

# **CAPÍTULO I**

## **EL PROYECTO**

## **Capítulo I El Proyecto**

### **1.1 Presentación del Proyecto**

#### **1.1.1 Título del Proyecto**

Mejoramiento en el control y administración de la información en la cooperativa de transporte Río Guadalquivir.

#### **1.1.2 Carrera y/o Unidad**

Ingeniería Informática.

#### **1.1.3 Facultad**

Ciencias y Tecnología.

#### **1.1.4 Duración del Proyecto**

La duración del proyecto es de 8 meses.

#### **1.1.5 Área y/o Línea de Investigación Priorizado**

Tecnologías de la Información y Comunicación / Desarrollo de Sistemas y Software / Investigación.

#### **1.1.6 Responsable del Proyecto**

Carrera de Ingeniería Informática – Taller III.

## **1.2 Personal Vinculado al Proyecto**

### **1.2.1 Director del Proyecto**

<b>Apellido Paterno:</b> Veliz	<b>Apellido Materno:</b> Cardozo	<b>Nombre:</b> Luis Miguel	<b>C.I.:</b> 7509740 chu.
<b>Carrera:</b> Ingeniería Informática		<b>Facultad:</b> Facultad de Ciencias y Tecnología	

Teléfono:	Celular:	Correo Electrónico:	Firma:
78674318	78674318	luisveliz32@gmail.com	.....

Tabla 1 Director del Proyecto

**1.2.2 Participantes del Equipo de Trabajo**

Categoría	Nombres y Apellidos	Carrera/Profesión	C.I.	Firma
Director	Luis Miguel Veliz Cardozo	Ingeniería Informática	7509740	
Asesor		ing. en Ingeniería Informática	7127541	
Asesor				

Tabla 2 Participantes del Equipo de Trabajo

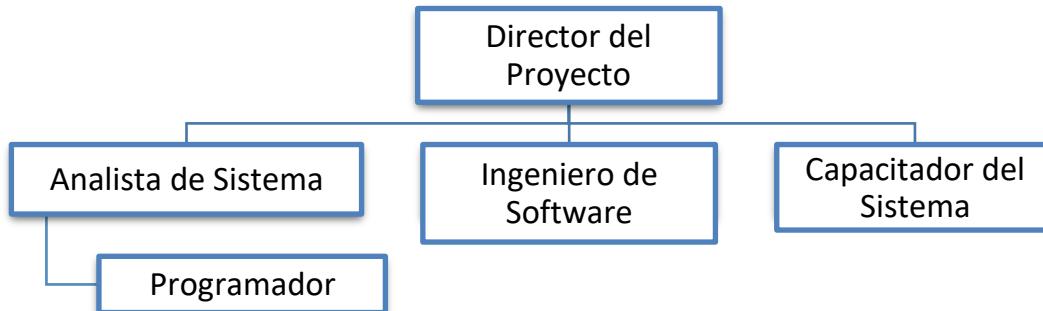
**1.2.3 Actividades Previstas para los Integrantes del Equipo de Investigación**

Responsable	Actividades
Director	<p><b>Jefe de Proyecto:</b></p> <p>Definir los objetivos del proyecto.</p> <p>Planificación del proyecto en todos sus aspectos, identificando las actividades a realizar.</p> <p>Establecer métodos, técnicas y herramientas por el equipo del</p>

	<p>proyecto.</p> <p>Supervisar el diseño, desarrollo, instalación y posterior mantenimiento de la solución.</p> <p>Presentación final del proyecto al cliente final.</p> <p><b>Analista de Sistema:</b></p> <p>Recopilación de la información requerida para la elaboración del Sistema mediante entrevistas, cuestionarios y formularios, etc.</p> <p>Definir las necesidades del usuario.</p> <p>Realizar el Análisis y Diseño del Sistema.</p> <p><b>Programador:</b></p> <p>Realizar la programación del Sistema Informático.</p> <p>Construcción de prototipos.</p> <p><b>Ingeniero de Software:</b></p> <p>Elaborar las pruebas funcionales del Sistema Informáticos.</p> <p>Verificar que el Sistema desarrollado cumpla con la legislación en vigor.</p>
<b>Asesor</b>	<p>Revisión del documento del proyecto.</p> <p>Indicar las diversas alternativas tecnologías a poder utilizar.</p> <p>Prepara al aspirante para la defensa final.</p>

Tabla 3 Actividades Previstas para los Integrantes del Equipo de Investigación

### 1.2.3.1 Unidades de Gestión Organigrama del Equipo del Proyecto



## 1.3 Descripción del Proyecto

### 1.3.1 Descripción, Fundamentación y Justificación del Proyecto

La cooperativa de transporte “Rio Guadalquivir” es una cooperativa que se dedica al transporte pesado (transporte de carga pesada para camiones) presenta la información de forma desordenada (en papeles, documentos Excel, documentos Word); Para ello se propone el proyecto que consiste en “**MEJORAMIENTO EN EL CONTROL Y ADMINISTRACIÓN DE LA INFORMACIÓN EN LA COOPERATIVA DE TRANSPORTE RÍO GUADALQUIVIR**” lo cual busca mantener de forma ordenada y segura a información de dicha cooperativa.

El proyecto cuenta con dos componentes:

Desarrollo del Sistema web y aplicación móvil para el control y administración de la información de la cooperativa de transporte Rio Guadalquivir

La realización de estos componentes permitirá a la cooperativa río Guadalquivir llevar un mejor control y administración de la información que se genera en la cooperativa de transporte Río Guadalquivir.

En la actualidad la cooperativa río Guadalquivir no cuenta con un sistema ya que dificulta el funcionamiento ocasionando problemas los cuales son:

Al registrar a un socio se hace de forma insegura ya que los datos están almacenados en papeles y de esta manera no es la forma más segura de guardar la información.

Es complicado al formar un reporte ya que se tarda mucho buscando los datos de los socios en los papeles o documentos Word, Excel.

La asignación de carga es de manera manual ya que un socio tiene que ir personalmente a la oficina de la cooperativa a averiguar si le toca su turno para cargar esto sucede cada día.

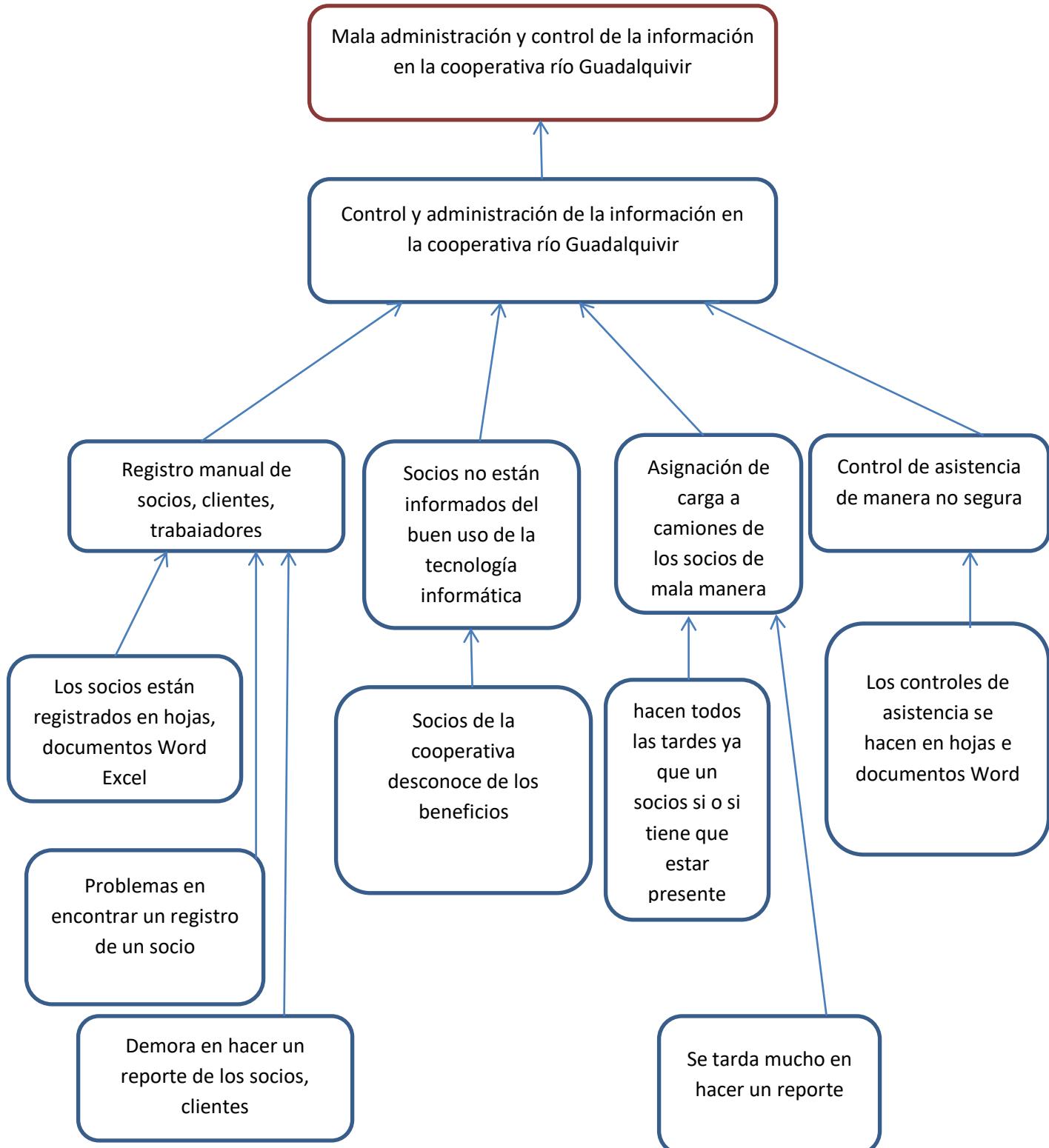
Las asignaciones de carga para los camiones pesados se hacen todas las tardes ya que un socio si o si tiene que estar presente de lo contrario no le tocaría carga.

Al hacer un historial de asignación de carga es complicado ya que los datos están en papeles y en documentos Word, Excel.

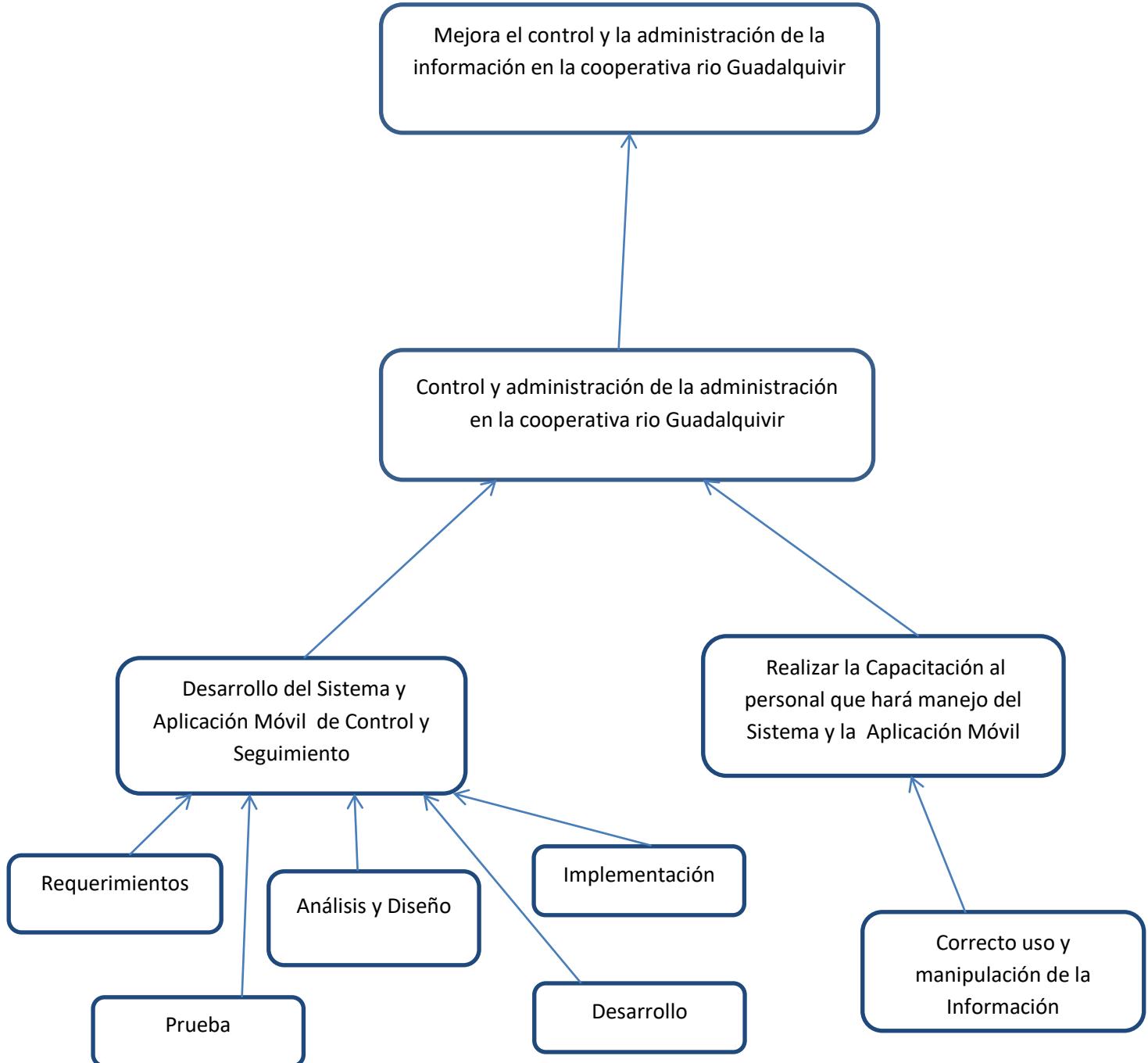
El control de asistencia en las reuniones no se hace de manera factible ya que guardada en papeles documentos Word.

No se cuenta con información detallada de los socios, trabajadores, clientes, listas de reuniones.

### 1.3.1.1 Análisis de Causas de Problemas



### 1.3.1.2 Análisis de Objetivos



### **1.3.2 Objetivos**

#### **1.3.2.1 Objetivo General**

Mejoramiento en el control y administración de la información en la cooperativa de transporte río Guadalquivir.

#### **1.3.2.2 Objetivos Específicos**

- Desarrollar un Sistema de control y administración de la información en la cooperativa de transporte río Guadalquivir .
- Desarrollar una aplicación móvil de control y administración de la información en la cooperativa de transporte río Guadalquivir.
- implementar una técnica de capacitación del Sistema y aplicación desarrollado.

#### **1.3.2.3 Limitaciones**

El sistema no contará con los siguientes componentes:

- No contará con el componente de planillas de sueldos a los trabajadores.
- No contará en el componente de contabilidad.
- No contará con el componente de facturación .

### **1.3.3 Metodología**

La forma de trabajo de este proyecto estará basada en:

#### **1.3.3.1 Metodología para el desarrollo de aplicaciones**

El software será desarrollado con la metodología RUP, Es un proceso de ingeniería de software, que hace una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización que desarrolla software. Su meta principal es asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios, con una planeación y presupuesto predecible.

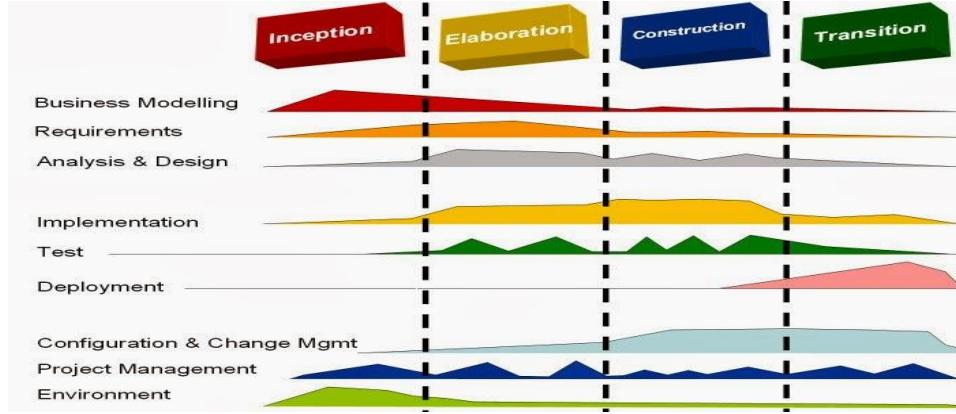


Figura 1 RUP

### 1.3.3.1.1 Características principales

#### 1.3.3.1.1.1 Proceso dirigido por Casos de Uso

Según Kruchten Philippe (2000), “los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar”.

Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema”.

Los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo.



Figura 2 Proceso dirigido por Casos de Uso

Los casos de uso Inician el proceso de desarrollo y proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Basándose en los casos de uso, se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso

#### **1.3.3.1.1.2 Proceso centrado en la arquitectura**

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

En el caso de RUP, además de utilizar los casos de uso para guiar el proceso, se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Existe una interacción entre los casos de uso y la arquitectura, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro.

#### **1.3.3.1.1.3 Proceso iterativo e incremental**

El equilibrio correcto entre los casos de uso y la arquitectura es muy parecido al equilibrio de la forma y la función en el desarrollo de un producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

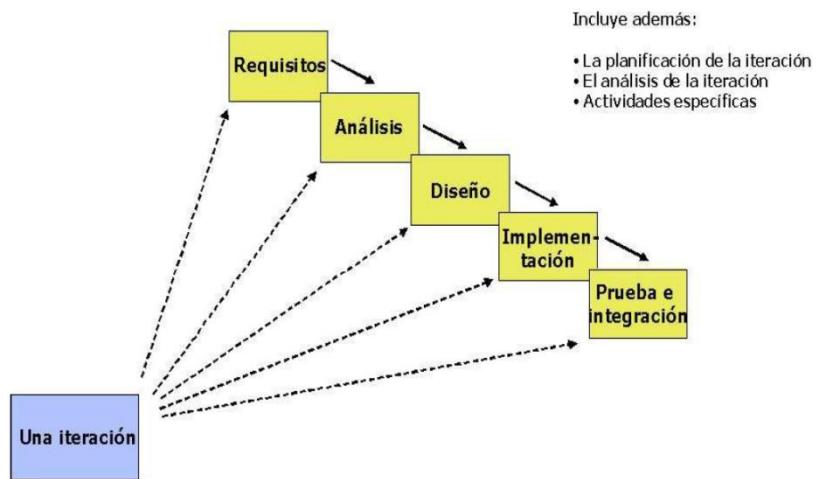


Figura 3 Proceso iterativo e incremental

Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando se termina. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones (que son una cantidad variable) según el proyecto, y en las que se hace un mayor o menor hincapié en los distintas actividades.

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” (base de referencia) de la arquitectura.

Para cada iteración se escogen algunos casos de uso, se refina su análisis y diseño, y se procede a su implementación y pruebas. Se realizan tantas 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, es el esfuerzo dedicado a una disciplina.

### **1.3.3.1.2 Fases**

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales. En cada extremo de una fase se realiza una evaluación para determinar si los objetivos de la fase se han cumplido. Una vez que la evaluación obtiene los resultados deseados, se procede a la siguiente fase

#### **1.3.3.1.2.1 Fase de inicio**

Define el ámbito y objetivos del proyecto, además de la funcionalidad y capacidades del producto.

#### **1.3.3.1.2.2 Elaboración**

Tanto la funcionalidad como el dominio del problema se estudian a profundidad. Se define una arquitectura básica y se planifica el proyecto considerando recursos disponibles.

#### **1.3.3.1.2.3 Construcción**

El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación. Las fases de concepción y elaboración sólo dieron una arquitectura básica que es refinada aquí de manera incremental conforme se construye (se permiten cambios en la estructura). Gran parte del trabajo es programación y pruebas, se documenta tanto el sistema construido como el manejo del mismo. En esta fase se hace una documentación junto con el producto.

#### **1.3.3.1.2.4 Transición**

Se libera el producto y se entrega al usuario para un uso real. Se incluyen tareas de mercadotecnia, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.

Los manuales de usuario se completan y refinan con la información anterior.

### **1.3.4 Transferencia de Resultado**

#### **1.3.4.1 Medios y Estrategias para la Transferencia de Resultados**

La Transferencia de resultados se realizara mediante un convenio establecido entre la Universidad Juan Misael Saracho y los beneficiarios, en este convenio también se definirán los medios que se utilizaron en el desarrollo del proyecto.

#### **1.3.4.2 Grupo de Beneficiarios**

Los beneficiarios con el desarrollo del proyecto son la cooperativa de transporte “RÍO GUADALQUIVIR”.

### **1.3.5 Resumen Ejecutivo del Proyecto**

La cooperativa de transporte “Río Guadalquivir” es una cooperativa que se dedica al transporte pesado (transporte de carga pesada para camiones) presenta la información de forma desordenada (en papeles, documentos Excel, documentos Word); Para ello se propone el proyecto que consiste en “**MEJORAMIENTO EN EL CONTROL Y ADMINISTRACIÓN DE LA INFORMACIÓN EN LA COOPERATIVA DE TRANSPORTE RIO GUADALQUIVIR**” lo cual busca mantener de forma ordenada y segura a información de dicha cooperativa.

El sistema control y administración de la información en la cooperativa de transporte río Guadalquivir permitirá mantener guardada la información de todos los socios, clientes, trabajadores y estar disponible cuando lo necesiten, también ayuda a que los socios estén actualizados a cerca de la asignación y programación de carga, reuniones.

También es una gran ventaja para la cooperativa “RIO GUADALQUIVIR”, en tanto que lo lleva un paso adelante en la modernización.

El desarrollo del sistema informático utilizará la metodología RUP Rational Unified Process (Proceso Unificado Racional) el lenguaje de desarrollo será PHP del lado del servidor con el framework LARAVEL 5.7 , COMPOUSER desarrollo web

JAVASCRIP con el framework JQUERY , framework BOOTSTRAP , framework IONIC, framework ANGULAR y base de datos MYSQL.

Desarrollar una aplicación móvil es muy importante en los últimos años debido al aumento masivo de uso de celulares inteligentes ésta ayuda a tener la información en la palma de la mano de todos los socios de la cooperativa.

La realización de la capacitación apoyada por distintas metodologías, permite capacitar al personal en el uso del software. De manera que la información automatizada, coadyuvara a que los procesos que con lleva la administración de la cooperativa “Rio Guadalquivir”.

### **1.3.6 Cronograma de Actividades**

### 1.3.7 Marco Lógico del Proyecto

<b>Resumen Narrativo del Proyecto</b>	<b>Indicadores</b>	<b>Medios de Verificación</b>	<b>Supuestos</b>
<b>Fin</b> Contribuir en el control y administración de la información en la cooperativa transporte Guadalquivir	Buen funcionamiento del sistema brindando información detallada y ordenada, implementando una tecnología al alcance del cliente con la información necesaria.	La satisfacción de los socios de la cooperativa Rio Guadalquivir al saber que la información están guarda y disponible.	Mantener la condición tecnológica y administrativa necesaria para el correcto funcionamiento del sistema.
<b>Objetivo General</b> Mejoramiento en el control y administración de la información en la cooperativa transporte Guadalquivir	Al finalizar el proyecto un 73 % de los procesos de la cooperativa serán automatizados (con un criterio de mejora)(Anexo1) $Pa = Npa / Ntp * 100$ $Pa = 11 / 15 * 100 = 73\%$	Carta de conformidad, emitido por el presidente de la cooperativa "Rio Guadalquivir", con toda la funcionalidad que se ha requerido y con el personal capacitado en su uso. (página 227,228)	Garantizar el mantenimiento del sistema a través de revisiones periódicas por parte del administrador.
<b>Objetivos Específicos</b> -Desarrollar un Sistema de control y administración de la información en la cooperativa de transporte rio Guadalquivir	-Al finalizar el proyecto, se ha desarrollado un Sistema y aplicación móvil para el control y administración de la información en la cooperativa de transporte rio Guadalquivir basados en los requerimientos	- Corrección de la documentación por el Ingeniero. Succi Aguirre Clovis Gustavo docente de la materia de Taller III, avalado por la universidad Juan Misal Saracho.	-El Equipo de desarrollo cuenta con todo el apoyo de la cooperativa "Rio Guadalquivir " para el correcto desarrollo del sistema de control y administración de la información en la cooperativa de transporte rio Guadalquivir

<p>-Desarrollar una aplicación móvil de control y administración de la información en la cooperativa de transporte rio Guadalquivir</p> <p>-Plantear una técnica de capacitación del Sistema Desarrollado.</p>	<p>expresados .</p> <p>-Al finalizar el proyecto, se ha planteado una técnica de capacitación sobre el uso del sistema informático y la aplicación móvil.</p>	<p>-informe sobre la capacitación realizada en la cooperativa rio Guadalquivir (página 270)</p> <p>-Fotografías de la Capacitación Realizada.(página 266)</p>	<p>-Interés y apoyo de los socios presentes en la capacitación.</p>
<p><b>Actividades</b></p> <p>Verificación del software</p> <p>Mantenimiento</p> <p>Actualización</p>	<p>El presupuesto total del Total: Bs 28,200.</p>	<p>-Instalador del sistema listo para ser utilizado.</p> <p>-Informe presupuestario avalado por la universidad Autónoma Juan Misael Saracho. (página 42)</p>	<p>Los desembolsos de realizar de acuerdo al cronograma</p>

Tabla 4 Marco Lógico del Proyecto

#### 1.4 Presupuesto / Justificación

ITEM	RUBROS	Apunte Universidad	Otro Apunte	TOTAL (Bs.)
<b>10000</b>	<b>SERVICIOS PERSONALES</b>			
	<b>12000 Empleados no Permanentes</b>	0		3000
	<b>Sub total rubro</b>			<b>3000</b>
<b>20000</b>	<b>SERVICIOS NO PERSONALES</b>	0		
	<b>21000. Servicios Básicos</b>	0		3000
	<b>22000. Servicios de transporte</b>	0		1000
	<b>23000. Alquileres</b>	0		4000
	<b>24000. Mantenimiento y reparación</b>	0		2000
	<b>25000. Servicios Profesionales y Comerciales</b>	0		2400
	<b>Sub total rubro</b>			<b>12400</b>
<b>30000</b>	<b>MATERIALES Y SUMINISTROS</b>			
	<b>32000. Productos de Papel, Cartón e Impresos</b>	0		1000
	<b>33000. Textiles y Vestuario.</b>	0		1000
	<b>39000. Productos Varios.</b>	0		800
	<b>Sub total rubro</b>			<b>2800</b>
<b>40000</b>	<b>ACTIVOS REALES</b>			
	<b>43000. Maquinaria y Equipo.</b>			9000
	<b>49000. Otros Activos</b>			1000
	<b>Sub total rubro</b>			<b>10000</b>
	<b>TOTAL</b>			<b>28200</b>

Tabla 5 Presupuesto / Justificación

## **GRUPO 10000. SERVICIOS PERSONALES**

### **1.4.1 SUB GRUPO 12000. Empleados no Permanentes**

<b>Partida</b>	<b>Personal</b>	<b>Remuneración</b>	<b>Tiempo/meses</b>	<b>Total</b>
12100	Personal Eventual			
	Ingeniero Junior área de Informática	600.-	5	3000
<b>Total</b>				<b>3000</b>

Tabla 6 SUB GRUPO 12000. Empleados no Permanentes

## **GRUPO 20000. SERVICIOS NO PERSONALES**

### **1.4.2 SUB GRUPO 21000. Descripción de los gastos de servicios básicos**

<b>Partida</b>	<b>Tipo de servicio básico *</b>	<b>Costo</b>	<b>Tiempo mes</b>	<b>Costo Total</b>
21100	Comunicación	200	5	1000
21200	Energía Eléctrica	200	5	1000
21300	Agua	250	2	500
21400	Servicios Telefónicos	250	2	500
<b>Total</b>				<b>3000</b>

Tabla 7 SUB GRUPO 21000. Descripción de los gastos de servicios básicos

### **1.4.3 SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal**

<b>Partida</b>	<b>Personal</b>	<b>Lugar</b>	<b>Nº de viajes</b>	<b>Costo unitario*</b>	<b>Costo total</b>
22100	Pasajes	Tarija	200	5	
<b>Total</b>					<b>1000</b>

Tabla 8 SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal

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

Partida	Alquiler de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
23100	Alquiler de Edificios	500	8	4000
23200	Alquiler de Equipos y Maquinaria			
23300	Alquiler de Tierras y Terrenos			
<b>Total</b>				<b>4000</b>

Tabla 9 SUB GRUPO 23000. Descripción de los gastos por concepto de alquileres de equipos y maquinarias

**1.4.5 SUB GRUPO 24000. Descripción mantenimiento y reparación**

Partida	Mantenimiento y reparación de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
24100	Mantenimiento y Reparación de Edificios y Equipos	500	2	1000
24300	Otros Gastos por Mantenimiento y Reparación	500	2	1000
<b>Total</b>				<b>2000</b>

Tabla 10 SUB GRUPO 24000. Descripción mantenimiento y reparación

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

Partida	Tipo de servicio profesional y comercial *	Cantidad	Costo unitario	Tiempo mes	Costo total
25200	Estudios e Investigaciones				
25500	Publicidad				
25600	Imprenta				
25700	Capacitación de Personal	1	600	1	2400

	Total			2400

Tabla 11 SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales

## GRUPO 30000. MATERIALES Y SUMINISTROS

### 1.4.8 SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio	2	500	1000
	Total			1000

Tabla 12 SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

Gastos de Productos de Papel, Cartón e Impresos.

### 1.4.9 SUB GRUPO 33000. Descripción del gasto en textiles y vestuario

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
33300	Prendas de vestir	3	150	450
33400	Calzados	2	275	550
	Total			1000

Tabla 13 SUB GRUPO 33000. Descripción del gasto en textiles y vestuario

### 1.4.10 SUB GRUPO 39000. Descripción del gasto en productos varios

Partida	Productos de cuero y caucho	Cantidad	Costo/Unitario	Total
39100	Material de Limpieza	3	200	600
39500	Útiles de Escritorio y de Oficina	5	40	200
	Total			800

Tabla 14 SUB GRUPO 39000. Descripción del gasto en productos varios

## **GRUPO 40000. ACTIVOS REALES**

### **1.4.12 SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo**

Partida	Tipos de productos	Cantidad	Costo/Unitario	Total
43100	Equipo de Oficina y Muebles	1	7000	7000
	Total			9000

Tabla 15 SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo

### **1.4.13 SUB GRUPO 49000. Descripción del gasto de Otros Activos**

Partida	Tipos de productos *	Cantidad	Costo/Unitario	Total
49900	Otros Activos	1	1000	1000
	Total			1000

Tabla 16 SUB GRUPO 49000. Descripción del gasto de Otros Activos

## **CAPÍTULO II**

## **COMPONENTES**

## **COMPONENTE I**

## **Capítulo II: Componentes**

**1.5 Componente 1:** Desarrollo de Sistema Informático para el control y administración de la información en la cooperativa de transporte rio Guadalquivir.

### **1.5.1 Marco Teórico**

#### **1.5.1.1 Objetivo General**

Desarrollar un Sistema Informático para el control y administración de la información en la cooperativa de transporte rio Guadalquivir.

#### **1.5.1.2 Objetivos Específicos**

- Recolectar de mejor manera la información para poder entender el funcionamiento de la cooperativa.
- Analizar la información recolectada para poder analizarla con el personal involucrado de la cooperativa, los requerimientos que cubrirá el sistema.
- Diseñar y estructurar el sistema de forma que cumpla con los requerimientos identificados y además pueda soportar futuros cambios .
- Elaborar el sistema tomando en cuenta las tecnologías que mejor se adapten a su propósito.

#### **1.5.1.3 Antecedentes**

Los Sistemas de Información son requeridos para poder dar apoyo al proceso de toma de decisiones de las organizaciones. Desde esta perspectiva debe ser absolutamente claro que toda organización requiere de Sistemas de Información, y los requiere durante toda su existencia. Lo único que va a cambiar en el tiempo, es la forma en que se implementarán estos sistemas.

Al analizar la llegada de la informática a las empresas, es bueno considerar dos ámbitos importantes, por una parte está la evolución de las teorías de la administración y en el otro frente de batalla, la llegada de la computación a las empresas. Sin pretender hacer un análisis exhaustivo de las teorías de administración

(materia propia de otros ramos), es necesario observar algunos hitos importantes que han influido en el acercamiento al término: "informática" y su importante relación con las organizaciones.

Las empresas buscan otorgar un soporte adecuado al proceso productivo, el cual se caracteriza (en la mayoría de los casos) por la repetición de tareas específicas y muy bien especificadas. Es decir, no sólo con pocos objetivos muy bien definidos, sino que con una definición muy precisa de la metodología a seguir para alcanzar el objetivo.

#### **1.5.1.4 Metodología de Desarrollo**

##### **1.5.1.4.1 Metodología RUP**

El software será desarrollado con la metodología RUP, Es un proceso de ingeniería de software, que hace una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización que desarrolla software. Su meta principal es **asegurar la producción de software de alta calidad** que cumpla con las necesidades de los usuarios, con una planeación y presupuesto predecible.

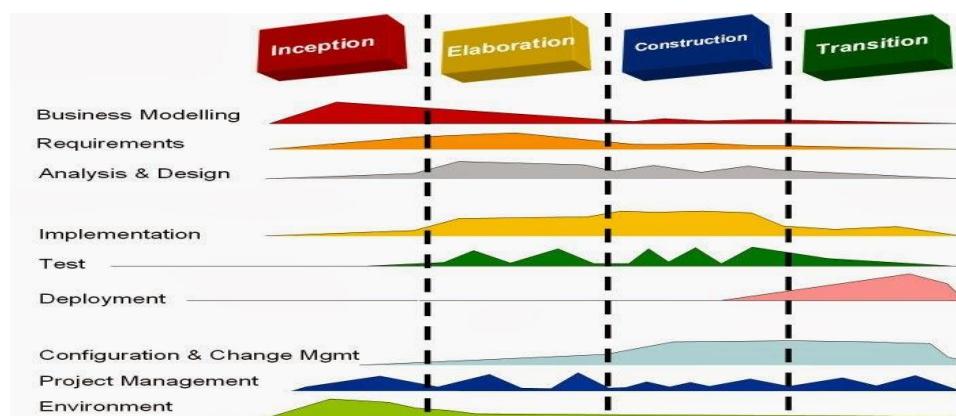


Figura 4 Metodología RUP

#### 1.5.1.4.1.1 Características principales

##### 1.5.1.4.1.1.1 Proceso dirigido por Casos de Uso

Según Kruchten Philippe (2000), “los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar”.

Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema”.

Los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo.

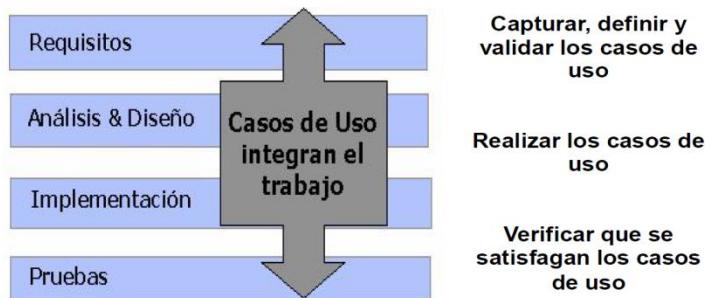


Figura 5 Proceso dirigido por Casos de Uso

Los casos de uso Inician el proceso de desarrollo y proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Basándose en los casos de uso, se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso

##### 1.5.1.4.1.1.2 Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados

(desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

En el caso de RUP, además de utilizar los casos de uso para guiar el proceso, se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Existe una interacción entre los casos de uso y la arquitectura, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro.

#### **1.5.1.4.1.1.3 Proceso iterativo e incremental**

El equilibrio correcto entre los casos de uso y la arquitectura es muy parecido al equilibrio de la forma y la función en el desarrollo de un producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

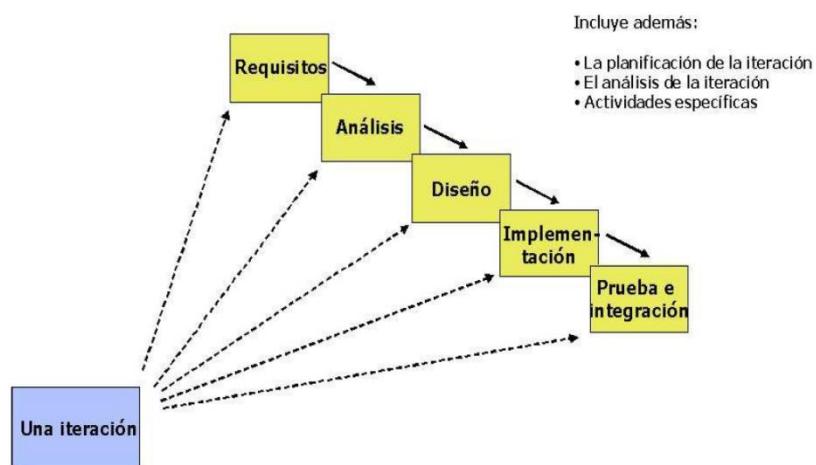


Figura 6 Proceso iterativo e incremental

Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando se termina. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones (que son una cantidad variable) según el proyecto, y en las que se hace un mayor o menor hincapié en los distintas actividades.

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” (base de referencia) de la arquitectura.

Para cada iteración se escogen algunos casos de uso, se refina su análisis y diseño, y se procede a su implementación y pruebas. Se realizan tantas 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, es el esfuerzo dedicado a una disciplina.

#### **1.5.1.4.1.2 Fases**

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales. En cada extremo de una fase se realiza una evaluación para determinar si los objetivos de la fase se han cumplido. Una vez que la evaluación obtiene los resultados deseados, se procede a la siguiente fase

#### **1.5.1.4.1.2.1 Fase de inicio**

Define el ámbito y objetivos del proyecto, además de la funcionalidad y capacidades del producto.

#### **1.5.1.4.1.2.2 Elaboración**

Tanto la funcionalidad como el dominio del problema se estudian a profundidad. Se define una arquitectura básica y se planifica el proyecto considerando recursos disponibles.

#### **1.5.1.4.1.2.3 Construcción**

El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación. Las fases de concepción y elaboración sólo dieron una arquitectura básica que es refinada aquí de manera incremental conforme se construye (se permiten cambios en la estructura). Gran parte del trabajo es programación y pruebas, se documenta tanto el sistema construido como el manejo del mismo. En esta fase se hace una documentación junto con el producto.

#### **1.5.1.4.1.2.4 Transición**

Se libera el producto y se entrega al usuario para un uso real. Se incluyen tareas de mercadotecnia, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.

Los manuales de usuario se completan y refinan con la información anterior.

#### **1.5.1.4.2 UML**

UML [UML] es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo

estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

Las diferencias son muy marcadas y afectan a todas las fases del proceso. El método del UML recomienda utilizar los procesos que otras metodologías tienen definidos.

#### **1.5.1.4.2.1 Los Inicios**

A partir del año 1994, Grady Booch [Booch96](precursor de Booch '93) y Jim Rumbaugh (creador de OMT) se unen en una empresa común, Rational Software Corporation, y comienzan a unificar sus dos métodos. Un año más tarde, en octubre de 1995, aparece UML (Unified Modeling Language) 0.8, la que se considera como la primera versión del UML. A finales de ese mismo año, Ivan Jacobson, creador de OOSE (Object Oriented Software Engineer) se añade al grupo.

Como objetivos principales de la consecución de un nuevo método que aunara los mejores aspectos de sus predecesores, sus protagonistas se propusieron lo siguiente:

- El método debía ser capaz de modelar no sólo sistemas de software sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).
- Crear un lenguaje para modelado utilizable a la vez por máquinas y por personas.
- Establecer un acoplamiento explícito de los conceptos y los artefactos ejecutables.
- Manejar los problemas típicos de los sistemas complejos de misión crítica.

Lo que se intenta es lograr con esto que los lenguajes que se aplican siguiendo los métodos más utilizados sigan evolucionando en conjunto y no por separado. Y además, unificar las perspectivas entre diferentes tipos de sistemas (no sólo software,

sino también en el ámbito de los negocios), al aclarar las fases de desarrollo, los requerimientos de análisis, el diseño, la implementación y los conceptos internos de la OO.

#### **1.5.1.4.2.2 Modelado de objetos**

En la especificación del UML podemos comprobar que una de las partes que lo componen es un meta modelo formal. Un meta modelo es un modelo que define el lenguaje para expresar otros modelos. Un modelo en OO es una abstracción cerrada semánticamente de un sistema y un sistema es una colección de unidades conectadas que son organizadas para realizar un propósito específico. Un sistema puede ser descripto por uno o más modelos, posiblemente desde distintos puntos de vista.

Una parte del UML define, entonces, una abstracción con significado de un lenguaje para expresar otros modelos (es decir, otras abstracciones de un sistema, o conjunto de unidades conectadas que se organizan para conseguir un propósito). Lo que en principio puede parecer complicado no lo es tanto si pensamos que uno de los objetivos del UML es llegar a convertirse en una manera de definir modelos, no sólo establecer una forma de modelo, de esta forma simplemente estamos diciendo que UML, además, define un lenguaje con el que podemos abstraer cualquier tipo de modelo.

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación.

Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.

Los modelos se utilizan en muchas actividades de la vida humana: antes de construir una casa el arquitecto utiliza un plano, los músicos representan la música en forma de notas musicales, los artistas pintan sobre el lienzo con carboncillos antes de empezar a utilizar los óleos, etc. Unos y otros abstraen una realidad compleja sobre unos bocetos, modelos al fin y al cabo. La OMT, por ejemplo, intenta abstraer la realidad utilizando tres clases de modelos OO: el modelo de objetos, que describe la estructura estática; el modelo dinámico, con el que describe las relaciones temporales entre objetos; y el modelo funcional que describe las relaciones funcionales entre valores. Mediante estas tres fases de construcción de modelos, se consigue una abstracción de la realidad que tiene en sí misma información sobre las principales características de ésta.

Los modelos además, al no ser una representación que incluya todos los detalles de los originales, permiten probar más fácilmente los sistemas que modelan y determinar los errores. Según se indica en la Metodología OMT (Rumbaugh), los modelos permiten una mejor comunicación con el cliente por distintas razones:

- Es posible enseñar al cliente una posible aproximación de lo que será el producto final.
- Proporcionan una primera aproximación al problema que permite visualizar cómo quedará el resultado.
- Reducen la complejidad del original en subconjuntos que son fácilmente tratables por separado.

