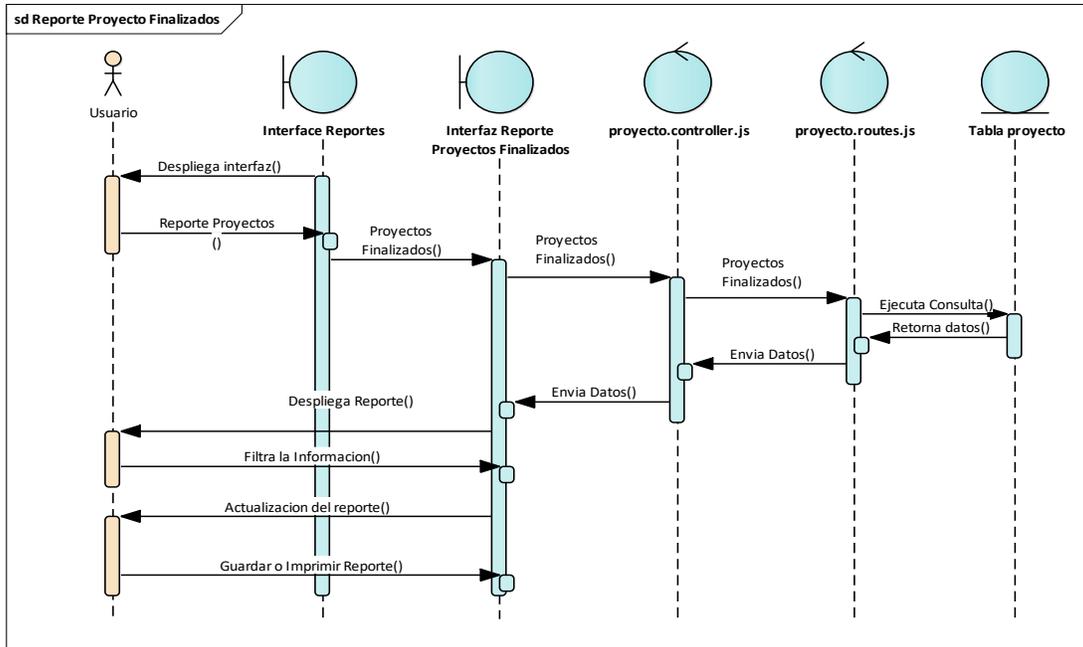
**Figura 55**

*Diagramas de secuencia Reporte de Proyectos Suspendidos*



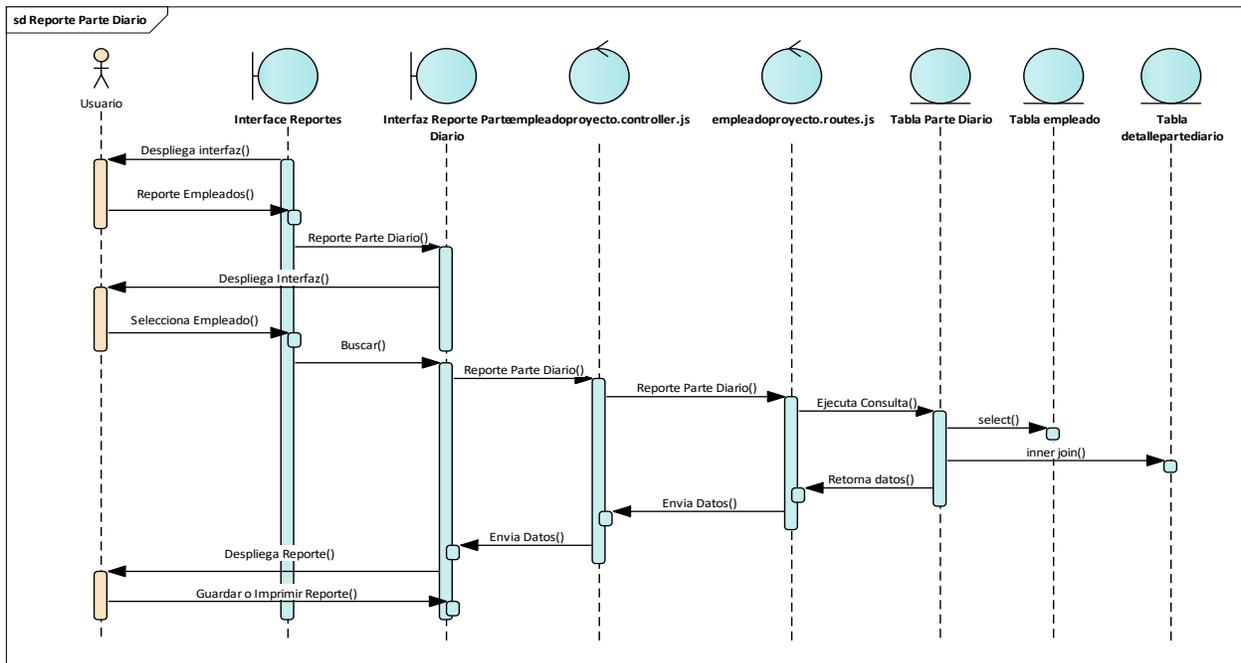
**Figura 56**

*Diagramas de secuencia Reporte de Proyectos Finalizados*



**Figura 57**

*Diagramas de secuencia Reporte de Parte Diario*



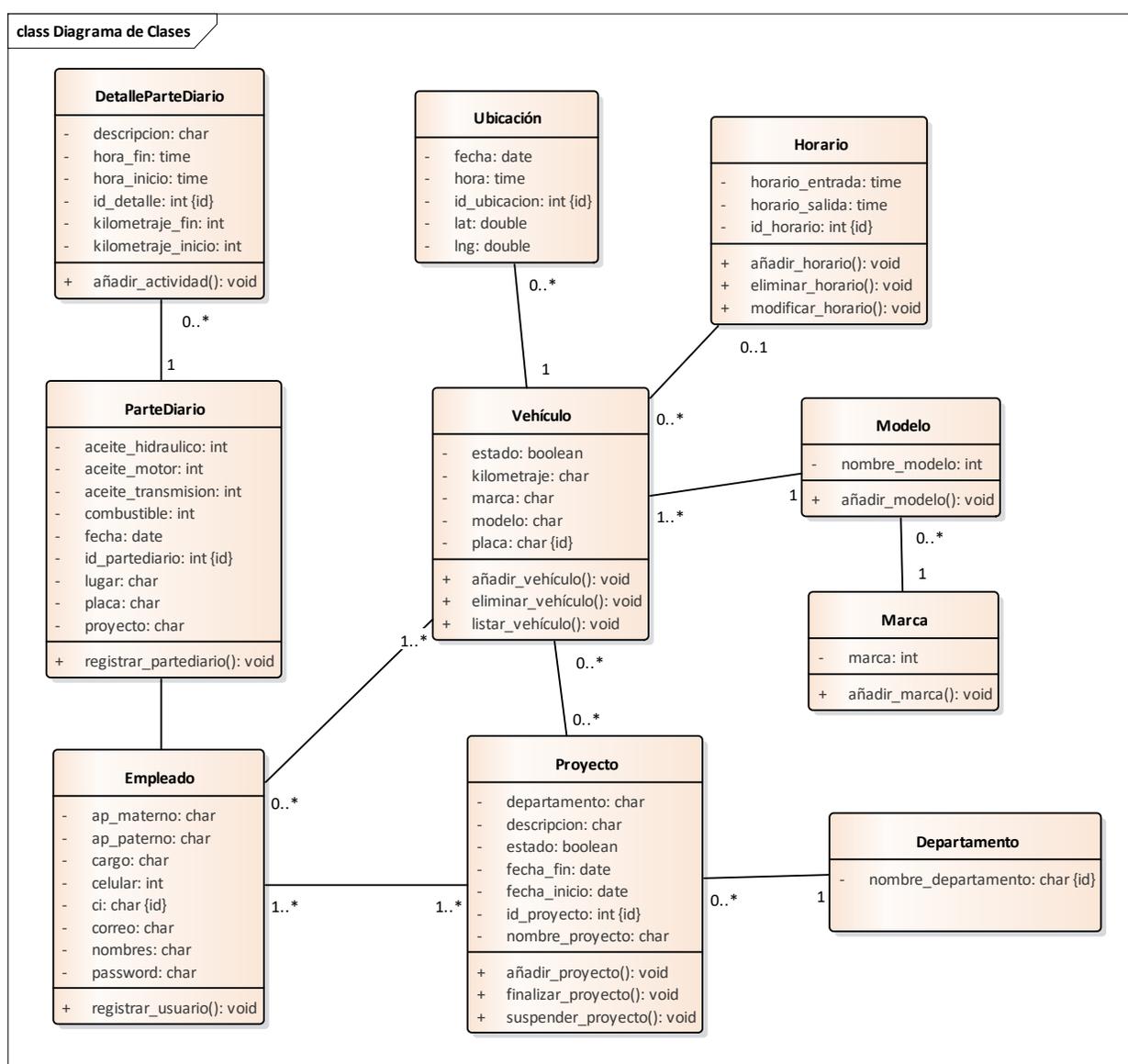
## II.1.2.12 Modelo de datos

El modelo de datos nos permite el análisis, diseño y comprensión de la estructura que tendrá la base de datos, la cual guardará toda la información recopilada por el sistema.