Se consigue un modelo completo de la realidad cuando el modelo captura los aspectos importantes del problema y omite el resto. Los lenguajes de programación que estamos acostumbrados a utilizar no son adecuados para realizar modelos completos de sistemas reales porque necesitan una especificación total con detalles que no son importantes para el algoritmo que están implementando. En OMT se modela un sistema desde tres puntos de vista diferentes donde cada uno representa una parte del sistema y una unión lo describe de forma completa. En esta técnica de modelado se utilizó una aproximación al proceso de implementación de software

habitual donde se utilizan estructuras de datos (modelo de objetos), las operaciones que se realizan con ellos tienen una secuencia en el tiempo (modelo dinámico) y se realiza una transformación sobre sus valores (modelo funcional).

UML utiliza parte de este planteamiento obteniendo distintos puntos de vista de la realidad que modela mediante los distintos tipos de diagramas que posee. Con la creación del UML se persigue obtener un lenguaje que sea capaz de abstraer cualquier tipo de sistema, sea informático o no, mediante los diagramas, es decir, mediante representaciones gráficas que contienen toda la información relevante del sistema. Un diagrama es una representación gráfica de una colección de elementos del modelo, que habitualmente toma forma de grafo donde los arcos que conectan sus vértices son las relaciones entre los objetos y los vértices se corresponden con los elementos del modelo. Los distintos puntos de vista de un sistema real que se quieren representar para obtener el modelo se dibuja de forma que se resaltan los detalles necesarios para entender el sistema. [8]

#### **1.5.1.4.2.3 Artefactos para el Desarrollo de Proyectos**

Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. Pueden ser artefactos un modelo, una descripción o un software. Los artefactos de UML se especifican en forma de diagramas, éstos, junto con la documentación sobre el sistema constituyen los artefactos principales que el modelador puede observar.

Se necesita más de un punto de vista para llegar a representar un sistema. UML utiliza los diagramas gráficos para obtener estos distintos puntos de vista de un sistema:

- **Diagramas de Implementación.**
- **Diagramas de Comportamiento o Interacción.**
- **Diagramas de Casos de uso.**
- **Diagramas de Clases.**

Ejemplo de algunos de los diagramas que utiliza UML.

### **Diagramas de Implementación**

Se derivan de los diagramas de proceso y módulos de la metodología de Booch, aunque presentan algunas modificaciones. Los diagramas de implementación muestran los aspectos físicos del sistema. Incluyen la estructura del código fuente y la implementación, en tiempo de implementación. Existen dos tipos:

#### **Diagramas de componentes**

##### **Diagrama de plataformas despliegue**

##### **Diagramas de componentes**

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

##### **Diagrama de plataformas o despliegue**

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

### **Diagramas de Interacción o Comportamiento**

Muestran las interacciones entre objetos ocurridas en un escenario (parte) del sistema.

Hay varios tipos:

- **Diagrama de secuencia.**
- **Diagrama de colaboración.**

- **Diagrama de estado.**
- **Diagrama de actividad.**
- **Diagrama de secuencia**

Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. En los diagramas de este tipo intervienen objetos, que tienen un significado parecido al de los objetos representados en los diagramas de colaboración, es decir son instancias concretas de una clase que participa en la interacción. El objeto puede existir sólo durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción. Un diagrama de secuencia representa una forma de indicar el período durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.

En este tipo de diagramas también intervienen los mensajes, que son la forma en que se comunican los objetos: el objeto origen solicita (llama a) una operación del objeto destino. Existen distintos tipos de mensajes según cómo se producen en el tiempo: simples, síncronos, y asíncronos.

Los diagramas de secuencia permiten indicar cuál es el momento en el que se envía o se completa un mensaje mediante el tiempo de transición, que se especifica en el diagrama.

#### **1.5.1.4.3 Tipos de Diagramas Utilizados**

##### **1.5.1.4.3.1 Diagrama de Caso de Uso**

Un diagrama de casos de uso actúa como foco en la descripción de los requisitos del usuario. En él se describen las relaciones entre los requisitos, los usuarios y los componentes principales. Los requisitos no se describen en detalle, ya que esto puede hacerse en otros diagramas o en documentos que pueden vincularse a cada caso de uso

#### **1.5.1.4.3.2 Diagrama de Clases**

Un diagrama de clases UML para describir los tipos de datos y sus relaciones con independencia de su implementación. El diagrama se utiliza para que la atención se centre en los aspectos lógicos de las clases en lugar de en su implementación.

#### **1.5.1.4.3.3 Diagrama de Actividades**

Un diagrama de actividades muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Las personas, los componentes de software o los equipos pueden realizar estas acciones.

#### **1.5.1.4.3.4 Diagrama de Secuencias**

Un diagrama de secuencia muestra una interacción, que representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores. El tiempo fluye por el diagrama y muestra el flujo de control de un participante a otro. Utilice diagramas de secuencia para visualizar instancias y eventos, en lugar de clases y métodos. En el diagrama, puede aparecer más de una instancia del mismo tipo. También puede haber más de una ocurrencia del mismo mensaje.

### **1.5.1.5 Herramientas de Construcción de Software**

#### **1.5.1.5.1 Enterprise Architect**

El modelado de sistemas utilizando 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 sistemas existentes. Los aspectos que pueden ser cubiertos por este tipo de modelado van desde el diseño de arquitecturas organizacionales o de sistemas, la reingeniería de procesos de negocios, el análisis de negocios y las arquitecturas orientadas a servicios y el modelado web, hasta el diseño de aplicaciones y bases de datos. y reingeniería, 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 a través de las fases de diseño, construcción, prueba y mantenimiento, con soporte para trazabilidad, gestión de proyectos y control de

cambios de estos procesos, así como instalaciones para el desarrollo de código de aplicación basado en modelos utilizando una plataforma interna de desarrollo integrado .

La base de usuarios abarca desde programadores y analistas de negocios hasta arquitectos empresariales, en organizaciones que van desde pequeñas empresas desarrolladoras, 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



Figura 7 Enterprise Architect

<https://sparxsystems.com/products/ea/>

#### 1.5.1.5.2 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS. 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,<sup>12</sup> aunque la descarga oficial está bajo software propietario.<sup>3</sup>

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar aplicaciones Node.js para el escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor ("Monaco") utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online)

Visual Studio Code es un editor de código fuente. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado, como se muestra en la siguiente tabla. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos .json (por ejemplo,

preferencias del usuario). La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo que está fuera de él. Esto también se aplica a los comandos que requieren mucho tiempo. Cuando esto sucede, el comando en progreso se cancela.

En el rol de editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una opción entre LF y CRLF) y el lenguaje de programación del documento activo

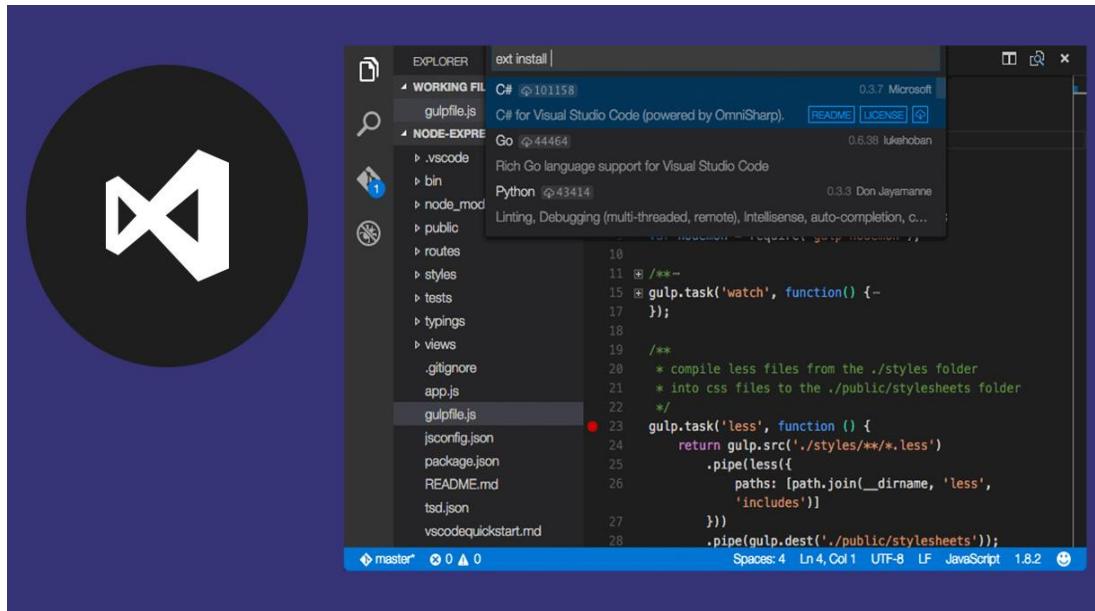


Figura 8 Visual Studio Code

<https://code.visualstudio.com/>

### 1.5.1.5.3 Mysql

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de doble

licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de GNU, versión , y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones Enterprise incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial. En 2009 se creó un fork denominado MariaDB por algunos desarrolladores (incluido algunos desarrolladores originales de MySQL) descontentos con el modelo de desarrollo y el hecho de que una misma empresa controle a la vez los productos MySQL y Oracle Database.

Está desarrollado en su mayor parte en ANSI C y C++. Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo LAMP y WAMP.

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr, y YouTube.



Figura 9 Mysql

<https://www.mysql.com/>

#### 1.5.1.5.4 Xampp

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP,

Perl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.



Figura 10 Xampp

<https://www.apachefriends.org/es/index.html>

#### 1.5.1.5.5 Composer

Composer es un administrador de paquetes a nivel de aplicación para el lenguaje de programación PHP que proporciona un formato estándar para administrar las dependencias del software PHP y las bibliotecas requeridas . Fue desarrollado por Nils Adermann y Jordi Boggiano, quienes continúan administrando el proyecto. Comenzaron el desarrollo en abril de 2011 y lanzado por primera vez el 1 de marzo de 2012. [1] Compositor está fuertemente inspirado por de Node.js " NGP " y Ruby " bundler ". El algoritmo de resolución de dependencias del proyecto comenzó como un puerto basado en PHP del libzypp satsolver de openSUSE .

Composer se ejecuta a través de la línea de comandos e instala dependencias (por ejemplo, bibliotecas) para una aplicación. También permite a los usuarios instalar

aplicaciones PHP que están disponibles en "Packagist" , que es su repositorio principal que contiene los paquetes disponibles. También proporciona capacidades de carga automática para bibliotecas que especifican información de carga automática para facilitar el uso del código de terceros.

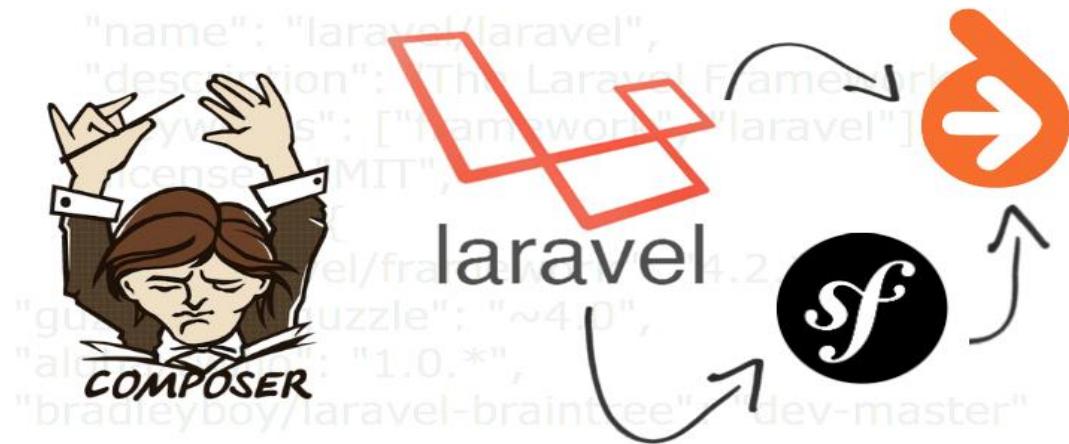


Figura 11 Composer

<https://getcomposer.org/>

<https://www.npmjs.com/>

### 1.5.1.6 Técnica

#### 1.5.1.6.1 Php

PHP, acrónimo recursivo en inglés de PHP Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser

usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP.3 Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3\_01, es una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, esta licencia no tiene restricciones de copyleft" asociadas con GPL.

El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. El número de sitios basados en PHP se ha visto reducido progresivamente en los últimos años, con la aparición de nuevas tecnologías como Node.JS, Golang, ASP.NET, etc. El sitio web de Wikipedia está desarrollado en PHP.6 Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

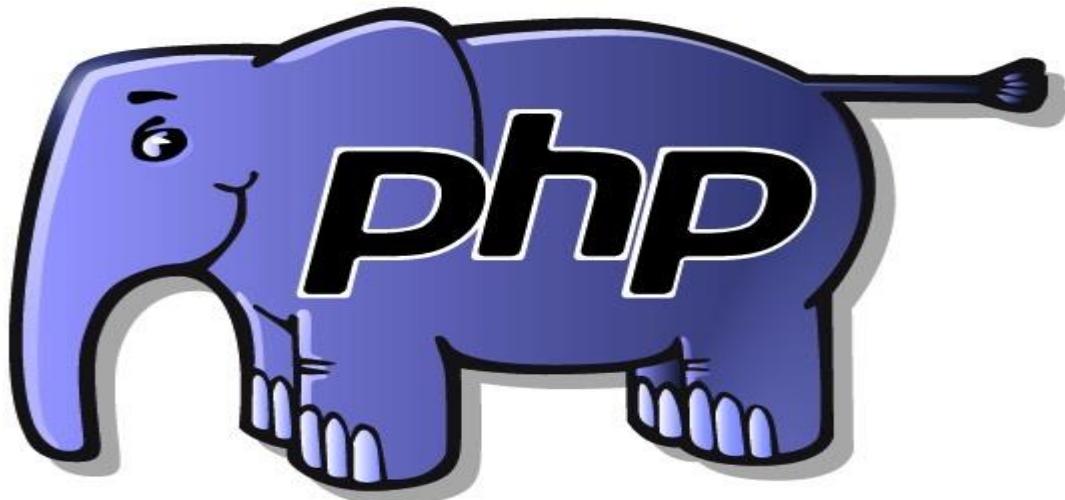


Figura 12 Php

<http://php.net/>

### 1.5.1.6.2 Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.<sup>2</sup>

Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

### Patrón MVC

Laravel propone en el desarrollo usar 'Routes with Closures', en lugar de un MVC tradicional con el objetivo de hacer el código más claro. Aun así permite el uso de MVC tradicional

### Modelo

Laravel incluye un sistema de mapeo de datos relacional llamado Eloquent ORM que facilita la creación de modelos. Este ORM se funda en patrón active record y su funcionamiento es muy sencillo. Es opcional el uso de Eloquent, pues también dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la creación de modelos.

### Vista

Laravel incluye de paquete un sistema de procesamiento de plantillas llamado Blade. Este sistema de [plantillas](#) favorece un código mucho más limpio en las Vistas, además

de incluir un sistema de Caché que lo hace mucho más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura.

### Controlador

Los controladores contienen la lógica de la aplicación y permiten organizar el código en clases sin tener que escribirlo todo en las rutas. Todos los controladores deben extenderse de la clase BaseController.además de eso



Figura 13 Laravel

<https://laravel.com/>

#### 1.5.1.6.3 Javascript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,<sup>3</sup> basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas<sup>4</sup> aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de javascript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

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).

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.

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó). En junio de 2015 se cerró y publicó la versión ECMAScript 6.7



Figura 14 Javascrip

#### 1.5.1.6.4 Jquery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas. Microsoft la añadirá en su IDE Visual Studio4 y la usará junto con los frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma



Figura 15 Jquery

<https://jquery.com/>

### 1.5.1.6.5 Html

TML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.



Figura 16 Html

### 1.5.1.6.6 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.<sup>2</sup> 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. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.



Figura 17 Css

### **1.5.1.7 Plan de Desarrollo de Software**

#### **1.5.1.7.1 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 ciencias y tecnología de la universidad autónoma Juan Misael Saracho este documento provee una visión global del enfoque de desarrollo propuesto.

El propósito del presente proyecto es el desarrollo de un sistema de control y administración de la información de la cooperativa rio Guadalquivir, con el proyecto propuesto se podrá eliminar y reducir los problemas de la cooperativa rio Guadalquivir.

#### **1.5.1.7.1.1 Propositivo**

El propósito del plan de desarrollo del software es proporcionar información necesaria para controlar el proyecto. En el que se describe el enfoque de desarrollo del software.

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

Los usuarios del plan de desarrollo del software son:

- El director responsable del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- El docente para evaluar el cumplimiento del proyecto.

#### **1.5.1.7.1.2 Alcance**

El plan de desarrollo de software describe el plan global para el desarrollo del proyecto titulado “**MEJORAMIENTO EN EL CONTROL Y ADMINISTRACIÓN DE LA INFORMACIÓN DE LA COOPERATIVA DE TRANSPORTE RÍO GUADALQUIVIR**”.

El detalle de las interacciones individuales se describe en los planes de cada iteración. Lo que se pretende alcanzar con este plan es documentar las mejoras planteadas en el control y administración de la información en la cooperativa rio Guadalquivir.

Posteriormente el avance del proyecto y seguimiento en cada una de las iteraciones ocasionara el ajuste de este documento produciendo nuevas versiones actualizadas siendo este documento la última versión.

#### **1.5.1.7.1.3 Resumen**

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

**Vista general del proyecto.-** Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

**Organización del proyecto.-** Describe la estructura organizacional del equipo de desarrollo.

#### **1.5.1.7.2 Vista General del Proyecto**

##### **1.5.1.7.2.1 Propósito**

Contribuir de manera eficiente y eficaz en la mejora y control de la administración de la información en la cooperativa rio Guadalquivir que permitirá otorgar información rápida y confiable a sus usuarios.

##### **1.5.1.7.2.2 Alcance**

El sistema desarrollado otorgara una interfaz amigable para que este pueda tener información actualizada y confiable.

Mediante el sistema se podrá realizar el registro de socios, clientes , trabajadores, asignación de carga a camiones, mostrara el rol de espera para cargar y aviso de reuniones.

### **1.5.1.7.2.3 Objetivos**

#### **1.5.1.7.2.3.1 Objetivo general**

Desarrollar un sistema informático para el control y administración de la información en la cooperativa nacional e internacional río Guadalquivir.

#### **1.5.1.7.2.3.2 Objetivo específico**

- Recolectar de mejor manera la información para poder entender el funcionamiento de la cooperativa.
- Analizar la información recolectada para poder analizarla con el personal involucrado de la cooperativa, los requerimientos que cubrirá el sistema.
- Diseñar y estructurar el sistema de forma que cumpla con los requerimientos identificados y además pueda soportar futuros cambios.
- Elaborar el sistema tomando en cuenta las tecnologías que mejor se adapten a su propósito.
- Ejecutar pruebas que garanticen la calidad del sistema e identifiquen los errores a conseguir antes de su implementación.
- Instrumentar el sistema con las siguientes tecnologías
  - Mysql
  - Php7.1
  - Html 5y css3
  - Laravel 5.7
  - Jquery3.2
- Aplicar el lenguaje de modelado unificado(UML) para el desarrollo del sistema.
- Aplicar la metodología RUP .
- Diseñar una interfaz gráfica interactiva, amigable y de fácil uso para el manejo del usuario.

### **1.5.1.7.2.4 Suposiciones y Restricciones**

#### **1.5.1.7.2.4.1 Suposiciones**

El personal involucrado apoya constantemente, proporciona la información de manera detallada y necesaria para un correcto desarrollo del sistema.

#### **1.5.1.7.2.4.2 Restricciones**

El sistema por lo menos debe contar con un administrador que será encargado del sistema para cumplir con su objetivo.

#### **1.5.1.7.3 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 contribuye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

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

- Plan de desarrollo del software

Es el presente documento.

- Modelo de casos de uso de negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos, permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se presenta con un diagrama de casos de uso usando estereotipos específicos para este modelo.

- Glosario

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

- Modelo de casos de uso

El modelo de casos de uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante diagramas de casos de uso.

- Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

- Descripción de casos de uso

Para los casos de uso que se requieran (cuya finalidad no sea evidente que no baste con una simple descripción) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones post-condiciones, flujo de eventos y requisitos no funciones asociados.

- Prototipo de interfaces de usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de la interface que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema

- Modelo de datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la presentación lógica de los datos persistentes, de acuerdo con el enfoque para el modelado relacional de datos.

- Casos de prueba

Cada prueba es especificado mediante un documento que establece las condiciones de ejecución, las entradas de la prueba de regresión en cada iteración. Cada caso de prueba cada caso de prueba es asociado a un procedimiento de prueba de las instrucciones para realizar la prueba.

#### **1.5.1.7.4 Glosario**

##### **1.5.1.7.4.1 Introducción**

El propósito es definir la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del sistema. Constituye una base de acuerdo a los requisitos del sistema.

##### **1.5.1.7.4.2 Propósito**

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo.

##### **1.5.1.7.4.3 Alcance**

El alcance del presente documento se extiende a todos los subsistemas definidos. de tal modo que la terminología empleada, se refleja con claridad en este documento.

##### **1.5.1.7.4.4 Referencias**

El presente glosario hace referencia a los siguientes documentos:

- Documento plan desarrollo del software del proyecto.
- Documento visión del proyecto.
- Documento especificación de casos de uso.

##### **1.5.1.7.4.5 Organización del glosario**

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

##### **1.5.1.7.4.6 Definiciones**

A continuación se presenta todos los términos manejados a lo largo de todo el proyecto de desarrollo.

**Administrador** persona encargada de dirigir la cooperativa y control de diferentes opciones en el sistema.

**Aplicación** un programa informático que lleva acabo una función con el objeto de ayudar a un usuario a realizar una determinada actividad.

**Cliente-servidor** modelo de comunicación entre ordenadores conectados a una red en el cual hay uno llamado cliente que satisface las peticiones realizadas por otro llamado servidor.

**Código fuente** conjunto de instrucciones que componen un programa informático. Estos programas se escriben en determinados lenguajes.

**Computadora** maquina electrónica capaz de procesar información siguiendo instrucciones almacenadas en programas.

**Diagrama** representación gráfica de un conjunto de elementos, representando la mayoría de las veces como un grafo conexo de nodos (elementos) y arcos (relaciones).

**Hardware** conjunto de los componentes que integran la parte material para el funcionamiento de una computadora .

**HTML** lenguaje de hipertextos en el cual se hacen las vistas del sistema.

**http** protocolo usado para la trasferencia de documentos WWW.

**Requerimientos** algo que se pide o solicita a alguien. En informática pueden ser funcionales o no funcionales.

**Rol** asignación de un cargo a los usuarios del negocio.

**RUP** es una metodología de desarrollo de software que integra aspectos todo el ciclo de vida del software.

**Software** conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en la computadora.

### **1.5.1.7.5 Visión**

#### **1.5.1.7.5.1 Propósito**

El propósito de este documento es recoger analizar y definir las necesidades de alto nivel y las características del sistema de control y administración de la información en

la cooperativa rio Guadalquivir. el documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

#### **1.5.1.7.5.2 Alcance**

El documento visión se ocupa, como ya se ha apuntado, del sistema de control y administración de la información en la cooperativa rio Guadalquivir. Dicho sistema será desarrollado por el universitario: Luis Miguel Veliz Cardozo, alumno de quinto año de la Carrera de Ingeniería Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho (U.A.J.M.S.) Tarija.

#### **1.5.1.7.5.3 Definiciones Acrónicas**

**RUP:** son siglas Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.

**UML:** son las siglas de Unified Modeling Lenguaje. Es un lenguaje grafico para visualizar especificar, contribuir y documentar un sistema software.

#### **1.5.1.7.5.4 Referencias**

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

#### **1.5.1.7.6 Modelo general de casos de uso del negocio**

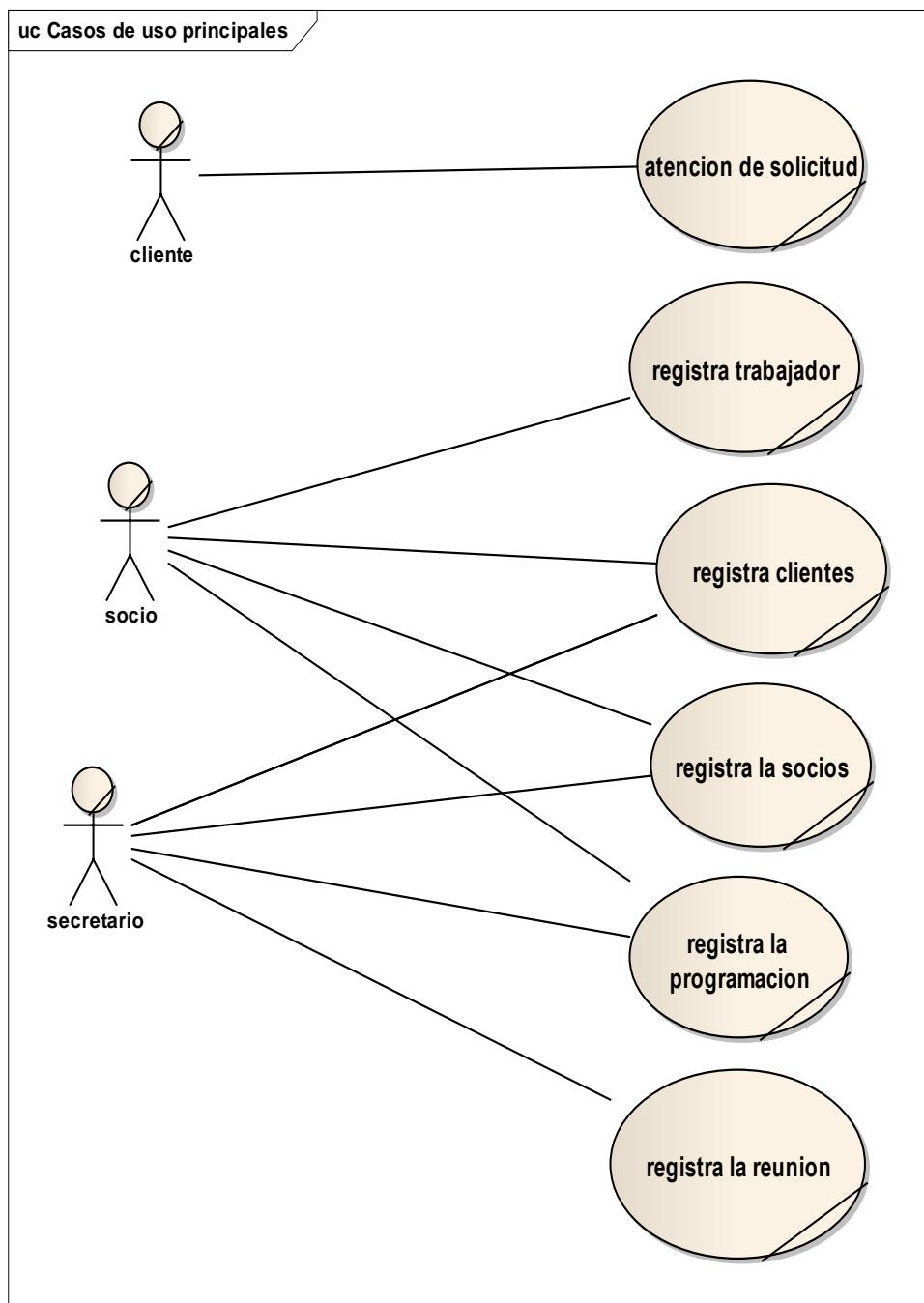


Figura 18 Caso de uso del negocio

### 1.5.1.7.7 Casos de uso

#### 1.5.1.7.7.1 Introducción

El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones, describe lo que hace el sistema para cada tipo de usuario es decir cada forma en los actores usan el sistema se representa con un caso de uso, los mismos que son fragmentos de funcionalidad, especifican una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

#### 1.5.1.7.7.2 Propósito

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

#### 1.5.1.7.7.3 Alcance

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

#### 1.5.1.7.7.4 Diagrama de casos de uso

##### 1.5.1.7.7.4.1 Identificación de actores

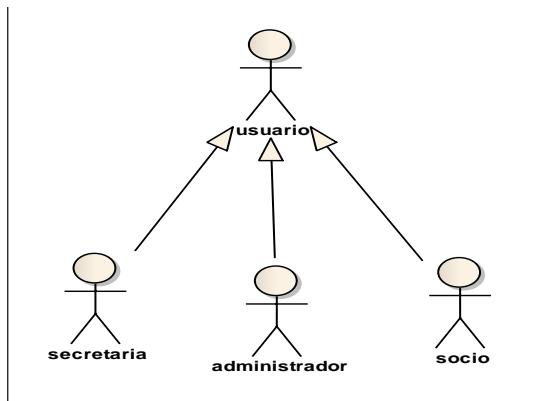


Figura 19 Identificación de actores

#### 1.5.1.7.7.4.2 Casos de uso general del sistema

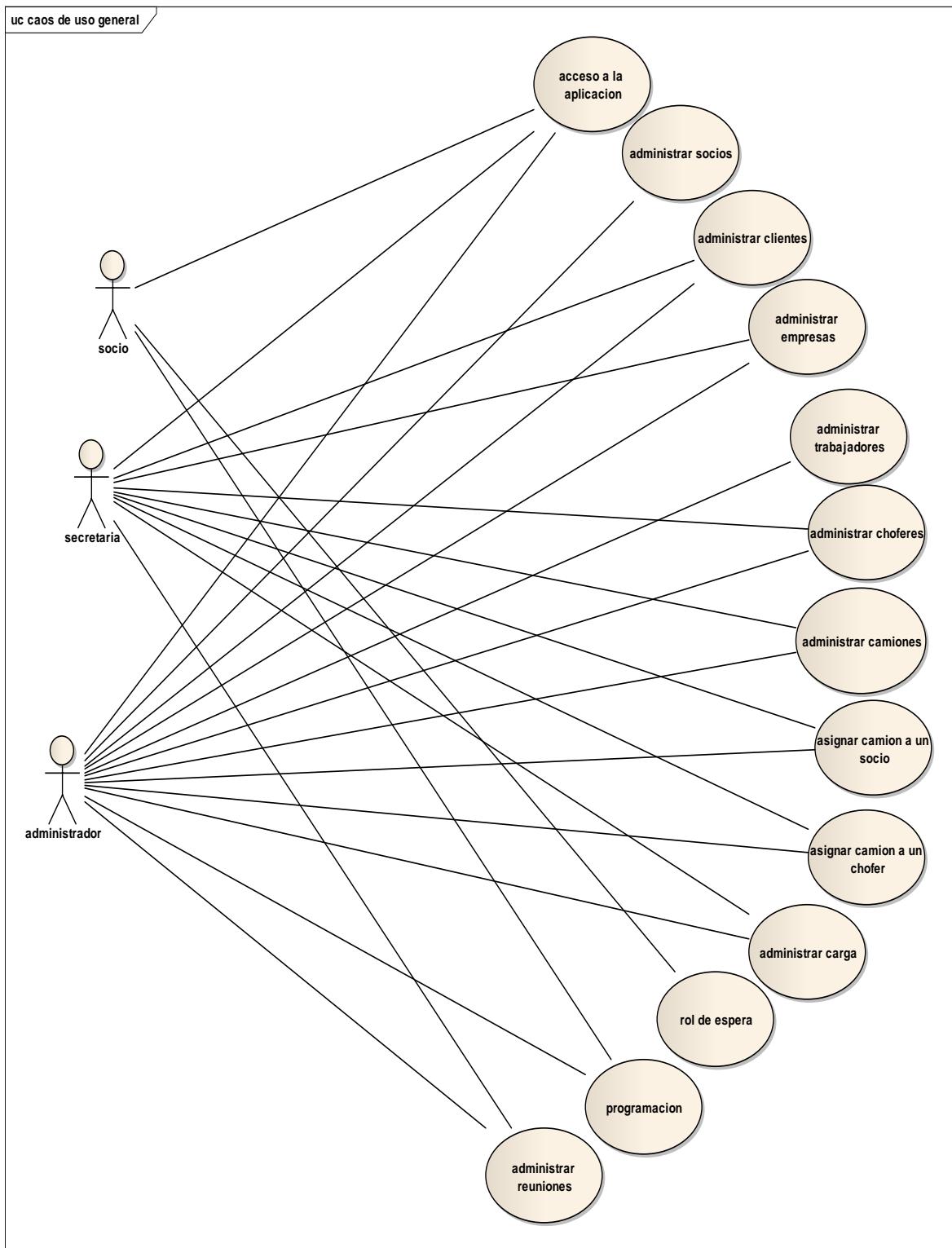


Figura 20 Casos de uso general del sistema

### 1.5.1.7.7.4.3 Explosión de casos de uso

#### 1.5.1.7.7.4.3.1 Caso de uso acceso al sistema

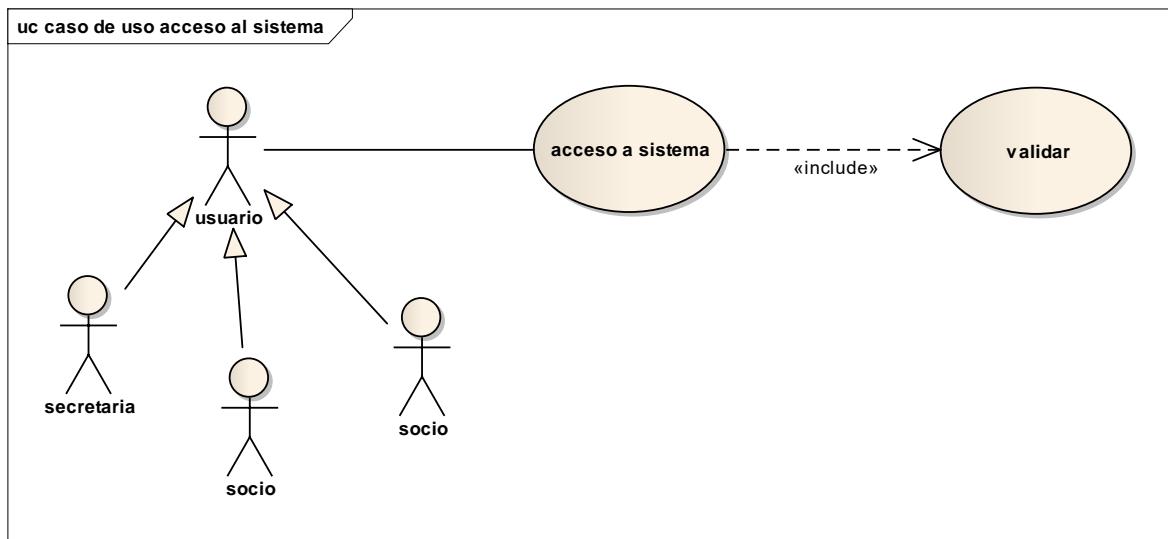


Figura 21 Caso de uso acceso al sistema

Caso de uso	Acceso al sistema
Actores	Secretaria socio administrador
Propósito	Controlar el acceso del personal autorizado para acceder al sistema
Resumen	Este caso de uso hace referencia al control de acceso del usuario autorizado al manejo del sistema, teniendo que introducir el usuario y clave de acceso, cada usuario que está autorizado para acceder al sistema tiene su respectivo rol, que puede ser secretaria, socio y administrador del sistema y de acuerdo a este se habilitarán o deshabilitarán algunas opciones.
Precondiciones	Se despliega la pantalla de “ingreso al sistema” y tener usuario y contraseña para poder ingresar al sistema.

Flujo Normal	<p>Se presenta al usuario la pantalla iniciar sesión al sistema</p> <p>1.- El usuario introduce nombre del usuario y contraseña.</p> <p>2.- Selecciona la opción “Ingresar”.</p> <p>3.- si presiona cancelar se cierra la ventana modal.</p>
Sub flujo	validar datos.

Tabla 17 Caso de uso acceso al sistema

**1.5.1.7.7.4.3.1.1 Caso de uso validar datos**

Caso de uso	Validar datos.
Actores	Administrador , secretaria, socios.
Propósito	Permite validar usuario y contraseña si existen o no.
Resumen	Este caso de uso hace referencia a validar los datos para de usuario y contraseña que sean escritos correctamente y que solo se ingresen números y letras nada mas.
Precondiciones	El usuario debe haber llenado el formulario de ingreso al sistema ingresando su usuario y contraseña.
Flujo Principal	<p>1.- Se validan los datos del usuario y la contraseña va a comparar con los datos de la tabla contrasena.</p> <p>2.- Si existe, se muestra la pantalla principal del sistema.</p>
Flujo alternativo	El sistema no encuentra al usuario y Se muestra mensaje “usuario o clave incorrectos” en la pantalla iniciar sesión

Tabla 18 Caso de uso validar dato

### 1.5.1.7.7.4.3.2 Caso de uso administrar socios

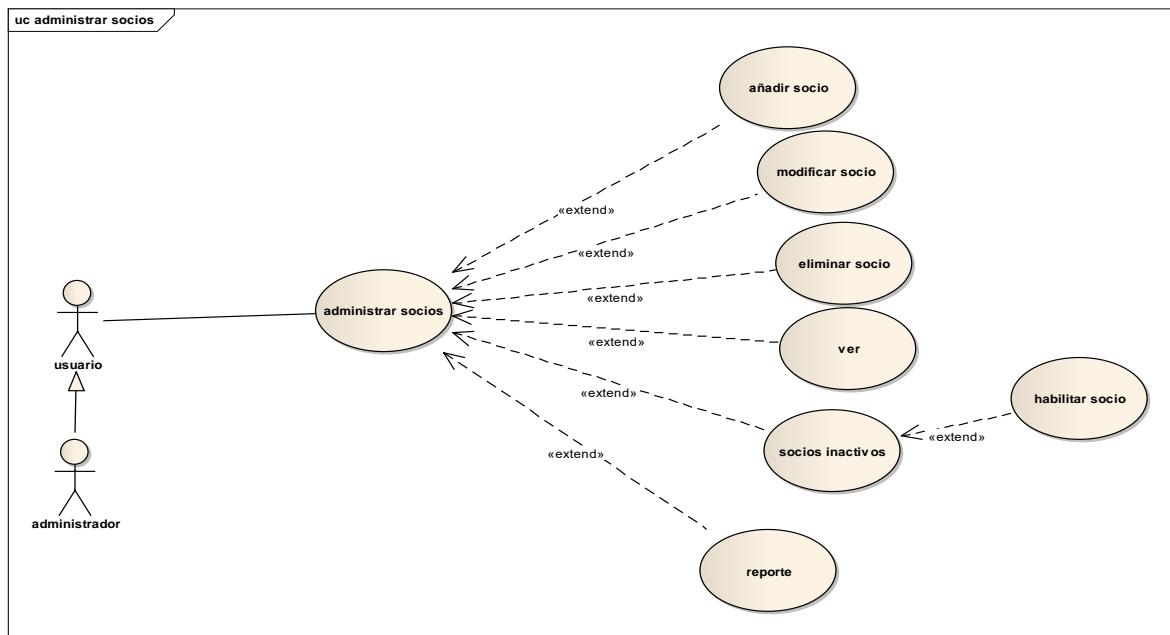


Figura 22 Caso de uso administrar socios

Caso de uso	Administrar socios
Actores	Administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los socios.
Resumen	Este caso de uso hace referencia a la administración de los socios.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar los socios y haber seleccionado del menú de la pantalla principal la opción administrar socios.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar socios con la lista de los socios que están guardados en la tabla usuarios_tra, socios también contara con un filtro de búsqueda para poder buscar a los socios.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los</p>

	diversos subflujos de este caso de uso.
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo socio” se ejecuta el subflujo añadir socio.</p> <p>Si el usuario selecciona la opción “Modificar socio” se ejecuta el subflujo modificar socio.</p> <p>Si el usuario selecciona la opción “Eliminar socio” se ejecuta el subflujo eliminar socio.</p> <p>Si el usuario selecciona la opción “clic en socios inactivos” se ejecuta el subflujo socios inactivos.</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo eliminar ver.</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 19 Caso de uso administrar socios

#### 1.5.1.7.7.4.3.2.1 Caso de uso añadir socio

Caso de uso	Añadir socio
Actores	Administrador
Propósito	Permite el registro de un nuevo socio.
Resumen	Este caso de uso hace referencia a la creación de un nuevo socio.
Precondiciones	El usuario debe haber ingresado al sistema estar autorizado para administrar socios y haber seleccionado la opción nuevo socio la pantalla administrar socios.

Flujo Principal	<p>Se presenta al usuario la pantalla de añadir socio.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla usuario_tra, socios, contraseña, camiones en la base de datos y retorna a la pantalla “administrar socios” si los campos no están completos o son incorrectos se genera una excepción, si el socio ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar socios”.</p>
-----------------	---

Tabla 20 Caso de uso añadir socio

**1.5.1.7.7.4.3.2.2 Caso de uso modificar socio**

Caso de uso	modificar socio
Actores	Administrador.
Propósito	Permite modificar los datos de un socio de la cooperativa.
Resumen	Este caso de uso hace referencia a la modificación de datos de los socios, para corregirlos o actualizarlos.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar socios, debe haber seleccionado de la lista un registro existente de un socio y seleccionar la opción modificar.
Flujo Principal	<p>Se presenta al usuario la pantalla de “modificar socio” con los datos actuales del socio.</p> <p>Se llenan los campos a modificar.</p> <p>Se presentan las siguientes opciones:</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla</p>

	<p>usuario_tra, socios, contraseña, camiones en la base de datos y retorna a la pantalla “administrar socios”, si los campos no están completos o son incorrectos se genera una excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administra socios”.</p>
--	---

Tabla 21 Caso de uso modificar socio

**1.5.1.7.7.4.3.2.3 Caso de uso eliminar socio**

Caso de uso	Eliminar socio
Actores	Administrador
Propósito	Permite eliminar un socio dándole de baja.
Resumen	Este caso de uso hace referencia a la eliminación de los socios, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar socios, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar socio.</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina lógicamente (cambia de estado de 1 a 0) de la tabla usuarios_tra, muestra un mensaje “transacción de realizo correctamente” luego retorna a la pantalla “administrar socios”.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar socios”.</p>

Tabla 22 Caso de uso eliminar socio

#### 1.5.1.7.7.4.3.2.4 Caso de uso ver

Caso de uso	Ver
Actores	Administrador
Propósito	Permite ver todos los datos del socio
Resumen	Este caso de uso hace referencia a ver los datos del socio con todos sus detalles.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar socios, y seleccionar administrar socios y seleccionar un registro existente de la lista y seleccionar la opción ver.
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles de los de los socios que están guardados en la tabla usuarios_tra , socio, contraseña.
Subflujos	Ninguna.

Tabla 23 Caso de uso ver

#### 1.5.1.7.7.4.3.2.5 Casos de uso socios inactivos

Caso de uso	Socios inactivos
Actores	Secretaria , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los socios inactivos.
Resumen	Este caso de uso hace referencia a la administración de los socios inactivos de la cooperativa rio Guadaluquivir.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para socios inactivos y haber seleccionado del menú de la pantalla principal la opción

	administrar socios y luego la opción socios inactivos.
Flujo Principal	<p>Se muestra al usuario la pantalla de socios inactivos con la lista de los socios registrados en la tabla usuarios_tra, socios también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “socios activos” se ejecuta el subflujo administrar socios.</p> <p>Si el usuario selecciona la opción “habilitar socio” se ejecuta el subflujo habilitar socio.</p>
Excepciones	Ninguna.