### II.1.2.12.1 Diagrama de clases

**Figura 58**

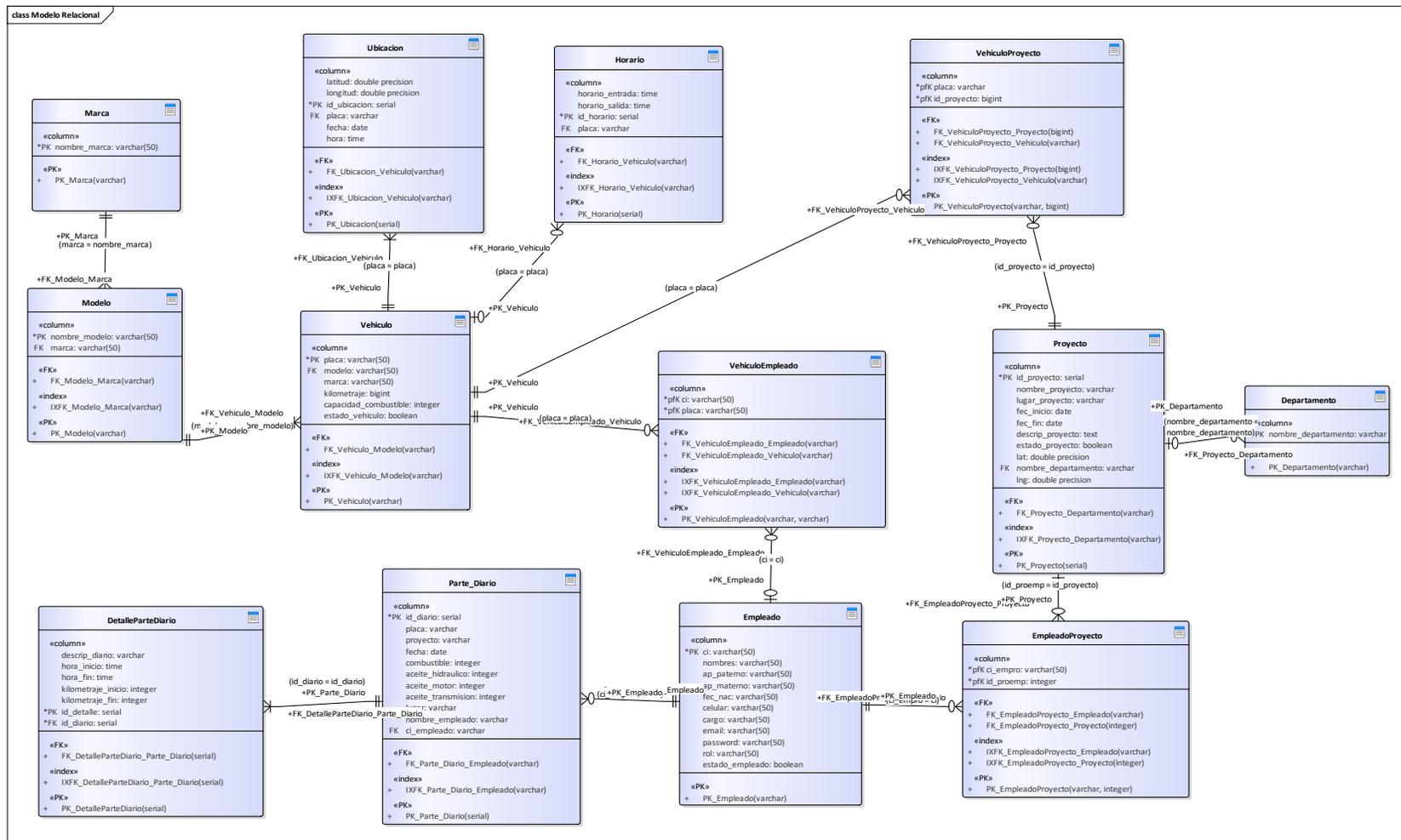
*Diagrama de clases*



## II.1.2.12.2 Modelo relacional

Figura 59

Modelo relacional



### II.1.2.12.3 Diccionario de datos

#### Descripción de las tablas

**Tabla 14**

*Tabla diccionario de datos empleado*

<b>Nombre: Empleado</b>			
<b>Descripción: Contiene datos personales de los empleados</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Nombres	Carácter	50	Nombres del empleado
Ap_paterno	Carácter	50	Apellido paterno del empleado
Ap_materno	Carácter	50	Apellido materno del empleado
CI	Carácter	10	Número de carnet de identificación del empleado e identificador del empleado
Celular	Carácter	50	Número de celular del empleado
Cargo	Carácter	50	Cargo o puesto que desarrolla el empleado
Email	Carácter	50	Email del empleado para el ingreso al sistema
Password	Carácter	50	Contraseña del empleado para ingresar al sistema
Estado	Lógico	1	(True)Activo o (False) Inactivo

**Tabla 15**

*Tabla diccionario de datos proyecto*

<b>Nombre: Proyecto</b>			
<b>Descripción: Contiene datos de los proyectos que realiza la empresa</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Id_Proyecto	Numérico		Identificador del proyecto
Nombre	Carácter	50	Nombre del proyecto
Departamento	Carácter	50	Departamento donde se está desarrollando el proyecto
Fecha_inicio	Fecha		Fecha de inicio del proyecto
Fecha_fin	Fecha		Fecha en la que se concluyó el proyecto
Estado	Lógico	1	(True)Activo o (False) Inactivo
Descripción	Carácter	100	Breve descripción del proyecto
Lat	Numérico		Coordenada que corresponde a la latitud donde se encuentra el proyecto
Lng	Numérico		Coordenada que corresponde a la longitud donde se encuentra el proyecto

**Tabla 16***Tabla diccionario de datos vehículo*

<b>Nombre: Vehículo</b>			
<b>Descripción: Contiene datos que caracterizan a los vehículos de la empresa Bristlecone con el fin de poder identificarlos</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Placa	Carácter	50	Placa del vehículo, identificador de la tabla vehículo
Marca	Carácter	20	Marca comercial del vehículo
Modelo	Carácter	20	Modelo del vehículo
Kilometraje	Numérico		Kilometraje del vehículo
Estado	Lógico	1	(True)Activo o (False) Inactivo

**Tabla 17***Tabla diccionario de datos horario*

<b>Nombre: Horario</b>			
<b>Descripción: Contiene datos que define un horario, que tiene que ser respetado por el vehículo</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Id_horario	Numérico		Identificador del horario
Horario_Entrada	Tiempo		Hora de entrada laboral del vehículo
Horario_Salida	Tiempo		Hora de salida laboral del vehículo
Placa	Carácter	50	Placa del vehículo al cual esta asignado el horario

**Tabla 18***Tabla diccionario de datos parte diario*

<b>Nombre: Parte diario</b>			
<b>Descripción: Contiene datos de los partes diarios que realizan los empleados</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Id_partediario	Numérico		Identificador del parte diario
Placa	Carácter	50	Placa del vehículo el cual está conduciendo el empleado
Proyecto	Carácter	50	Nombre del proyecto en el cual está trabajando el empleado
Fecha	Fecha		Fecha en la que se realizó el parte

			diario
Combustible	Numérico		Combustible con el que cuenta el vehículo al momento del registro del parte diario
Aceite hidráulico	Numérico		Aceite hidráulico con el que cuenta el vehículo al momento del registro del parte diario
Aceite motor	Numérico		Aceite motor con el que cuenta el vehículo al momento del registro del parte diario
Aceite transmisión	Numérico		Aceite transmisión con el que cuenta el vehículo al momento del registro del parte diario
Lugar	Carácter	50	Lugar en el que se encuentra el empleado
CI	Carácter	50	Carnet de identidad del empleado que realiza el parte diario
Nombre_empleado	Carácter	50	Nombre del empleado

**Tabla 19**

Tabla diccionario de datos detalle parte diario

<b>Nombre: Detalle Parte diario</b>			
<b>Descripción: Contiene datos de los detalles de los partes diarios que realizan los empleados</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Id_partediario	Numérico		Identificador del parte diario
Id_detalle	Numérico		Identificador del detalle
Kilometraje Inicio	Carácter	50	Kilometraje al iniciar la actividad
Kilometraje Fin	Fecha		Kilometraje al finalizar la actividad
Hora inicio	Numérico		Hora de inicio de la actividad
Hora fin	Numérico		Hora al finalizar la actividad
Descripción	Numérico		Descripción de la actividad realizada

**Tabla 20***Tabla diccionario de datos marca*

<b>Nombre: Marca</b>			
<b>Descripción: Contiene datos de las marcas de vehículos</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Nombre	Carácter	50	Nombre de la marca

**Tabla 21***Tabla diccionario de datos modelo*

<b>Nombre: Modelo</b>			
<b>Descripción: Contiene datos de los modelos de vehículos</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Nombre	Carácter	50	Nombre del modelo
Marca	Carácter	50	Nombre de la marca asociada al modelo

**Tabla 22***Tabla diccionario de datos ubicación*

<b>Nombre: Ubicación</b>			
<b>Descripción: Contiene datos que permitirán definir la ubicación actual y los trayectos recorridos por un vehículo</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Id_ubicacion	Numérico		Identificador del trayecto
Lat	Numérico		Coordenada que corresponden a la latitud en la que se encontraba el vehículo
Lng	Numérico		Coordenada que corresponden a la longitud en la que se encontraba el vehículo
Placa	Carácter	50	Placa del vehículo que fue localizado en las coordenadas respectivas
Fecha	Fecha		Fecha en la que se realizó el registro de las coordenadas
Hora	Tiempo		Hora en la que se realizó el registro de las coordenadas

**Tabla 23**

Tabla diccionario de datos empleado-proyecto

<b>Nombre: Empleado-Proyecto</b>			
<b>Descripción: Contiene relacionales de las tablas empleado y proyecto</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
CI	Carácter	50	Carnet de identidad del empleado
Id_proyecto	Numérico		Identificador del proyecto

**Tabla 24**

Tabla diccionario de datos empleado vehículo

<b>Nombre: Empleado-Vehículo</b>			
<b>Descripción: Contiene relacionales de las tablas empleado y vehículo</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
CI	Carácter	50	Carnet de identidad del empleado
Placa	Carácter	50	Placa del vehículo

**Tabla 25**

Tabla diccionario de datos proyecto vehículo

<b>Nombre: Proyecto-Vehículo</b>			
<b>Descripción: Contiene relacionales de las tablas proyecto y vehículo</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Placa	Carácter	50	Placa del vehículo
Id_proyecto	Numérico		Identificador del proyecto

**Tabla 26**

Tabla diccionario de datos departamento

<b>Nombre: Departamento</b>			
<b>Descripción: Departamentos de Bolivia para definir el lugar en donde se desarrolla un proyecto</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
Nombre_departamento	Carácter	50	Departamento de Bolivia

#### II.1.2.12.4 Script de la base de datos

```
CREATE TABLE Departamento
```

```
(  
    nombre_departamento varchar(50) NOT NULL
```

```
)
```

```
;
```

```
CREATE TABLE DetalleParteDiario
```

```
(  
    descripc_diario varchar(50) NULL,  
    hora_inicio time(6) without time zone NULL,  
    hora_fin time(6) without time zone NULL,  
    kilometraje_inicio integer NULL,  
    kilometraje_fin integer NULL,  
    id_detalle serial NOT NULL,  
    id_diario serial NOT NULL
```

```
)
```

```
;
```

```
CREATE TABLE Empleado
```

```
(  
    ci varchar(50) NOT NULL,  
    nombres varchar(50) NULL,  
    ap_paterno varchar(50) NULL,
```

```
ap_materno varchar(50) NULL,  
fec_nac varchar(50) NULL,  
celular varchar(50) NULL,  
cargo varchar(50) NULL,  
email varchar(50) NULL,  
password varchar(50) NULL,  
)  
;
```

```
CREATE TABLE EmpleadoProyecto
```

```
(  
    ci_empro varchar(50) NOT NULL,  
    id_proemp integer NOT NULL  
)  
;
```

```
CREATE TABLE Horario
```

```
(  
    horario_entrada time(6) without time zone NULL,  
    horario_salida time(6) without time zone NULL,  
    id_horario serial NOT NULL,  
    placa varchar(50) NULL  
)
```

;

```
CREATE TABLE Marca
```

```
(
```

```
    nombre_marca varchar(50) NOT NULL
```

```
)
```

;

```
CREATE TABLE Modelo
```

```
(
```

```
    nombre_modelo varchar(50) NOT NULL,
```

```
    marca varchar(50) NULL
```

```
)
```

;

```
CREATE TABLE Parte_Diario
```

```
(
```

```
    id_diario serial NOT NULL,
```

```
    placa varchar(50) NULL,
```

```
    proyecto varchar(50) NULL,
```

```
    fecha date NULL,
```

```
    combustible integer NULL,
```

```
    aceite_hidraulico integer NULL,
```

```
    aceite_motor integer NULL,  
    aceite_transmision integer NULL,  
    lugar varchar(50) NULL,  
    nombre_empleado varchar(50) NULL,  
    ci_empleado varchar(50) NULL  
)  
;
```

```
CREATE TABLE Proyecto
```

```
(  
    id_proyecto serial NOT NULL,  
    nombre_proyecto varchar(50) NULL,  
    lugar_proyecto varchar(50) NULL,  
    fec_inicio date NULL,  
    fec_fin date NULL,  
    descrip_proyecto text NULL,  
    estado_proyecto boolean NULL  
)  
;
```

```
CREATE TABLE Ubicacion
```

```
(  
    latitud double precision NULL,
```

```
    longitud double precision NULL,  
    id_ubicacion serial NOT NULL,  
    placa varchar(50) NULL,  
    fecha date NULL,  
    hora time without time zone NULL  
)  
;
```

```
CREATE TABLE Vehiculo
```

```
(  
    placa varchar(50) NOT NULL,  
    modelo varchar(50) NULL,  
    marca varchar(50) NULL,  
    kilometraje bigint NULL,  
    capacidad_combustible integer NULL,  
    estado_vehiculo boolean NULL  
)  
;
```

```
CREATE TABLE VehiculoEmpleado
```

```
(  
    ci varchar(50) NOT NULL,  
    placa varchar(50) NOT NULL
```

)

;

CREATE TABLE VehiculoProyecto

(

placa varchar(50) NOT NULL,

id\_proyecto bigint NOT NULL

)

;

*/\* Create Primary Keys, Indexes, Uniques, Checks \*/*

ALTER TABLE Departamento ADD CONSTRAINT PK\_Departamento

PRIMARY KEY (nombre\_departamento)

;

ALTER TABLE DetalleParteDiario ADD CONSTRAINT PK\_DetalleParteDiario

PRIMARY KEY (id\_detalle)

;

CREATE INDEX IXFK\_DetalleParteDiario\_Parte\_Diario ON DetalleParteDiario (id\_diario  
ASC)

;

ALTER TABLE Empleado ADD CONSTRAINT PK\_Empleado

PRIMARY KEY (ci)

;

```
ALTER TABLE EmpleadoProyecto ADD CONSTRAINT PK_EmpleadoProyecto  
    PRIMARY KEY (ci_empro,id_proemp)
```

;

```
CREATE INDEX IXFK_EmpleadoProyecto_Empleado ON EmpleadoProyecto (ci_empro ASC)
```

;

```
CREATE INDEX IXFK_EmpleadoProyecto_Proyecto ON EmpleadoProyecto (id_proemp ASC)
```

;

```
ALTER TABLE Horario ADD CONSTRAINT PK_Horario  
    PRIMARY KEY (id_horario)
```

;

```
CREATE INDEX IXFK_Horario_Vehiculo ON Horario (placa ASC)
```

;

```
ALTER TABLE Marca ADD CONSTRAINT PK_Marca  
    PRIMARY KEY (nombre_marca)
```

;

```
ALTER TABLE Modelo ADD CONSTRAINT PK_Modelo
```

```
    PRIMARY KEY (nombre_modelo)
```

```
;
```

```
CREATE INDEX IXFK_Modelo_Marca ON Modelo (marca ASC)
```

```
;
```

```
ALTER TABLE Parte_Diario ADD CONSTRAINT PK_Parte_Diario
```

```
    PRIMARY KEY (id_diario)
```

```
;
```

```
CREATE INDEX IXFK_Parte_Diario_Empleado ON Parte_Diario (ci_empleado ASC)
```

```
;
```

```
ALTER TABLE Proyecto ADD CONSTRAINT PK_Proyecto
```

```
    PRIMARY KEY (id_proyecto)
```

```
;
```

```
ALTER TABLE Ubicacion ADD CONSTRAINT PK_Ubicacion
```

```
    PRIMARY KEY (id_ubicacion)
```

```
;
```

```
CREATE INDEX IXFK_Ubicacion_Vehiculo ON Ubicacion (placa ASC)
```

;

```
ALTER TABLE Vehiculo ADD CONSTRAINT PK_Vehiculo
```

```
    PRIMARY KEY (placa)
```

;

```
CREATE INDEX IXFK_Vehiculo_Modelo ON Vehiculo (modelo ASC)
```

;

```
ALTER TABLE VehiculoEmpleado ADD CONSTRAINT PK_VehiculoEmpleado
```

```
    PRIMARY KEY (ci,placa)
```

;

```
CREATE INDEX IXFK_VehiculoEmpleado_Empleado ON VehiculoEmpleado (ci ASC)
```

;

```
CREATE INDEX IXFK_VehiculoEmpleado_Vehiculo ON VehiculoEmpleado (placa ASC)
```

;

```
ALTER TABLE VehiculoProyecto ADD CONSTRAINT PK_VehiculoProyecto
```

```
    PRIMARY KEY (placa,id_proyecto)
```

;

```
CREATE INDEX IXFK_VehiculoProyecto_Proyecto ON VehiculoProyecto (id_proyecto ASC)
```

```
;
```

```
CREATE INDEX IXFK_VehiculoProyecto_Vehiculo ON VehiculoProyecto (placa ASC)
```

```
;
```

```
/* Create Foreign Key Constraints */
```

```
ALTER TABLE Proyecto ADD CONSTRAINT FK_Proyecto_Departamento
```

```
    FOREIGN KEY (nombre_departamento) REFERENCES Departamento  
(nombre_departamento) ON DELETE No Action ON UPDATE No Action
```

```
;
```

```
ALTER TABLE DetalleParteDiario ADD CONSTRAINT FK_DetalleParteDiario_Parte_Diario
```

```
    FOREIGN KEY (id_diario) REFERENCES Parte_Diario (id_diario) ON DELETE No  
Action ON UPDATE No Action
```

```
;
```

```
ALTER TABLE EmpleadoProyecto ADD CONSTRAINT FK_EmpleadoProyecto_Empleado
```

```
    FOREIGN KEY (ci_empro) REFERENCES Empleado (ci) ON DELETE No Action ON  
UPDATE No Action
```

```
;
```

```
ALTER TABLE EmpleadoProyecto ADD CONSTRAINT FK_EmpleadoProyecto_Proyecto
```

```
    FOREIGN KEY (id_proemp) REFERENCES Proyecto (id_proyecto) ON DELETE No  
Action ON UPDATE No Action
```

;

```
ALTER TABLE Horario ADD CONSTRAINT FK_Horario_Vehiculo
```

```
    FOREIGN KEY (placa) REFERENCES Vehiculo (placa) ON DELETE No Action ON  
UPDATE No Action
```

;

```
ALTER TABLE Modelo ADD CONSTRAINT FK_Modelo_Marca
```

```
    FOREIGN KEY (marca) REFERENCES Marca (nombre_marca) ON DELETE No  
Action ON UPDATE No Action
```

;

```
ALTER TABLE Parte_Diario ADD CONSTRAINT FK_Parte_Diario_Empleado
```

```
    FOREIGN KEY (ci_empleado) REFERENCES Empleado (ci) ON DELETE No Action  
ON UPDATE No Action
```