Tabla 24 Caso de uso socios inactivos

#### 1.5.1.7.7.4.3.2.6 Casos de uso habilitar socio

Caso de uso	Habilitar socio
Actores	Administrador , secretaria
Propósito	Permite hacer habilitar a un socio.
Resumen	Este caso de uso tiene la función de habilitar a un socio cambiar su estado de desactivado a activado.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para habilitar a un socio, seleccionar administrar socios, seleccionar socios inactivos y seleccionar habitar socio.

Flujo Principal	Este caso de uso tiene la función de activar a un socio que se encuentra desactivado en la tabla usuarios_tra.
Subflujos	Ninguna.

Tabla 25 Caso de uso habilitar socio

**1.5.1.7.7.4.3.2.7 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de todos los socios del registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar socios, y seleccionar administrar socios y seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los socios registrados en el la tabla usuarios_tra, socios.
Subflujos	Ninguna.

Tabla 26 Casos de uso reporte

**1.5.1.7.7.4.3.3 Casos de uso administrar clientes**

Caso de uso	Administrar clientes
Actores	Secretaria, socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los clientes.
Resumen	Este caso de uso hace referencia a la administración de los clientes.

Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar los clientes y haber seleccionado del menú de la pantalla principal la opción administrar clientes.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar clientes con la lista de los clientes registrados en las tablas usuarios_tra, clientes, empresas con el rol de clientes también contara con un filtro de búsqueda para poder buscar a los clientes.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo cliente” se ejecuta el subflujo añadir cliente .</p> <p>Si el usuario selecciona la opción “Modificar cliente” se ejecuta el subflujo modificar cliente.</p> <p>Si el usuario selecciona la opción “Eliminar cliente” se ejecuta el subflujo eliminar cliente.</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver</p> <p>Si el usuario selecciona la opción “clientes inactivos” se ejecuta el subflujo clientes inactivos.</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte</p>
Excepciones	Ninguna.

Tabla 27 Casos de uso administrar clientes

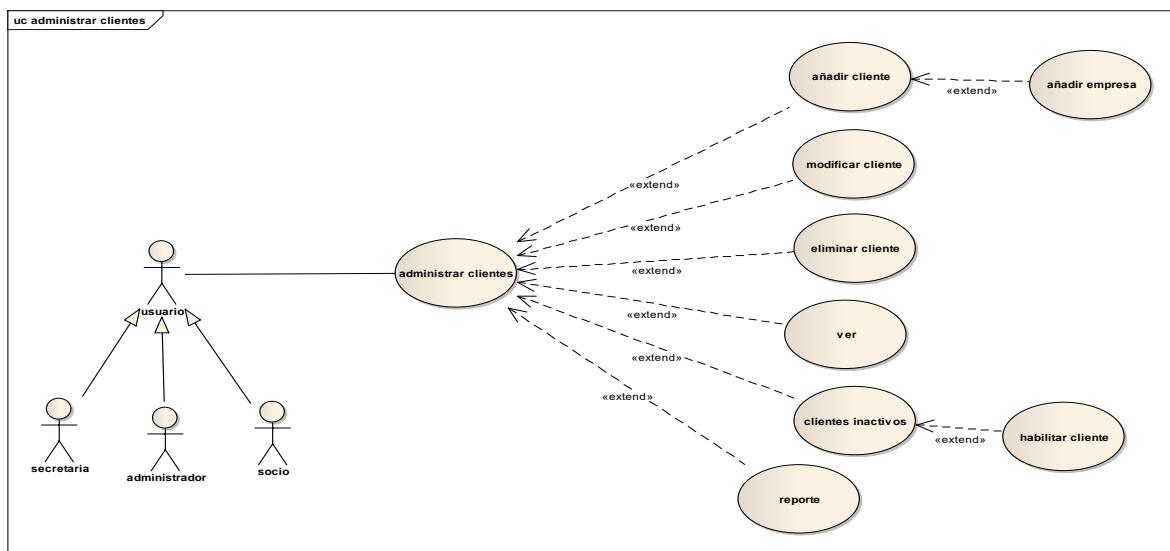


Figura 23 Caso de uso administrar clientes

#### 1.5.1.7.7.4.3.3.1 Casos de uso añadir cliente

Caso de uso	Añadir cliente
Actores	Secretaria, socio, administrador
Propósito	Permite el registro de un nuevo cliente.
Resumen	Este caso de uso hace referencia a la creación de un nuevo cliente de la cooperativa.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar clientes y haber seleccionado la opción añadir cliente en la pantalla administrar clientes.
Flujo Principal	<p>Se presenta al usuario la pantalla de añadir cliente.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de : “Guardar” los datos introducidos se validan y se guardaran en la tabla clientes, usuarios_tra y empresas de la base de datos y retorna a la pantalla “administrar clientes” si los campos no están completos o son incorrectos se genera una excepción, si el cliente ya existe se genera la</p>

	<p>excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar clientes”.</p>
Subflojo	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo empresa” se ejecuta el subflujo añadir empresa.</p>

Tabla 28 Casos de uso añadir cliente

#### 1.5.1.7.7.4.3.3.2 Caso de uso añadir empresa

Caso de uso	Añadir empresa
Actores	Secretaria, socio, administrador
Propósito	Permite el registro de una nueva empresa.
Resumen	Este caso de uso hace referencia a la creación de una nueva empresa
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar clientes y haber seleccionado la opción añadir cliente luego seleccionar añadir empresa.
Flujo Principal	<p>Se presenta al usuario la pantalla de añadir empresa.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla empresas de la base de datos y retorna a la pantalla “añadir cliente” si los campos no están completos o son incorrectos se genera una excepción, si el cliente ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “añadir cliente”.</p>

Tabla 29 caso de uso añadir empresa

### 1.5.1.7.7.4.3.3.3 Casos de uso modificar cliente

Caso de uso	Modificar cliente
Actores	Secretaria, socio, administrador
Propósito	Permite modificar los datos de un cliente.
Resumen	Este caso de uso hace referencia a la modificación de datos de los clientes, para corregirlos o actualizarlos.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar clientes, debe haber seleccionado de la lista un registro existente de un cliente y seleccionar la opción modificar.
Flujo Principal	<p>Se presenta al usuario la pantalla de “modificar cliente” de un cliente con los datos actuales del cliente.</p> <p>Se llenan los campos a modificar.</p> <p>Se presentan las siguientes opciones:</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla usuarios_tra, clientes, empresas en la base de datos y retorna a la pantalla “administrar clientes”, si los campos no están completos o son incorrectos se genera una excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administra clientes”.</p>

Tabla 30 Casos de uso modificar cliente

### 1.5.1.7.7.4.3.3.4 Casos de uso eliminar cliente

Caso de uso	Eliminar cliente
Actores	Secretaria, socio, administrador
Propósito	Permite eliminar un cliente dándole de baja.

Resumen	Este caso de uso hace referencia a la eliminación de los clientes, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar clientes, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar cliente.</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina lógicamente de la tabla usuarios_tra de la base de datos, muestra un mensaje de “se eliminó correctamente” luego retorna a la pantalla “administrar clientes”</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar clientes”.</p>

Tabla 31 Casos de uso eliminar cliente

#### 1.5.1.7.7.4.3.3.5 Caso de uso ver

Caso de uso	Ver
Actores	Administrador , socios, secretaria
Propósito	Permite ver todos los datos de los clientes
Resumen	Este caso de uso hace referencia a ver los datos del cliente con todos sus detalles
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar cliente, y seleccionar administrar clientes y seleccionar un registro existente de la lista y seleccionar la opción ver
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles del cliente

	guardados en la tabla usuarios_tra, clientes, empresas
Subflujos	Ninguna.

Tabla 32 Caso de uso ver

### 1.5.1.7.7.4.3.3.6 Casos de uso clientes inactivos

Caso de uso	Clientes inactivos
Actores	Secretaria , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los clientes inactivos.
Resumen	Este caso de uso hace referencia a la administración de los clientes inactivos de la cooperativa rio Guadaluquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para clientes inactivos y haber seleccionado del menú de la pantalla principal la opción administrar clientes y luego la opción clientes inactivos.
Flujo Principal	<p>Se muestra al usuario la pantalla de clientes inactivos con la lista de los clientes registrados en la tabla usuarios_tra, clientes también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “clientes activos” se ejecuta el subflujo administrar clientes</p> <p>Si el usuario selecciona la opción “habilitar cliente” se ejecuta el subflujo habilitar cliente</p>

Excepciones	Ninguna.
-------------	----------

Tabla 33 Caso de uso clientes inactivos

**1.5.1.7.7.4.3.3.7 Casos de uso habilitar cliente**

Caso de uso	Habilitar cliente
Actores	Administrador , secretaria.
Propósito	Permite hacer habilitar a un cliente.
Resumen	Este caso de uso tiene la función de habilitar a un cliente cambiar su estado de desactivado ha activado.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para habilitar a un cliente, seleccionar administrar clientes, seleccionar clientes inactivos y seleccionar habitar cliente.
Flujo Principal	Este caso de uso tiene la función de activar a un cliente que se encuentra desactivado en la tabla usuarios_tra
Subflujos	Ninguna.

Tabla 34 Caso de uso habilitar cliente

**1.5.1.7.7.4.3.3.8 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de todos los clientes registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar clientes, y seleccionar administrar clientes y seleccionar

	reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los clientes registrados en la tabla usuarios_tra, clientes, empresas
Subflujos	Ninguna.

Tabla 35 Casos de uso reporte

#### 1.5.1.7.7.4.3.4 Casos de uso administrar empresas

Caso de uso	Administrar empresas
Actores	Secretaria, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de las empresas
Resumen	Este caso de uso hace referencia a la administración de las empresas de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar empresas y haber seleccionado del menú de la pantalla principal la opción administrar empresas.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar empresas con la lista de las empresas registrados en la tabla empresas también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nueva empresa” se ejecuta el subflujo añadir empresa.</p> <p>Si el usuario selecciona la opción “Modificar empresa” se ejecuta el</p>

	<p>subflujo modificar empresa</p> <p>Si el usuario selecciona la opción “Eliminar empresa” se ejecuta el subflujo eliminar empresa.</p> <p>Si el usuario selecciona la opción “empresa inactivas” se ejecuta el subflujo empresas inactivas.</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 36 Caso de uso administrar empresas

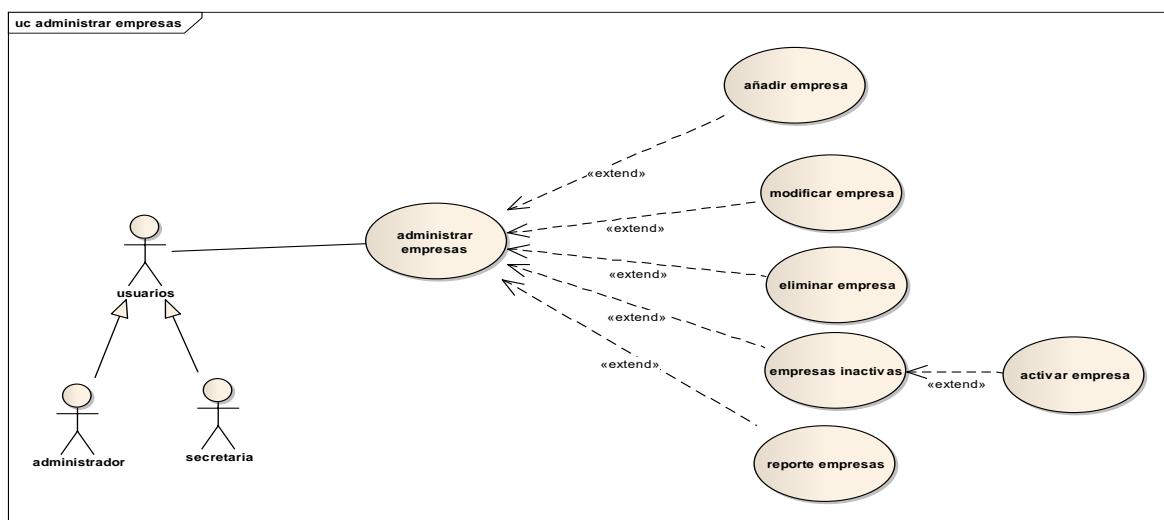


Figura 24 Casos de uso administrar empresas

#### 1.5.1.7.7.4.3.4.1 Casos de uso añadir empresa

Caso de uso	Añadir empresa
Actores	Secretaria, administrador
Propósito	Permite el registro de una nueva empresa.
Resumen	Este caso de uso hace referencia a la creación de una nueva empresa

Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar empresas y haber seleccionado la opción añadir empresa en la pantalla administrar empresas.
Flujo Principal	<p>Se presenta al usuario la pantalla de añadir empresa.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla empresas de la base de datos y retorna a la pantalla “administrar empresas” si los campos no están completos o son incorrectos se genera una excepción, si la empresa ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar empresas”.</p>
Subfijo	ninguno

Tabla 37 Casos de uso añadir empresa

#### 1.5.1.7.7.4.3.4.2 Casos de uso modificar empresa

Caso de uso	Modificar empresa
Actores	Secretaria, administrador
Propósito	Permite modificar los datos de una empresa.
Resumen	Este caso de uso hace referencia a la modificación de datos de una empresa, para corregirlos o actualizarlos.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar empresas, debe haber seleccionado de la lista un registro existente de una empresa y seleccionar la opción modificar.
Flujo Principal	Se presenta al usuario la pantalla de “modificar empresa” con los datos actuales de la empresa.

	<p>Se llenan los campos a modificar.</p> <p>Se presentan las siguientes opciones:</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla empresas en la base de datos y retorna a la pantalla “administrar empresas”, si los campos no están completos o son incorrectos se genera una excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administra empresas”.</p>
--	---

Tabla 38 Casos de uso modificar empresa

**1.5.1.7.7.4.3.4.3 Casos de uso eliminar empresa**

Caso de uso	Eliminar empresa
Actores	Secretaria, administrador
Propósito	Permite eliminar una empresa dándole de baja.
Resumen	Este caso de uso hace referencia a la eliminación de las empresas, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar empresas, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar empresa.</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro cambiara de estado de 1 a 0 de la tabla empresas de la base de datos, muestra un mensaje de “se eliminó correctamente” luego retorna a la pantalla “administrar empresas”</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla</p>

	“administrar empresas”.
--	-------------------------

Tabla 39 Casos de uso eliminar empresa

**1.5.1.7.7.4.3.4.4 Casos de uso empresas inactivas**

Caso de uso	Empresas inactivas
Actores	Secretaria , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los empresas inactivas.
Resumen	Este caso de uso hace referencia a la administración de las empresas inactivas de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para empresas inactivas y haber seleccionado del menú de la pantalla principal la opción administrar empresas y luego la opción empresas inactivas.
Flujo Principal	<p>Se muestra al usuario la pantalla de empresas inactivas con la lista de las empresas registrados en la tabla empresas también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “empresas activos” se ejecuta el subflujo administrar empresas</p> <p>Si el usuario selecciona la opción “habilitar empresa” se ejecuta el subflujo habilitar empresa</p>
Excepciones	Ninguna.

Tabla 40 Casos de uso empresas inactivas

#### **1.5.1.7.7.4.3.4.5 Casos de uso habilitar empresa**

Caso de uso	Habilitar empresa
Actores	Administrador , secretaria
Propósito	Permite hacer habilitar a una empresa
Resumen	Este caso de uso tiene la función de habilitar a una empresa cambiar su estado de desactivado ha activado
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para habilitar a una empresa, seleccionar administrar empresa, seleccionar empresas inactivas y seleccionar habitar empresa
Flujo Principal	Este caso de uso tiene la función de activar a una empresa que se encuentra desactivado en la tabla empresas
Subflujos	Ninguna.

Tabla 41 Casos de uso habilitar empresa

#### **1.5.1.7.7.4.3.4.6 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de todas las empresas registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar empresas, y seleccionar administrar empresas y seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de las empresas

	registrados en la tabla empresas
Subflujos	Ninguna.

Tabla 42 Casos de uso reporte

### 1.5.1.7.7.4.3.5 Casos de uso administrar trabajadores

Caso de uso	Administrar trabajadores
Actores	Administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los trabajadores.
Resumen	Este caso de uso hace referencia a la administración de los trabajadores de la cooperativa rio Guadalquivir.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar trabajadores y haber seleccionado del menú de la pantalla principal la opción administrar trabajadores.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar trabajadores con la lista de los trabajadores registrados en la tabla usuarios_tra también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo trabajador” se ejecuta el subflujo añadir trabajador.</p> <p>Si el usuario selecciona la opción “Modificar trabajador” se ejecuta el</p>

	<p>subflujo modificar trabajador.</p> <p>Si el usuario selecciona la opción “Eliminar trabajador” se ejecuta el subflujo eliminar trabajador.</p> <p>Si el usuario selecciona la opción “trabajadores inactivos” se ejecuta el subflujo trabajadores inactivos</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 43 Casos de uso administrar trabajadores

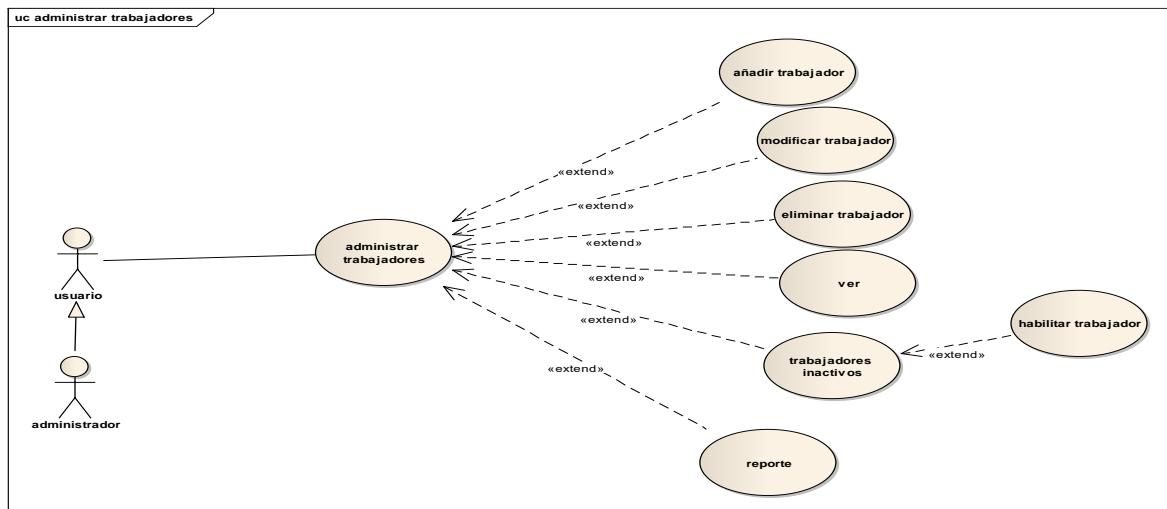


Figura 25 Casos de uso administrar trabajadores

#### 1.5.1.7.7.4.3.5.1 Casos de uso añadir trabajador

Caso de uso	Añadir trabajador
Actores	Administrador, socio
Propósito	Permite el registro de un nuevo trabajador

Resumen	Este caso de uso hace referencia a la creación de un nuevo trabajador
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar trabajador.” y haber seleccionado la opción nuevo trabajador en la pantalla administrar trabajador.
Flujo Principal	<p>Se presenta al usuario la pantalla de “añadir trabajador”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla usuarios_tra de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar trabajador.”, si los campos no están completos o son incorrectos se genera una excepción, si el trabajador ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar trabajador”,</p>
Subfluxos	Ninguna

Tabla 44 Casos de uso añadir trabajador

#### 1.5.1.7.7.4.3.5.2 Caso de uso modificar trabajador

Caso de uso	Modificar trabajador
Actores	Administrador, socio
Propósito	Permite Modificar trabajador
Resumen	Este caso de uso hace Modificar trabajador
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar trabajadores” y haber seleccionado la opción Modificar trabajador en la pantalla administrar trabajador..
Flujo Principal	Se presenta al usuario la pantalla de “Modificar trabajador.”.

	<p>Luego se completan los datos que hay en pantalla y salen opciones de : “Guardar” los datos introducidos se validan y se guardaran en la tabla usuarios_tra de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar trabajador”, si los campos no están completos o son incorrectos se genera una excepción, si la trabajador ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar trabajador”,</p>
Subflujos	Ninguna

Tabla 45 Caso de uso modificar trabajador

#### 1.5.1.7.7.4.3.5.3    Caso de uso eliminar trabajador

Caso de uso	Eliminar trabajador
Actores	Administrador socio
Propósito	Permite eliminar trabajador.
Resumen	Este caso de uso hace referencia a la eliminación trabajador, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar trabajador, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar trabajador con el nombre de la eliminar trabajador</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina lógicamente de la tabla usuarios_tra de la base de datos y muestra mensaje de “su transacción se realizó con exito” y se retorna a la pantalla “administrar trabajadores”.</p>

	“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar trabajador”.
Subflujos	Ninguna.

Tabla 46 Caso de uso eliminar trabajador

**1.5.1.7.7.4.3.5.4 Caso de uso ver**

Caso de uso	Ver
Actores	Administrador , socio
Propósito	Permite ver todos los datos de los trabajadores
Resumen	Este caso de uso hace referencia a ver los datos del trabajador con todos sus detalles
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar trabajadores, y seleccionar administrar trabajadores y seleccionar un registro existente de la lista y seleccionar la opción ver
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles del trabajador guardados en la tabla usuarios_tra
Subflujos	Ninguna.

Tabla 47 Caso de uso ver

**1.5.1.7.7.4.3.5.5 Casos de uso trabajadores inactivos**

Caso de uso	Trabajadores inactivos
Actores	Administrador

Propósito	Permite visualizar y realizar diferentes acciones en una lista de los trabajadores inactivos.
Resumen	Este caso de uso hace referencia a la administración de los trabajadores inactivos de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para trabajadores inactivos y haber seleccionado del menú de la pantalla principal la opción administrar trabajadores y luego la opción trabajadores inactivos.
Flujo Principal	<p>Se muestra al usuario la pantalla de trabajadores inactivos con la lista de los trabajadores registrados en la tabla usuarios_tra también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “trabajadores activos” se ejecuta el subflujo administrar trabajadores</p> <p>Si el usuario selecciona la opción “habilitar trabajador” se ejecuta el subflujo habilitar trabajador</p>
Excepciones	Ninguna.

Tabla 48 Casos de uso trabajadores inactivos

#### 1.5.1.7.7.4.3.5.6 Casos de uso habilitar trabajador

Caso de uso	Habilitar trabajador
Actores	Administrador

Propósito	Permite habilitar a un trabajador
Resumen	Este caso de uso tiene la función de habilitar trabajadores cambiar su estado de desactivado a activado
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para habilitar trabajador, seleccionar administrar trabajadores, seleccionar trabajadores inactivos y seleccionar habilitar trabajador
Flujo Principal	Este caso de uso tiene la función de activar a un trabajador que se encuentra desactivado en la tabla usuario_tra
Subflujos	Ninguna.

Tabla 49 Casos de uso habilitar trabajador

#### 1.5.1.7.7.4.3.5.7 Casos de uso reporte

Caso de uso	Reporte
Actores	Administrador , socio
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se genera un reporte de todos los trabajadores registrados en el sistema.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar trabajadores, y seleccionar administrar trabajadores y seleccionar reporte.
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los trabajadores registrados en la tabla usuarios_tra
Subflujos	Ninguna.

Tabla 50 Casos de uso reporte

### 1.5.1.7.7.4.3.6 Casos de uso administrar choferes

Caso de uso	Administrador choferes
Actores	Socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los choferes.
Resumen	Este caso de uso hace referencia a la administración de los choferes de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar choferes y haber seleccionado del menú de la pantalla principal la opción administrar trabajadores.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar choferes con la lista de los choferes registrados en la tabla usuarios_tra también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo chofer” se ejecuta el subflujo añadir chofer.</p> <p>Si el usuario selecciona la opción “Modificar trabajador” se ejecuta el subflujo modificar trabajador.</p> <p>Si el usuario selecciona la opción “Eliminar trabajador” se ejecuta el subflujo eliminar trabajador.</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver</p> <p>Si el usuario selecciona la opción “choferes inactivos” se ejecuta el</p>

	subflujo choferes inactivos  Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.
Excepciones	Ninguna.

Tabla 51 Casos de uso administrar choferes

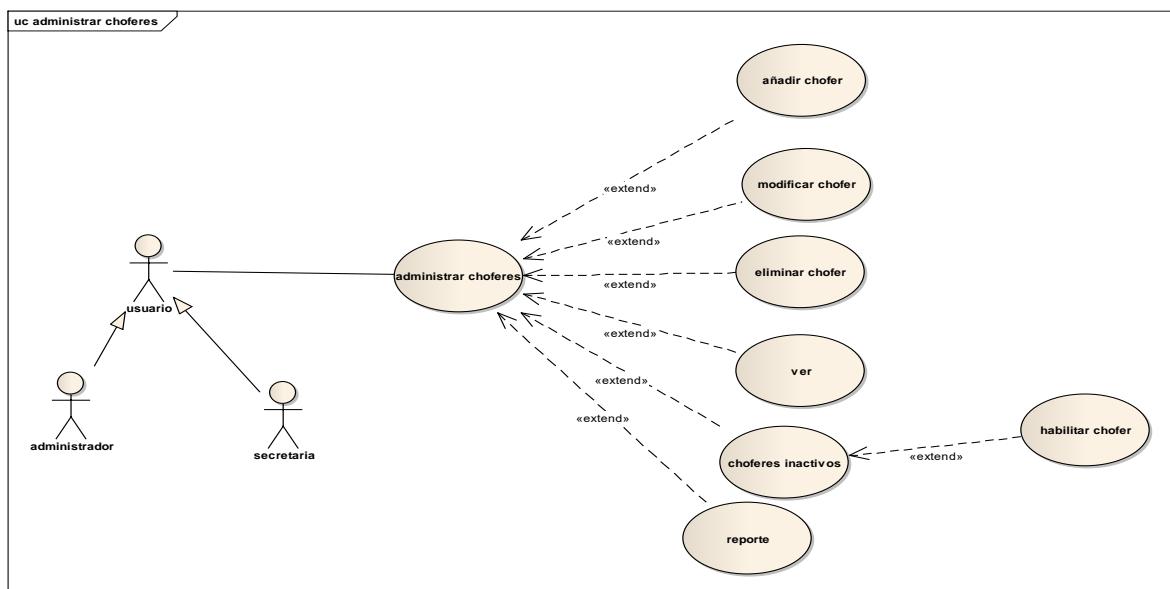


Figura 26 Casos de uso administrar choferes

#### 1.5.1.7.7.4.3.6.1 Casos de uso añadir chofer

Caso de uso	Añadir chofer
Actores	Administrador , Secretaria
Propósito	Permite el registro de un nuevo chofer
Resumen	Este caso de uso hace referencia a la creación de una nuevo chofer
Precondiciones	El usuario debe haber ingresado a sistema, estar autorizado para “administrar choferes.” y haber seleccionado la opción nuevo chofer. en

	la pantalla administrar choferes.
Flujo Principal	<p>Se presenta al usuario la pantalla de “añadir chofer”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla usuarios_tra, choferes de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar choferes.”, si los campos no están completos o son incorrectos se genera una excepción, si el chofer ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar choferes”,.</p>
Subflujos	Ninguna

Tabla 52 Casos de uso añadir chofer

**1.5.1.7.7.4.3.6.2 Casos de uso modificar chofer**

Caso de uso	Modificar chofer
Actores	Administrador , socio
Propósito	Permite Modificar chofer
Resumen	Este caso de uso hace Modificar chofer
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar choferes” y haber seleccionado la opción Modificar chofer en la pantalla administrar choferes..
Flujo Principal	<p>Se presenta al usuario la pantalla de “Modificar chofer.”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla usuarios_tra de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar choferes”, si los</p>

	campos no están completos o son incorrectos se genera una excepción, si la chofer ya existe se genera la excepción.  “Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar choferes”,
Subflujos	Ninguna

Tabla 53 Casos de uso modificar chofer

**1.5.1.7.7.4.3.6.3 Casos de uso eliminar chofer**

Caso de uso	Eliminar chofer
Actores	Administrador secretaria
Propósito	Permite eliminar chofer.
Resumen	Este caso de uso hace referencia a la eliminación chofer, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar choferes, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	Se muestra al usuario la pantalla de dialogo de confirmación eliminar chofer con el nombre del chofer  El usuario puede seleccionar las siguientes opciones:  “Aceptar” El registro se elimina lógicamente de la tabla choferes de la base de datos y muestra mensaje de “su transacción se realizó con exito” y se retorna a la pantalla “administrar choferes”.  “Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar choferes”.
Subflujos	Ninguna.

Tabla 54 Casos de uso eliminar chofer

#### 1.5.1.7.7.4.3.6.4 Casos de uso ver

Caso de uso	Ver
Actores	Administrador , secretaria
Propósito	Permite ver todos los datos de los choferes
Resumen	Este caso de uso hace referencia a ver los datos del chofer con todos sus detalles
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar choferes, y seleccionar administrar choferes y seleccionar un registro existente de la lista y seleccionar la opción ver
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles del chofer guardados en la tabla usuarios_tra
Subflujos	Ninguna.

Tabla 55 Casos de uso ver

#### 1.5.1.7.7.4.3.6.5 Casos de uso choferes inactivos

Caso de uso	Choferes inactivos
Actores	Secretaria , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los choferes inactivos.
Resumen	Este caso de uso hace referencia a la administración de los choferes inactivos de la cooperativa rio Guadaluquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para choferes inactivos y haber seleccionado del menú de la pantalla principal la opción administrar choferes y luego la opción choferes inactivos.

Flujo Principal	<p>Se muestra al usuario la pantalla de choferes inactivos con la lista de los choferes registrados en la tabla usuarios_tra también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “choferes activos” se ejecuta el subflujo administrar choferes</p> <p>Si el usuario selecciona la opción “transferencia” se ejecuta el subflujo transferencia</p> <p>Si el usuario selecciona la opción “reporte inactivos” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 56 Casos de uso choferes inactivos

#### 1.5.1.7.7.4.3.6.6 Casos de uso habilitar chofer

Caso de uso	Habilitar chofer
Actores	Administrador, Secretaria
Propósito	Permite habilitar a un chofer
Resumen	Este caso de uso tiene la función de habilitar chofer cambiar su estado de desactivado a activado
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para habilitar chofer, seleccionar administrar choferes, seleccionar choferes inactivos y seleccionar habitar chofer

Flujo Principal	Este caso de uso tiene la función de activar a un chofer de la tabla choferes de la base de datos que se encuentra desactivado ya sea por equivocación
Subflujos	Ninguna.

Tabla 57 Casos de uso habilitar chofer

**1.5.1.7.7.4.3.6.7 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador , secretaria
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de todos los choferes registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar choferes, y seleccionar administrar choferes y seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los choferes registrados en la tabla usuarios_tra
Subflujos	Ninguna.

Tabla 58 Casos de uso reporte

**1.5.1.7.7.4.3.7 Casos de uso administrar camiones**

Caso de uso	Administrar camiones
Actores	Socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los camiones.

Resumen	Este caso de uso hace referencia a la administración de los camiones de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar camiones y haber seleccionado del menú de la pantalla principal la opción administrar camiones.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar camiones con la lista de los camiones registrados en la tabla camiones también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo camión” se ejecuta el subflujo añadir camión.</p> <p>Si el usuario selecciona la opción “modificar camión” se ejecuta el subflujo modificar camión.</p> <p>Si el usuario selecciona la opción “eliminar camión” se ejecuta el subflujo eliminar camión.</p> <p>Si el usuario selecciona la opción “camiones inactivos” se ejecuta el subflujo camiones inactivos</p> <p>Si el usuario selecciona la opción “habilitar camión” se ejecuta el subflujo habilitar camión</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 59 Casos de uso administra camiones

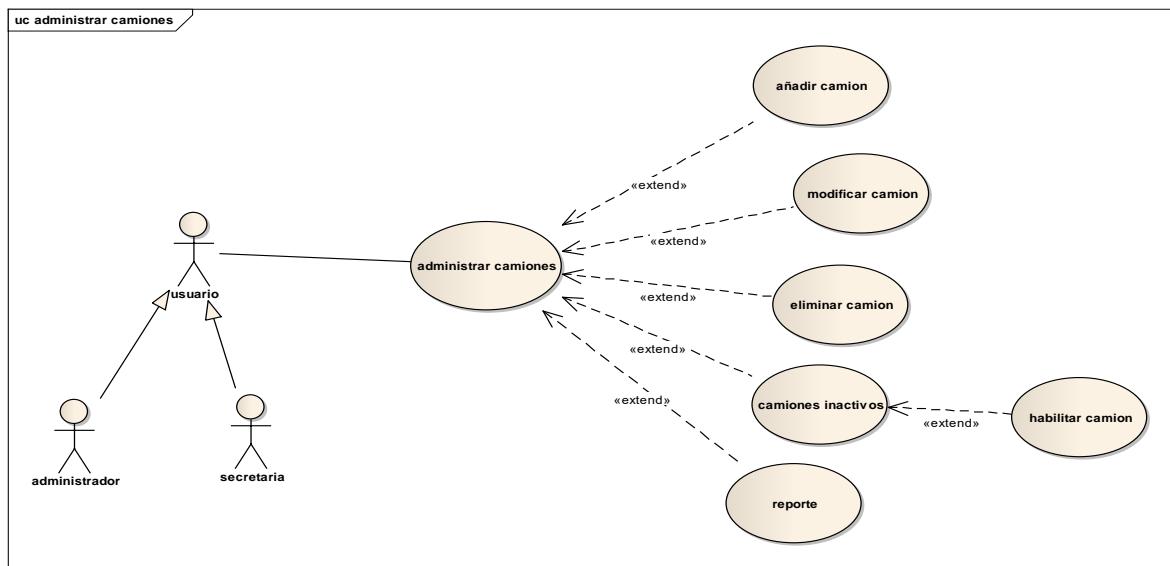


Figura 27 Casos de uso administrar choferes

#### 1.5.1.7.7.4.3.7.1 Caso de uso añadir camión

Caso de uso	Añadir camión
Actores	Administrador , secretaria
Propósito	Permite el registro de un nuevo camión
Resumen	Este caso de uso hace referencia a la creación de una nuevo camión
Precondiciones	El usuario debe haber ingresado a sistema, estar autorizado para “administrar camiones.” y haber seleccionado la opción nuevo camión. en la pantalla administrar camiones.
Flujo Principal	<p>Se presenta al usuario la pantalla de “añadir camión”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla camiones de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar camiones.”, si los campos no están completos o son incorrectos se genera una excepción, si</p>

	el camión ya existe se genera la excepción.  “Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar camiones”,.
Subflujos	Ninguna

Tabla 60 Casos de uso añadir camión

**1.5.1.7.7.4.3.7.2 Casos de uso modificar camión**

Caso de uso	Modificar camión
Actores	administrador, secretaria
Propósito	Permite Modificar camión
Resumen	Este caso de uso hace Modificar camión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar camiones” y haber seleccionado la opción Modificar camión en la pantalla administrar camiones..
Flujo Principal	<p>Se presenta al usuario la pantalla de “Modificar camión.”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla camiones de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar camiones”, si los campos no están completos o son incorrectos se genera una excepción, si la chofer ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar camiones”,</p>
Subflujos	Ninguna

Tabla 61 Casos de uso modificar camión

### 1.5.1.7.7.4.3.7.3 Casos de uso eliminar camión

Caso de uso	Eliminar camión
Actores	Administrador secretaria
Propósito	Permite eliminar camión.
Resumen	Este caso de uso hace referencia a la eliminación de un camión, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar camiones, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar camión con la placa del camión</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina lógicamente de la tabla camiones de la base de datos y muestra mensaje de “su transacción se realizó con exito” y se retorna a la pantalla “administrar camiones”.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar camiones”.</p>
Subflujos	Ninguna.

Tabla 62 Casos de uso eliminar camión

### 1.5.1.7.7.4.3.7.4 Casos de uso camiones inactivos

Caso de uso	Camiones inactivos
Actores	Socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los camiones inactivos.

Resumen	Este caso de uso hace referencia a la administración de los camiones inactivos de la cooperativa rio Guadaluquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para camiones inactivos y haber seleccionado del menú de la pantalla principal la opción administrar camiones y luego la opción camiones inactivos.
Flujo Principal	<p>Se muestra al usuario la pantalla de camiones inactivos con la lista de los camiones registrados en la tabla camiones también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “camiones activos” se ejecuta el subflujo administrar camiones</p> <p>Si el usuario selecciona la opción “habilitar camión ” se ejecuta el subflujo habilitar camión</p>
Excepciones	Ninguna.

Tabla 63 Casos de uso camiones inactivos

#### 1.5.1.7.7.4.3.7.5 Casos de uso habilitar camión

Caso de uso	Habilitar camión
Actores	Administrador, Secretaria
Propósito	Permite habilitar a un camión
Resumen	Este caso de uso tiene la función de habilitar camión cambiar su estado de desactivado a activado

Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para habilitar camión, seleccionar administrar camiones, seleccionar camiones inactivos y seleccionar habitar camión
Flujo Principal	Este caso de uso tiene la función de activar a un camión de la tabla camiones de la base de datos que se encuentra desactivado ya sea por equivocación
Subflujos	Ninguna.

Tabla 64 Casos de uso habilitar camión

**1.5.1.7.7.4.3.7.6 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador , secretaria
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de todos los camiones activos registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar camiones, seleccionar administrar camiones, seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los camiones activos registrados en la tabla camiones
Subflujos	Ninguna.

Tabla 65 Casos de uso reporte

**1.5.1.7.7.4.3.8 Casos de uso asignar camión a un socio**

Caso de uso	Asignar camión a un socio
-------------	---------------------------

Actores	Secretaria , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los camiones con sus dueños(socios).
Resumen	Este caso de uso hace referencia a la administración de los camiones de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para asignar camión a un socio y haber seleccionado del menú de la pantalla principal la opción asignar camión a un socio
Flujo Principal	<p>Se muestra al usuario la pantalla de asignar camión a un socio con la lista de los camiones con su dueños (socio) registrados en la tabla sociocamion ,socios, camiones también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nueva asignación” se ejecuta el subflujo añadir asignación.</p> <p>Si el usuario selecciona la opción “eliminar asignación” se ejecuta el subflujo eliminar asignación.</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 66 Casos de uso asignar camión a un socio

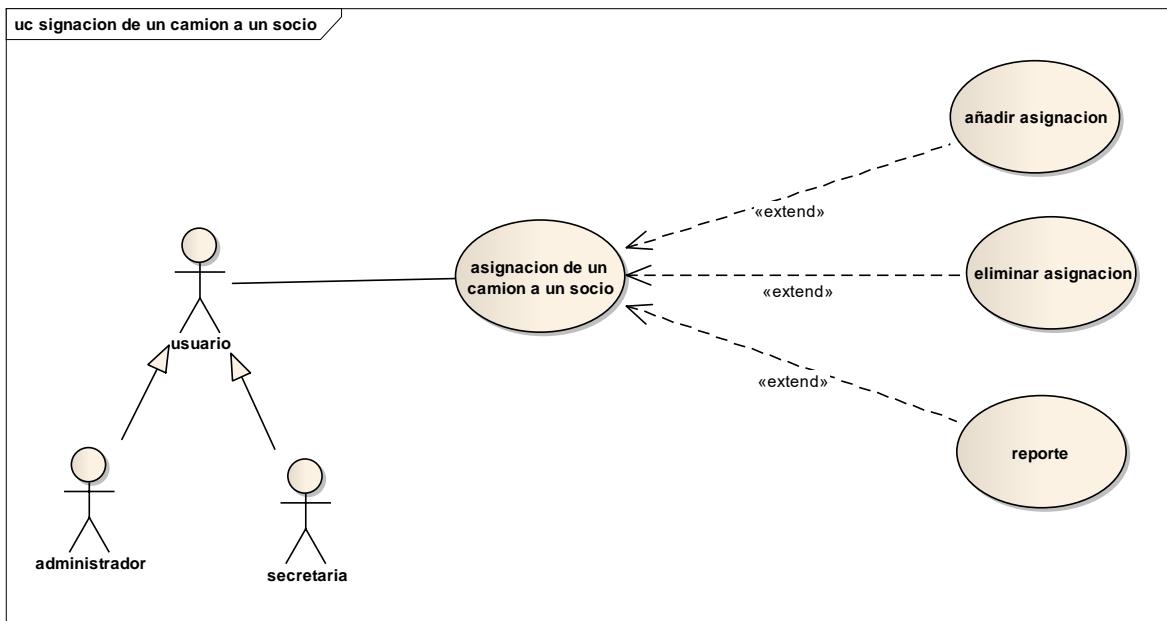


Figura 28 Casos de uso asignar camión a un socio

#### 1.5.1.7.7.4.3.8.1 Caso de uso añadir asignación

Caso de uso	Añadir asignación
Actores	Administrador, secretaria
Propósito	Permite la asignación de un camión a un socio
Resumen	Este caso de uso hace referencia a la asignación de un camión a un socio
Precondiciones	El usuario debe haber ingresado a sistema, estar autorizado para “asignar camión a un socio.” y haber seleccionado la opción nueva asignación. en la pantalla asignar camión a un socio.
Flujo Principal	<p>Se presenta al usuario la pantalla de “nueva asignación”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla sociocamion de la base de datos, muestra mensaje de “su transacción se</p>

	realizó con éxito” y retorna a la pantalla “asignar camión a un socio”, si los campos no están completos o son incorrectos se genera una excepción, si el camión ya existe se genera la excepción.  “Cancelar” no se realiza ningún cambio. Retornando a la pantalla “asignar camión a un socio”,.
Subflujos	Ninguna

Tabla 67 Casos de uso añadir asignación

**1.5.1.7.7.4.3.8.2 Casos de uso eliminar asignación**

Caso de uso	Eliminar asignación
Actores	Administrador secretaria
Propósito	Permite eliminar una asignación.
Resumen	Este caso de uso hace referencia a la eliminación de una, asignación. Que el socio este sin camión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para asignar camión a un socio, seleccionar un registro existente de la lista y seleccionar la opción eliminar asignación.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar asignación</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina lógicamente de la tabla sociocamion de la base de datos y muestra mensaje de “su transacción se realizó con exito” y se retorna a la pantalla “asignar camión a un socio”.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “asignar camión a un socio”.</p>

Subflujos	Ninguna.
-----------	----------

Tabla 68 Casos de uso eliminar asignación

**1.5.1.7.7.4.3.8.3 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador , secretaria
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso seara un reporte de todos los camiones con su dueño(socio) registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para asignar camión a un socio, seleccionar asignar camión a un socio, seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los camiones y sus dueños(socios) registrados en la tabla camiones, sociocamion, socios, usuarios_tra
Subflujos	Ninguna.