;

```
ALTER TABLE Ubicacion ADD CONSTRAINT FK_Ubicacion_Vehiculo
```

```
    FOREIGN KEY (placa) REFERENCES Vehiculo (placa) ON DELETE No Action ON  
UPDATE No Action
```

;

```
ALTER TABLE Vehiculo ADD CONSTRAINT FK_Vehiculo_Modelo
```

```
    FOREIGN KEY (modelo) REFERENCES Modelo (nombre_modelo) ON DELETE No  
Action ON UPDATE No Action
```

;

```
ALTER TABLE VehiculoEmpleado ADD CONSTRAINT FK_VehiculoEmpleado_Empleado
    FOREIGN KEY (ci) REFERENCES Empleado (ci) ON DELETE No Action ON
UPDATE No Action
```

;

```
ALTER TABLE VehiculoEmpleado ADD CONSTRAINT FK_VehiculoEmpleado_Vehiculo
    FOREIGN KEY (placa) REFERENCES Vehiculo (placa) ON DELETE No Action ON
UPDATE No Action
```

;

```
ALTER TABLE VehiculoProyecto ADD CONSTRAINT FK_VehiculoProyecto_Proyecto
    FOREIGN KEY (id_proyecto) REFERENCES Proyecto (id_proyecto) ON DELETE No
Action ON UPDATE No Action
```

;

```
ALTER TABLE VehiculoProyecto ADD CONSTRAINT FK_VehiculoProyecto_Vehiculo
    FOREIGN KEY (placa) REFERENCES Vehiculo (placa) ON DELETE No Action ON
UPDATE No Action
```

;

### **II.1.2.13      Prototipos de interfaces de usuario**

Prototipos de las pantallas del sistema que plasman las funcionalidades descritas en los casos de uso, los prototipos serán de referencia para el desarrollo del sistema.

El desarrollo del sistema no está sujeto a los prototipos, es decir que la versión estable que será presentada del sistema no será igual a estos prototipos, algunos serán mejorados y otros incluso serán desechados o no tomados en cuenta.

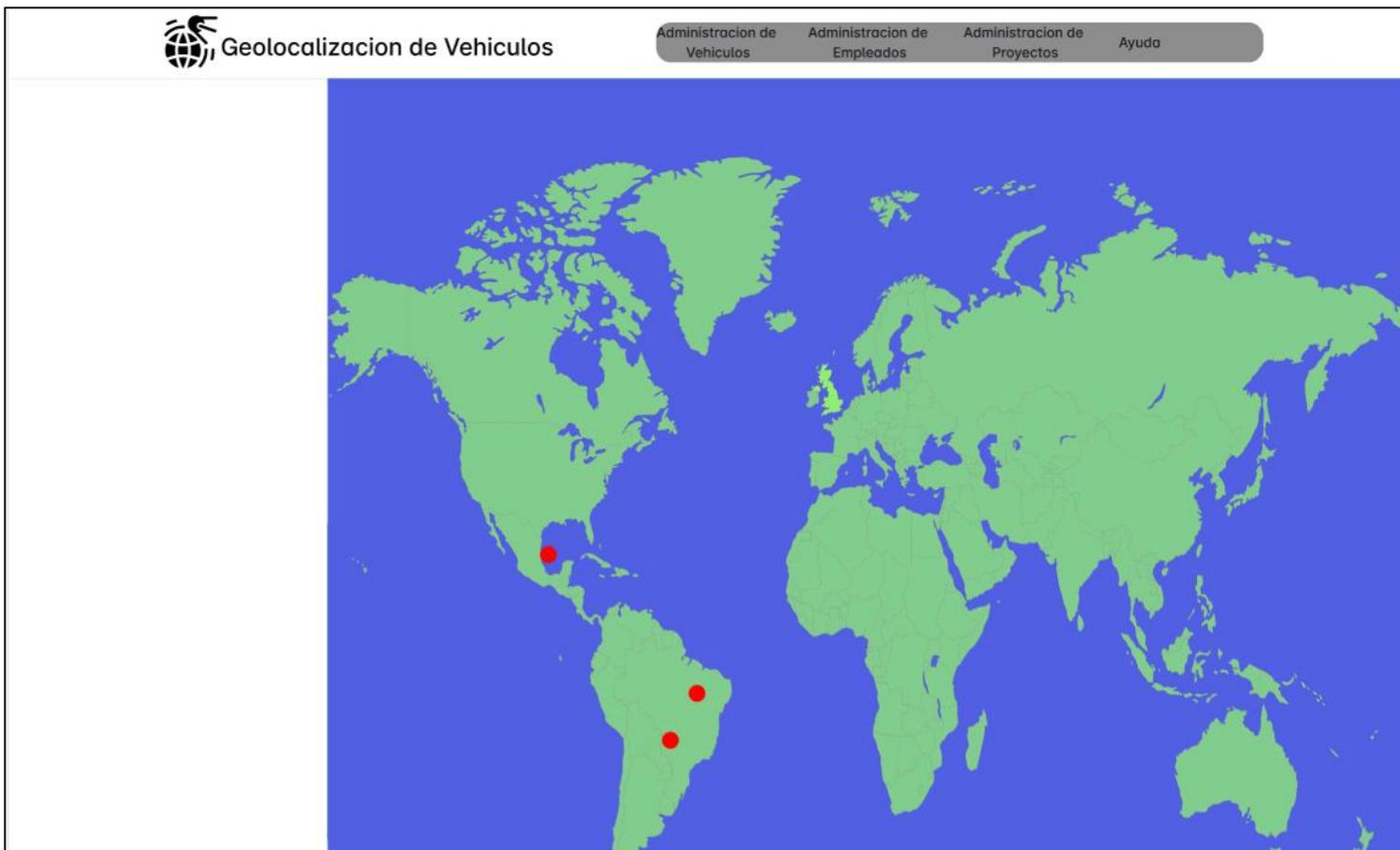
**Figura 60**

*Prototipo interfaz Inicio de Sesión*

El prototipo de interfaz de inicio de sesión para GEOBRIS presenta un diseño limpio y moderno. El fondo es un color gris claro con un elemento decorativo en la parte inferior que combina un arco de color verde oscuro y un triángulo de color verde brillante. El título "GEOBRIS" está centrado en la parte superior en una tipografía sans-serif azul oscura. Debajo del título, se encuentran dos campos de entrada de texto con un fondo azul oscuro y texto blanco: "Usuario" y "Contraseña". A continuación, hay un campo de selección con un pequeño cuadrado azul oscuro a la izquierda y el texto "Recordar". Finalmente, un botón de "Iniciar Sesión" con un fondo verde brillante y texto blanco está centrado al final del formulario.

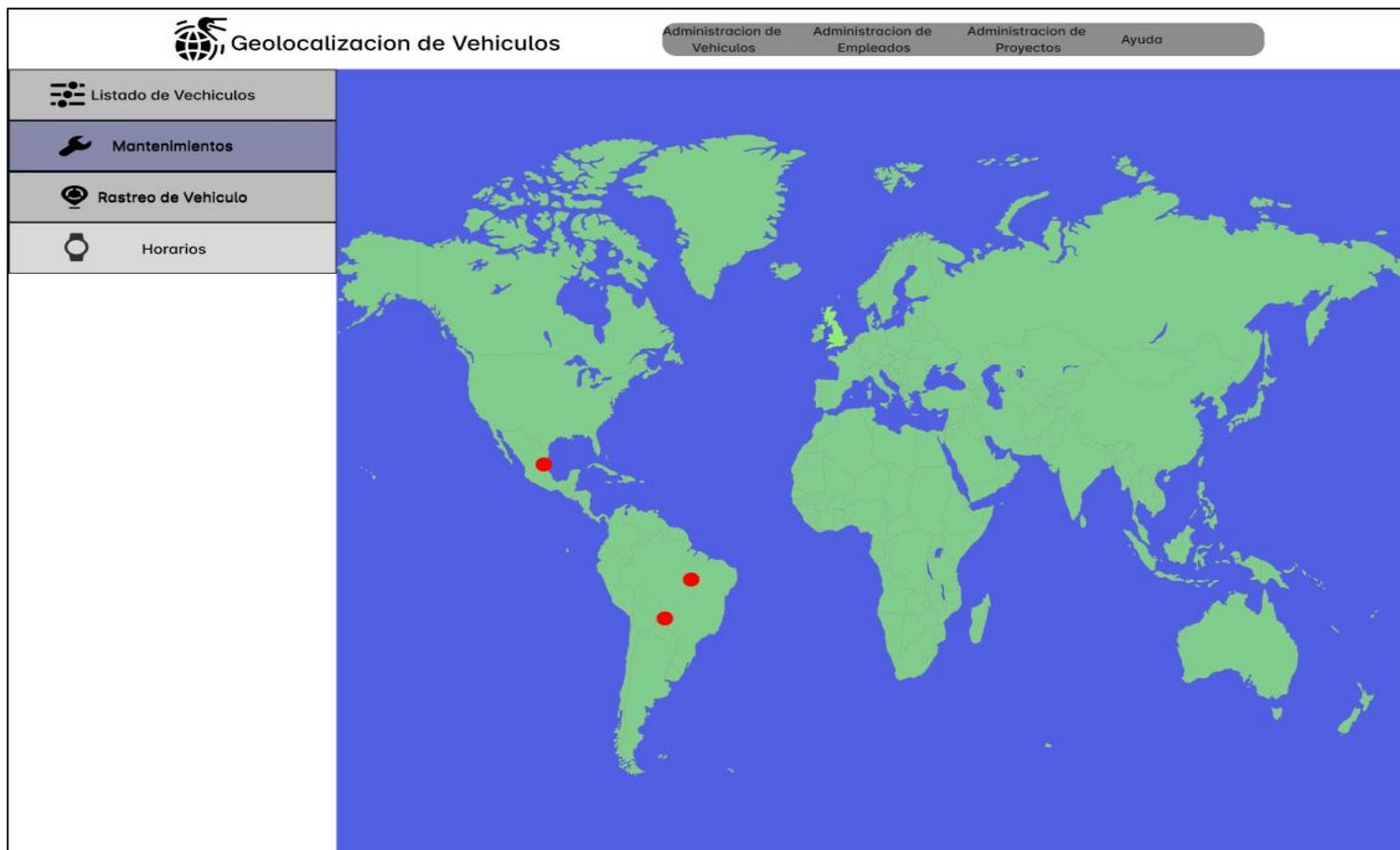
**Figura 61**

*Prototipo interfaz Página principal del sistema*



**Figura 62**

*Prototipo interfaz Administración de vehículos*



**Figura 63***Prototipo interfaz Listado de vehículos*

Geolocalizacion de Vehiculos

Administracion de Vehiculos | Administracion de Empleados | Administracion de Proyectos | Ayuda

Listado de Vehiculos

Mantenimientos

Rastreo de Vehiculo

Horarios

+ Añadir Vehiculo

PLACA	MARCA	DESCRIPCION	CHOFER	PROYECTO	ESTADO
1595-FSA	Nissan Condor	Tolva 2 cubos	Juan Martinez	Sistema de Riego Saladillo	ACTIVO
5555-PLP	Finning Cat	Retroexcavadora	Luis Perez	Sistema de Riego Saladillo	ACTIVO
3215-MTS	Toyota	Camioneta	Pedro Lopez	Ninguno	INACTIVO

**Figura 64**

*Prototipo interfaz Añadir vehículo*

Geolocalizacion de Vehiculos

Administracion de Vehiculos | Administracion de Empleados | Administracion de Proyectos | Ayuda

Listado de Vehiculos

Mantenimientos

Rastreo de Vehiculo

Horarios

### Añadir Vehiculo

Numero de Placa

Modelo

Marca

Tipo

Chofer

Proyecto

Horario

**GUARDAR**

**CANCELAR**

**Figura 65**

*Prototipo interfaz Modificar vehículo*

Geolocalizacion de Vehiculos

Administracion de Vehiculos | Administracion de Empleados | Administracion de Proyectos | Ayuda

Listado de Vehiculos

Mantenimientos

Rastreo de Vehiculo

Horarios

### Modificar Vehiculo

Numero de Placa: 3215-MTS

Modelo: Hilux

Marca: Toyota

Tipo: Camioneta

Chofer: Pedro Lopez

Proyecto:

Horario: 7:00 - 19:00

**GUARDAR**

**CANCELAR**

**Figura 66**

*Prototipo interfaz Eliminar vehículo*

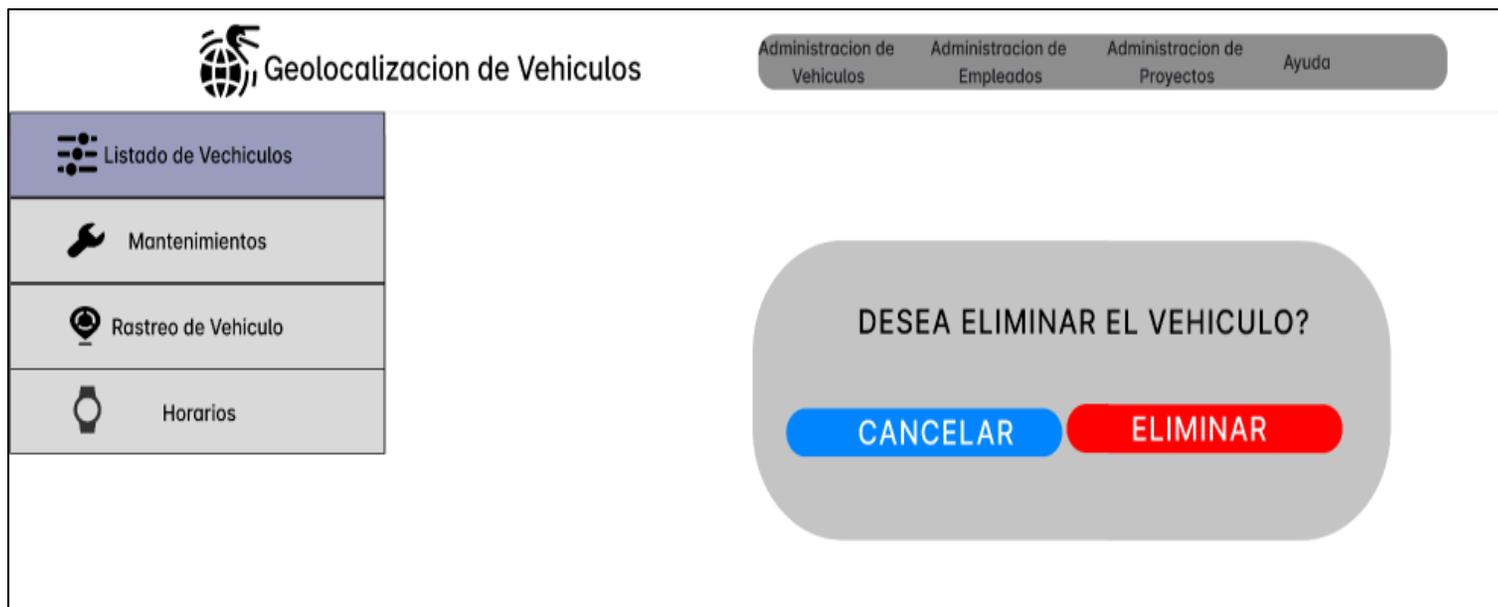


Figura 67

Prototipo interfaz Ubicación geográfica

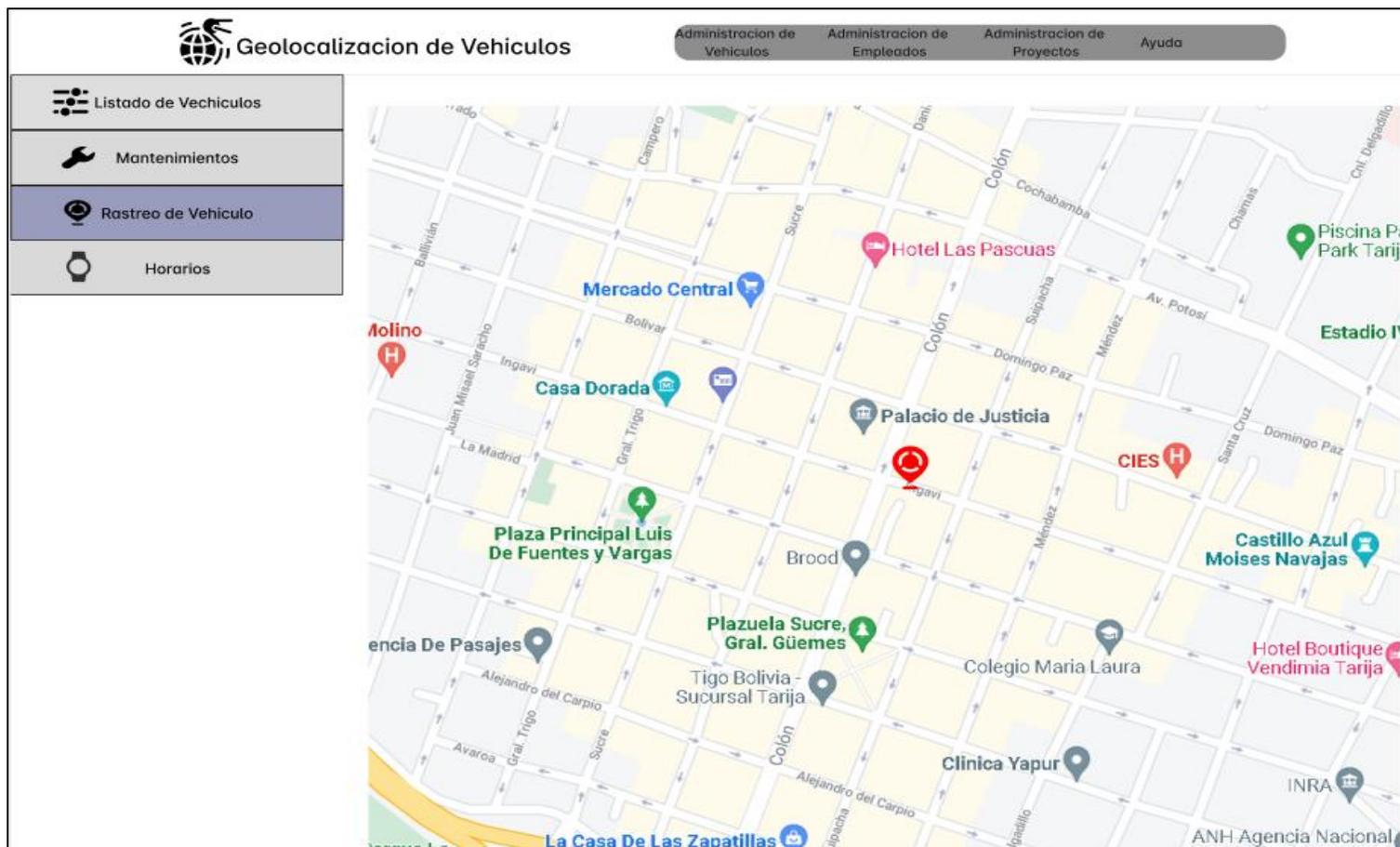


Figura 68

Prototipo interfaz Trayecto

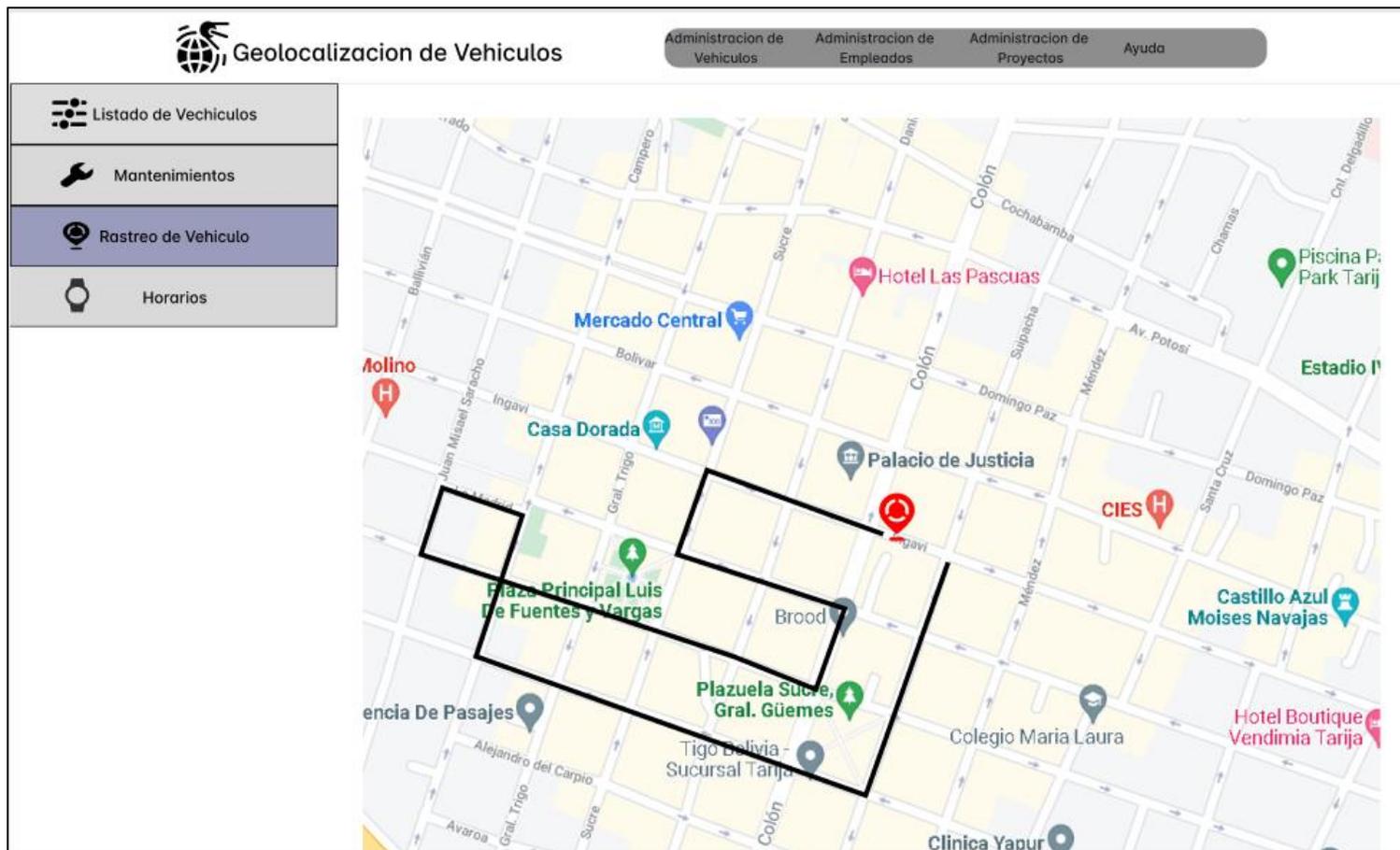


Figura 69

*Prototipo interfaz Horarios*

Geocalizacion de Vehiculos

Administracion de Vehiculos    Administracion de Empleados    Administracion de Proyectos    Ayuda

Listado de Vehiculos

Mantenimientos

Rastreo de Vehiculo

Horarios

+ Añadir Horario

NOMBRE	HORARIO ENTRADA	HORARIO SALIDA		
Horario Camionetas	7:00	18:00		
Horario Volquetas	7:00	18:00		
Horario Vagonetas	8:00	20:00		

**Figura 70**

*Prototipo interfaz Añadir horario*

Geolocalización de Vehículos

Administración de Vehículos   Administración de Empleados   Administración de Proyectos   Ayuda

Listado de Vehículos

Mantenimientos

Rastreo de Vehículo

Horarios

**Añadir Horario**

Nombre

Hora de Entrada

Hora de Salida

**GUARDAR**

**CANCELAR**

Detailed description: The image shows a software prototype for a vehicle management system. At the top left is a logo with a globe and a hand, followed by the text 'Geolocalización de Vehículos'. To the right of this is a navigation bar with four items: 'Administración de Vehículos', 'Administración de Empleados', 'Administración de Proyectos', and 'Ayuda'. Below the navigation bar is a vertical sidebar menu with four options: 'Listado de Vehículos', 'Mantenimientos', 'Rastreo de Vehículo', and 'Horarios'. The 'Horarios' option is highlighted with a blue background. The main content area on the right features a form titled 'Añadir Horario'. This form contains three input fields: 'Nombre', 'Hora de Entrada', and 'Hora de Salida'. Below the input fields are two buttons: a blue 'GUARDAR' button and a red 'CANCELAR' button.

**Figura 71**

*Prototipo interfaz Modificar horario*

The image shows a web application interface for 'Geolocalizacion de Vehiculos'. At the top, there is a navigation bar with the title 'Geolocalizacion de Vehiculos' and a logo of a hand holding a globe. To the right of the title are four menu items: 'Administracion de Vehiculos', 'Administracion de Empleados', 'Administracion de Proyectos', and 'Ayuda'. Below the navigation bar is a sidebar with four menu items: 'Listado de Vehiculos', 'Mantenimientos', 'Rastreo de Vehiculo', and 'Horarios'. The 'Horarios' menu item is highlighted. The main content area displays a 'Modificar Horario' form. The form has a title 'Modificar Horario' and three input fields: 'Nombre' with the value 'Horario Vaganetas', 'Hora de Entrada' with the value '7:00', and 'Hora de Salida' with the value '19:00'. Below the form are two buttons: a blue 'GUARDAR' button and a red 'CANCELAR' button.

Geolocalizacion de Vehiculos

Administracion de Vehiculos    Administracion de Empleados    Administracion de Proyectos    Ayuda

Listado de Vehiculos

Mantenimientos

Rastreo de Vehiculo

Horarios

Modificar Horario

Nombre  
Horario Vaganetas

Hora de Entrada  
7:00

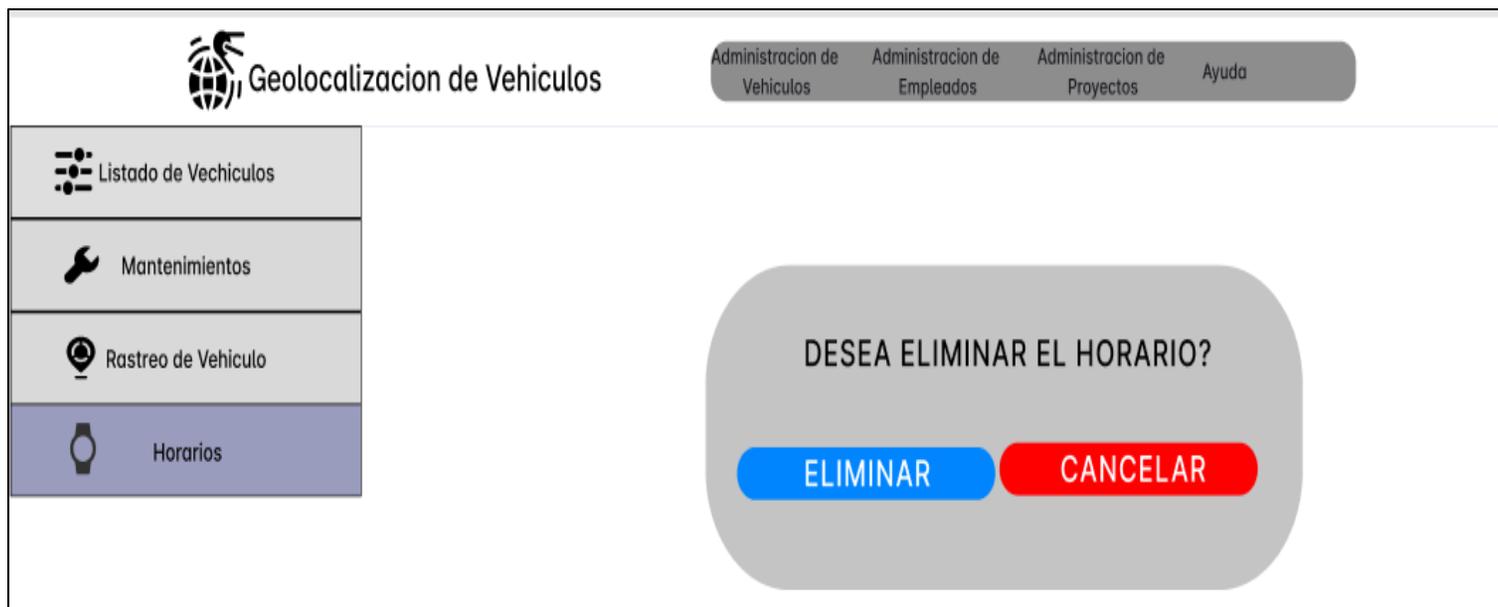
Hora de Salida  
19:00

GUARDAR

CANCELAR

**Figura 72**

*Prototipo interfaz Eliminar horario*



**Figura 73***Prototipo interfaz Listado de empleados*

Geocalización de Vehículos

Administración de Vehículos   Administración de Empleados   Administración de Proyectos   Ayuda