Tabla 69 Casos de uso reporte

**1.5.1.7.7.4.3.9 Casos de uso asignar camión a un chofer**

Caso de uso	Asignar camión a un chofer
Actores	Secretaria , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los camiones con su chofer

Resumen	Este caso de uso hace referencia a la asignar camión a un chofer de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para asignar camión a un chofer y haber seleccionado del menú de la pantalla principal la opción asignar camión a un chofer
Flujo Principal	<p>Se muestra al usuario la pantalla de asignar camión a un chofer con la lista de los camiones con su chofer registrados en la tabla chofer_camion, camiones, choferes, usuarios_tra también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nueva asignación” se ejecuta el subflujo añadir asignación.</p> <p>Si el usuario selecciona la opción “eliminar asignación” se ejecuta el subflujo eliminar asignación.</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 70 Casos de uso asignar camión a un chofer

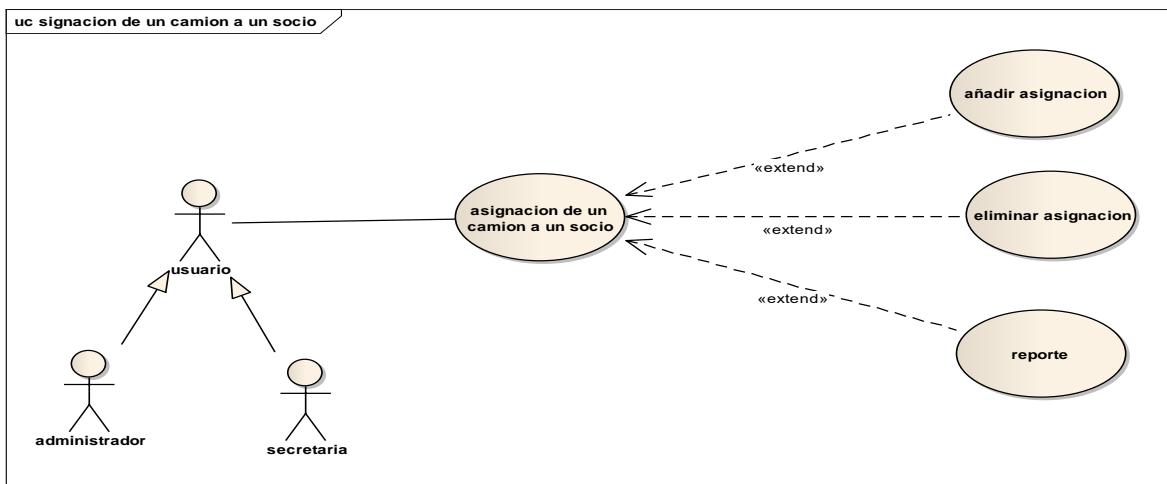


Figura 29 Casos de uso asignar camión a un chofer

#### 1.5.1.7.7.4.3.9.1 Caso de uso añadir asignación

Caso de uso	Añadir asignación
Actores	Administrador, secretaria
Propósito	Permite la asignar camión a un chofer
Resumen	Este caso de uso hace referencia a la asignación de un camión a un chofer
Precondiciones	El usuario debe haber ingresado a sistema, estar autorizado para “asignar camión a un chofer.” y haber seleccionado la opción nueva asignación. en la pantalla asignar camión a un chofer.
Flujo Principal	<p>Se presenta al usuario la pantalla de “nueva asignación”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla chofer_camion de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “asignar camión a un chofer”, si los campos no están completos o son incorrectos se genera una excepción, si el camión ya existe se genera la excepción.</p>

	“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “asignar camión a un chofer”,.
Subflujos	Ninguna

Tabla 71 Casos de uso añadir asignación

#### 1.5.1.7.7.4.3.9.2 Casos de uso eliminar asignación

Caso de uso	Eliminar asignación
Actores	Administrador secretaria
Propósito	Permite eliminar una asignación.
Resumen	Este caso de uso hace referencia a la eliminación de una asignación. Que el chofer este sin camión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para asignar camión a un chofer, seleccionar un registro existente de la lista y seleccionar la opción eliminar asignación.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo de confirmación eliminar asignación</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina lógicamente de la tabla chofer_camion de la base de datos y muestra mensaje de “su transacción se realizó con exito” y se retorna a la pantalla “asignar camión a un chofer”.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “asignar camión a un chofer”.</p>
Subflujos	Ninguna.

Tabla 72 Casos de uso eliminar asignación

### 1.5.1.7.7.4.3.9.3 Casos de uso reporte

Caso de uso	Reporte
Actores	Administrador , secretaria
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de todos los camiones con su chofer registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para asignar camión a un socio, seleccionar asignar camión a un socio, seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los camiones y su chofer registrados en la tabla camiones, chofer_amion, choferes, usuarios_tra
Subflujos	Ninguna.

Tabla 73 Casos de uso reporte

### 1.5.1.7.7.4.3.10 Casos de uso administrar carga

Caso de uso	Administrar carga
Actores	Secretaria, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de las cargas.
Resumen	Este caso de uso hace referencia a la administración de carga de la cooperativa.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar carga y haber seleccionado del menú de la pantalla principal la opción administrar carga.

Flujo Principal	<p>Se muestra al usuario la pantalla de administrar carga con la lista de los cargas registrados en la tabla cargas también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nuevo carga” se ejecuta el subflujo añadir carga.</p> <p>Si el usuario selecciona la opción “Modificar carga” se ejecuta el subflujo modificar carga.</p> <p>Si el usuario selecciona la opción “Eliminar carga” se ejecuta el subflujo eliminar carga.</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver.</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte.</p>
Excepciones	Ninguna.

Tabla 74 Casos de uso administrar carga

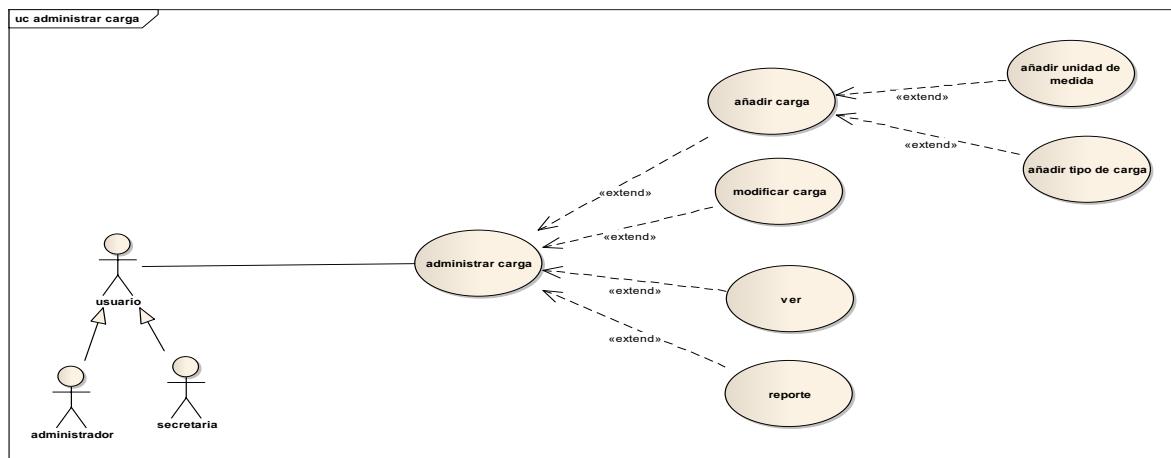


Figura 30 Casos de uso administrar carga

### 1.5.1.7.7.4.3.10.1 Caso de uso añadir carga

Caso de uso	Añadir carga
Actores	Secretaria, administrador
Propósito	Permite el registro de una nueva carga
Resumen	Este caso de uso hace referencia a la creación de una nueva carga y se asigna automáticamente al socio que está primero en la fila
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar carga.” y haber seleccionado la opción añadir carga. en la pantalla administrar carga.
Flujo Principal	<p>Se presenta al usuario la pantalla de “añadir carga”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de : “Guardar” los datos introducidos se validan y se guardaran en la tabla carga una vez guardado en la base de datos saco el código de carga que recién se ingresó ,saco el carnet de identidad del socio en la fila y los guardo en la tabla cargas de la base de datos , muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “administrar carga.”, si los campos no están completos o son incorrectos se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar carga”.</p>
	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “añadir unidad de medida” se ejecuta el subflujo añadir unidad de medida.</p> <p>Si el usuario selecciona la opción “añadir tipo de carga” se ejecuta el subflujo añadir tipo de carga.</p>

Tabla 75 Caso de uso añadir carga

**1.5.1.7.7.4.3.10.2 Caso de uso añadir unidad de medida**

Caso de uso	Añadir unidad de medida
Actores	Secretaria, administrador
Propósito	Permite el registro de una nueva unidad de medida
Resumen	Este caso de uso hace referencia a la creación de una nueva unidad de medida
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar carga.” y haber seleccionado la opción añadir carga. en la pantalla administrar carga luego seleccionar añadir unidad de medida en la pantalla añadir carga.
Flujo Principal	Se presenta al usuario la pantalla de “añadir unidad de medida”. Luego se completan los datos que hay en pantalla y salen opciones de : “Guardar” los datos introducidos se validan y se guardaran en la tabla unidades de medida de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “añadir carga”, si los campos no están completos o son incorrectos se genera la excepción. “Cancelar” no se realiza ningún cambio. Retornando a la pantalla “añadir carga”,.
Subflujos	ninguno

Tabla 76 Caso de uso añadir carga

**1.5.1.7.7.4.3.10.3 Caso de uso añadir tipo de carga**

Caso de uso	Añadir tipo de carga
-------------	----------------------

Actores	Secretaria, administrador
Propósito	Permite el registro de un nuevo tipo de carga
Resumen	Este caso de uso hace referencia a la creación de un nuevo tipo de carga
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar carga.” y haber seleccionado la opción añadir carga. en la pantalla administrar carga luego seleccionar añadir tipo de carga en la pantalla añadir carga.
Flujo Principal	<p>Se presenta al usuario la pantalla de “añadir tipo de carga”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de : “Guardar” los datos introducidos se validan y se guardaran en la tabla tipo_carga de la base de datos, muestra mensaje de “su transacción se realizó con éxito” y retorna a la pantalla “añadir carga”, si los campos no están completos o son incorrectos se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “añadir carga”,.</p>
Subflujos	ninguno

Tabla 77 Añadir tipo de carga

#### 1.5.1.7.7.4.3.10.4 Casos de uso modificar carga

Caso de uso	Modificar carga
Actores	Secretaria, administrador
Propósito	Permite Modificar carga
Resumen	Este caso de uso hace Modificar carga
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar carga” y haber seleccionado la opción Modificar carga en

	la pantalla administrar carga..
Flujo Principal	<p>Se presenta al usuario la pantalla de “Modificar carga.”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla carga de la base de datos, muestra mensaje de “se guardó correctamente” y retorna a la pantalla “administrar carga”, si los campos no están completos o son incorrectos se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar carga”,</p>
Subflujos	Ninguna

Tabla 78 Casos de uso modificar carga

**1.5.1.7.7.4.3.10.5 Caso de uso ver**

Caso de uso	Ver
Actores	Administrador , secretaria
Propósito	Permite ver todos los datos de las cargas disponibles para los camiones de los socios de la cooperativa
Resumen	Este caso de uso hace referencia a ver los datos del cargas disponibles para los camiones de los socios de la cooperativa
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar cargas, y seleccionar administrar cargas y seleccionar un registro existente de la lista y seleccionar la opción ver
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles de la carga registrado en la tabla cargas
Subflujos	Ninguna.

Tabla 79 Caso de uso ver

**1.5.1.7.7.4.3.10.6 Casos de uso reporte**

Caso de uso	Reporte
Actores	Administrador , socio
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso seara u reporte de todas las cargas registrados en el sistema
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar cargas, y seleccionar administrar cargas y seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de las cargas registrados en el sistema
Subflujos	Ninguna.

Tabla 80 Casos de uso reporte

**1.5.1.7.7.4.3.11 Caso de uso rol de espera**

Caso de uso	Rol de espera
Actores	Socio , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de socios que están haciendo fila para recibir carga.
Resumen	Este caso de uso hace referencia a una lista donde los socios están esperando carga ordenados por fecha de su última carga
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para rol de

	espera y haber seleccionado del menú de la pantalla principal la opción rol de espera.
Flujo Principal	<p>Se muestra al usuario la pantalla de rol de espera con la lista de los socios que están esperando carga ordenador por la fecha de su última carga registrados en la tabla socios, usuarios_tra también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “cambio de estado” se ejecuta el subflujo cambio de estado.</p>
Excepciones	Ninguna.

Tabla 81 Caso de uso rol de espera

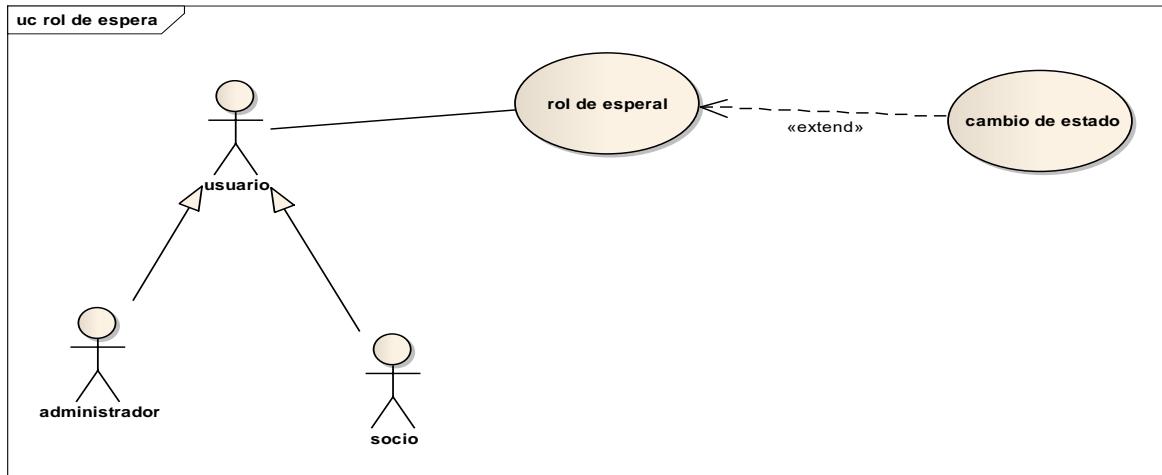


Figura 31 Caso de uso rol de espera

#### 1.5.1.7.7.4.3.11.1 Caso de uso cambio de estado

Caso de uso	Cambio de estado
Actores	Administrador, socio

Propósito	Permite cambiar de estado.
Resumen	<p>Este caso de uso hace referencia a cambiar de estado.</p> <p>Cuando su camión este fregado o en mantenimiento el socio puede cambiar su estado para no recibirá una carga y puede activar su estado cuando su camión este de listo para transportar una carga</p>
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para rol de espera, seleccionar su nombre para poder cambiar de estado.
Flujo Principal	<p>Se muestra al usuario la pantalla de dialogo “su camión está en mantenimiento señor y se muestra los datos del socio”</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“si” se hace el cambio de estado de 1 a 0 El registro se guardara en la tabla socio de la base de datos y muestra mensaje de “su transacción se realizó correctamente” y se retorna a la pantalla “rol de espera”.</p> <p>“no” no se realiza ningún cambio. Retornando a la pantalla “rol de espera”.</p>

Tabla 82 Cambio de estado

#### 1.5.1.7.7.4.3.12 Caso de uso programación

Caso de uso	Programación
Actores	Socio , administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los socios ya con su carga asignada
Resumen	Este caso de uso hace referencia a una lista donde ya se asignó la carga a cada socio y el socio puede ver que carga le toco transportar
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado programación y haber seleccionado del menú de la pantalla principal la opción

	programación.
Flujo Principal	<p>Se muestra al usuario la pantalla de programación con la lista de los socios registrados en las tablas socios, usuarios_tra, cargas y asignado una carga también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver</p>
Excepciones	Ninguna.

Tabla 83 Caso de uso programación

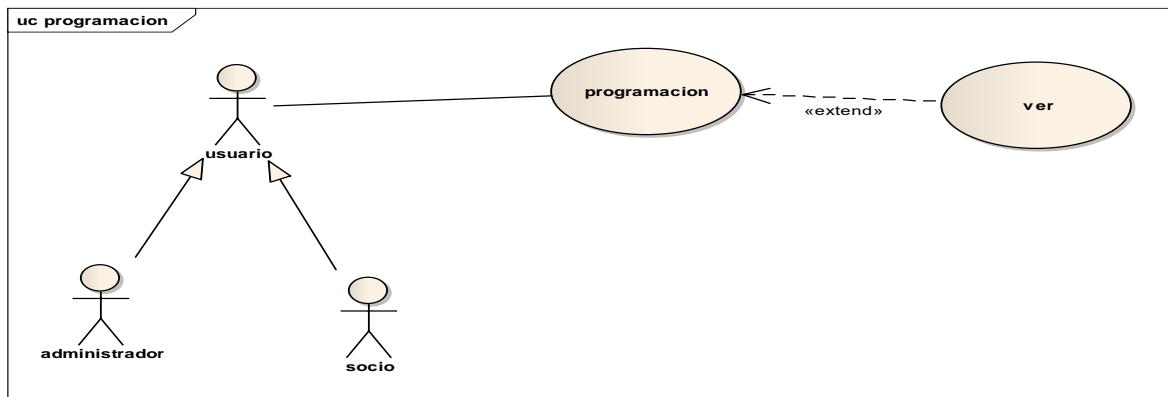


Figura 32 Caso de uso programación

#### 1.5.1.7.7.4.3.12.1 Casos de uso ver

Caso de uso	Ver
Actores	Administrador , socio
Propósito	Permite ver todos los datos de la programación
Resumen	Este caso de uso hace referencia a ver los datos de la carga que le toco a un determinado socio

Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para programación, y seleccionar programación y seleccionar un registro existente de la lista y seleccionar la opción ver
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles de la carga que le toco recuperados de la tabla cargas
Subflujos	Ninguna.

Tabla 84 Casos de uso ver

#### 1.5.1.7.7.4.3.13 Caso de uso administrar reuniones

Caso de uso	Administrar reuniones
Actores	Secretaria, socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de las reuniones.
Resumen	Este caso de uso hace referencia a la administración de reuniones del la cooperativa.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reuniones y haber seleccionado del menú de la pantalla principal la opción administrar reuniones.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar reuniones con la lista de los reuniones registradas en la tabla reuniones también contara con un filtro de búsqueda para poder buscarlos.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “nueva reunión” se ejecuta el subflujo</p>

	<p>añadir reunión.</p> <p>Si el usuario selecciona la opción “Modificar reunión” se ejecuta el subflujo modificar reunión.</p> <p>Si el usuario selecciona la opción “Eliminar reunión” se ejecuta el subflujo eliminar reunión.</p> <p>Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver</p> <p>Si el usuario selecciona la opción “reporte” se ejecuta el subflujo reporte</p>
Excepciones	Ninguna.

Tabla 85 Caso de uso administrar reuniones

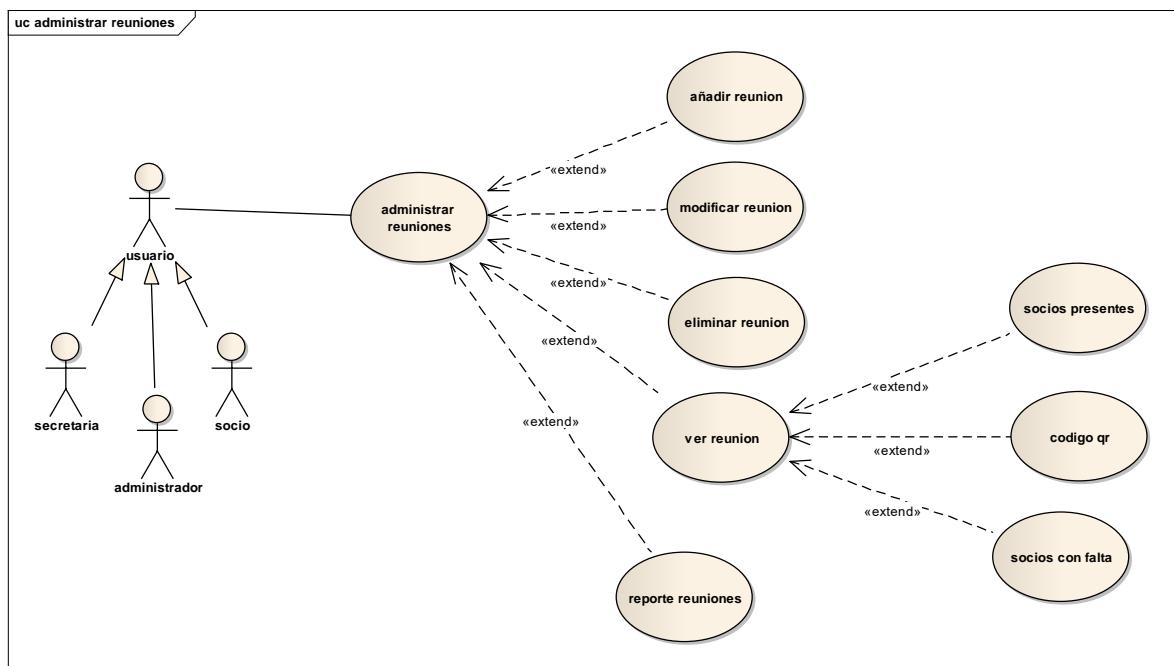


Figura 33 Caso de uso administrar reuniones

### 1.5.1.7.7.4.3.13.1 Caso de uso añadir reunión

Caso de uso	Añadir reunión
Actores	Secretaria, administrador
Propósito	Permite el registro de una nueva reunión
Resumen	Este caso de uso hace referencia a la creación de una nueva reunión de los socios haciéndoles recuerdo tal fecha es la reunión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar reuniones.” y haber seleccionado la opción añadir reunión. en la pantalla administrar reuniones.
Flujo Principal	<p>Se presenta al usuario la pantalla de “añadir reunión”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla reuniones de la base de datos, muestra mensaje de “se guardo correctamente” y retorna a la pantalla “administrar reuniones.”, si los campos no están completos o son incorrectos se genera una excepción, si la reunión ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar reuniones”.</p>
Subflujos	Ninguna

Tabla 86 Caso de uso añadir reunión

### 1.5.1.7.7.4.3.13.2 Casos de uso modificar reunión

Caso de uso	Modificar reunión
Actores	Secretaria, administrador
Propósito	Permite Modificar reunión

Resumen	Este caso de uso hace Modificar reunión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para “administrar reuniones” y haber seleccionado la opción Modificar reunión en la pantalla administrar reunión..
Flujo Principal	<p>Se presenta al usuario la pantalla de “Modificar reunión.”.</p> <p>Luego se completan los datos que hay en pantalla y salen opciones de :</p> <p>“Guardar” los datos introducidos se validan y se guardaran en la tabla reuniones de la base de datos, muestra mensaje de “se guardó correctamente” retorna a la pantalla “administrar reuniones”, si los campos no están completos o son incorrectos se genera una excepción, si la reunión ya existe se genera la excepción.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar reuniones”,</p>
Subflujos	Ninguna

Tabla 87 Casos de uso modificar reunión

#### 1.5.1.7.7.4.3.13.3 Caso de uso eliminar reunión

Caso de uso	Eliminar reunión
Actores	Secretaria, administrador
Propósito	Permite eliminar carga.
Resumen	Este caso de uso hace referencia a la eliminación de la reunión, dándolos de baja.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reunión, seleccionar un registro existente de la lista y seleccionar la opción eliminar.
Flujo Principal	Se muestra al usuario la pantalla de dialogo de confirmación eliminar

	<p>reunión</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“Aceptar” El registro se elimina física o lógicamente de la tabla reuniones de la base de datos y muestra mensaje de “se eliminó correctamente” se retorna a la pantalla “administrar reuniones”.</p> <p>“Cancelar” no se realiza ningún cambio. Retornando a la pantalla “administrar reuniones”.</p>
Subflujos	Ninguna.

Tabla 88 Caso de uso eliminar reunión

**1.5.1.7.7.4.3.13.4 Caso de uso ver reunión**

Caso de uso	Administrar reuniones
Actores	Secretaria, socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en la reunión
Resumen	Este caso de uso hace referencia a la ver de reuniones de la cooperativa.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reuniones y haber seleccionado del menú de la pantalla principal la opción administrar reuniones y seleccionar un reunión con la opción ver reunión.
Flujo Principal	<p>Se muestra al usuario la pantalla de ver reuniones con todos los detalles de la reunión registrados en la tabla reuniones</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “socios presentes” se ejecuta el subflujo socios presentes.</p>

	<p>Si el usuario selecciona la opción “código qr” se ejecuta el subflujo código qr.</p> <p>Si el usuario selecciona la opción “socios con falta” se ejecuta el subflujo socios con falta.</p>
Excepciones	Ninguna.

Tabla 89 Caso de uso ver reunión

**1.5.1.7.7.4.3.13.5 Caso de uso socios presentes**

Caso de uso	Socios presentes
Actores	Secretaria, administrador,
Propósito	Permite ver a los socios que estuvieron presentes en la reunión.
Resumen	Este caso de uso hace referencia ver a los socios que estuvieron presentes en dicha reunión seleccionada.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reunión, seleccionar un registro existente de la lista y seleccionar la opción ver luego seleccionar la opción socios presentes.
Flujo Principal	<p>Se muestra al usuario la pantalla de socios presentes reunión registrados en la tabla reuniones, asistencia, socios, usuarios_tra</p> <p>Con los nombre y apellidos de los socios presentes en la reunión</p>
Subflujos	Ninguna.

Tabla 90 Caso de uso socios presentes

**1.5.1.7.7.4.3.13.6 Caso de uso código qr**

Caso de uso	Código qr
Actores	Secretaria, administrador
Propósito	Permite controlar si un socios a asistido o no a la reunión.

Resumen	Este caso de uso hace referencia al control de asistencia para poder saber que socios asistieron o no a dicha reunión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reunión, seleccionar un registro existente de la lista y seleccionar la opción ver luego seleccionar la opción código qr
Flujo Principal	<p>Se muestra a los socios un código qr que al ingreso de una reunión deberán escanear para que el sistema les ponga presentes</p> <p>El código qr genera un código cada 7 segundo el código se guarda en la tabla cel pero cada 7 segundos se cambia a otro código el socio deberá escanear con su celular y con la aplicación desarrollada en este proyecto</p>
Subflujos	Ninguna.

Tabla 91 Caso de uso código qr

#### 1.5.1.7.7.4.3.13.7 Caso de uso socios con falta

Caso de uso	Socios con falta
Actores	Secretaria, administrador,
Propósito	Permite ver a los socios que no estuvieron presentes en la reunión.
Resumen	Este caso de uso hace referencia ver a los socios que no estuvieron presentes en dicha reunión seleccionada.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reunión, seleccionar un registro existente de la lista y seleccionar la opción ver luego seleccionar la opción socios con falta.
Flujo Principal	<p>Se muestra al usuario la pantalla de socios con falta registrados en la tabla reuniones, asistencia, socios, usuarios_tra</p> <p>Con los nombre y apellidos de los socios con falta en la reunión</p>
Subflujos	Ninguna.

Tabla 92 Caso de uso socios con falta

**1.5.1.7.7.4.3.13.8 Casos de uso reporte**

Caso de uso	reporte
Actores	Administrador , secretaria, socio
Propósito	Permite hacer un reporte en formato pdf
Resumen	Este caso de uso se podrá hacer un reporte de la de los socios que asistieron a dicha reunión
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para administrar reuniones, y seleccionar administrar reuniones y seleccionar reporte
Flujo Principal	Se muestra al usuario la pantalla reporte con un listado de los socios que asistieron a esa reunión registrados en la tabla reuniones, asistencias, socios, usuarios_tra
Subflujos	Ninguna.

Tabla 93 Casos de uso reporte

### **1.5.1.7.8 Modelo de datos**

#### **1.5.1.7.8.1 Introducción**

El modelado de diagrama de clases es uno de los diagramas requeridos en la fase de análisis/diseño de la metodología RUP la cual implementando.

Previamente que la información del sistema será soportado por una clase de datos relacional, este modelo describe la presentación lógica de los datos persistentes, de acuerdo con el enfoque para el modelado relacional de datos. Para expresar este modelo se realiza un diagrama de clases.

Los diagramas de clases son diagramas de estructura estática que muestra las clases del sistema y sus interrelaciones (incluye herencia, agregación y asociación, etc.). Los diagramas de clases son el pilar fundamental del modelo UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer(análisis), como para mostrar cómo puede ser construido.

#### **1.5.1.7.8.2 Propósito**

Comprender la estructura del sistema deseado para la cooperativa.

#### **1.5.1.7.8.3 Alcance**

Describir las clases y los objetos de diseño del sistema en sus segunda interacción.

Identificar y definir los objetos del sistema según los objetivos del sistema deseado aprobado por la organización.

### 1.5.1.7.8.4 Diagrama de clases

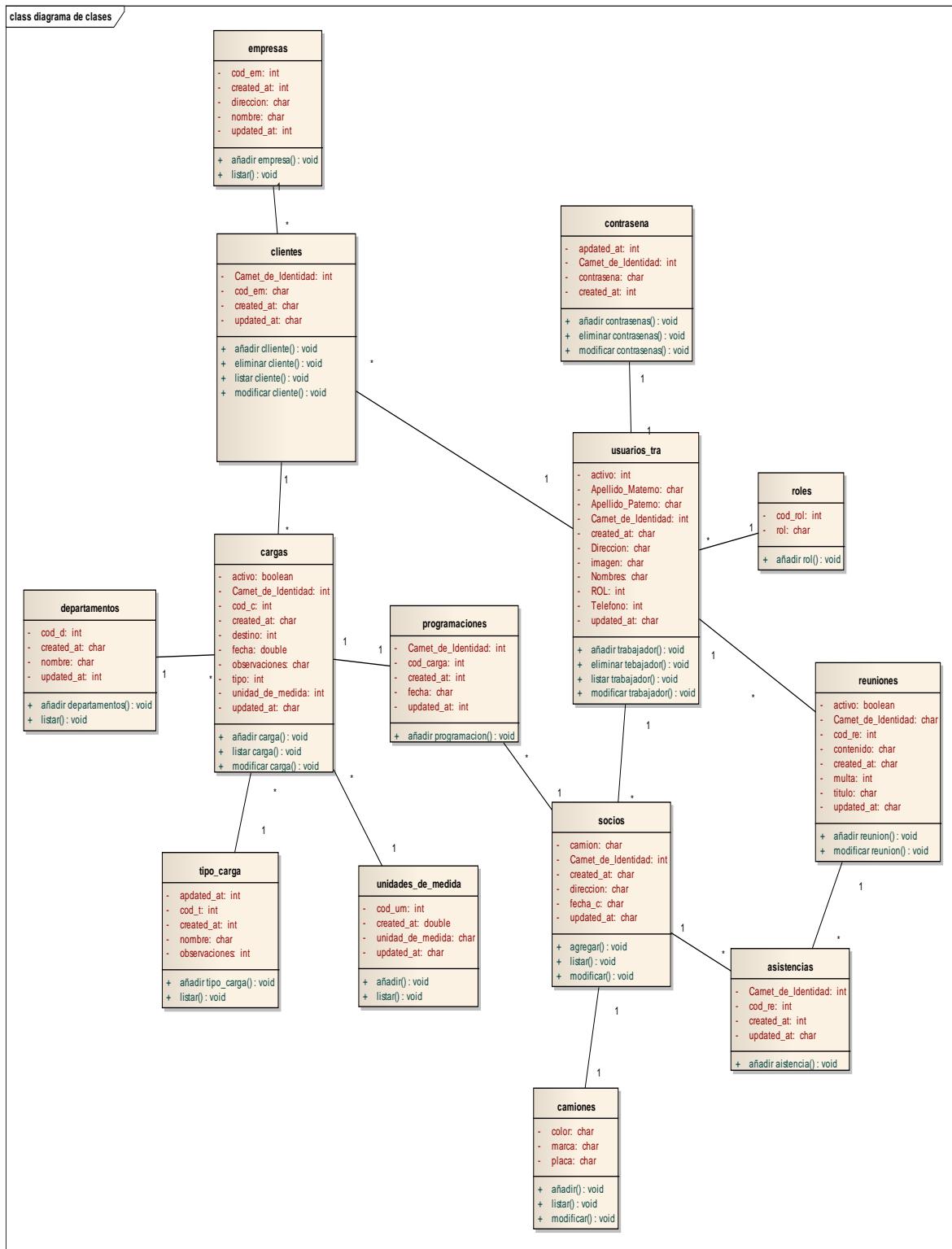


Figura 34 Diagrama de clases

### 1.5.1.7.8.5 Entidad relación

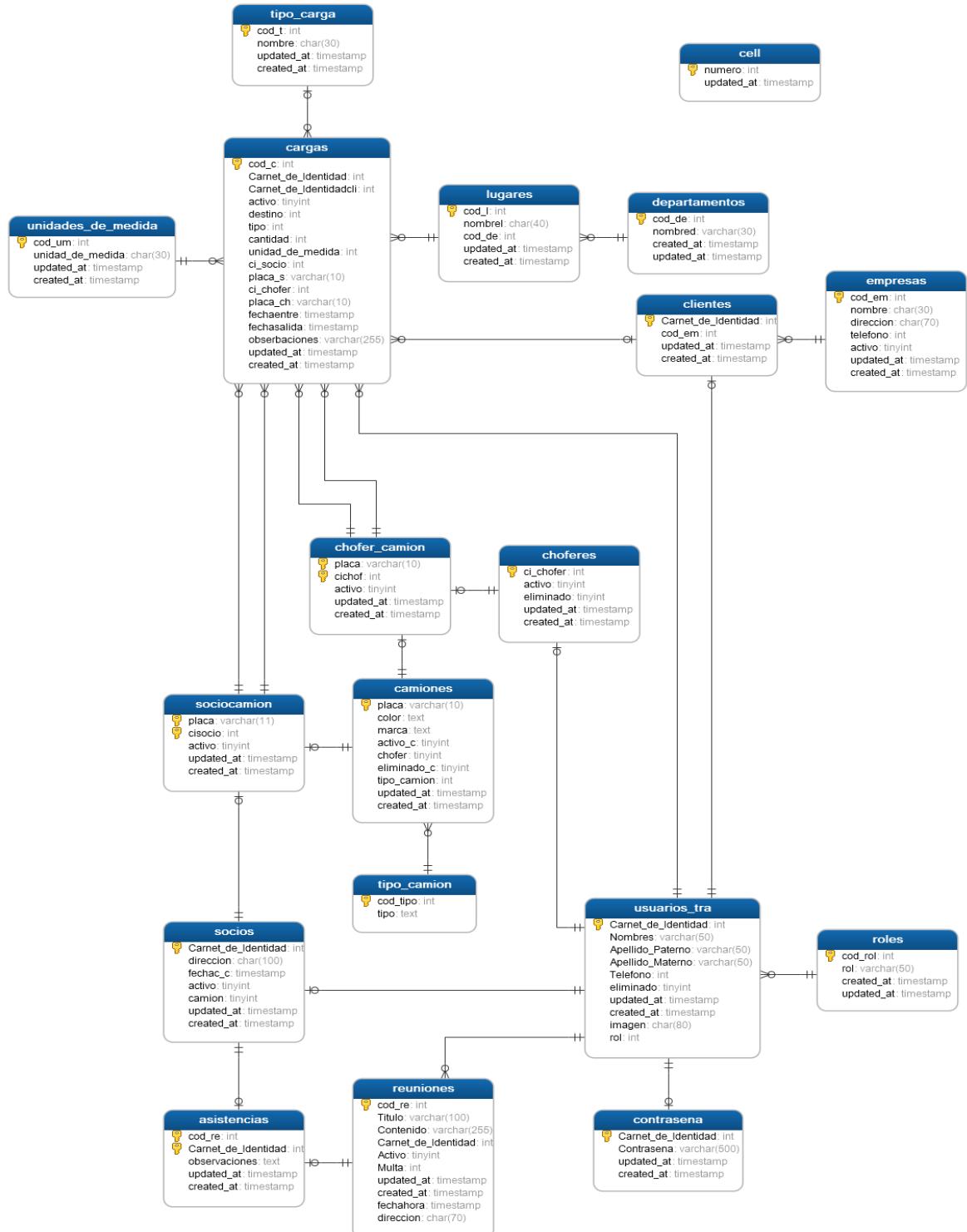


Figura 35 Entidad relación

**1.5.1.7.8.6 Diccionario de datos****Tabla Contrasena**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Carnet_de_identidad	Entero	11	No	si	si	Carnet de identidad de la persona
Contrasenia	Varchar	50	No	No	No	Contrasenia de la persona para ingresar al sistema
Updated_at	timestamp		No	No	No	Fecha de actualización del dato
Created_at	Timestamp		No	No	No	Fecha de creación del dato

Tabla 94 Contrasena

**Tabla Usuarios\_tra**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Carnet_de_Identidad	Entero	11	No	Si	No	Carnet de identidad del usuario del sistema
Nombres	varchar	50	No	No	No	Nombre de la usuario del sistema
Apellido_Paterno	varchar	50	Si	No	No	Apellido del usuario del sistema
Apellido_Materno	Varchar	50	si	No	No	Apellido materno del usuario del sistema
Telefono	Entero	11	No	No	No	Teléfono del usuario del sistema
Eliminado	Entero	1	No	No	No	Para eliminado lógico del usuario
Imagen	Texto	30	No	No	No	Para guardar el nombre de la imagen

Rol	Entero	11	No	No	si	Hace referencia al rol del usuario
Updated_at	Timestamp		No	No	No	Para guardar la fecha de actualización del usuario
Created_at	Timestamp		No	No	No	Para guardar la fecha de creación del usuario

Tabla 95 Usuarios\_tra

**Tabla Unidades\_de\_medida**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_um	Entero	11	No	Si	No	Id de la tabla unidades_de_medida
Unidad_de_medida	Texto	30	No	No	No	Para guardar el Nombre de la unidad de medida
Updated_at	Timestamp		No	No	No	Fecha de actualización de la unidad de medida
Created_at	Timestamp		No	No	No	Fecha de creación de la unidad de medida

Tabla 96 Unidades de medida

**Tabla Tipo\_carga**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_t	Entero	11	No	Si	no	Id de la tabla tipo_carga
Nombre	Texto	30	No	No	No	Nombre del tipo de carga
Updated_at	Timestamp		No	No	No	Fecha de actualización del tipo de carga
Created_at	Timestamp		No	No	No	Fecha de creación del tipo de carga

Tabla 97 Tipo\_carga

### Tabla Socios

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Carnet_de_Identidad	Entero	11	No	si	si	Es el carné de identidad de la tabla usuarios_tra y es el id de la tabla socios
Direccion	Texto	100	No	No	No	Para guardar la dirección del socio
fecha_c	Timestamp	11	No	No	No	Para guardar la última fecha de creación y con esta fecha el socio hace fila para cargar
Activo	tinyint	1	No	No	No	Para saber si está listo para cargar o su camión está en reposo
Camión	tinyint	1	No	No	No	Para saber si tiene camión o no
Updated_at	Timestamp		No	No	No	Para guardar la fecha de actualización del socio
Created_at	Timestamp		No	No	No	Para guardar la fecha de creación del socio

Tabla 98 Socios

### Tabla Choferes

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Ci_chofer	Entero	11	No	si	si	Es el carné de identidad de la tabla usuarios_tra y es el id de la tabla choferes
Activo	tinyint	1	No	No	No	Para saber si está con un camión
Eliminado	tinyint	1	No	No	No	Para saber si está eliminado o no
Updated_at	Timestamp		No	No	No	Para guardar la fecha de actualización del socio

Created_at	Timestamp		No	No	No	Para guardar la fecha de creación del socio
------------	-----------	--	----	----	----	---

Tabla 99 Socios

**Tabla Roles**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_rol	Entero	11	No	Si	No	Id de la tabla roles
Rol	Texto	50	No	No	No	Nombre del rol
Updated_at	Timestamp		No	No	No	Para guardar la fecha de actualización del rol
Created_at	Timestamp		No	No	No	Para guardar la fecha de creación del rol

Tabla 100 Roles

**Tabla Reuniones**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_re	Entero	11	No	Si	No	Id de la tabla reuniones
Titulo	Varchar	100	No	No	No	Título de la reunión
Contenido	Varchar	255	No	No	No	Contenido de la reunión
Activo	Entero	1	No	No	No	Eliminar lógicamente la reunión
Multa	Entero	11	No	No	No	Para sabes la multa en caso de no venir a dicha reunión
Fechahora	Timestamp		No	No	No	Para guardar la fecha y hora de la reunión
Direccion	Texto	70	No	No	No	Para guardar la dirección de la reunión
Updated_at	Timestamp		No	No	No	Para guardar la fecha de actualización de la reunión
Created-at	Timestamp		No	No	No	Para guardar la fecha de creación de la reunión

Tabla 101 Reuniones

### Tabla Empresas

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_em	Entero	11	No	Si	No	Id de la tabla empresas
Nombre	Texto	30	No	No	No	nombre de la empresa
Direccion	Texto	70	No	No	No	Dirección de la empresa
Telefono	Entero	11	No	No	No	Para guardar el teléfono de la empresa
Activo	Entero	1	No	No	No	Para saber si la empresa esta activa o eliminada
Updated_at	Timestamp		No	No	No	Para saber la actualización de la empresa
Created_at	Timestamp		No	No	No	Para saber la creación de la empresa

Tabla 102 Empresas

### Tabla Departamentos

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_de	Entero	11	No	Si	No	Id de la tabla departamentos
Nombred	Texto	20	No	No	No	Nombre del departamento
Updated_at	Timestamp		No	No	No	Para saber la fecha de actualización del departamento
Created_at	Timestamp		No	No	No	Para saber la fecha de creación del departamento

Tabla 103 Departamentos

### Tabla Clientes

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Carnet_de_identidad	Entero	11	No	Si	Si	Id de la tabla clientes y heredado de la tabla usuarios_tra
Cod_em	Entero	11	No	No	Si	Heredo el código de la tabla empresas
Updated_at	Timestamp		No	No	No	Para saber la fecha de actualización cliente

Created_at	Timestamp		No	No	No	Para saber la fecha de creacion del cliente
------------	-----------	--	----	----	----	---

Tabla 104 Clientes

**Tabla Cell**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Numero	Entero	11	No	Si	No	Código aleatorio para el control de asistencia
Updated_at	Timestamp		No	No	No	Para la actualización del código