Listado de Empleados

+ Añadir Empleado

CI	NOMBRE	APELLIDOS	CARGO	PROYECTO	ESTADO	
1277777	Juan	Martinez	Ingeniero Civil	Sistema de Riego Saladillo	ACTIVO	
1277588	Luis	Perez	Chofer	Sistema de Riego Saladillo	ACTIVO	
1299555	Pedro	Lopez	Contador	Ninguno	INACTIVO	

**Figura 74**

*Prototipo interfaz Añadir empleado*

Geolocalizacion de Vehiculos

Administracion de Vehiculos | Administracion de Empleados | Administracion de Proyectos | Ayuda

Listado de Empleados

**Añadir Empleado**

Nombre

Apellidos

Carnet de Identidad

Fecha de Nacimiento

Cargo Laboral

**GUARDAR**

**CANCELAR**

**Figura 75**

*Prototipo interfaz Modificar empleado*

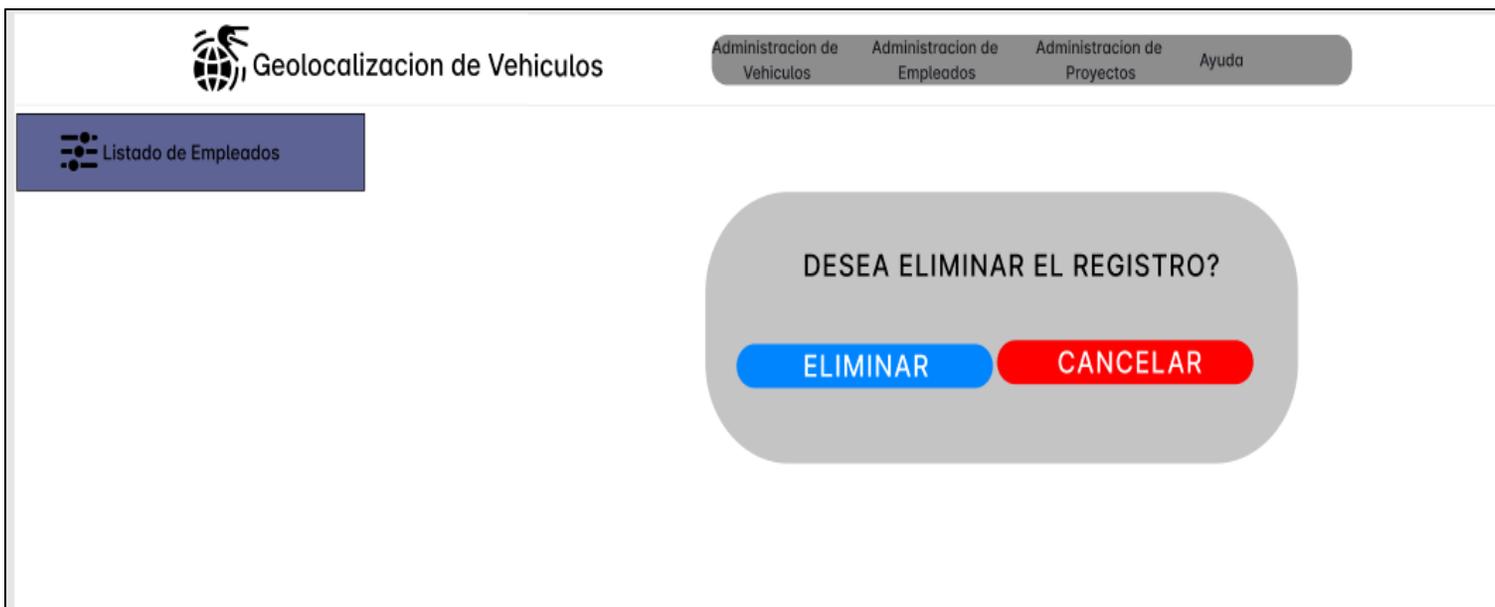
The image shows a web application interface for 'Geolocalizacion de Vehiculos'. The interface includes a navigation menu with 'Listado de Empleados' selected. A 'Modificar Empleado' form is displayed with the following fields and values:

Field	Value
Nombre	Juan
Apellidos	Martinez
Carnet de Identidad	12777777
Fecha de Nacimiento	11-09-1975
Cargo Laboral	Supervisor

Below the form are two buttons: 'GUARDAR' (blue) and 'CANCELAR' (red).

**Figura 76**

*Prototipo interfaz Eliminar empleado*



**Figura 77***Prototipo interfaz Listado Proyectos*

Geolocalizacion de Vehiculos

Administracion de Vehiculos | Administracion de Empleados | Administracion de Proyectos | Ayuda

Listado de Proyectos

+ Añadir Proyecto

ID	NOMBRE	ENCARGADO	UBICACION	DURACION	ESTADO
1	Sistema de Riego Saladillo	Juan Martinez	Uriondo-Tarija	1 años	ACTIVO
2	Agua Potable	Daniel Rodriguez	Tacmari-Potosi	2 años	ACTIVO
3	Centro de Salud	Rodrigo Almazan	Santa Ana-Tarija	1 año	FINALIZADO

**Figura 78**

*Prototipo interfaz Añadir proyecto*

Geolocalización de Vehículos

Administración de Vehículos | Administración de Empleados | Administración de Proyectos | Ayuda

Listado de Proyectos

### Añadir Proyecto

Nombre

Ubicación

Fecha Inicio

Fecha Fin

Descripción

**GUARDAR**

**CANCELAR**

**Figura 79**

Prototipo interfaz Modificar proyecto

Geolocalizacion de Vehiculos

Administracion de Vehiculos | Administracion de Empleados | Administracion de Proyectos | Ayuda

Listado de Proyectos

### Modificar Proyecto

Nombre

Ubicacion

Fecha Inicio

Fecha Fin

Descripcion

**GUARDAR**

**CANCELAR**

#### **II.1.2.14 Manual de usuario e instalación**

Este documento explica el uso correcto del sistema y la instalación del mismo.

Debido a la extensión de este documento ha sido ubicado en los Anexos.

#### **II.1.3 Medios de verificación**

Informe de entrega final del sistema informático, avalado por el docente de Taller III:

### **CERTIFICADO**

Tarija. 21 de noviembre del 2022

El que suscribe, Lic. Efraín Torrejón Tejerina, docente de la materia DIS501 Taller III, grupo 1, habiendo realizado la revisión del sistema y documentación correspondiente al proyecto “Mejorar el control y administración de la empresa Consultora y Constructora Bristlecone SRL a través de las TIC”, dirigido por el Univ. Saúl Ancasi Zegarra, CERTIFICA que ha cumplido con los requerimientos del sistema expresados bajo la norma IEEE-830.

**Lic. Efraín Torrejón Tejerina**

**DOCENTE DIS501 TALLER III GRUPO 1**

## **II.2. Componente 2: Capacitación**

### **II.2.1 Introducción**

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

El propósito del proyecto es: “Mejorar el control y administración de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L.”; la capacitación en el uso del sistema informático al personal afectado por el proyecto se convierte en un componente fundamental para el logro del mismo.

El componente capacitación, se encamina hacia el siguiente objetivo: usar adecuadamente el sistema informático “GEOBRIS” por el personal de la empresa Consultora y Constructora Bristlecone S.R.L. explotando las fortalezas del mismo.

La Capacitación será presencial dada la corta duración de la misma, la disponibilidad de ambientes, de materiales didácticos y la importancia de posibilitar que el alumno (usuario) reciba asesoramiento oportuno ante cualquier consulta.

### **II.2.2 Contexto**

La Capacitación se desarrollará en dos partes: la primera parte tiene como objetivo que el personal a capacitar conozca en forma global los alcances y beneficios que el sistema informático “GEOBRIS” aporta a la empresa Consultora y Constructora Bristlecone S.R.L. así como los cambios positivos y responsabilidades que esto implica para la empresa.

Se realizarán actividades de capacitación personalizadas de acuerdo al cargo que a cada uno le compete.

En este contexto el capacitador confeccionó la capacitación tomando en cuenta los diferentes niveles de preparación del usuario final.

El rol del capacitador estará en función a las categorías de los usuarios según el siguiente detalle:

Nivel ejecutivo:

- Se mostrará la importancia de la capacitación, objetivos y participación del personal seleccionado.

Personal Técnico:

Se realizarán actividades de capacitación acorde a las siguientes categorías:

- Personal de soporte técnico al usuario final (si corresponde)

Usuarios Finales:

- Administrador
- Empleados

### **II.2.3 Propuesta pedagógica**

La propuesta pedagógica a utilizar dada las características de los usuarios del sistema “GEOBRIS”, tendrá en cuenta sus particularidades, el rol que juega dentro de la organización y niveles de conocimiento.

Los métodos de enseñanza a utilizar pondrán su énfasis principalmente en tres teorías de aprendizajes: la cognitiva, con su máximo exponente en el constructivismo, la colaborativa, fundamentalmente para ser explotada con intensidad en la formación del personal técnico y finalmente la significativa, aunque también estará presente en la formación del personal de las categorías de nivel ejecutivo y de usuarios finales.

Finalmente se pone de manifiesto el aprendizaje significativo porque el alumno tiene que incorporar los nuevos conocimientos en forma sustantiva en su estructura cognitiva. Esto se logra cuando el alumno relaciona los nuevos conocimientos con los anteriormente adquiridos; pero también es necesario que el alumno se interese por aprender lo que se le está mostrando. De esta forma el alumno no solo obtendrá resultados satisfactorios en un trabajo final, sino que será capaz de enfrentarse a diversas situaciones donde podrá aplicar los conocimientos adquiridos.