Tabla 105 Cell

**Tabla Cargas**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_c	Entero	11	No	Si	No	Id de la tabla cargas
Observaciones	Varchar	255	No	No	No	Escribir alguna observación de la carga
Carnet_de_Identidad	Entero	11	No	No	Si	Heredamos de la tabla usuarios_tra para saber quién añadio la carga
Carnet_de_Identidadcli	Entero	11	No	No	Si	Heredamos el ci del cliente
Activo	Entero	1	No	No	No	Para saber si la carga a sido entregado o no
Destino	Entero	11	No	No	Si	Heredamos de la tabla lugares
Tipo	Entero	11	No	No	Si	Heredamos de la tabla tipo_carga
Cantidad	Entero	11	No	No	No	Para saber la cantidad de carga
Unidad_de_medidad	Entero	11	No	No	Si	Para saber la unidad de medida de la carga
Ci_socio	Entero	11	No	No	Si	Heredamos de la el ci del socio

Placa_s	Varchar	10	No	No	Si	Heredamos la placa del camión del socio
Ci_chofer	Entero	10	No	No	Si	Heredamos el ci del chofer
Placa_ch	Varchar	10	No	No	Si	Heredamos la placa del camión del chofer
fechaentre	Timestamp		Si	No	No	La fecha de entrega de la carga
Fechasalida	Timestamp		Si	No	No	Fecha de salida de la carga
Updated_at	Timestamp		No	No	No	Para saber la actualización de la carga
Created_at	Timestamp		No	No	No	Para saber la creación de la carga

Tabla 106 Cargas

**Tabla Camiones**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Placa	Varchar	11	No	Si	No	Id de la tabla camiones y almacena la placa del camión
Color	Texto	20	No	No	No	Color del camión
Marca	Texto	20	No	No	No	Marca del camión
Activi_c	tinyint	1	No	No	No	Para saber si el camión está activo o no
Chofer	tinyint	1	No	No	No	Para saber si el camión tiene o no chofer
Tipo_camion	Entero	11	No	No	Si	Heredamos el tipo de camión
Updated_at	Timestamp		No	No	No	Para saber la fecha de actualización del camión
Created_at	Timestamp		No	No	No	Para saber la fecha de creación del camión

Tabla 107 Camiones

### Tabla Asistencias

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_re	Entero	11	No	Si	Si	Id de la tabla asistencia y heredamos de la tabla reuniones
Observacion	Texto	100	No	No	No	Para poner alguna observación en la asistencia
Carnet_de_Identidad	Entero	11	No	Si	Si	Id de la tabla asistencias y heredamos de la tabla socios
Updated_at	Timestamp		No	No	No	Fecha de actualización de la asistencias
Created_at	Timestamp		No	No	No	Fecha de creación de la asistencia

Tabla 108 Asistencias

### Tabla Tipo\_camion

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_tipo	Entero	11	No	Si	Si	Id de la tabla tipo_camion
Tipo	Text	30	No	No	no	El nombre del tipo de camión
Updated_at	Timestamp		No	No	No	Fecha de actualización
Created_at	Timestamp		No	No	No	Fecha de creación

Tabla 109 Tipo\_camion

### Tabla Sociocamion

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
placa	varchar	11	No	Si	Si	Id de la tabla socio camión
cisocio	Entero	11	No	Si	Si	Id de la tabla, Ci del socio
Activo	tinyint	1	No	No	No	Para saber con qué camión se encuentra el socio
Updated_at	Timestamp		No	No	No	Fecha de actualización

Created_at	Timestamp		No	No	No	Fecha de creación
------------	-----------	--	----	----	----	-------------------

Tabla 110 Sociocamion

**Tabla Chofer\_camion**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
placa	varchar	11	No	Si	Si	Id de la tabla socio camión
Cichof	Entero	11	No	Si	Si	Id de la tabla, Ci del chofer
Activo	tinyint	1	No	No	No	Para saber con qué camión se encuentra el socio
Updated_at	Timestamp		No	No	No	Fecha de actualización
Created_at	Timestamp		No	No	No	Fecha de creación

Tabla 111 Chofer\_camion

**Tabla lugares**

Nombre	Tipo	Tamaño	Nulo	Pk	Fk	Descripción
Cod_l	Entero	11	No	Si	Si	Id de la tabla lugares
nombrel	Texto	30	No	No	No	Nombre del lugar
Cod_de	Entero	11	No	No	No	Heredamos el id de la tabla departamentos
Updated_at	Timestamp		No	No	No	Fecha de actualización
Created_at	Timestamp		No	No	No	Fecha de creación

Tabla 112 Lugares

### 1.5.1.7.8.7 Script SQL para crear la base de datos

```
CREATE TABLE `asistencias` (
  `cod_re` int(11) NOT NULL,
  `Carnet_de_Identidad` int(11) NOT NULL,
  `observaciones` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `camiones` (
  `placa` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `color` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `marca` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `activo_c` tinyint(1) NOT NULL,
  `chofer` tinyint(1) NOT NULL,
  `eliminado_c` tinyint(4) NOT NULL,
  `tipo_camion` int(11) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `cargas` (
  `cod_c` int(11) NOT NULL,
  `Carnet_de_Identidad` int(11) NOT NULL,
  `Carnet_de_Identidadcl` int(11) DEFAULT NULL,
  `activo` tinyint(1) NOT NULL,
  `destino` int(11) NOT NULL,
  `tipo` int(11) DEFAULT NULL,
  `cantidad` int(11) NOT NULL,
  `unidad_de_medida` int(11) NOT NULL,
  `ci_socio` int(11) NOT NULL,
  `placa_s` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `ci_chofer` int(11) NOT NULL,
  `placa_ch` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `fecharentre` timestamp NULL DEFAULT '0000-00-00 00:00:00',
  `fechasalida` timestamp NULL DEFAULT NULL,
  `obserbaciones` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
```

```

`updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
`created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

CREATE TABLE `cell` (
  `numero` int(11) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

INSERT INTO `cell` (`numero`, `updated_at`) VALUES
(1554263280, '2019-04-03 03:48:00');

```

```

CREATE TABLE `choferes` (
  `ci_chofer` int(11) NOT NULL,
  `activo` tinyint(1) NOT NULL,
  `eliminado` tinyint(1) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

CREATE TABLE `chofer_camion` (
  `placa` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `cichof` int(11) NOT NULL,
  `activo` tinyint(1) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

CREATE TABLE `clientes` (
  `Carnet_de_Identidad` int(11) NOT NULL,
  `cod_em` int(11) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```
CREATE TABLE `contrasena` (
  `Carnet_de_Identidad` int(11) NOT NULL,
  `Contrasena` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `departamentos` (
  `cod_de` int(11) NOT NULL,
  `nombred` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
INSERT INTO `departamentos` (`cod_de`, `nombred`, `created_at`, `updated_at`) VALUES
(1, 'Santa Cruz', '2019-03-25 23:41:03', '2019-03-25 04:00:00'),
(2, 'La Paz', '2019-03-25 23:41:06', '2019-03-25 04:00:00'),
(3, 'Cochabamba', '2019-03-25 23:41:09', '2019-03-25 04:00:00'),
(4, 'Oruro', '2019-03-25 23:41:13', '2019-03-25 04:00:00'),
(5, 'Potosí', '2019-03-25 23:41:18', '2019-03-25 04:00:00'),
(6, 'Chuquisaca', '2019-03-25 23:41:21', '2019-03-25 04:00:00'),
(7, 'Tarija', '2019-03-25 23:41:25', '2019-03-25 04:00:00'),
(8, 'Pando', '2019-03-25 23:41:28', '2019-03-25 04:00:00'),
(9, 'Beni', '2019-03-25 23:41:31', '2019-03-25 04:00:00');
```

```
CREATE TABLE `empresas` (
  `cod_em` int(11) NOT NULL,
  `nombre` char(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `direccion` char(70) COLLATE utf8mb4_unicode_ci NOT NULL,
  `telefono` int(11) NOT NULL,
  `activo` tinyint(4) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `lugares` (
  `cod_l` int(11) NOT NULL,
  `nombrel` char(40) COLLATE utf8mb4_unicode_ci NOT NULL,
  `cod_de` int(11) NOT NULL,
```

```

`updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
`created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

INSERT INTO `lugares` (`cod_l`, `nombrel`, `cod_de`, `updated_at`, `created_at`) VALUES
(7, 'Santa Cruz de la sierra', 1, '2019-03-25 04:00:00', '2019-03-25 04:00:00'),
(8, 'montero', 1, '2019-03-25 04:00:00', '2019-03-25 04:00:00'),
(9, 'Ciudad de la paz', 2, '2019-03-26 01:42:12', '2019-03-26 01:42:12'),
(10, 'san julian', 1, '2019-03-26 01:42:51', '2019-03-26 01:42:51'),
(11, 'ciudad de sucre', 6, '2019-03-26 01:47:06', '2019-03-26 01:47:06'),
(12, 'ciudad de potosi', 5, '2019-03-27 00:38:54', '2019-03-27 00:38:54'),
(13, 'ciudad de cochabamba', 3, '2019-03-27 21:39:59', '2019-03-27 21:39:59');

CREATE TABLE `reuniones` (
`cod_re` int(11) NOT NULL,
`Titulo` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
`Contenido` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
`Carnet_de_Identidad` int(11) NOT NULL,
`Activo` tinyint(1) NOT NULL,
`Multas` int(11) NOT NULL,
`updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
`created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
`fechahora` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
`direccion` char(70) COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `roles` (
`cod_rol` int(11) NOT NULL,
`rol` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
`created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
`updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

INSERT INTO `roles` (`cod_rol`, `rol`, `created_at`, `updated_at`) VALUES
(1, 'administrador', '2018-12-03 00:40:19', '0000-00-00 00:00:00'),
(2, 'socio', '2018-12-03 00:40:44', '0000-00-00 00:00:00'),
(3, 'secretaria', '2018-12-03 00:40:44', '0000-00-00 00:00:00'),
(4, 'cliente', '2018-12-03 00:40:57', '0000-00-00 00:00:00'),
(5, 'chofer', '2018-12-17 02:58:40', '0000-00-00 00:00:00');

```

```

CREATE TABLE `sociocamion` (
  `placa` varchar(11) COLLATE utf8mb4_unicode_ci NOT NULL,
  `cisocio` int(11) NOT NULL,
  `activo` tinyint(1) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `socios` (
  `Carnet_de_Identidad` int(11) NOT NULL,
  `direccion` char(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `fechac_c` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `activo` tinyint(1) NOT NULL,
  `camion` tinyint(1) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `tipo_camion` (
  `cod_tipo` int(11) NOT NULL,
  `tipo` text COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
INSERT INTO `tipo_camion` (`cod_tipo`, `tipo`) VALUES
(1, 'Camion con Acople'),
(2, 'Camion Trailer');

CREATE TABLE `tipo_carga` (
  `cod_t` int(11) NOT NULL,
  `nombre` char(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `unidades_de_medida` (
  `cod_um` int(11) NOT NULL,
  `unidad_de_medida` char(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

```

```

`created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `usuarios_tra` (
  `Carnet_de_Identidad` int(11) NOT NULL,
  `Nombres` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Apellido_Paterno` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Apellido_Materno` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Telefono` int(11) NOT NULL,
  `eliminado` tinyint(1) NOT NULL,
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `Imagen` char(80) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `rol` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-
-- ALTER TABLE `asistencias`
ADD PRIMARY KEY (`cod_re`,`Carnet_de_Identidad`),
ADD KEY `cod_re` (`cod_re`),
ADD KEY `Carnet_de_Identidad` (`Carnet_de_Identidad`);

--
-- Indices de la tabla `camiones`
-- ALTER TABLE `camiones`
ADD PRIMARY KEY (`placa`),
ADD KEY `tipo_camion` (`tipo_camion`);

--
-- Indices de la tabla `cargas`
-- ALTER TABLE `cargas`
ADD PRIMARY KEY (`cod_c`),
ADD UNIQUE KEY `UQ_cargas_cod_c` (`cod_c`),

```

```

ADD KEY `Carnet_de_Identidadcli` (`Carnet_de_Identidadcli`),
ADD KEY `unidad_de_medida` (`unidad_de_medida`),
ADD KEY `destino` (`destino`),
ADD KEY `tipo` (`tipo`),
ADD KEY `Carnet_de_Identidad` (`Carnet_de_Identidad`),
ADD KEY `ci_socio` (`ci_socio`,`placa_s`,`ci_chofer`,`placa_ch`),
ADD KEY `ci_chofer` (`ci_chofer`),
ADD KEY `placa_ch` (`placa_ch`),
ADD KEY `placa_s` (`placa_s`);


```

```

ALTER TABLE `cell`
ADD PRIMARY KEY (`numero`);


```

--

-- Indices de la tabla `choferes`

--

```

ALTER TABLE `choferes`


```

```

ADD PRIMARY KEY (`ci_chofer`),
ADD KEY `ci_chofer` (`ci_chofer`);


```

--

-- Indices de la tabla `chofer\_camion`

--

```

ALTER TABLE `chofer_camion`


```

```

ADD PRIMARY KEY (`placa`,`cichof`),
ADD KEY `placa` (`placa`,`cichof`),
ADD KEY `cichof` (`cichof`);


```

--

-- Indices de la tabla `clientes`

--

```

ALTER TABLE `clientes`


```

```

ADD PRIMARY KEY (`Carnet_de_Identidad`),
ADD UNIQUE KEY `UQ_clientes_Carnet_de_Identidad` (`Carnet_de_Identidad`),
ADD KEY `cod_em` (`cod_em`),
ADD KEY `Carnet_de_Identidad` (`Carnet_de_Identidad`);


```

--

```

-- Indices de la tabla `contrasena`

-- ALTER TABLE `contrasena`
ADD PRIMARY KEY (`Carnet_de_Identidad`),
ADD KEY `Carnet_de_Identidad` (`Carnet_de_Identidad`);

-- Indices de la tabla `departamentos`

-- ALTER TABLE `departamentos`
ADD PRIMARY KEY (`cod_de`);

-- Indices de la tabla `empresas`

-- ALTER TABLE `empresas`
ADD PRIMARY KEY (`cod_em`),
ADD UNIQUE KEY `UQ_empresas_cod_em` (`cod_em`);

-- Indices de la tabla `lugares`

-- ALTER TABLE `lugares`
ADD PRIMARY KEY (`cod_l`),
ADD UNIQUE KEY `UQ_departamentos_cod_l` (`cod_l`) USING BTREE,
ADD KEY `cod_de` (`cod_de`);

-- Indices de la tabla `reuniones`

-- ALTER TABLE `reuniones`
ADD PRIMARY KEY (`cod_re`),
ADD UNIQUE KEY `UQ_reuniones_cod_re` (`cod_re`),
ADD KEY `Carnet_de_Identidad` (`Carnet_de_Identidad`);

-- Indices de la tabla `roles`

-- ALTER TABLE `roles`

```

```

ADD PRIMARY KEY (`cod_rol`),
ADD UNIQUE KEY `UQ_roles_cod_rol` (`cod_rol`);

-- Indices de la tabla `sociocamion`

ALTER TABLE `sociocamion`
ADD PRIMARY KEY (`placa`, `cisocio`),
ADD KEY `cisocio` (`cisocio`),
ADD KEY `placa` (`placa`);

-- Indices de la tabla `socios`

ALTER TABLE `socios`
ADD PRIMARY KEY (`Carnet_de_Identidad`),
ADD UNIQUE KEY `UQ_socios_Carnet_de_Identidad` (`Carnet_de_Identidad`),
ADD KEY `Carnet_de_Identidad` (`Carnet_de_Identidad`);

-- Indices de la tabla `tipo_camion`

ALTER TABLE `tipo_camion`
ADD PRIMARY KEY (`cod_tipo`);

-- Indices de la tabla `tipo_carga`

ALTER TABLE `tipo_carga`
ADD PRIMARY KEY (`cod_t`),
ADD UNIQUE KEY `UQ_tipo_carga_cod_t` (`cod_t`);

-- Indices de la tabla `unidades_de_medida`

ALTER TABLE `unidades_de_medida`
ADD PRIMARY KEY (`cod_um`),
ADD UNIQUE KEY `UQ_unidades_de_medida_cod_um` (`cod_um`);

```

```

-- Indices de la tabla `usuarios_tra`  

--  

ALTER TABLE `usuarios_tra`  

ADD PRIMARY KEY (`Carnet_de_Identidad`),  

ADD UNIQUE KEY `UQ_usuarios_tra_Carnet_de_Identidad` (`Carnet_de_Identidad`),  

ADD KEY `rol` (`rol`);  

--  

-- AUTO_INCREMENT de las tablas volcadas  

--  

--  

-- AUTO_INCREMENT de la tabla `cargas`  

--  

ALTER TABLE `cargas`  

MODIFY `cod_c` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;  

--  

-- AUTO_INCREMENT de la tabla `departamentos`  

--  

ALTER TABLE `departamentos`  

MODIFY `cod_de` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;  

--  

-- AUTO_INCREMENT de la tabla `empresas`  

--  

ALTER TABLE `empresas`  

MODIFY `cod_em` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;  

--  

-- AUTO_INCREMENT de la tabla `lugares`  

--  

ALTER TABLE `lugares`  

MODIFY `cod_l` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;  

--  

-- AUTO_INCREMENT de la tabla `reuniones`  

--  

ALTER TABLE `reuniones`  

MODIFY `cod_re` int(11) NOT NULL AUTO_INCREMENT;  

--  

-- AUTO_INCREMENT de la tabla `roles`  


```

```

-- ALTER TABLE `roles`
  MODIFY `cod_rol` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
-- AUTO_INCREMENT de la tabla `tipo_camion`

-- ALTER TABLE `tipo_camion`
  MODIFY `cod_tipo` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
-- AUTO_INCREMENT de la tabla `tipo_carga`

-- ALTER TABLE `tipo_carga`
  MODIFY `cod_t` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
-- AUTO_INCREMENT de la tabla `unidades_de_medida`

-- ALTER TABLE `unidades_de_medida`
  MODIFY `cod_um` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
-- Restricciones para tablas volcadas

-- Filtros para la tabla `asistencias`

-- ALTER TABLE `asistencias`
  ADD CONSTRAINT `FK_asistencias_reuniones` FOREIGN KEY (`cod_re`) REFERENCES `reuniones`(`cod_re`) ON UPDATE CASCADE,
  ADD CONSTRAINT `FK_asistencias_socios` FOREIGN KEY (`Carnet_de_Identidad`) REFERENCES `socios`(`Carnet_de_Identidad`) ON UPDATE CASCADE;

-- Filtros para la tabla `camiones`

-- ALTER TABLE `camiones`
  ADD CONSTRAINT `camiones_ibfk_1` FOREIGN KEY (`tipo_camion`) REFERENCES `tipo_camion`(`cod_tipo`) ON DELETE CASCADE
  ON UPDATE CASCADE;

-- Filtros para la tabla `cargas`

```

```

ALTER TABLE `cargas`
ADD CONSTRAINT `FK_cargas_clientes` FOREIGN KEY (`Carnet_de_Identidadcli`) REFERENCES `clientes` (`Carnet_de_Identidad`) ON UPDATE CASCADE,
ADD CONSTRAINT `FK_cargas_departamentos` FOREIGN KEY (`destino`) REFERENCES `lugares` (`cod_l`) ON UPDATE CASCADE,
ADD CONSTRAINT `FK_cargas_tipo_carga` FOREIGN KEY (`tipo`) REFERENCES `tipo_carga` (`cod_t`) ON UPDATE CASCADE,
ADD CONSTRAINT `FK_cargas_unidades_de_medida` FOREIGN KEY (`unidad_de_medida`) REFERENCES `unidades_de_medida` (`cod_um`) ON UPDATE CASCADE,
ADD CONSTRAINT `FK_cargas_usuarios_tra` FOREIGN KEY (`Carnet_de_Identidad`) REFERENCES `usuarios_tra` (`Carnet_de_Identidad`) ON UPDATE CASCADE,
ADD CONSTRAINT `cargas_ibfk_1` FOREIGN KEY (`ci_chofer`) REFERENCES `chofer_camion` (`cichof`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `cargas_ibfk_2` FOREIGN KEY (`placa_ch`) REFERENCES `chofer_camion` (`placa`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `cargas_ibfk_3` FOREIGN KEY (`placa_s`) REFERENCES `sociocamion` (`placa`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `cargas_ibfk_4` FOREIGN KEY (`ci_socio`) REFERENCES `sociocamion` (`cisocio`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Filtros para la tabla `choferes`

ALTER TABLE `choferes`
ADD CONSTRAINT `choferes_ibfk_1` FOREIGN KEY (`ci_chofer`) REFERENCES `usuarios_tra` (`Carnet_de_Identidad`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Filtros para la tabla `chofer_camion`

ALTER TABLE `chofer_camion`
ADD CONSTRAINT `chofer_camion_ibfk_2` FOREIGN KEY (`placa`) REFERENCES `camiones` (`placa`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `chofer_camion_ibfk_3` FOREIGN KEY (`cichof`) REFERENCES `choferes` (`ci_chofer`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Filtros para la tabla `clientes`

ALTER TABLE `clientes`
ADD CONSTRAINT `FK_clientes_empresas` FOREIGN KEY (`cod_em`) REFERENCES `empresas` (`cod_em`) ON UPDATE CASCADE,
ADD CONSTRAINT `FK_clientes_usuarios_tra` FOREIGN KEY (`Carnet_de_Identidad`) REFERENCES `usuarios_tra` (`Carnet_de_Identidad`) ON UPDATE CASCADE;

-- Filtros para la tabla `contrasena`

```

```

-- ALTER TABLE `contrasena`  

ADD CONSTRAINT `FK_contrasena_usuarios_tra` FOREIGN KEY (`Carnet_de_Identidad`) REFERENCES `usuarios_tra`  

(`Carnet_de_Identidad`) ON UPDATE CASCADE;  

-- Filtros para la tabla `lugares`  

--  

ALTER TABLE `lugares`  

ADD CONSTRAINT `lugares_ibfk_1` FOREIGN KEY (`cod_de`) REFERENCES `departamentos` (`cod_de`) ON DELETE CASCADE ON  

UPDATE CASCADE;  

-- Filtros para la tabla `reuniones`  

--  

ALTER TABLE `reuniones`  

ADD CONSTRAINT `FK_reuniones_usuarios_tra` FOREIGN KEY (`Carnet_de_Identidad`) REFERENCES `usuarios_tra`  

(`Carnet_de_Identidad`) ON UPDATE CASCADE;  

-- Filtros para la tabla `sociocamion`  

--  

ALTER TABLE `sociocamion`  

ADD CONSTRAINT `sociocamion_ibfk_1` FOREIGN KEY (`cisocio`) REFERENCES `socios` (`Carnet_de_Identidad`) ON DELETE  

CASCADE ON UPDATE CASCADE,  

ADD CONSTRAINT `sociocamion_ibfk_2` FOREIGN KEY (`placa`) REFERENCES `camiones` (`placa`) ON DELETE CASCADE ON  

UPDATE CASCADE;  

-- Filtros para la tabla `socios`  

--  

ALTER TABLE `socios`  

ADD CONSTRAINT `FK_socios_usuarios_tra` FOREIGN KEY (`Carnet_de_Identidad`) REFERENCES `usuarios_tra` (`Carnet_de_Identidad`)  

ON UPDATE CASCADE;  

-- Filtros para la tabla `usuarios_tra`  

--  

ALTER TABLE `usuarios_tra`  

ADD CONSTRAINT `FK_usuarios_tra_roles` FOREIGN KEY (`rol`) REFERENCES `roles` (`cod_rol`) ON UPDATE CASCADE;  

COMMIT;

```

### **1.5.1.7.9 Diagrama de secuencia**

#### **1.5.1.7.9.1 Introducción**

Un diagrama de secuencia del sistema es un artefacto creado de manera rápida y fácil que muestra los eventos de entrada y salida relacionados con el sistema que se está estudiando. UML incluye la notación de los diagramas de secuencia. Los diagramas de secuencia es un dibujo que muestra, para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas.

#### **1.5.1.7.9.2 Propósito**

- Comprender la estructura y dinámica del sistema deseado para la organización
- Comprender la interacción de los actores del sistema

#### **1.5.1.7.9.3 Alcance**

- Describe un escenario específico de un caso de uso
- Representa las interacciones entre actores y operaciones que inician