## **II.2.4 Contenido de la capacitación**

### Lección 1: Ingresar al sistema

- Crear cuenta
- Inicio de sesión

### Lección 2: Administración de vehículos

- Listado de vehículos
- Añadir vehículos
- Modificar vehículos
- Eliminar vehículos
- Asignar conductor
- Añadir horario
- Modificar horario
- Eliminar horario
- Ubicación en tiempo real de los vehículos
- Trayecto recorrido por los vehículos

### Lección 3: Administración de proyectos

- Listado de proyectos
- Añadir proyectos
- Modificar proyectos

- Asignar vehículos
- Asignar empleados

#### Lección 4: Administración de empleados

- Listado de empleados
- Eliminar empleados

#### Lección 5: Administración de partes diarios

- Listado partes diarios
- Añadir partes diarios
- Añadir actividades
- Eliminar actividades

#### Lección 6: Reportes

- Reporte vehículos
- Reporte proyectos
- Reportes empleados
- Reporte partes diarios

## II.2.5 Plan de clases

**Tabla 27**

*Plan de clases capacitación*

<b>Nro.</b>	<b>Contenido</b>	<b>Objetivo</b>	<b>Fecha</b>	<b>Duración (minutos)</b>	<b>Medios de enseñanza - aprendizaje</b>	<b>Destinatario</b>
1	Lección 1: Ingreso al sistema	Que el usuario se identifique y pueda entrar al sistema de acuerdo al cargo que ocupa.	19/11/2022	10	Computadora	Gerente general o administrador, empleados
2	Lección 2: Administración de vehículos	Que el usuario pueda administrar y controlar los vehículos mediante las distintas funcionalidades que presenta el sistema tales como: la ubicación en tiempo real, trayecto recorrido, asignación de conductores, etc.	19/11/2022	10	Computadora	Gerente general o administrador
3	Lección 3: Administración	Que el usuario pueda administrar los proyectos de	19/11/2022	10	Computadora	Gerente general o

	de proyectos	la empresa, mediante las funcionalidades del sistema como ser: la asignación de los vehículos y los empleados a cualquier proyecto				administrador
4	Lección 4: Administración de empleados	Que el usuario pueda administrar a los empleados.	19/11/2022	10	Computadora	Gerente general o administrador
5	Lección 5: Administración de partes diarios	Teniendo en cuenta el cargo, que el gerente general pueda administrar los partes diarios de los empleados o los empleados pueden realizar sus partes diarios.	19/11/2022	10	Computadora	Gerente general o administrador, empleados
6	Lección 6: Reportes	Que el usuario pueda obtener reportes de la información recabada por el sistema.	19/11/2022	5	Computadora	Gerente general o administrador

## II.2.6 Cronograma

**Tabla 28**

*Cronograma de actividades capacitación del sistema*

Actividad	Duración (Minutos)	19 de noviembre 2022					
		08:00	08:10	08:20	08:30	08:40	08:50
Lección 1	5						
Lección 2	10						
Lección 3	10						
Lección 4	10						
Lección 5	10						
Lección 6	5						

## II.2.7 Resultados esperados

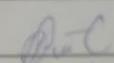
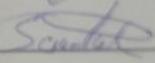
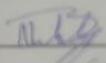
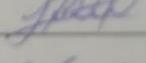
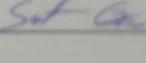
Interés por parte del personal de la empresa en la capacitación para el uso del sistema informático.

## II.2.8 Medios de verificación

### II.2.8.1 Lista de asistencia a la capacitación

#### Figura 80

*Lista de asistencia a la capacitación*

Lista de Asistencia Capacitación Software GeoBris		
Lugar: Consultora y Constructora Bristlecone SRL		
Lista de asistencia al personal de la empresa Consultora y Constructora Bristlecone SRL		
Nro	Nombres y apellidos	Firma
1	Romario Ovando Sanchez	
2	Santos Lopez Cordasi	
3	Niver Cardozo Narvaes	
4	Alex Caro Soruco	
5	Erick Jairo	
6	José A Lopez Guz	
7	Siles Ayarde Soruco	
8		
9		

Informe de la gerencia de la empresa Consultora y Constructora Bristlecone S.R.L. respecto a la capacitación realizada:

**Tarija, 15 de abril de 2023**

Señor:

**Lic. Efraín Torrejón Tejerina**

**Director del Departamento de la carrera de Ingeniería Informática**

Presente:

**Ref.: Capacitación de personal, uso del sistema web GEOBRIS**

A través de la presente, le informo que el UNIV. Saúl Ancasi Zegarra, realizó de manera responsable y satisfactoria la capacitación de nuestro personal para hacer uso del sistema web GEOBRIS.

Es cuanto certifico en honor a la verdad para fines que pudieran serle útil al interesado

Agradeciendo de antemano su atención me despido

Atentamente:

---

**Ing. Sergio Maldonado Duchén**

**REPRESENTANTE LEGAL**

**CONSULTORA Y CONSTRUCTORA BRISTLECONE S.R.L.**

## **II.2.9 Conclusiones**

Finalizada la capacitación, el personal de la empresa Consultora y Constructora Bristlecone S.R.L. expreso su conformidad y opinión respecto al sistema “GEOBRIS”, además de sugerir cambios y mejoras al sistema, todas las sugerencias serán analizadas y tomadas en cuenta para posteriores versiones del sistema “GEOBRIS”.

## **Capítulo III: Conclusiones y Recomendaciones**

### **III.1 CONCLUSIONES Y RECOMENDACIONES**

#### **III.1.1 Conclusiones**

Finalizado el proyecto que consiste en un sistema web GEOBRIS para el control y administración de los vehículos de la empresa Consultora y Constructora Bristlecone S.R.L. se cumplió con el propósito planteado, llegando a automatizar parte de los procesos que tienen que ver con administración de los vehículos de la empresa brindando facilidad al personal.

Mencionar que este es el primer sistema que usará la empresa Consultora y Constructora Bristlecone S.R.L. desarrollado de acuerdo a sus requerimientos, por tanto, esto puede llegar a significar un antes y un después para la empresa en cuanto a crecimiento e innovación tecnológica.

Teniendo en cuenta lo mencionado de igual manera se concluye que:

- A través de la metodología RUP se pudo realizar un análisis y diseño del sistema web de acuerdo a los requerimientos de la empresa Consultora y Constructora Bristlecone S.R.L.
- El uso del Lenguaje UML permitió y facilitó el diseño estructural del sistema permitiendo una mejor comprensión e interpretación del sistema.
- Se logró desarrollar una versión beta del sistema web GEOBRIS, pero no una definitiva, por lo cual el sistema web se verá expuesto a distintas correcciones y mejoras con el pasar de los años.
- El sistema GEOBRIS brinda facilidad en distintos procesos administrativos relacionados a los vehículos que competen a la empresa Consultora y Constructora Bristlecone S.R.L.

- Al tener un sistema donde parte del backend está configurado en NodeJS y el frontend en el framework React, a futuro se tendrá facilidad para el desarrollo e implementación en otras plataformas como los dispositivos móviles o aplicaciones de escritorio.
- Para el desarrollo del sistema se implementó PERN (PostgreSQL, Express, React, NodeJS) Stack que permitió el desarrollo del sistema web facilitando la incorporación de distintas librerías.
- El uso del API de Google Maps facilitó de cierta manera el desarrollo de los distintos módulos que requerían funciones relacionadas a la geolocalización.
- Con el sistema se dará cierta facilidad para realizar la documentación de los partes diarios de los empleados de la empresa, con esto se espera que el desarrollo de los proyectos.
- Los reportes que pueden ser generados con el sistema brindan al personal administrativo documentación de respaldo que será usada de acuerdo a la necesidad de la empresa.
- El uso del sistema logrará de cierta manera reducir el uso de recursos materiales, que vendrían a ser las planillas de partes diarios, debido a que ya no se realizarán de forma manual sino digital.
- Durante la capacitación del personal hubo muy pocas dudas respecto al funcionamiento del sistema y la manera de usarlo, dando a entender que es un sistema intuitivo.

### III.1.2 Recomendaciones

Tras haber finalizado el proyecto se proponen las siguientes recomendaciones:

- Cada cierto tiempo presentar avances del sistema tanto al tutor de la materia como a la institución que requiere el sistema.
- Realizar un investigación y análisis previo de las tecnologías existentes para el desarrollo de software, con el fin de usar las que mejor se adecúen a nuestra forma de trabajo.
- En lo posible realizar copias de seguridad de la base de datos a modo de prevenir problemas durante el desarrollo del sistema.
- Es preferible no usar librerías que no se exploten al máximo, debido a que estas pueden llegar a ralentizar el despliegue del sistema.
- Se recomienda uso Firebase Authentication si se requiere, debido a que facilitó el desarrollo de la autenticación de usuarios.
- Se recomienda analizar que lenguaje usar para el desarrollo con el framework React Js, puede ser JavaScript o TypeScript.
- Se recomienda el uso de Nodemon si se llega a usar un servidor en NodeJs, Nodemon nos facilita el reinicio del servidor cuando se hacen modificaciones en el sistema.
- Se recomienda dedicarle tiempo al diseño de la base de datos y que esta cumpla con la normalización de bases de datos.
- Se recomienda optimizar y mejorar el sistema, para que no quede obsoleto en poco tiempo.
- Cuando se requiera alguna mejora o modificación del sistema se recomienda leer la documentación y proceder de acuerdo a la metodología empleada en este proyecto.

- Se recomienda tomar y mantener todas las condiciones tecnológicas para el funcionamiento del sistema.
- Para el desarrollo de futuros proyectos se recomienda optar por el desarrollo de sistemas multiplataforma.
- En lo posible usar librerías que sigan teniendo soporte.
- Se recomienda el cuidado de sus equipos ante posibles cortes o bajones de energía eléctrica.