#### 1.5.1.7.9.4 Modelado de diagramas de secuencia

##### 1.5.1.7.9.4.1 Acceso al sistema

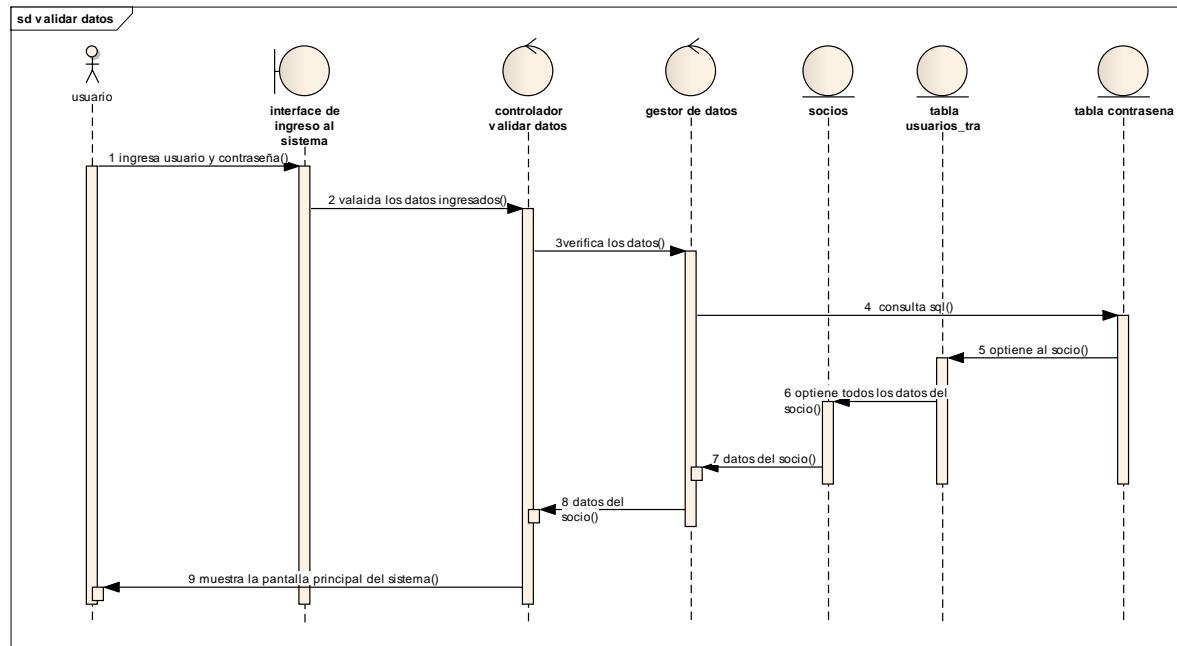


Figura 36 Acceso al sistema

##### 1.5.1.7.9.4.2 Añadir socio

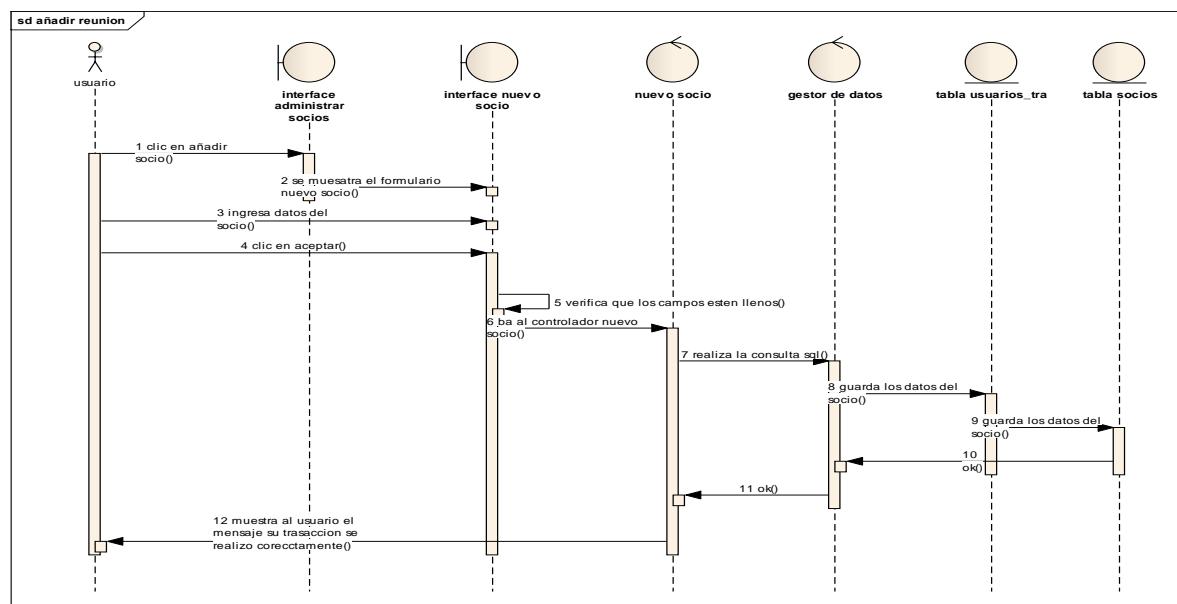


Figura 37 Añadir socio

### 1.5.1.7.9.4.3 Modificar socio

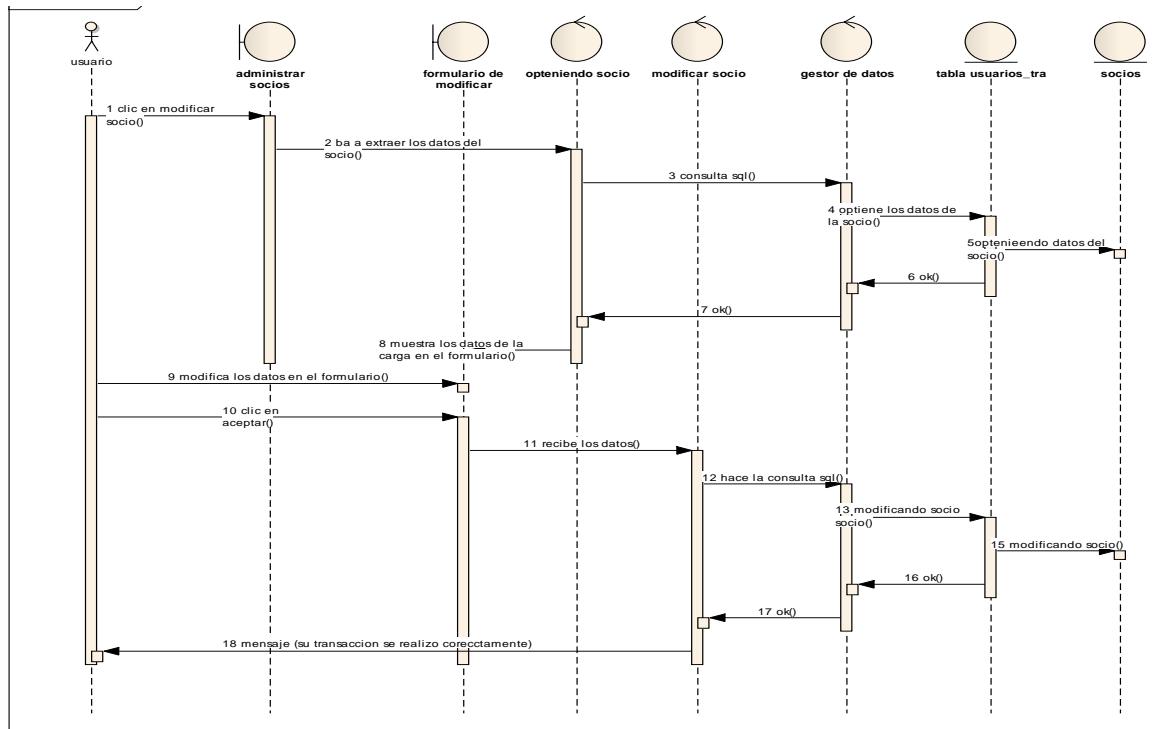


Figura 38 Modificar socio

### 1.5.1.7.9.4.4 Eliminar socio

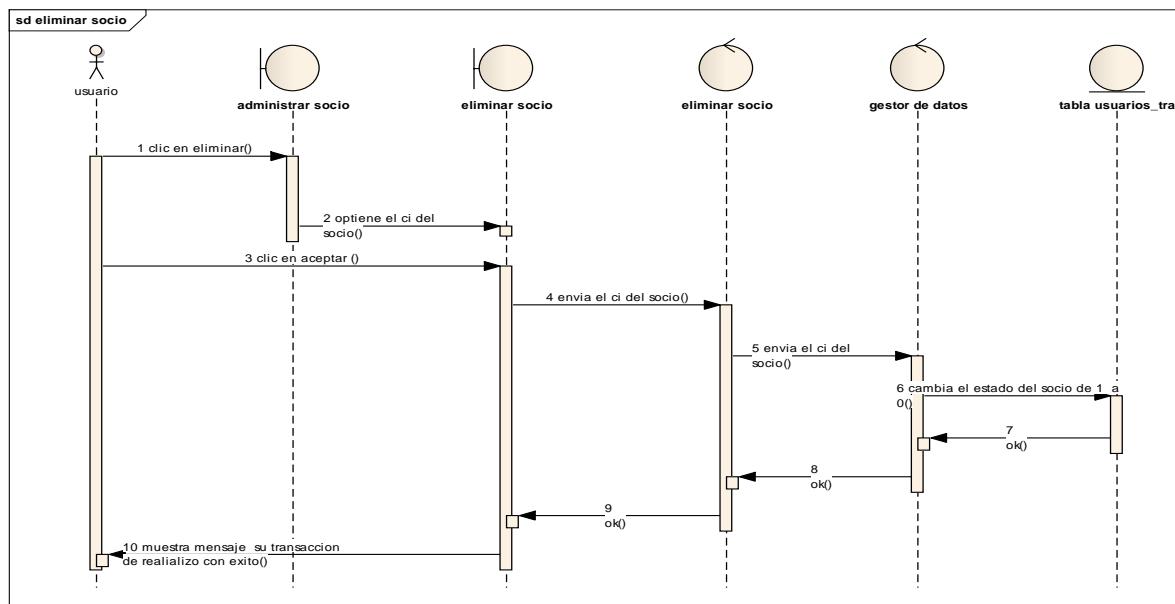


Figura 39 Eliminar socio

#### 1.5.1.7.9.4.5 Ver socio

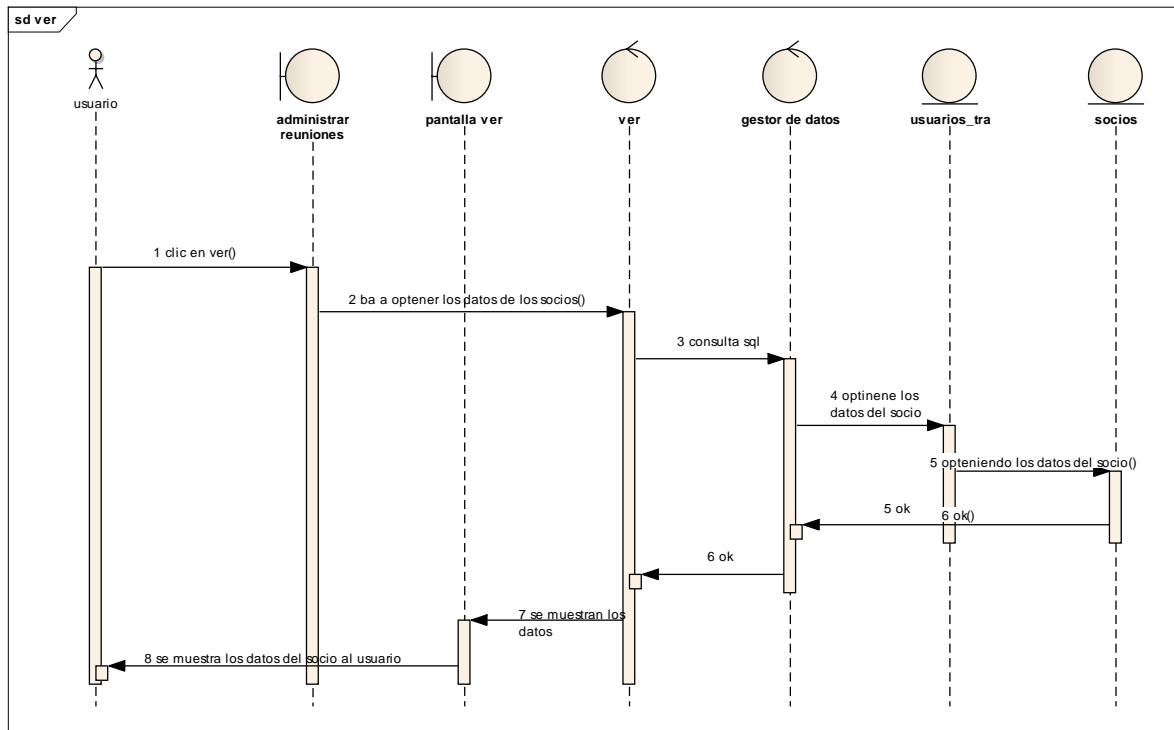


Figura 40 Ver socio

#### 1.5.1.7.9.4.6 Socios inactivos

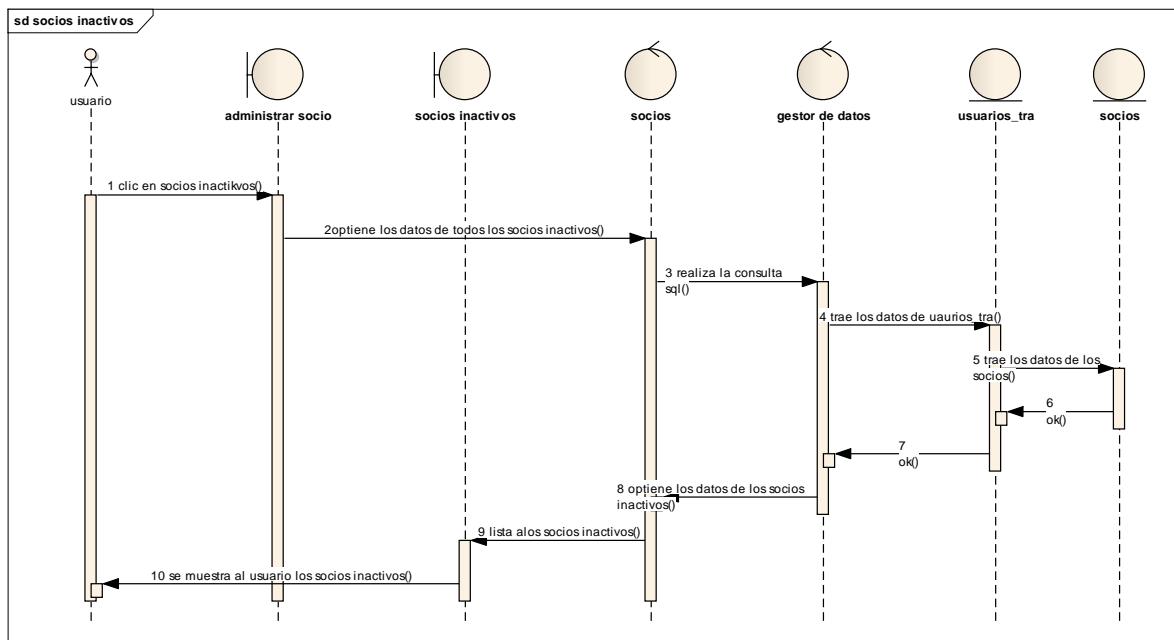


Figura 41 Ver socios inactivos

### 1.5.1.7.9.4.7 Habilitar socio

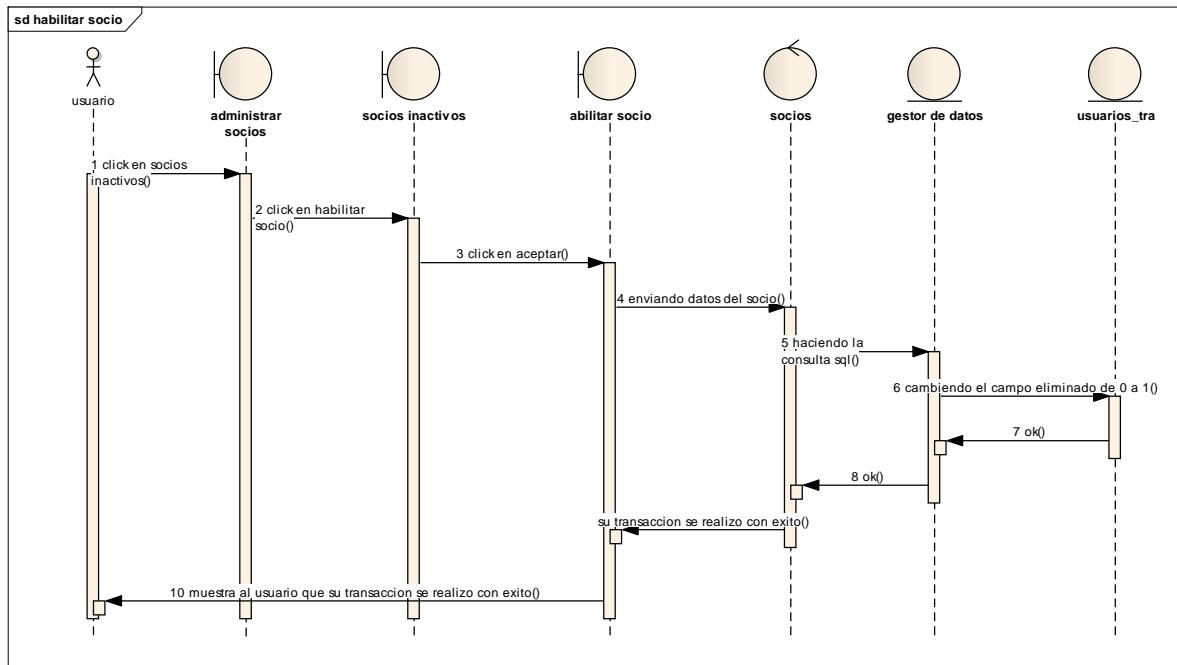


Figura 42 Habilitar socio

### 1.5.1.7.9.4.8 Reporte socio

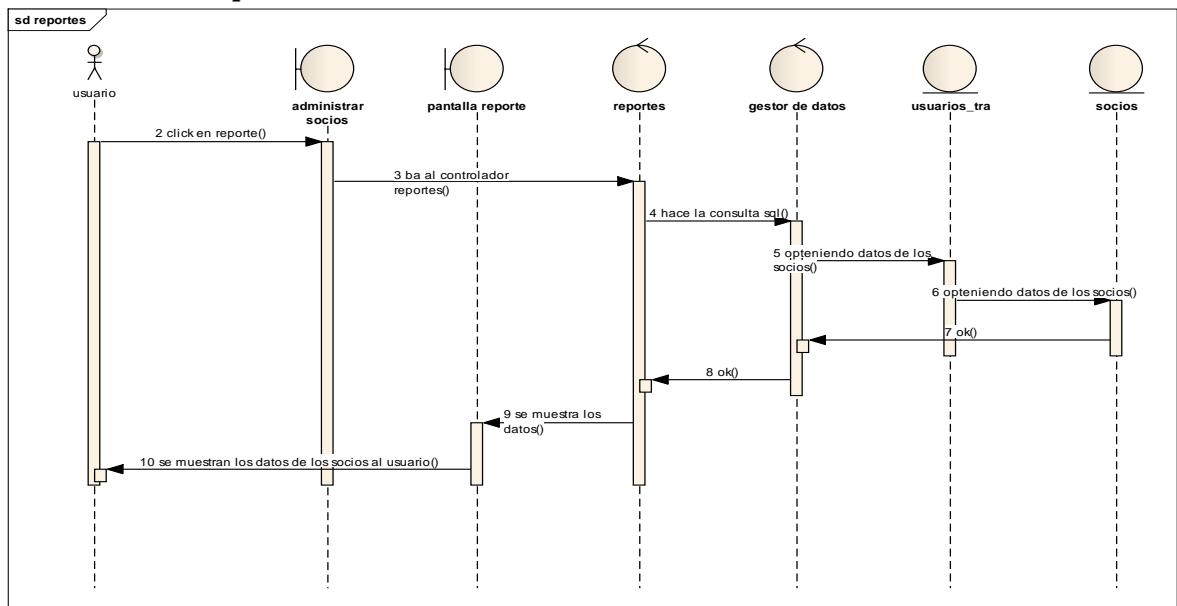


Figura 43 Reporte socio

### 1.5.1.7.9.4.9 Añadir cliente

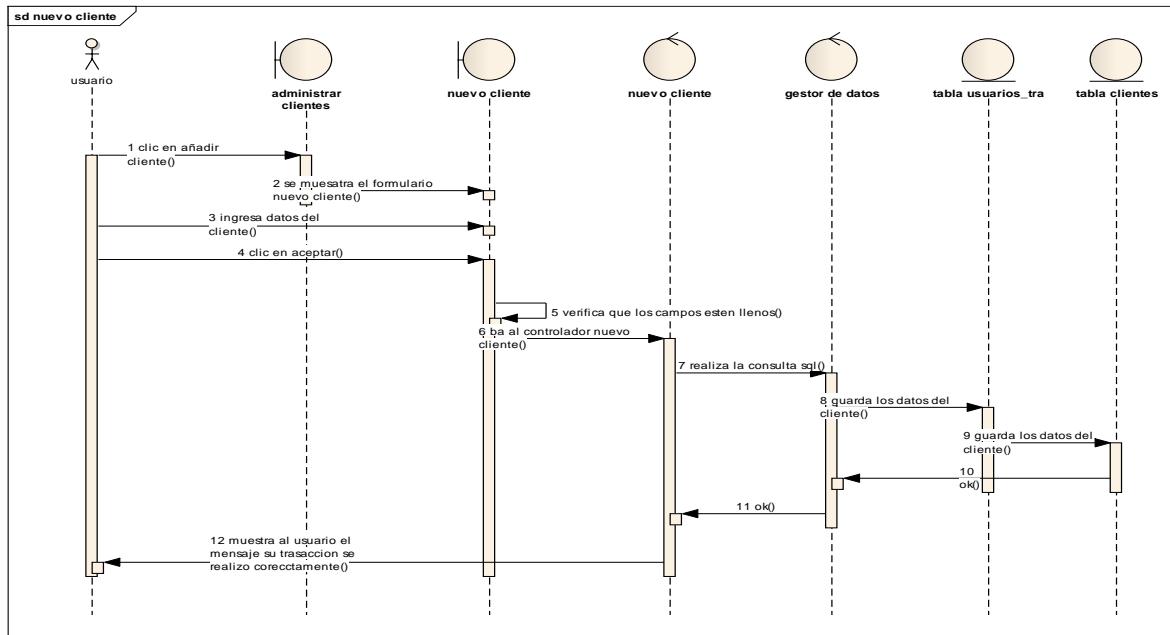


Figura 44 Añadir cliente

### 1.5.1.7.9.4.10 Añadir empresa

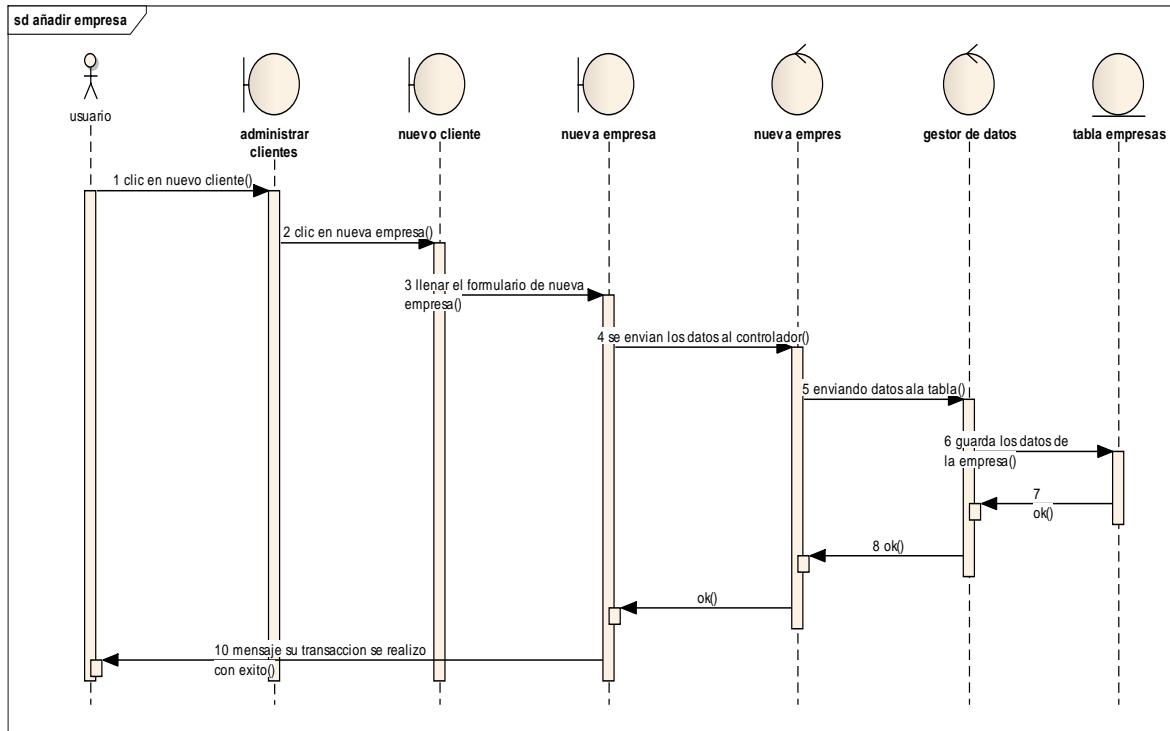


Figura 45 Añadir empresa

### 1.5.1.7.9.4.11 Modificar cliente

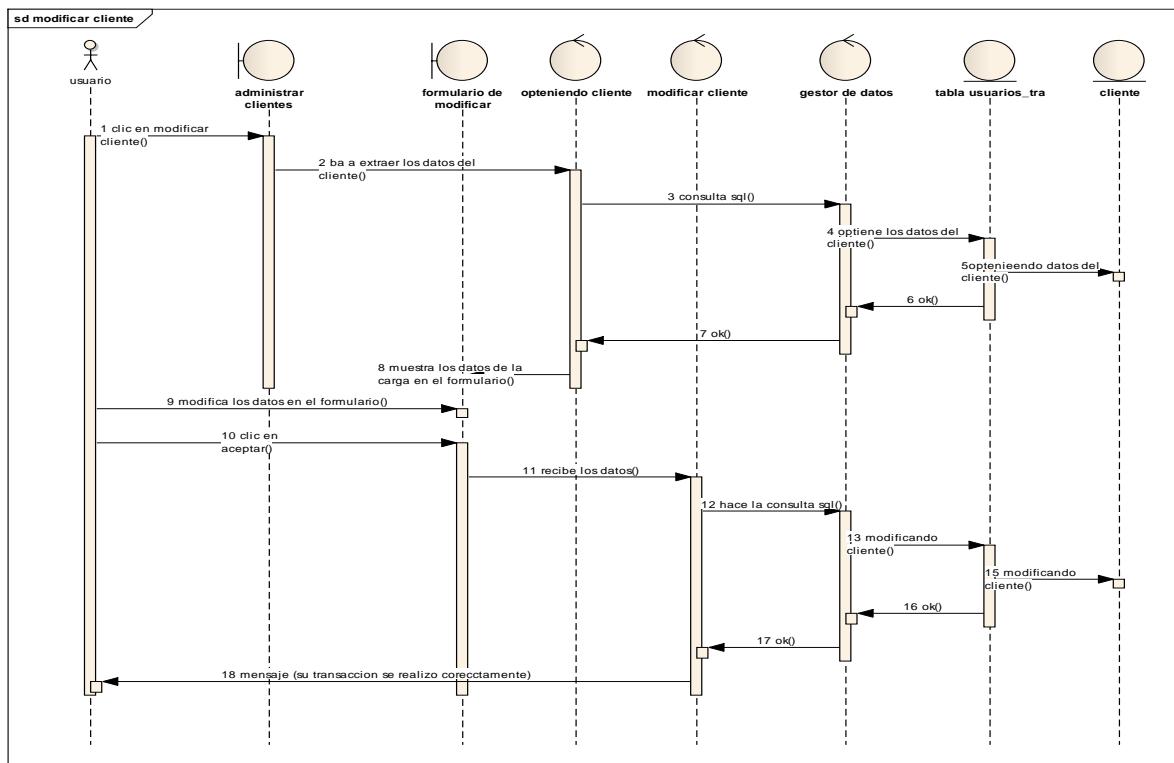


Figura 46 Modificar cliente

### 1.5.1.7.9.4.12 Eliminar cliente

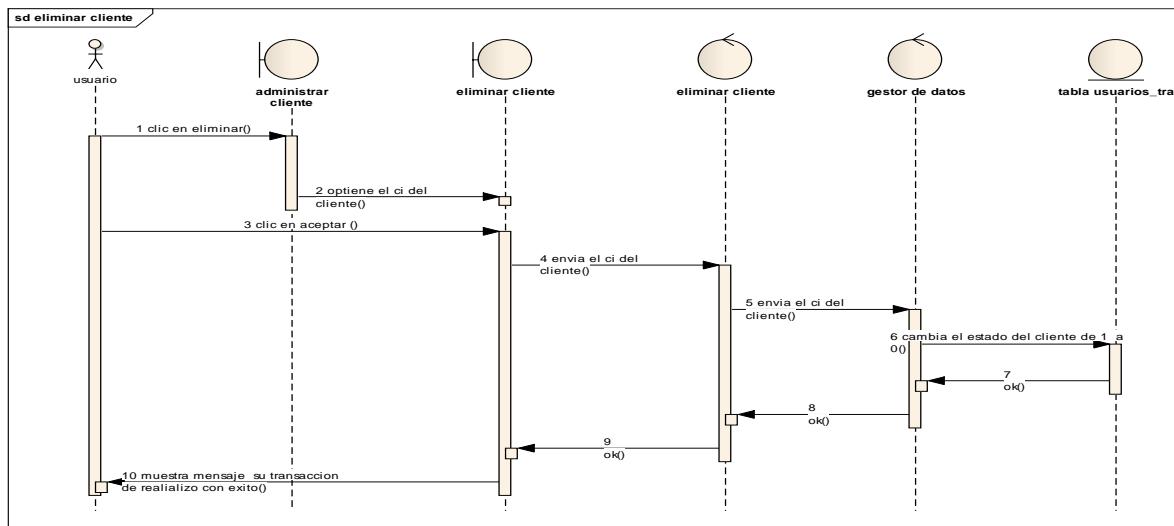


Figura 47 Eliminar cliente

### 1.5.1.7.9.4.13 Ver cliente

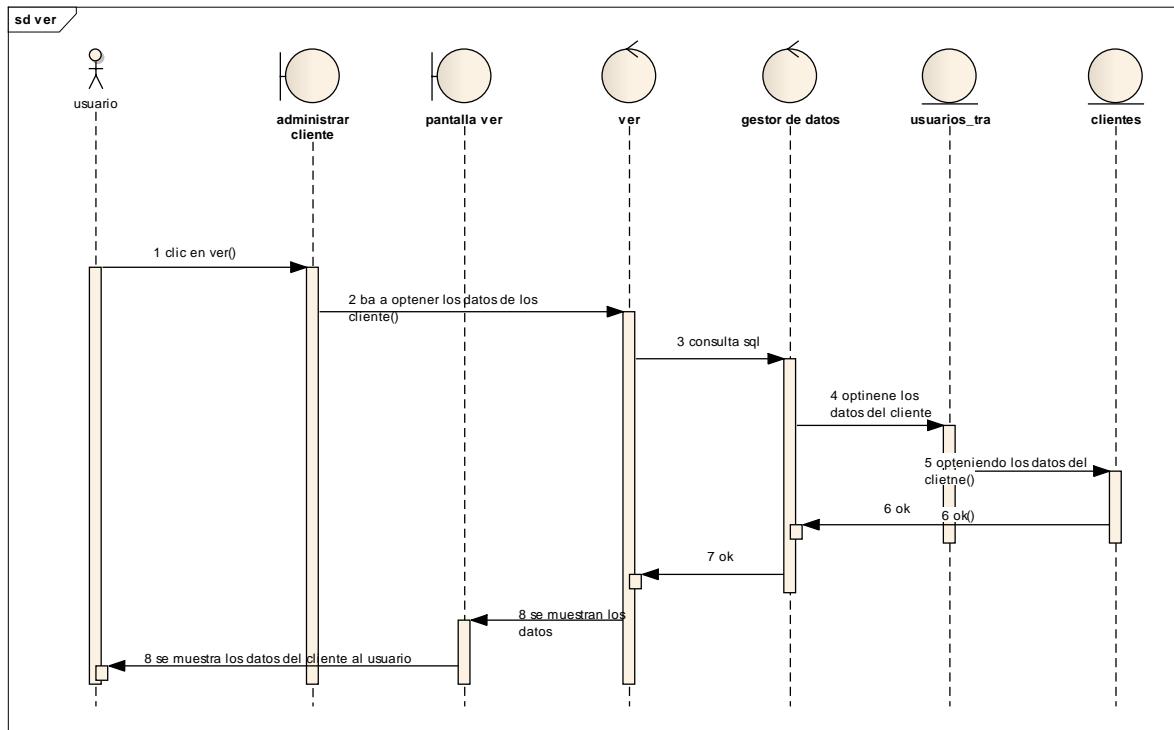


Figura 48 Ver cliente

### 1.5.1.7.9.4.14 Clientes inactivos

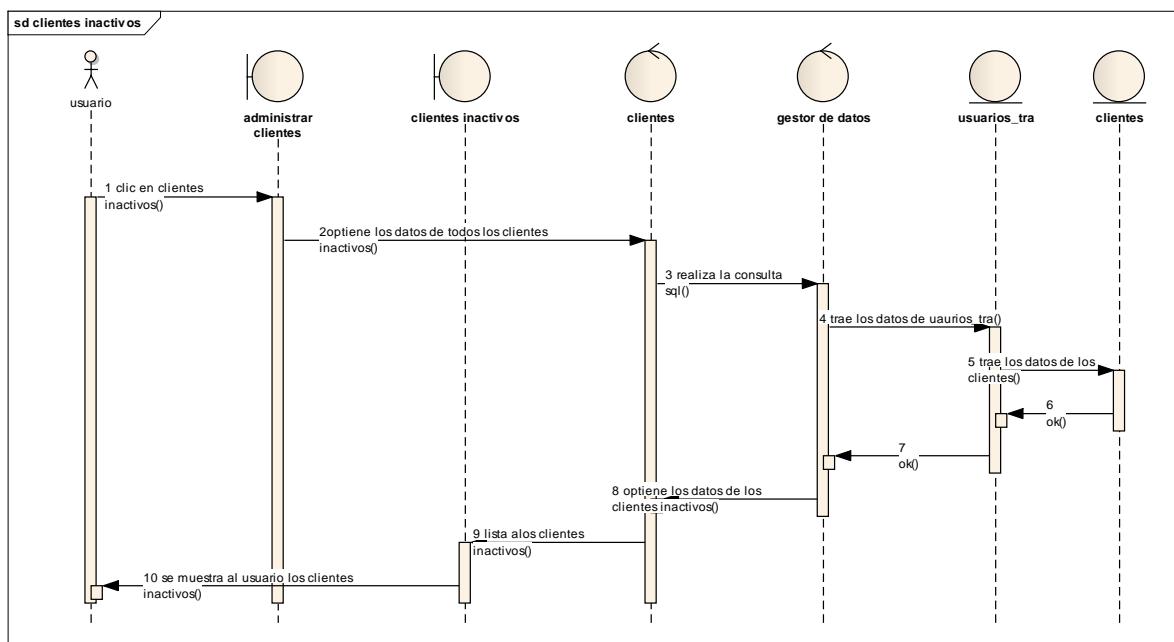


Figura 49 Clientes inactivos

### 1.5.1.7.9.4.15 Habilitar cliente

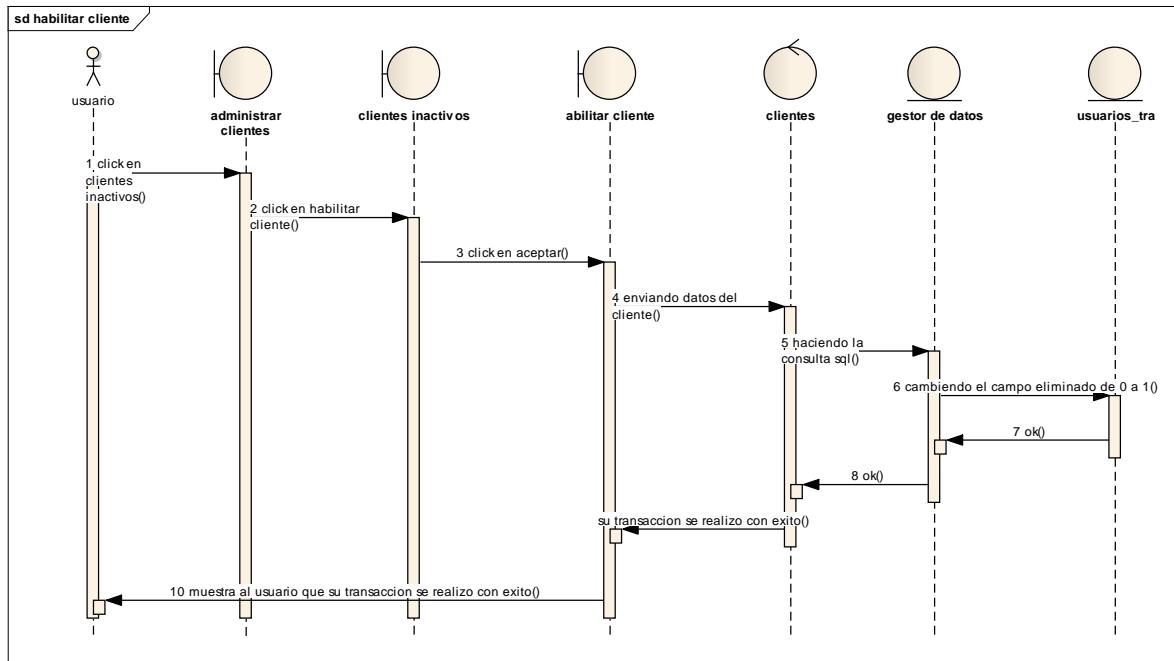


Figura 50 Habilitar cliente

### 1.5.1.7.9.4.16 Reporte cliente

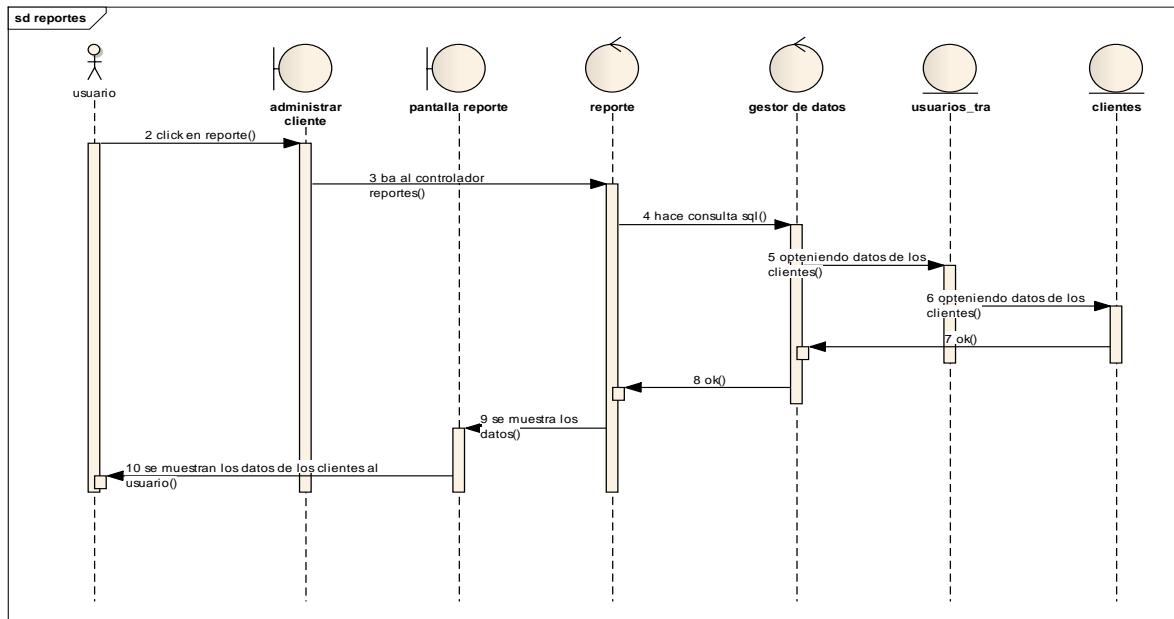


Figura 51 Reporte cliente

### 1.5.1.7.9.4.17 Añadir empresa

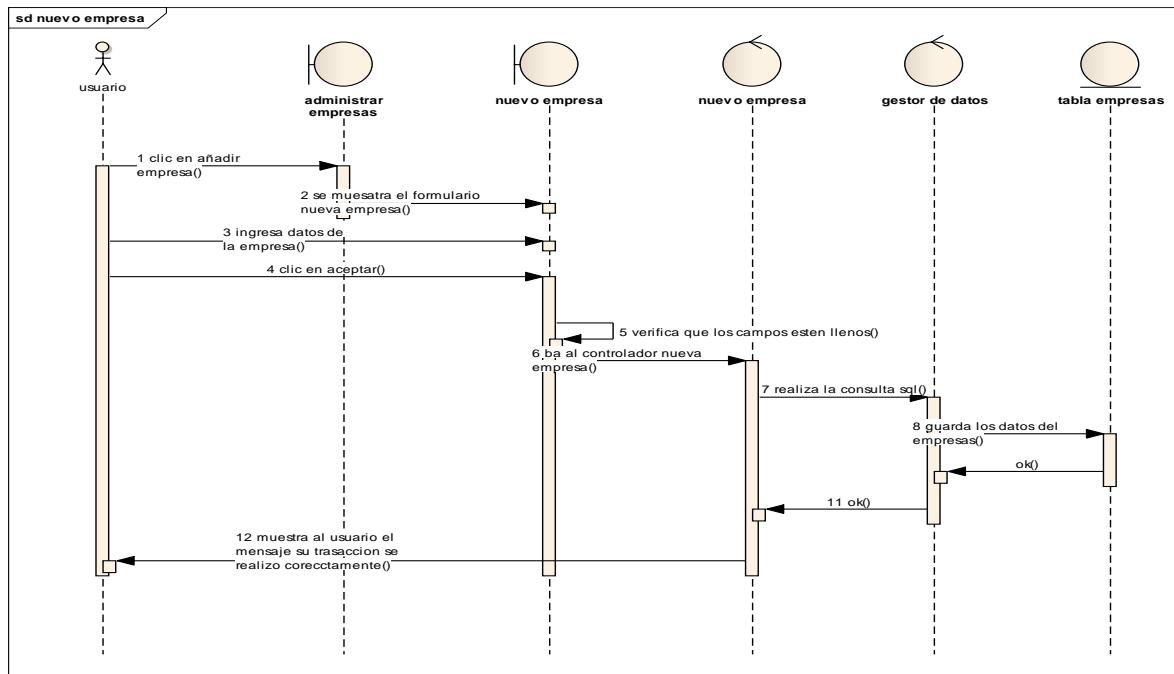


Figura 52 Añadir empresa

### 1.5.1.7.9.4.18 Modificar empresa

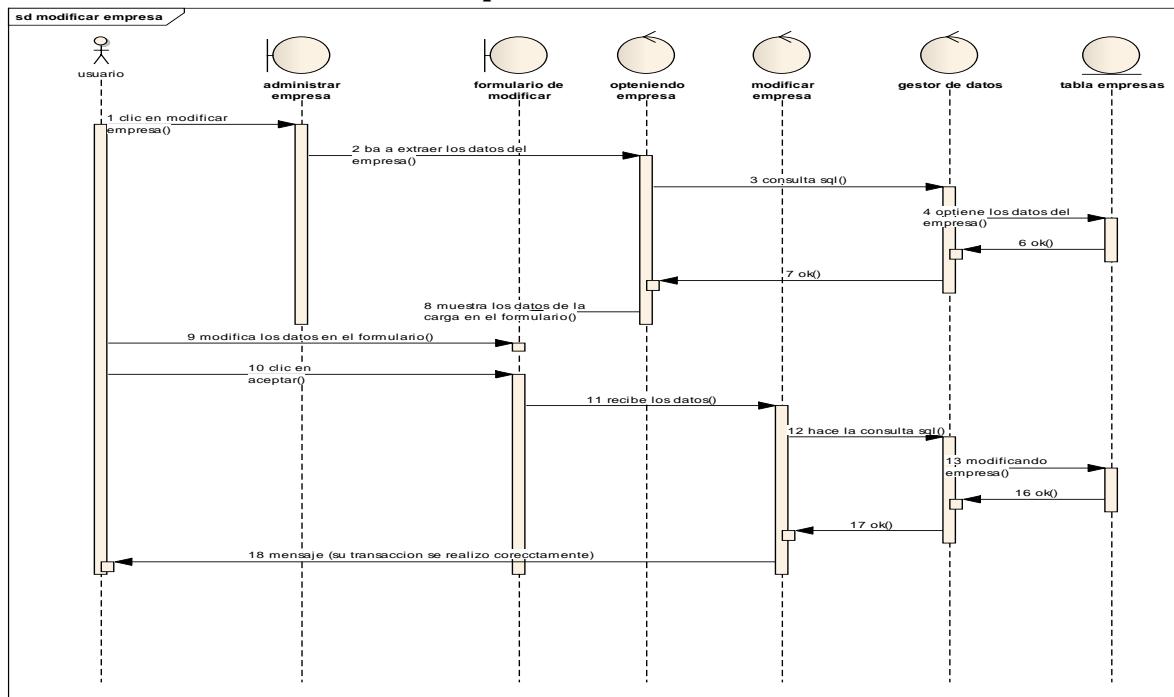


Figura 53 Modificar empresa

### 1.5.1.7.9.4.19 Eliminar empresa

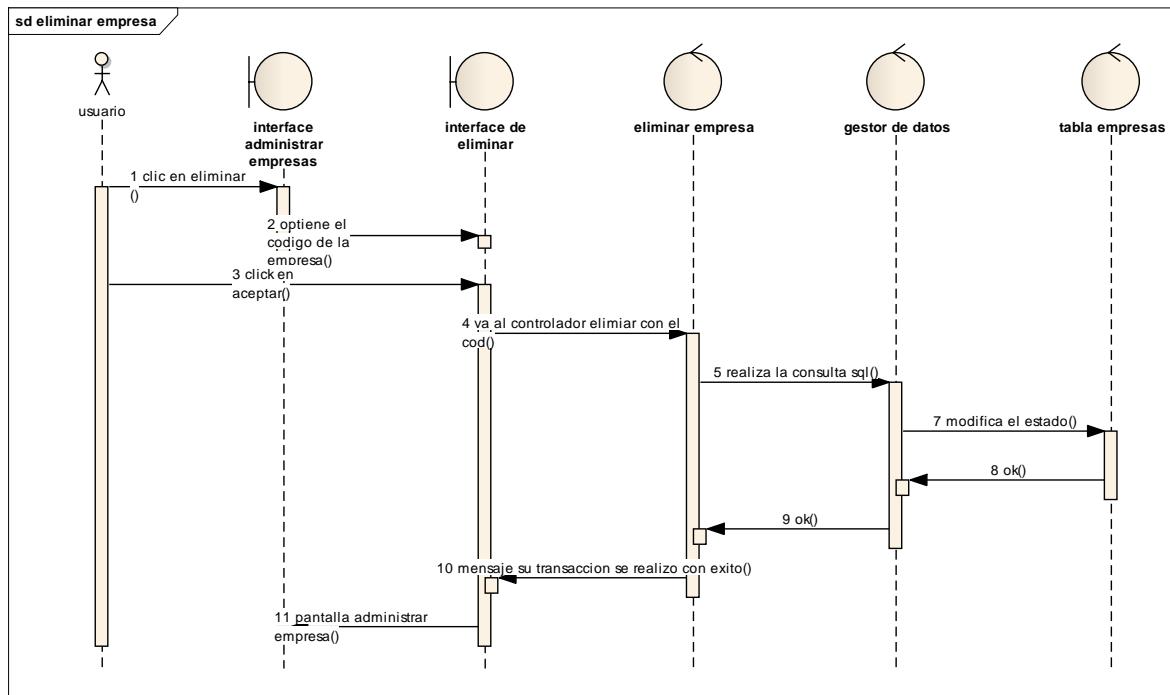


Figura 54 Eliminar empresa

### 1.5.1.7.9.4.20 Empresas inactivas

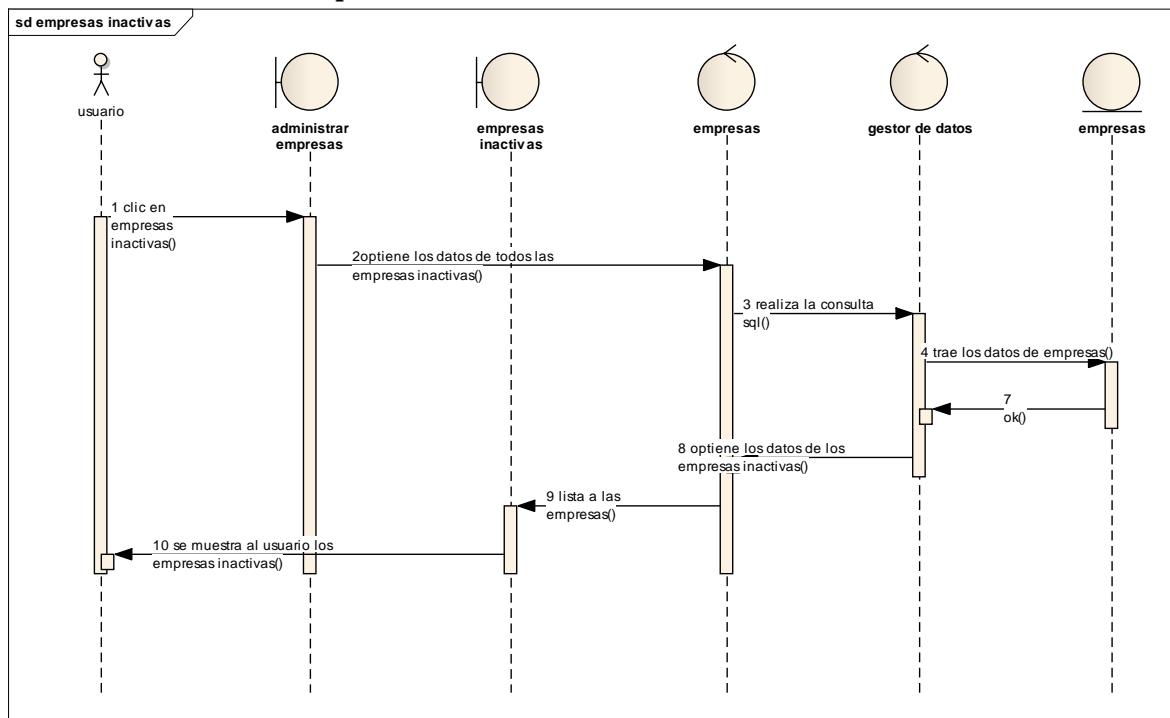


Figura 55 Empresas Inactivas

### 1.5.1.7.9.4.21 Habilitar empresa

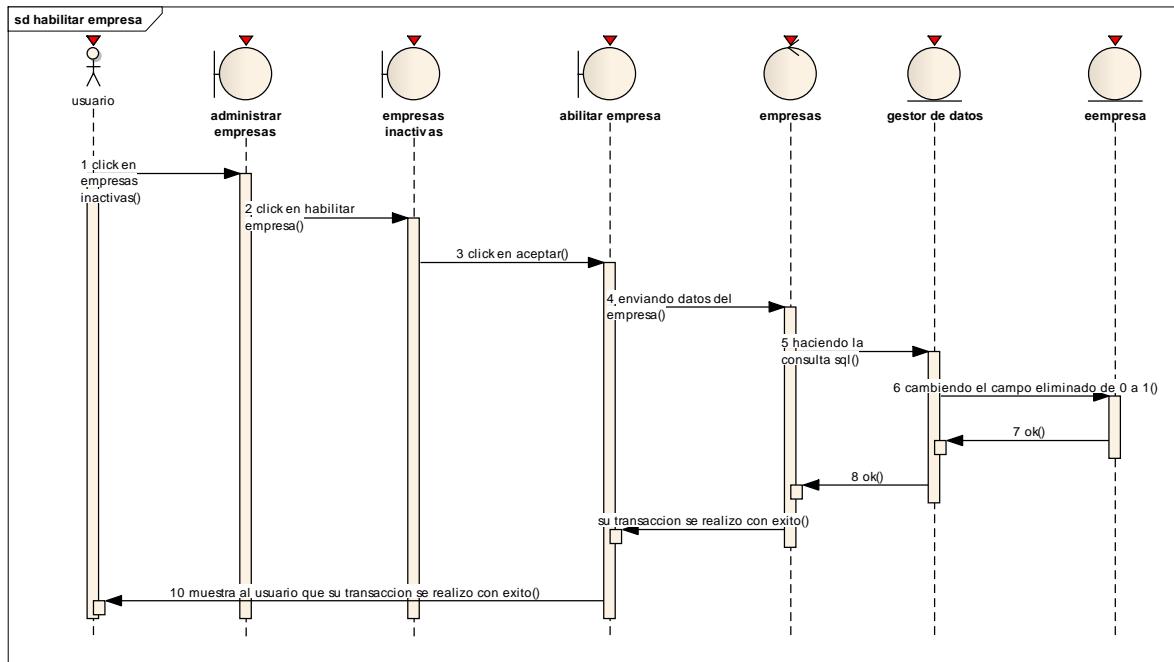


Figura 56 Habilitar empresa

### 1.5.1.7.9.4.22 Reporte

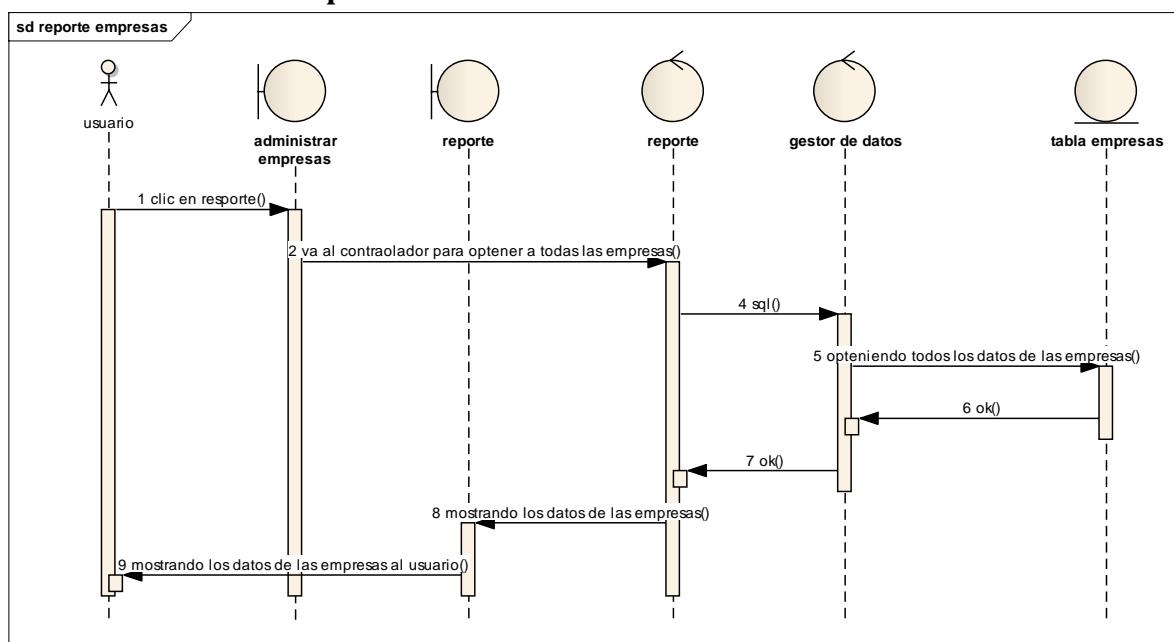


Figura 57 Reporte

### 1.5.1.7.9.4.23 Añadir trabajador

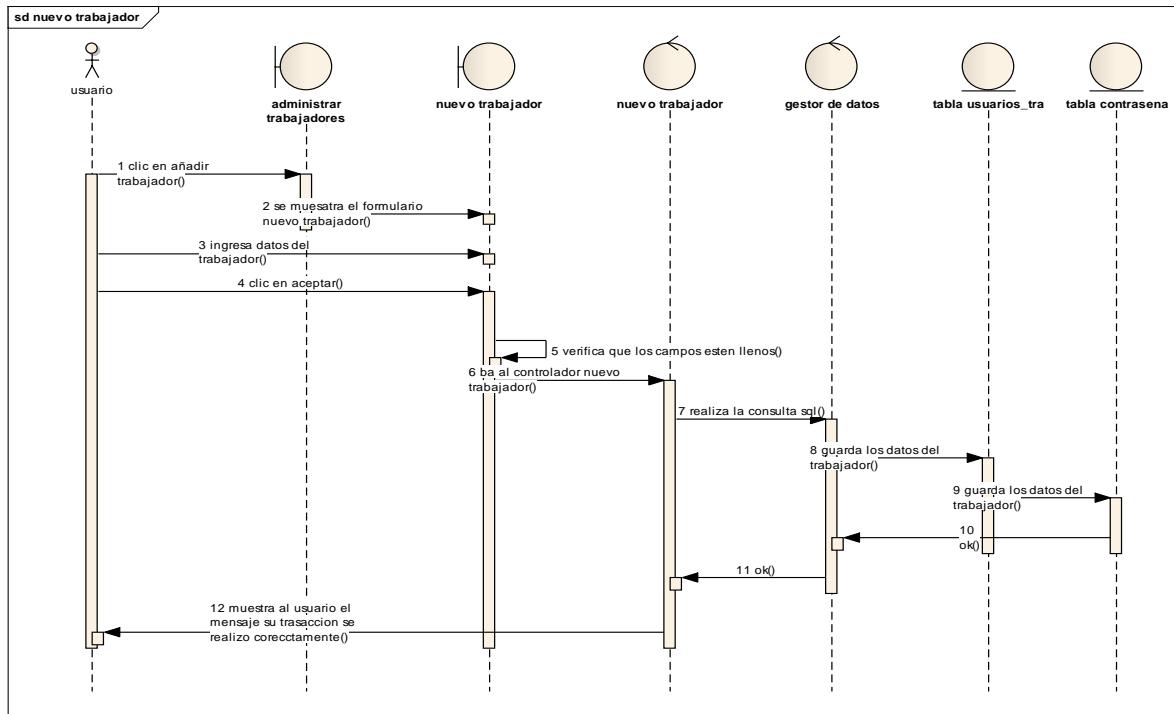


Figura 58 Añadir trabajador

### 1.5.1.7.9.4.24 Modificar trabajador

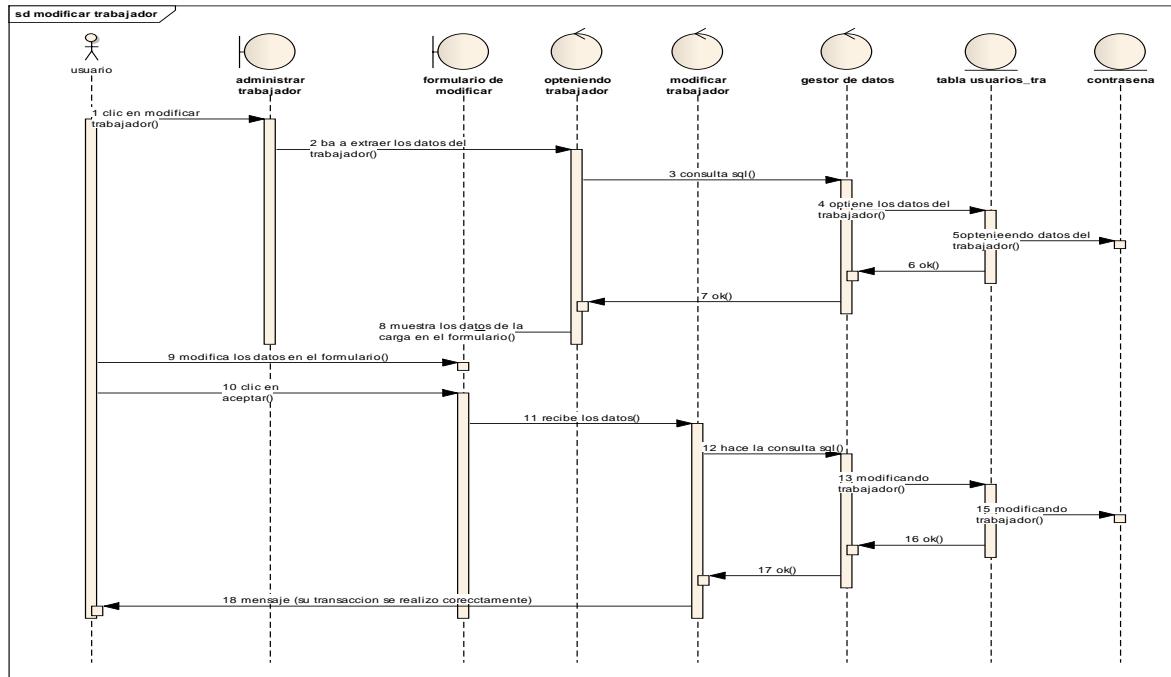


Figura 59 Modificar trabajador

### 1.5.1.7.9.4.25 Eliminar trabajador

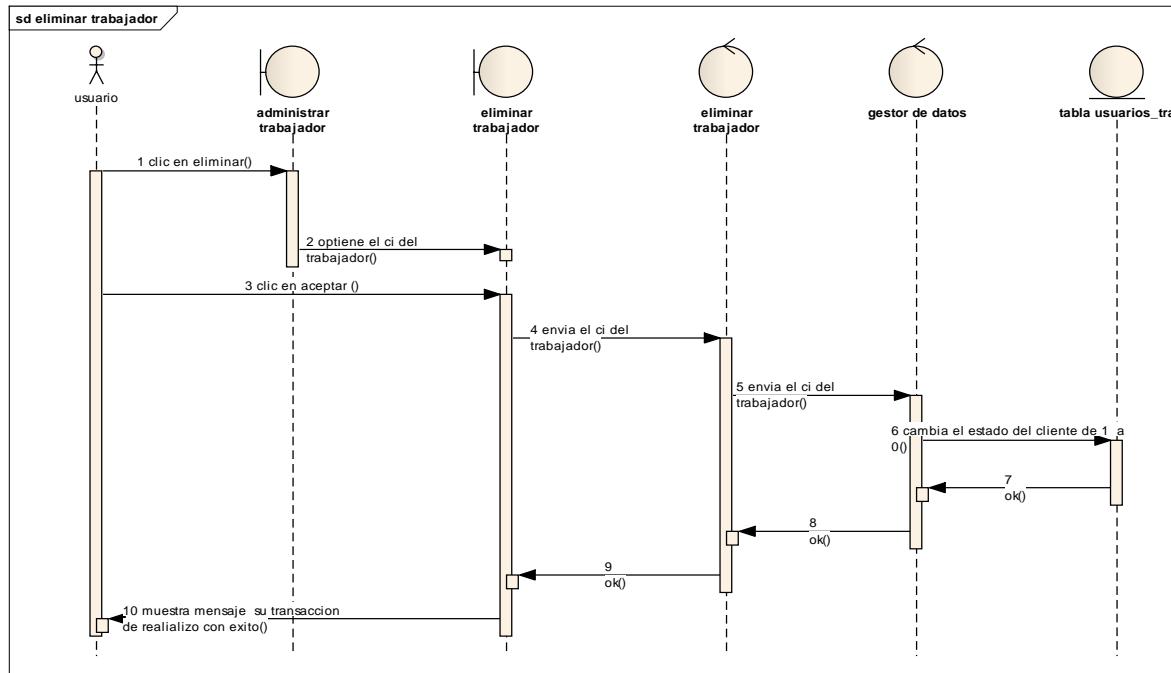


Figura 60 Eliminar trabajador

### 1.5.1.7.9.4.26 Ver trabajador

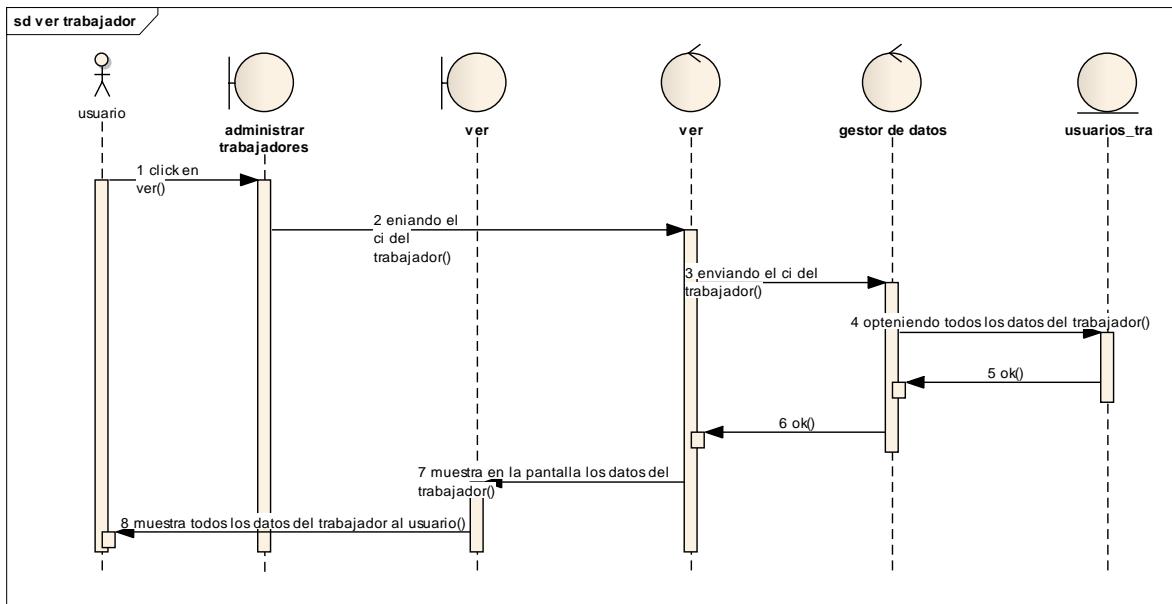


Figura 61 Ver trabajador

### 1.5.1.7.9.4.27 Trabajadores inactivos

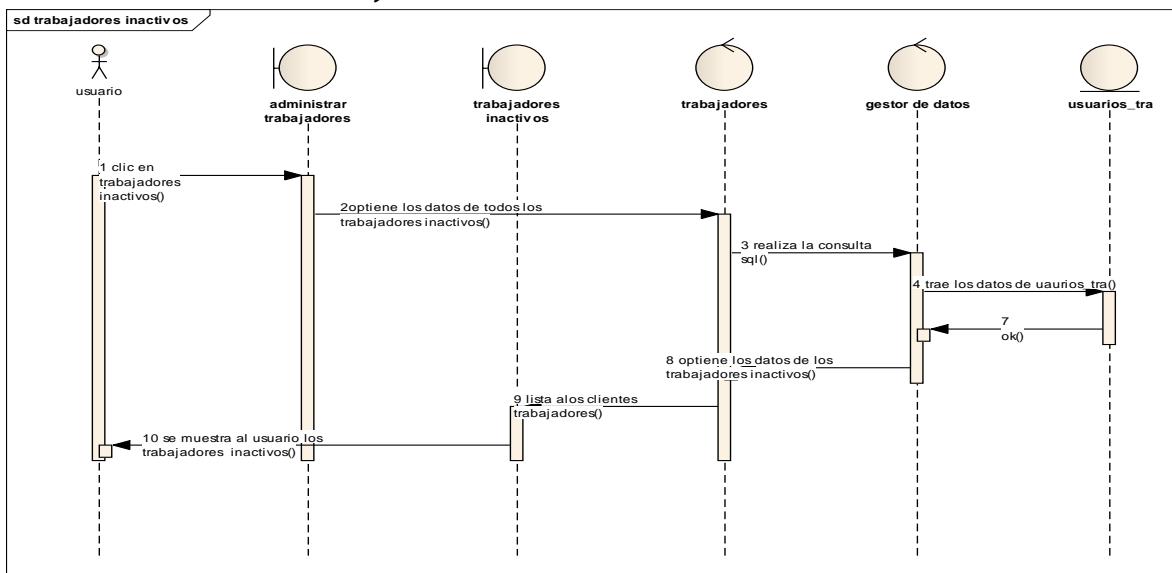


Figura 62 Trabajadores inactivos

### 1.5.1.7.9.4.28 Habilitar trabajador

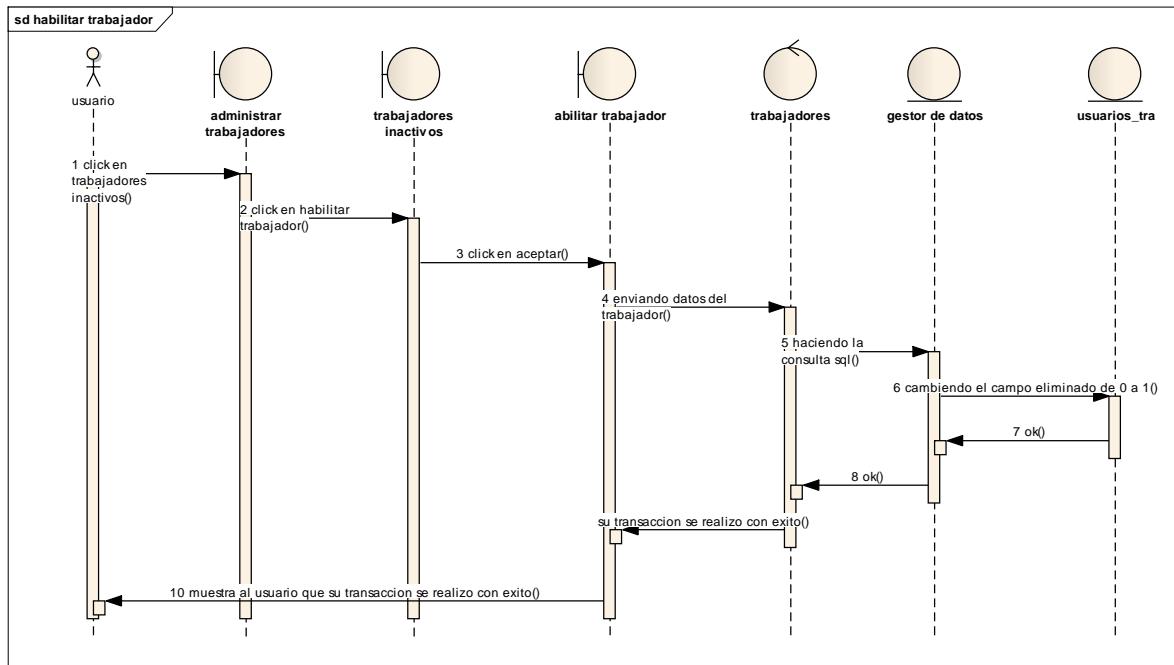


Figura 63 Habilitar trabajador

### 1.5.1.7.9.4.29 Reporte trabajador

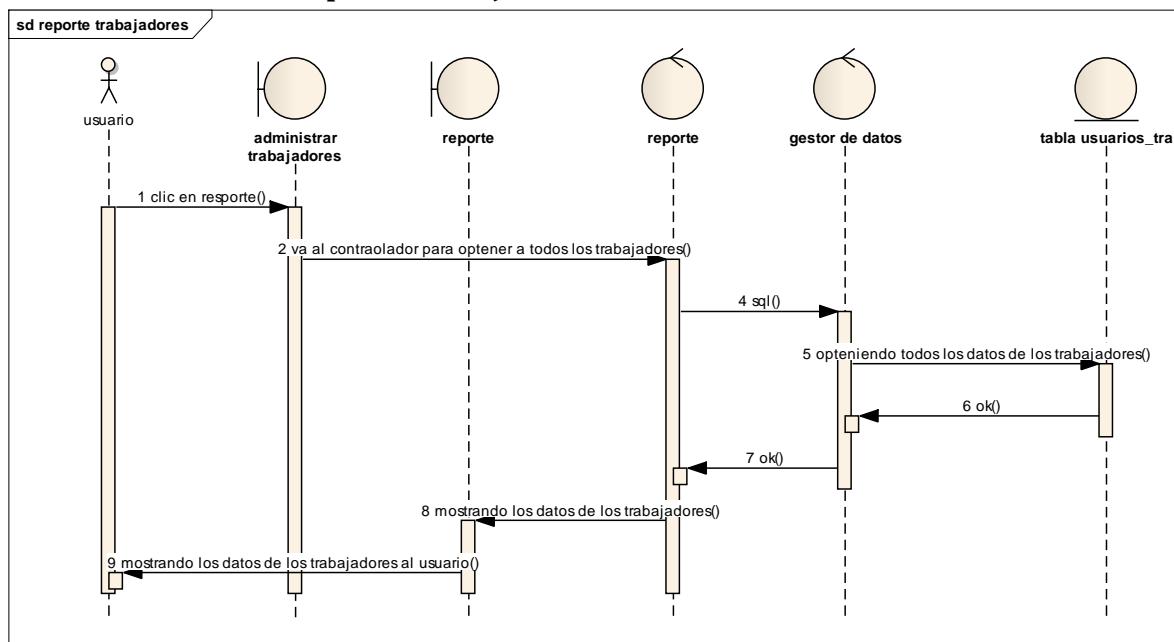


Figura 64 Reporte trabajador

### 1.5.1.7.9.4.30 Añadir chofer

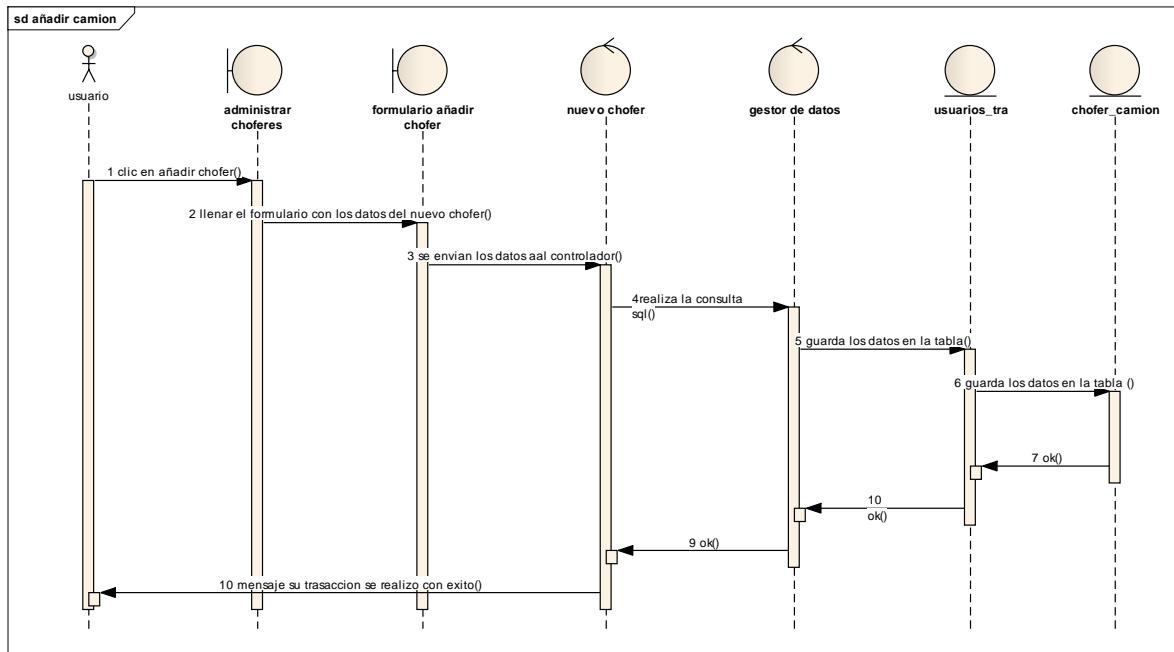


Figura 65 Añadir chofer

### 1.5.1.7.9.4.31 Modificar chofer

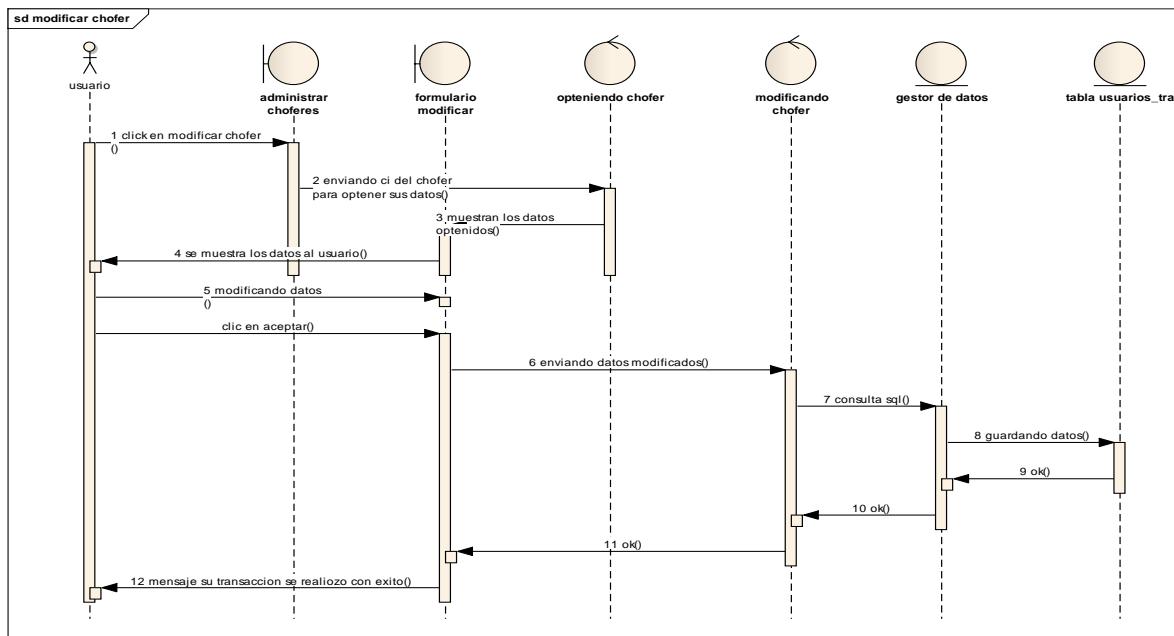


Figura 66 Modificar chofer

### 1.5.1.7.9.4.32 Eliminar chofer

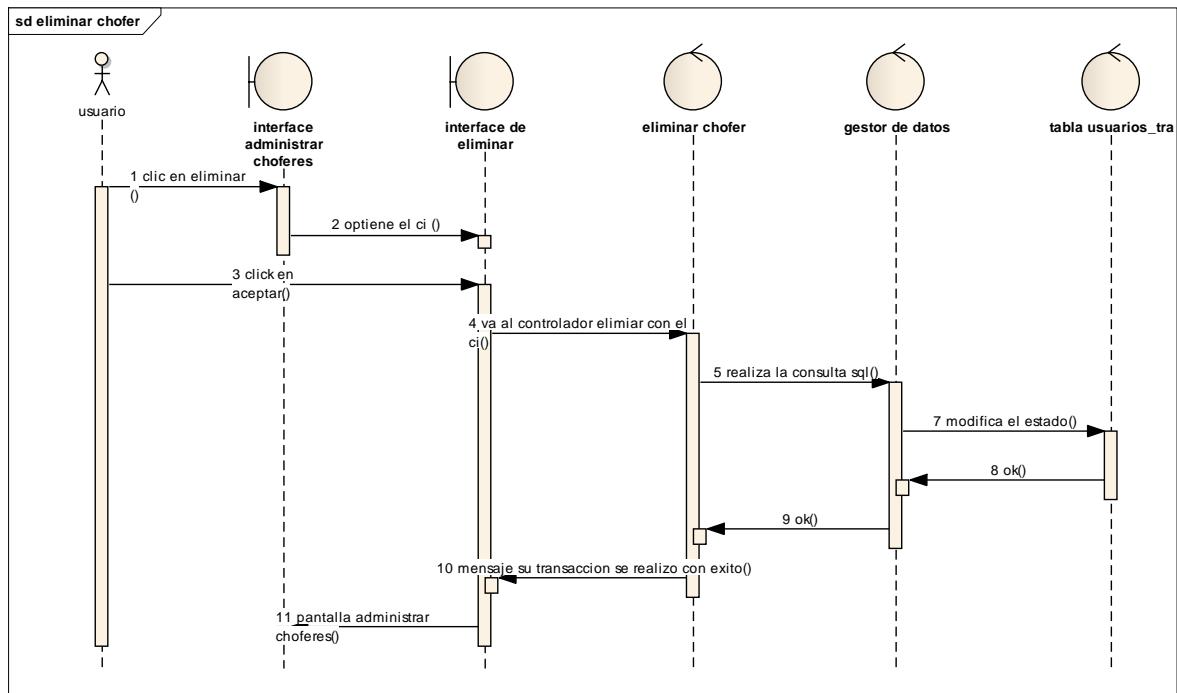


Figura 67 Eliminar chofer

### 1.5.1.7.9.4.33 Ver chofer

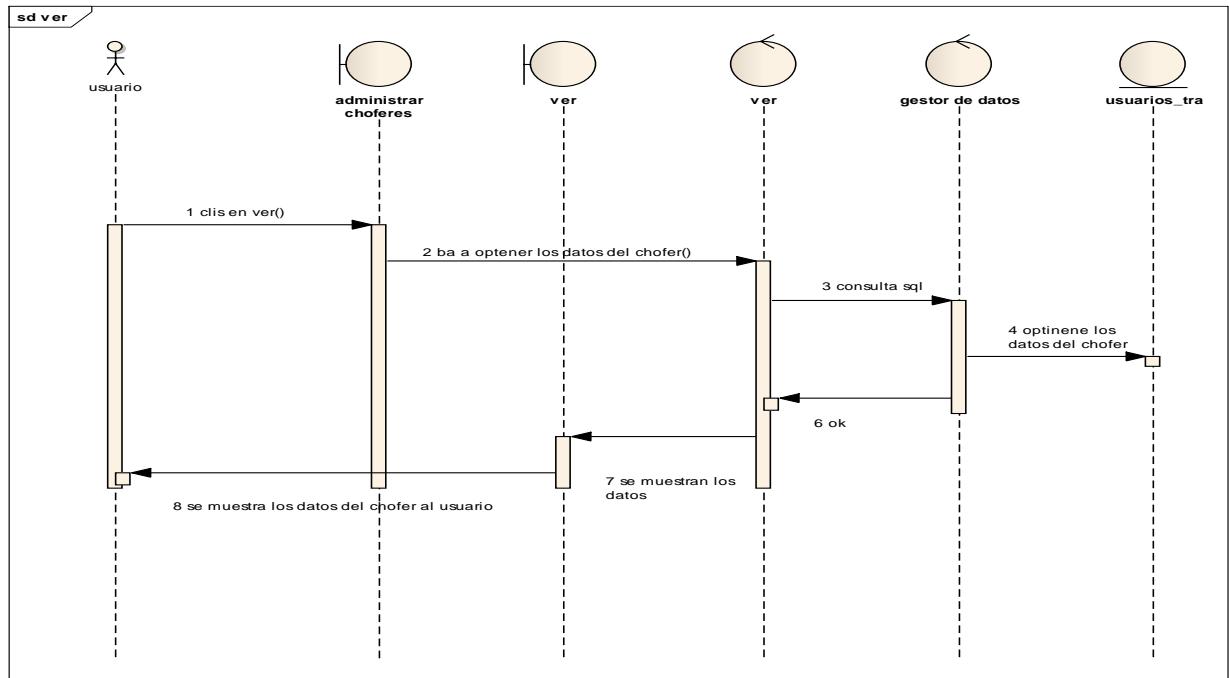


Figura 68 Ver chofer

### 1.5.1.7.9.4.34 Choferes inactivos

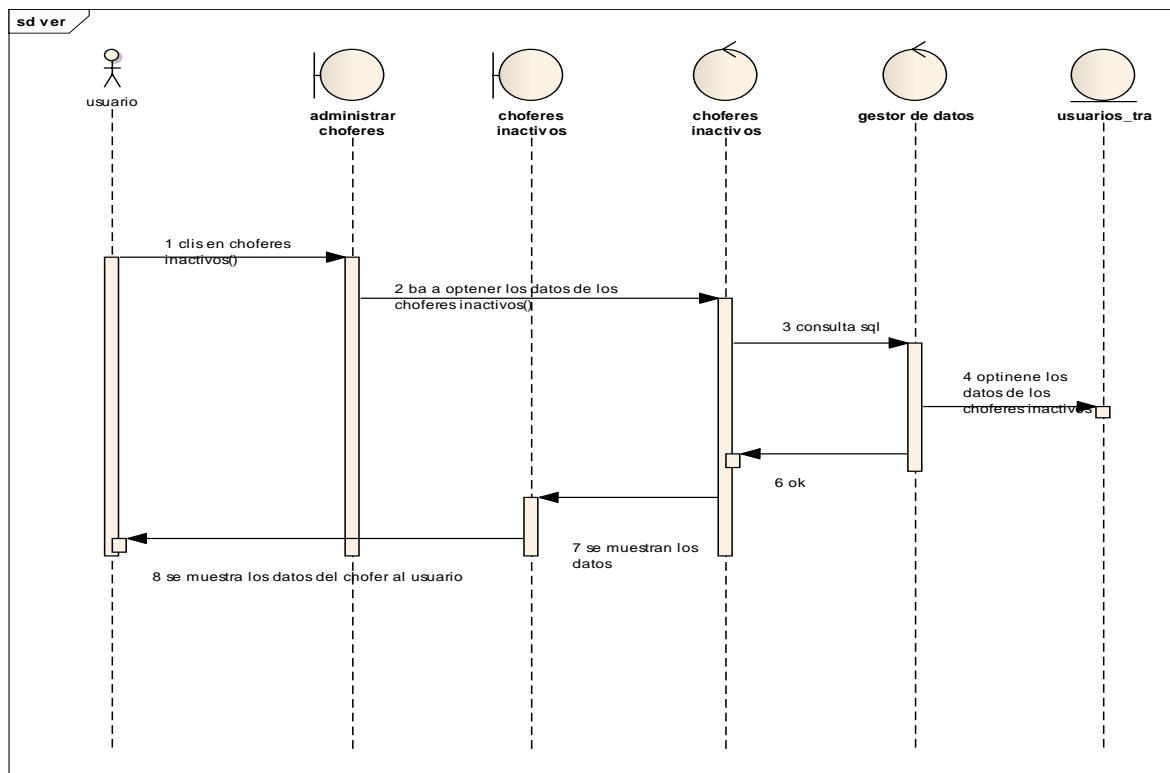


Figura 69 Choferes inactivos

### 1.5.1.7.9.4.35 Habilitar chofer

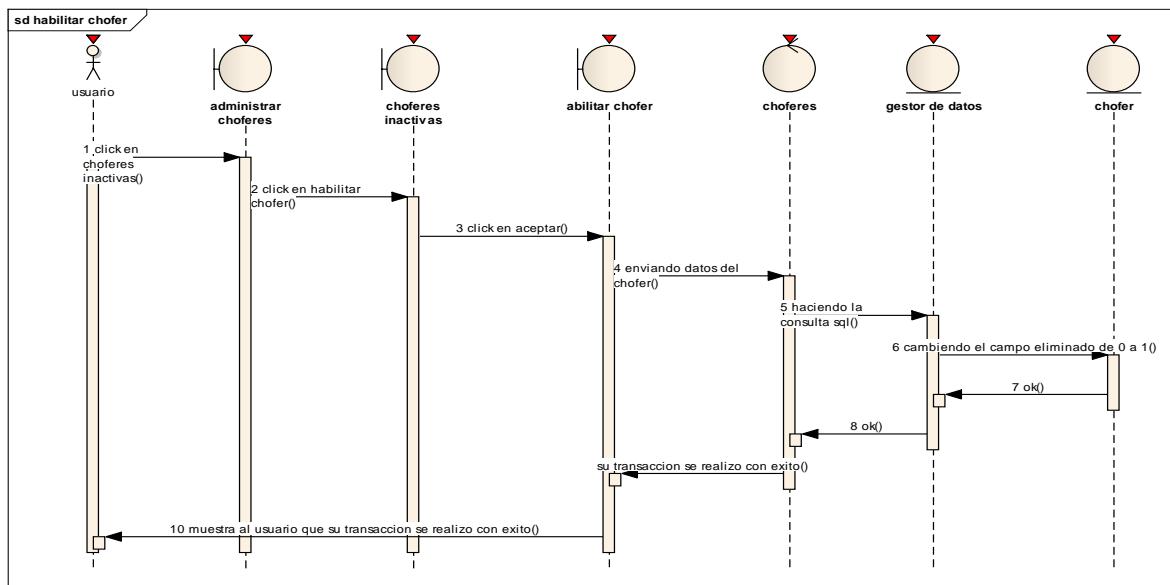


Figura 70 Habilitar chofer

### 1.5.1.7.9.4.36 Reporte

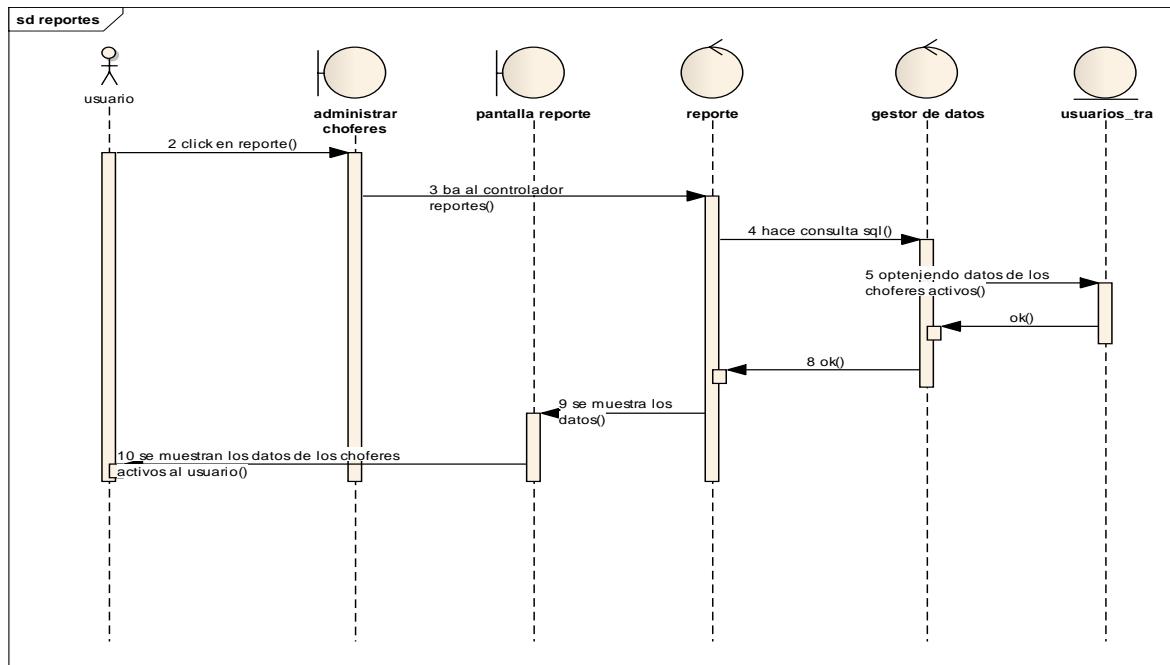


Figura 71 Reporte

### 1.5.1.7.9.4.37 Añadir camión

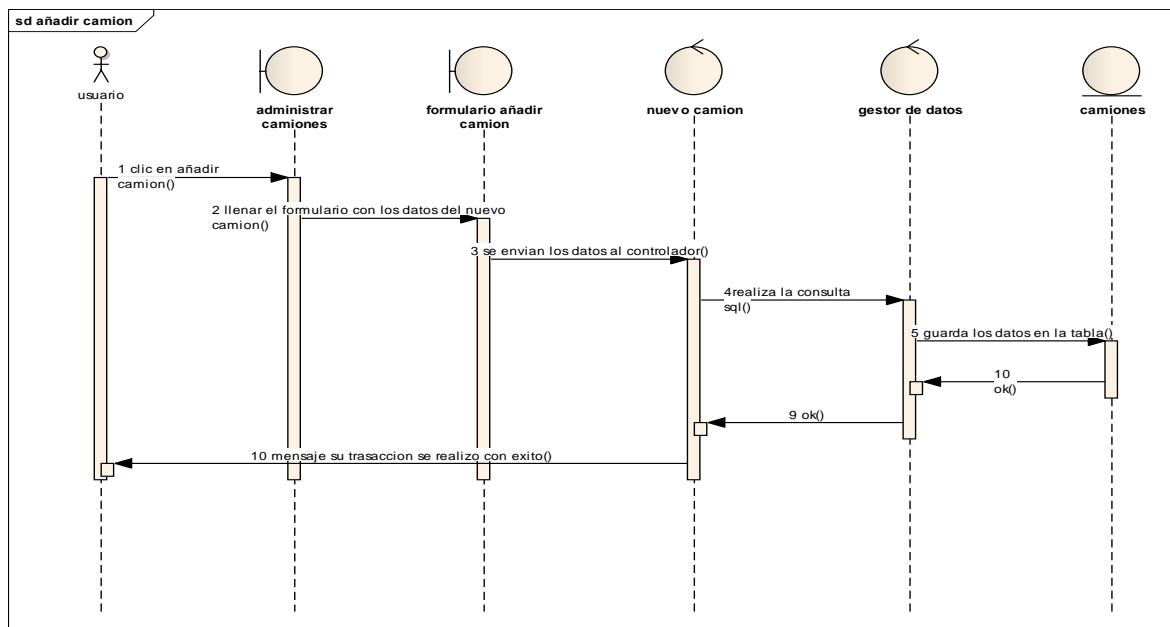


Figura 72 Añadir camión

### 1.5.1.7.9.4.38 Modificar camión

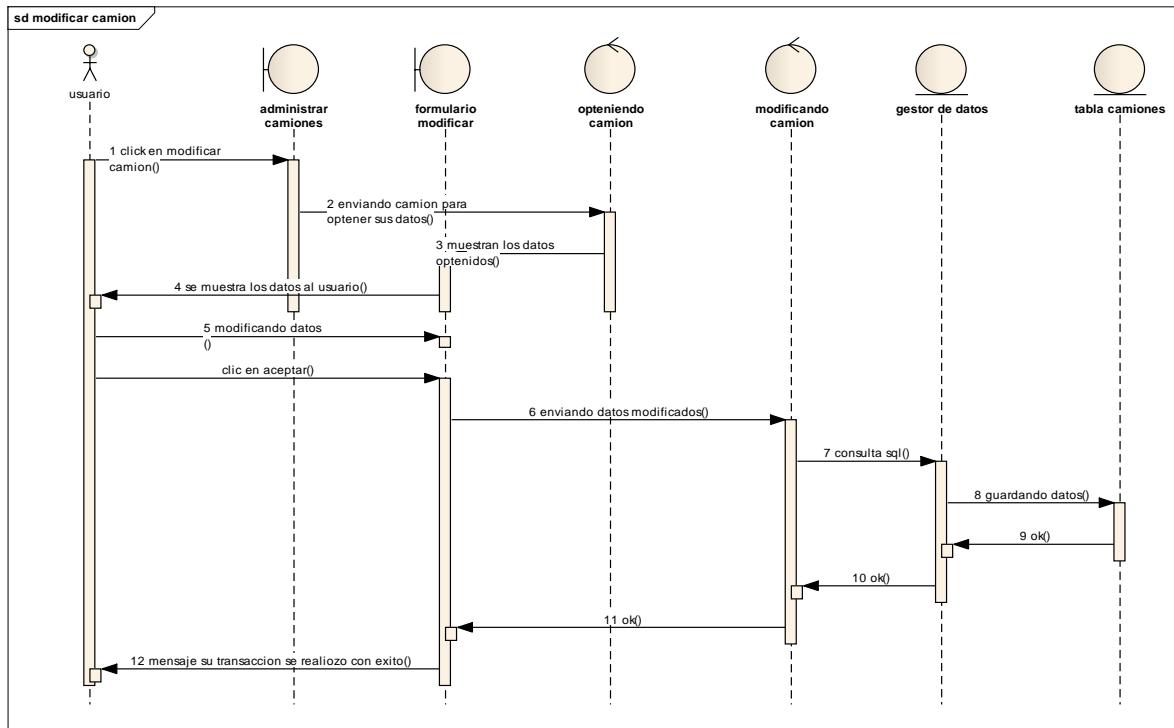


Figura 73 Modificar camión

### 1.5.1.7.9.4.39 Eliminar camión

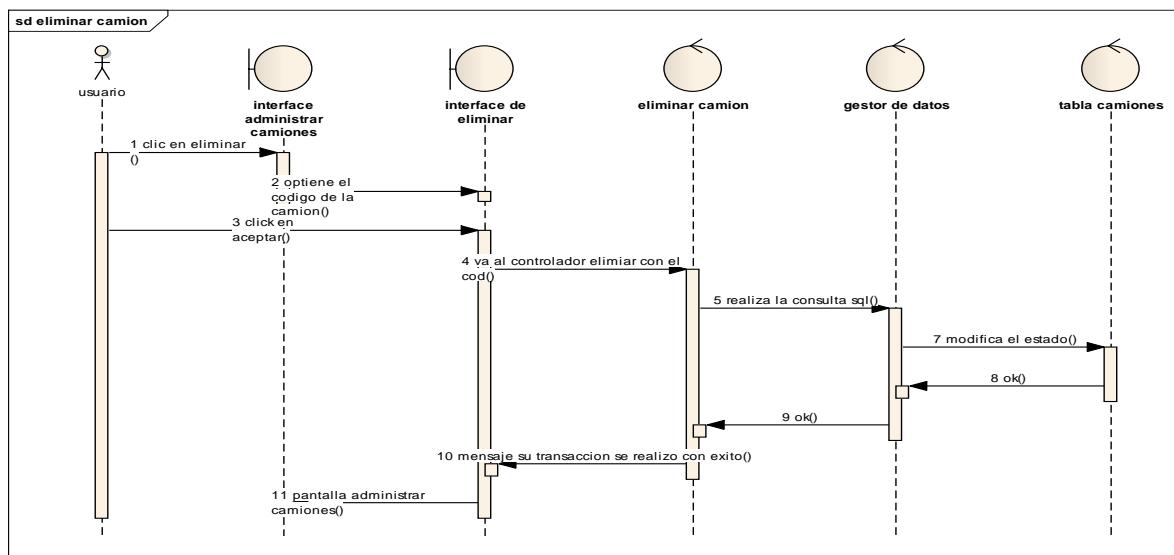


Figura 74 Eliminar camión

### 1.5.1.7.9.4.40 Camiones inactivos

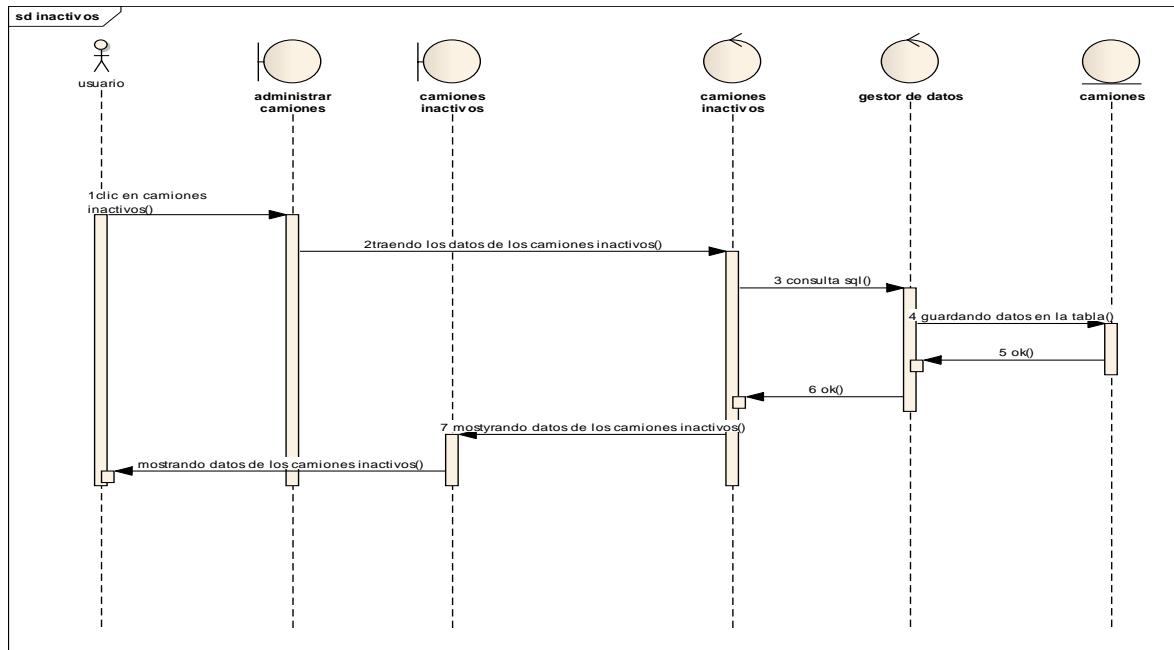


Figura 75 Camiones inactivos

### 1.5.1.7.9.4.41 Habilitar camión

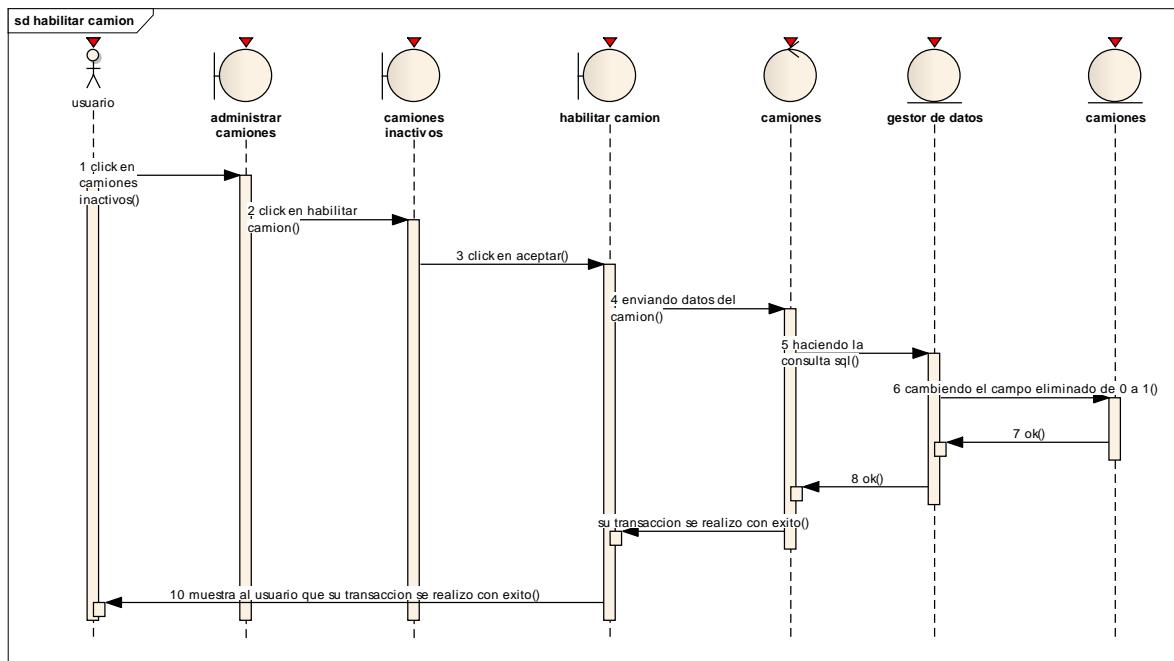


Figura 76 Habilitar camión

### 1.5.1.7.9.4.42 Reporte

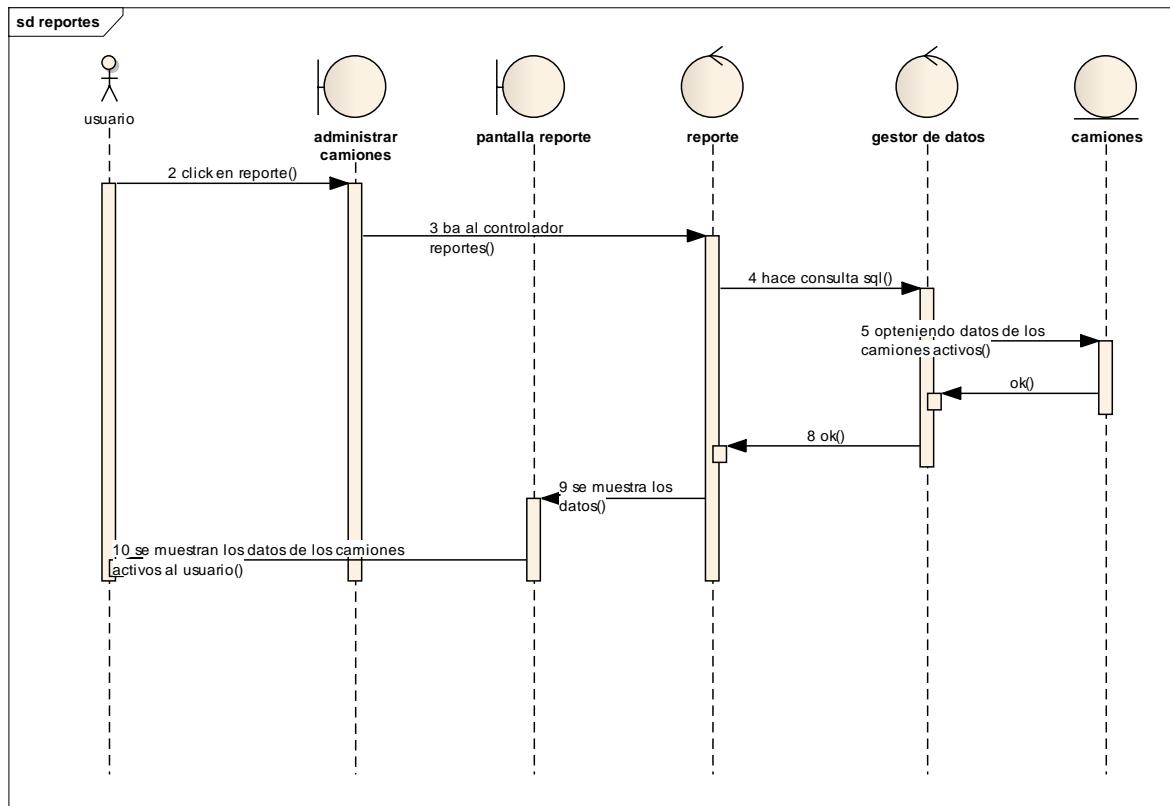


Figura 77 Reporte

### 1.5.1.7.9.4.43 Asignar camión a un socio

#### 1.5.1.7.9.4.43.1 Añadir asignación

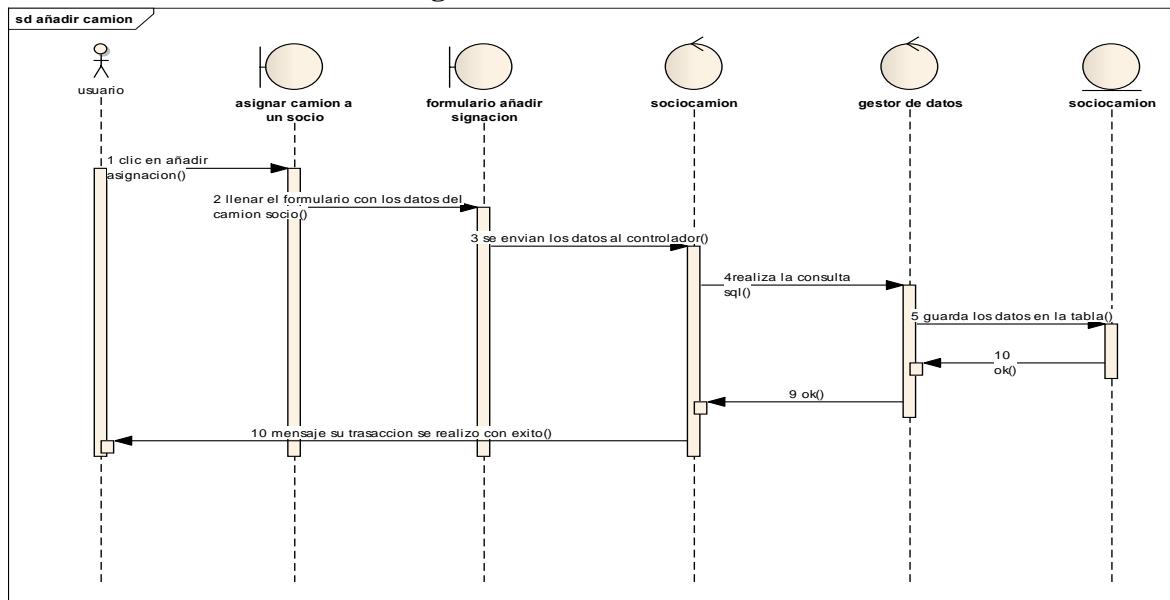


Figura 78 Añadir asignación

### 1.5.1.7.9.4.43.2 Eliminar asignación

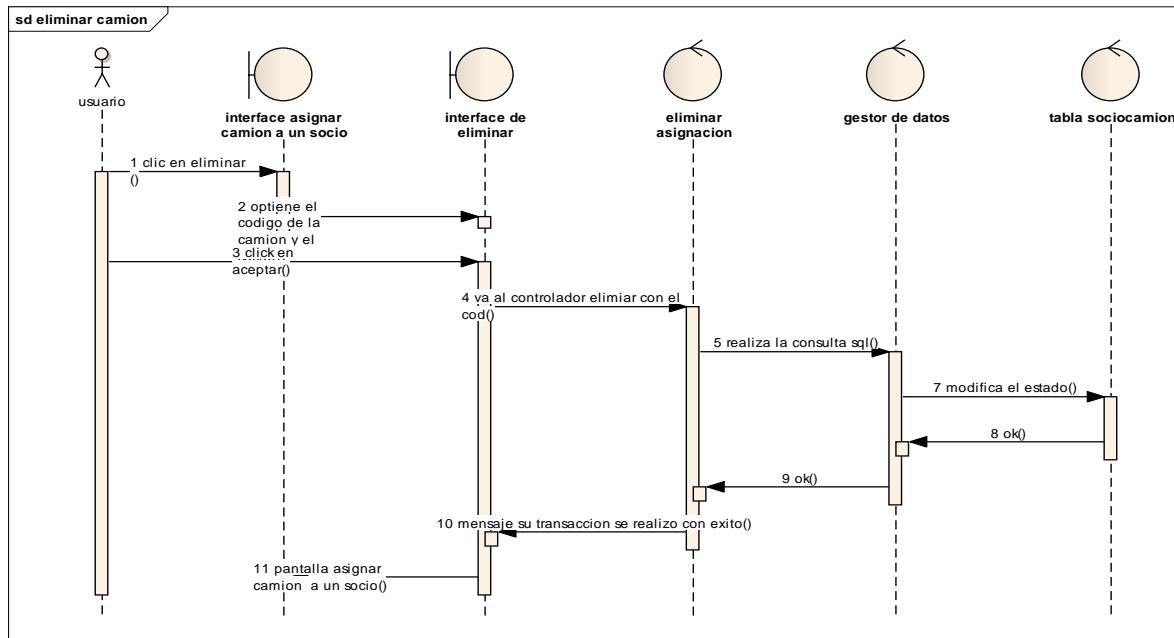


Figura 79 Eliminar asignación

### 1.5.1.7.9.4.44 Asignar camión a un chofer

### 1.5.1.7.9.4.44.1 Añadir asignación

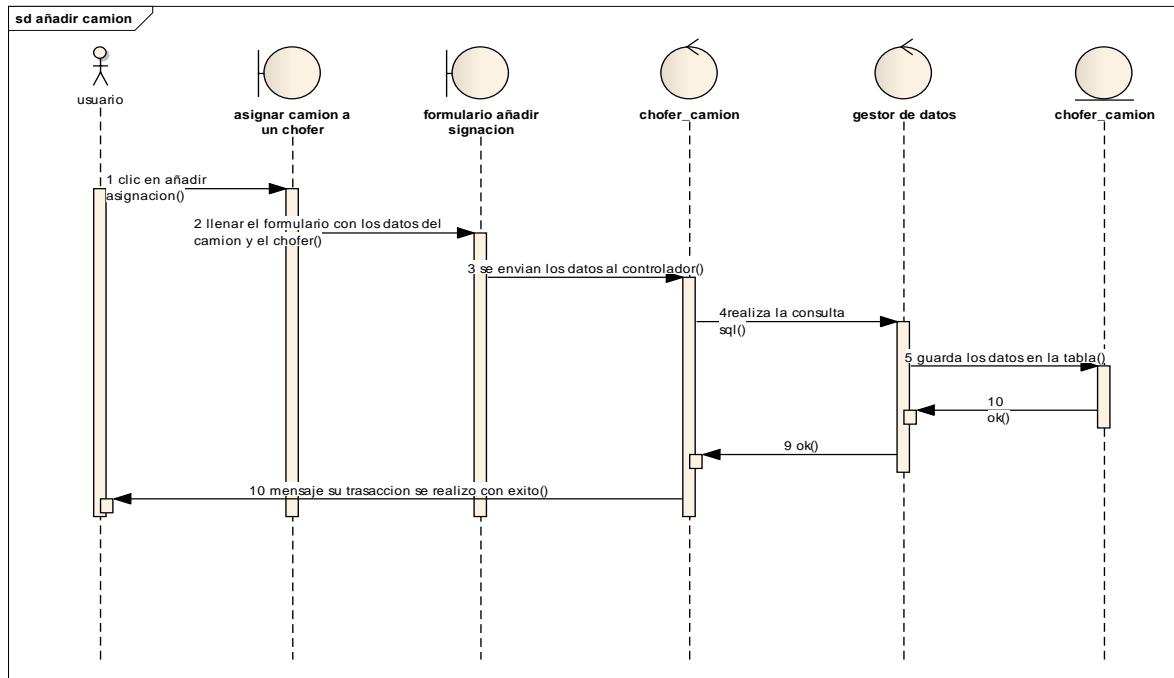


Figura 80 Añadir asignación

### 1.5.1.7.9.4.44.2 Eliminar asignación

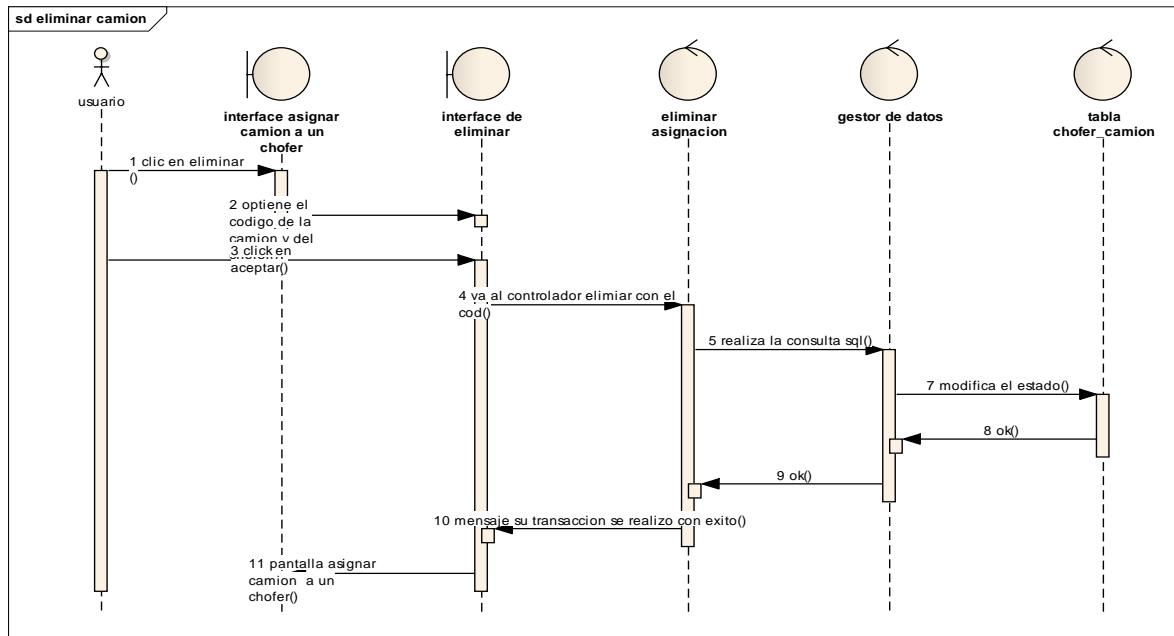


Figura 81 Eliminar asignación

### 1.5.1.7.9.4.45 Añadir carga

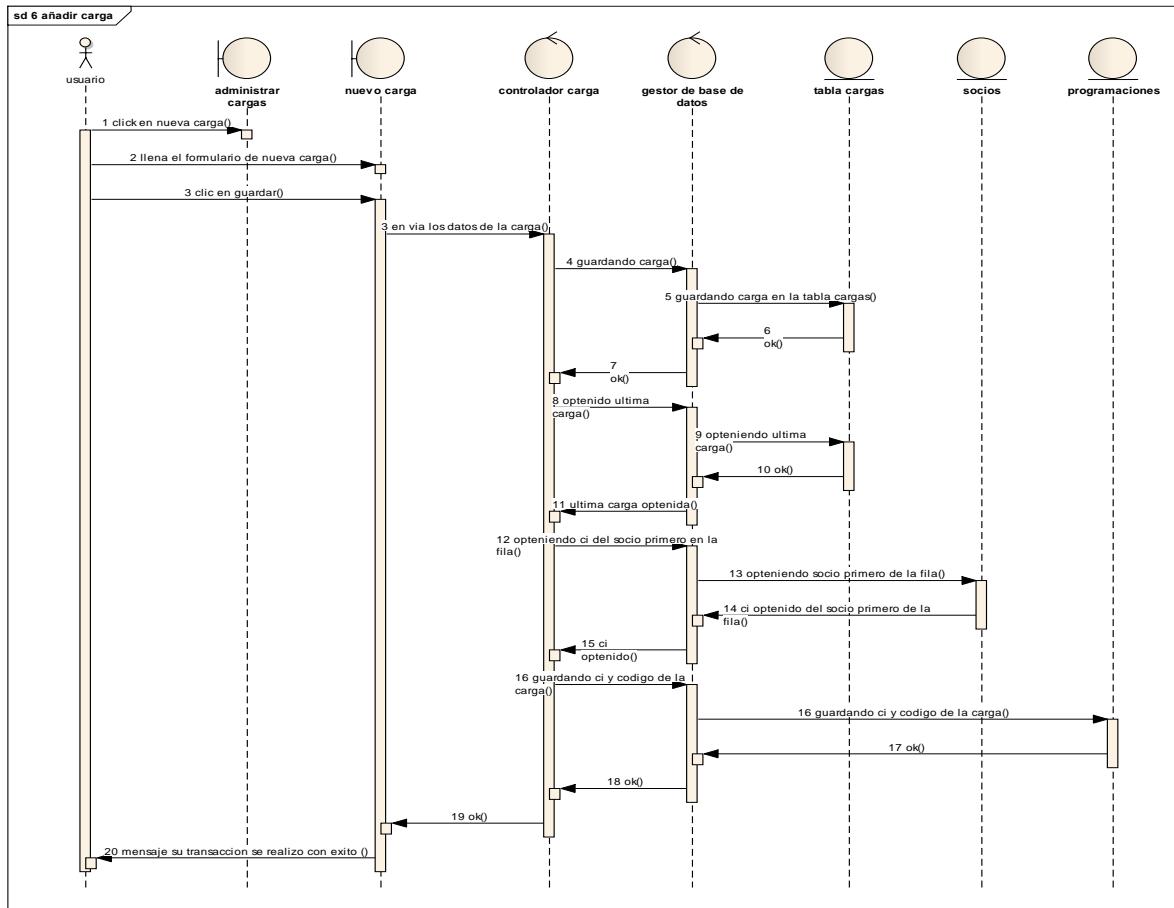


Figura 82 Añadir carga

### 1.5.1.7.9.4.46 Añadir tipo de carga

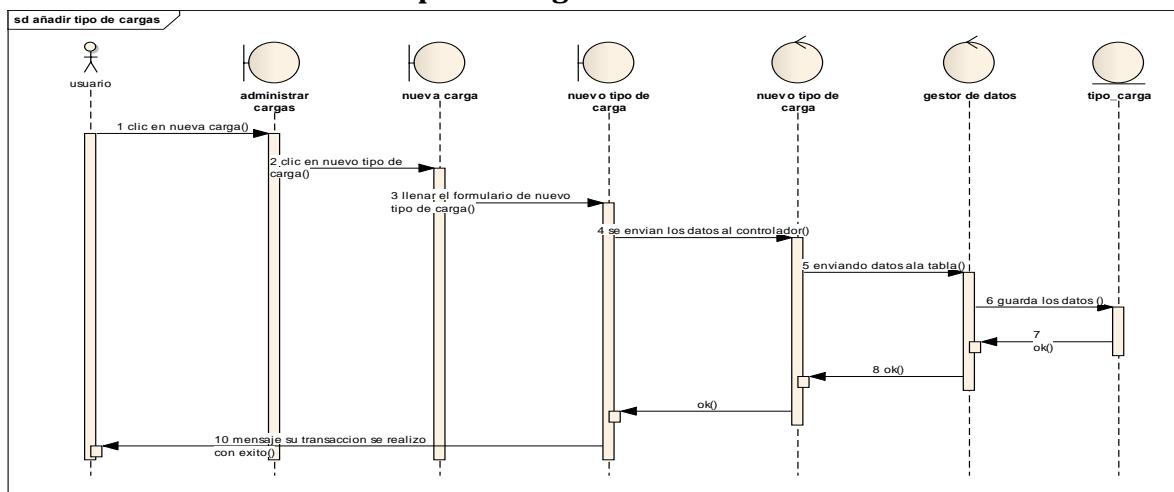


Figura 83 Añadir tipo de carga

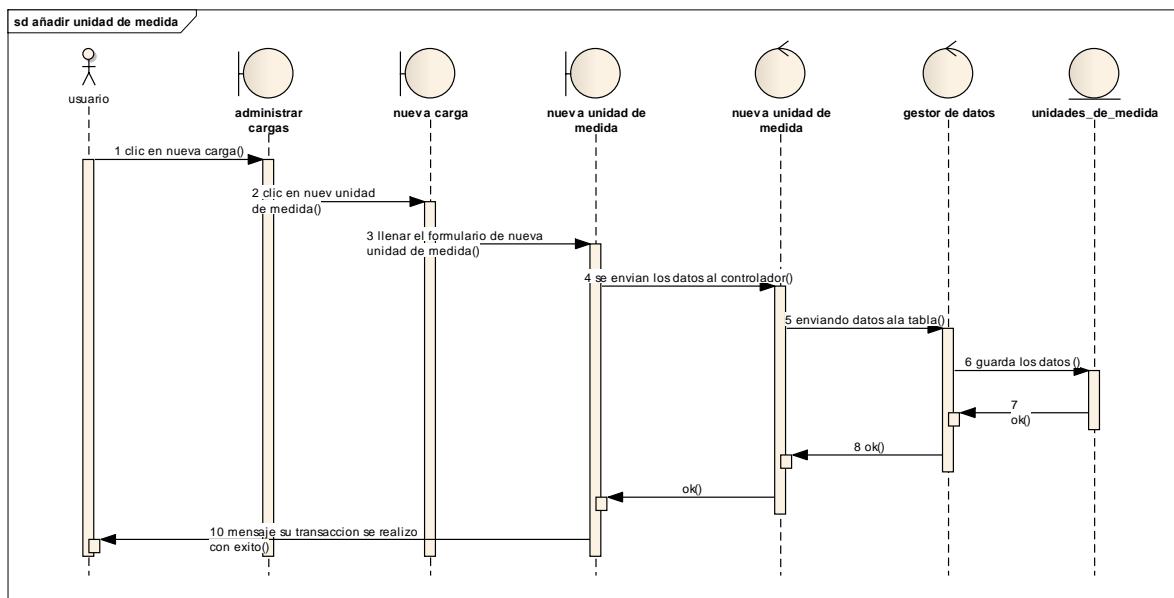
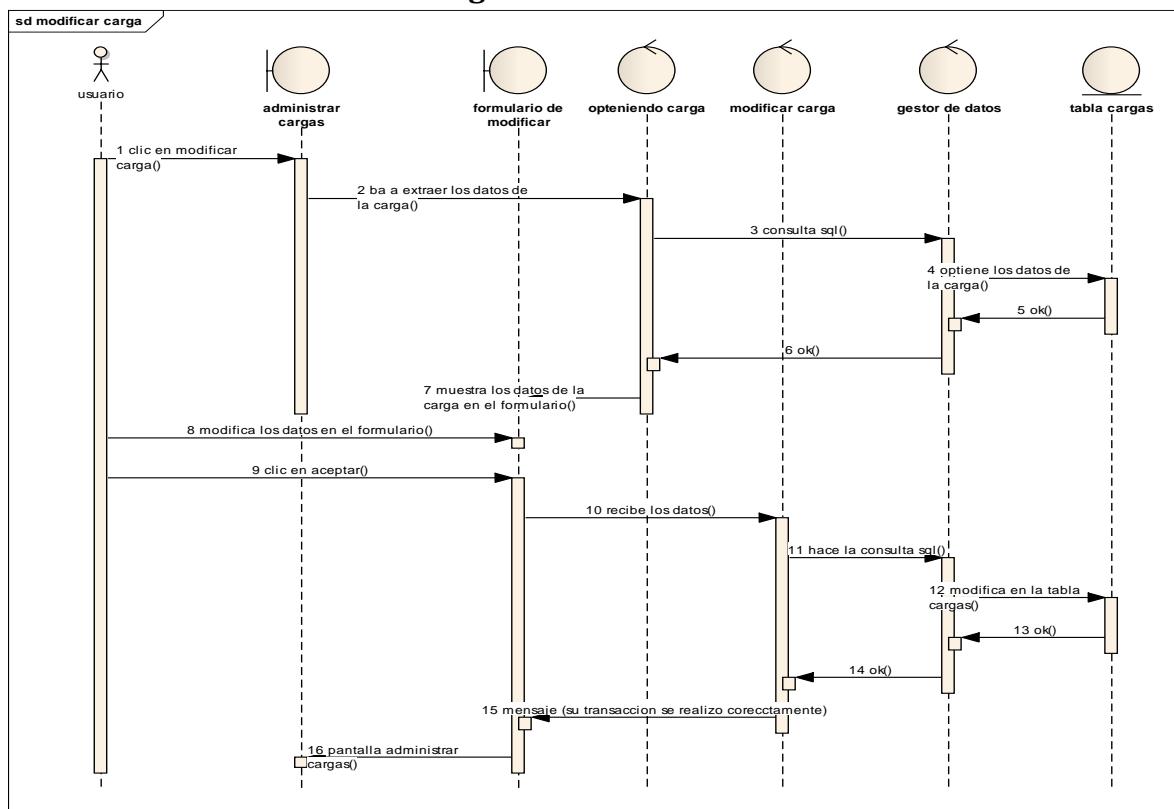
**1.5.1.7.9.4.47****Añadir unidad de medida****1.5.1.7.9.4.48****Modificar caga**

Figura 84 Modificar caga

### 1.5.1.7.9.4.49 Ver cargas

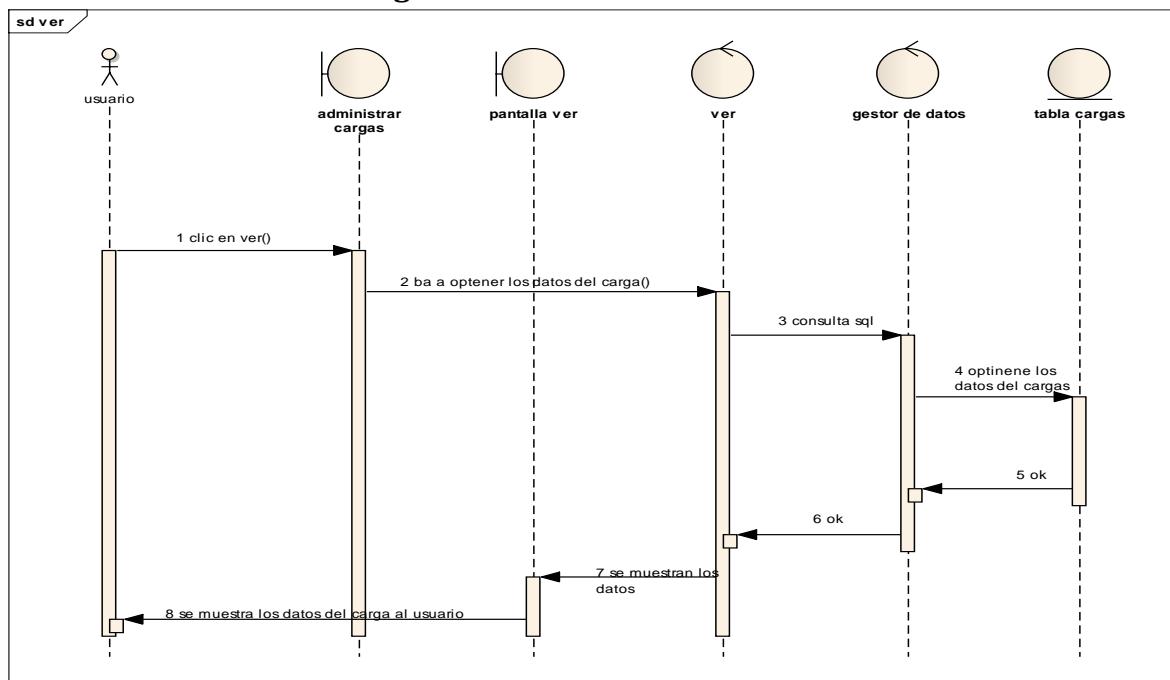


Figura 85 Ver cargas

### 1.5.1.7.9.4.50 Reporte cargas

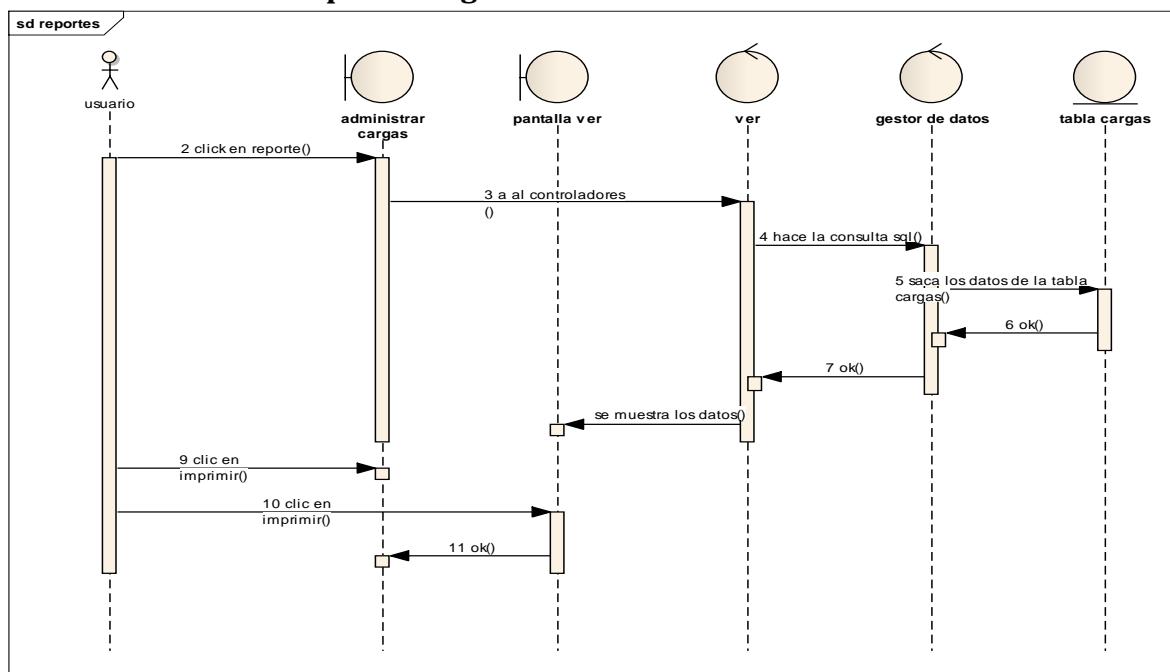


Figura 86 Reporte cargas

### 1.5.1.7.9.4.51 Rol de espera

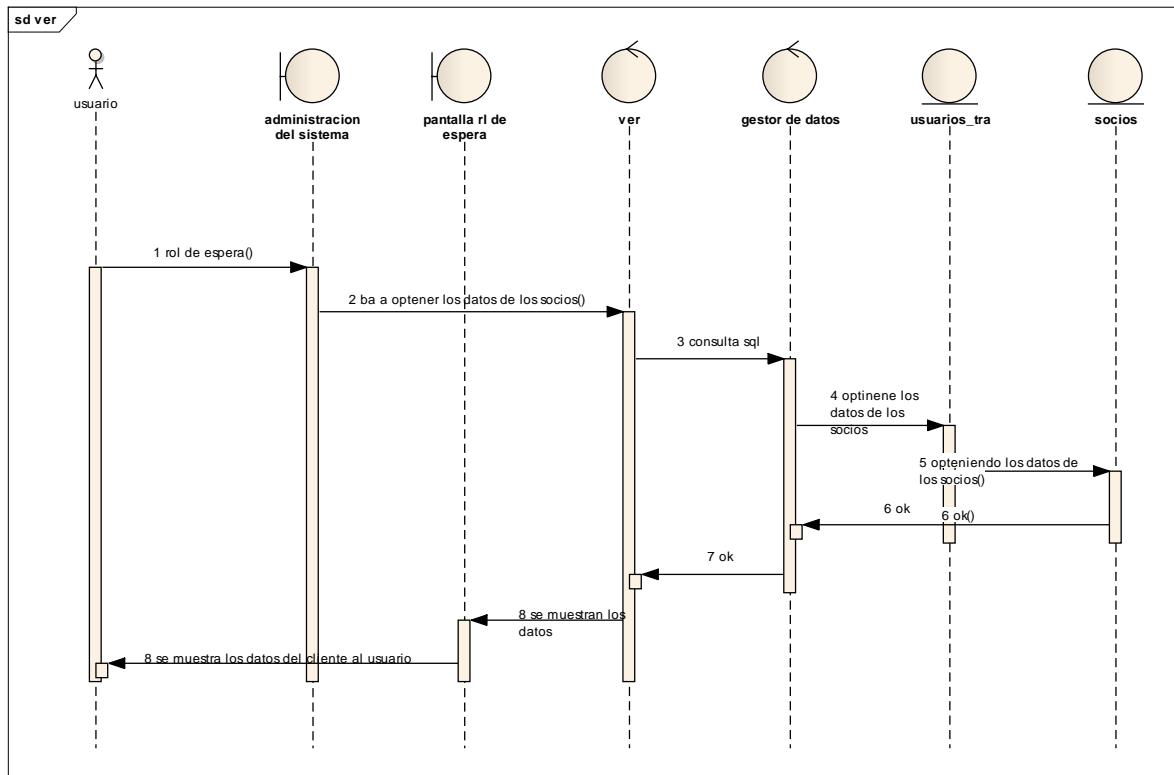


Figura 87 Rol de espera

### 1.5.1.7.9.4.52 Cambio de estado

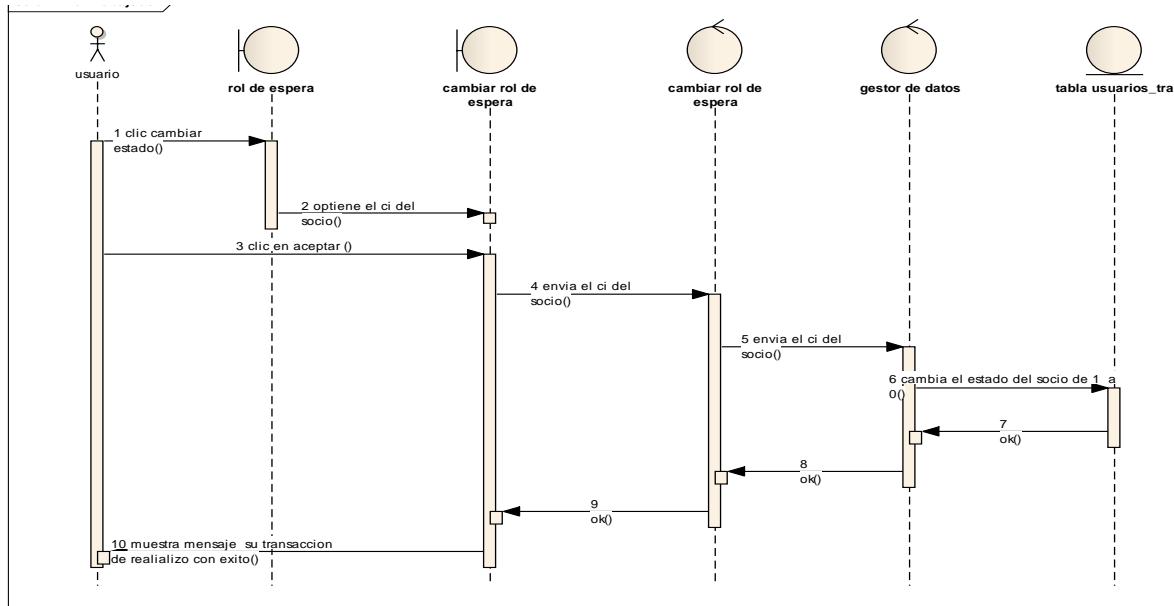


Figura 88 Cambio de estado

### 1.5.1.7.9.4.53 Programación

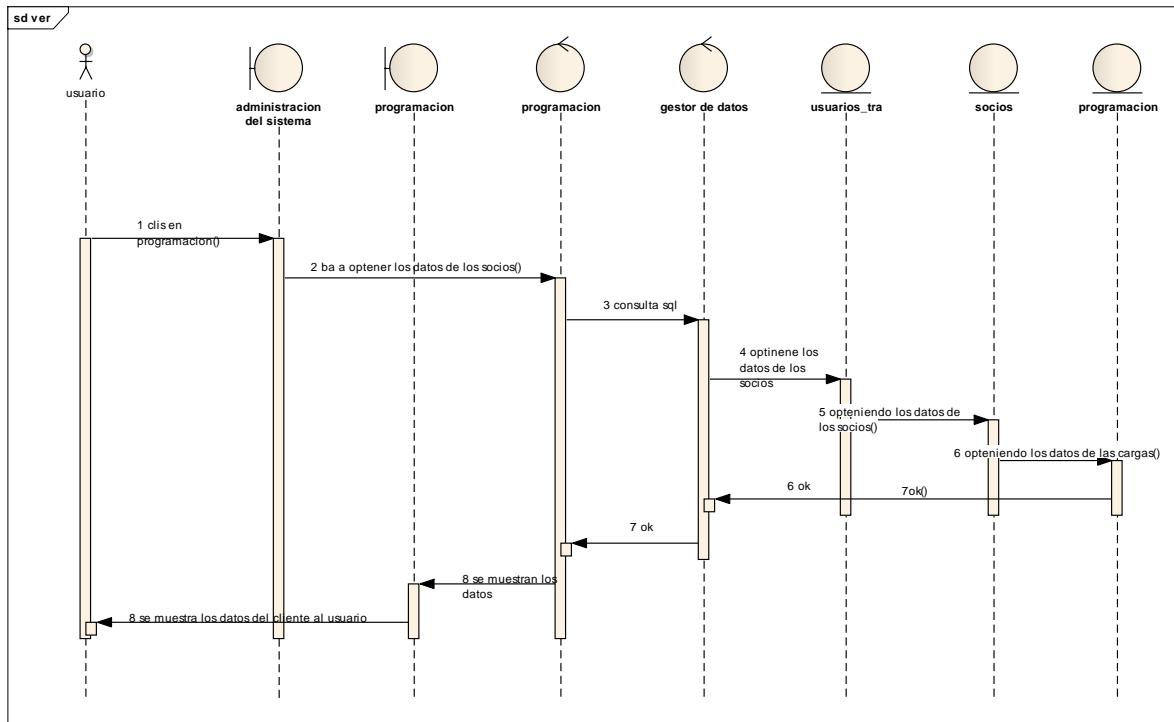


Figura 89 Programación

### 1.5.1.7.9.4.54 Ver programación

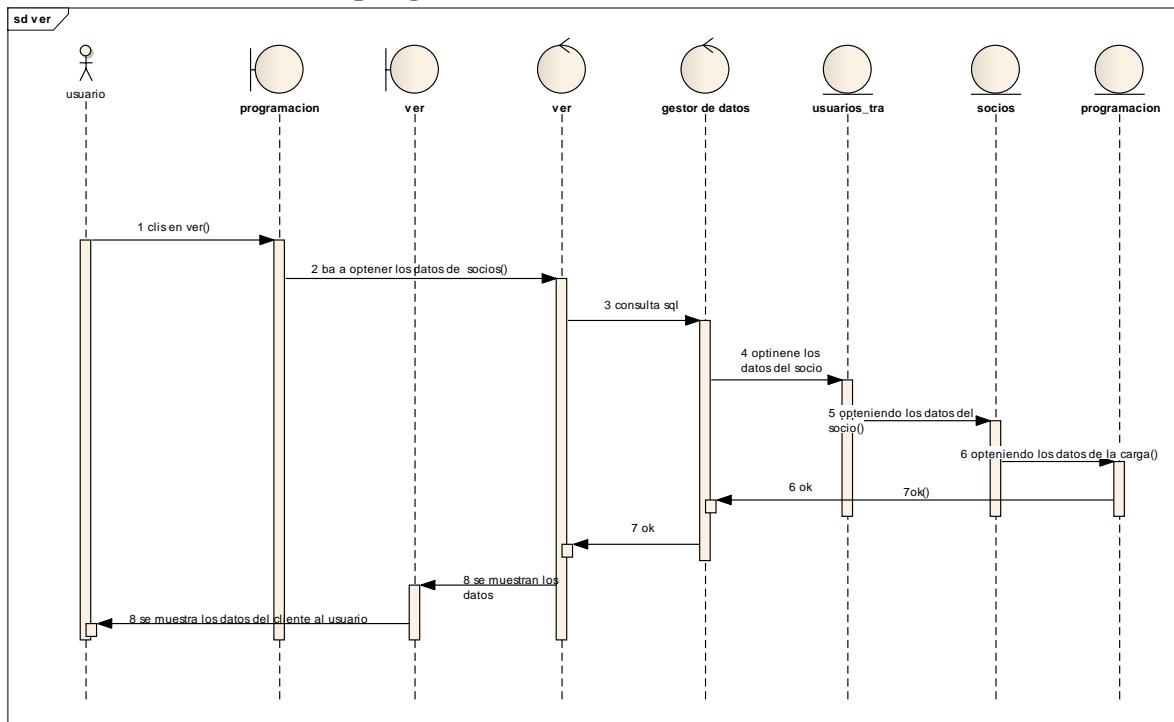


Figura 90 Ver programación

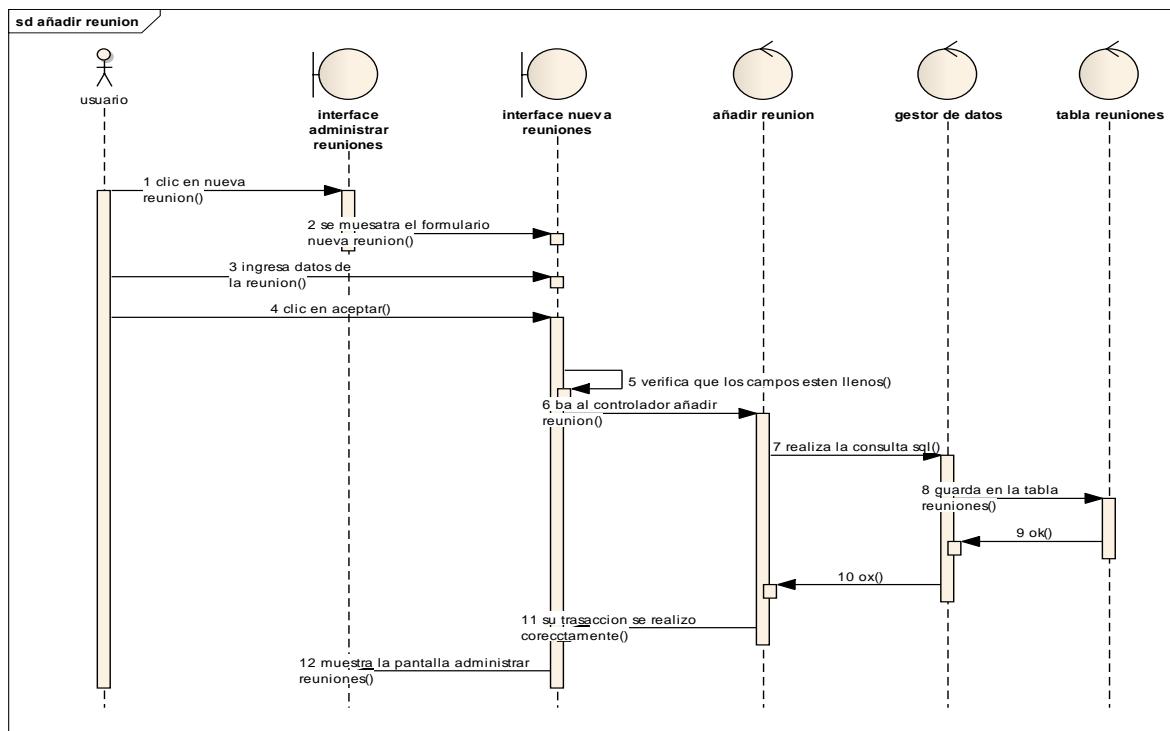
**1.5.1.7.9.4.55 Añadir reuniones**

Figura 91 Añadir reuniones

### 1.5.1.7.9.4.56 Modificar reuniones

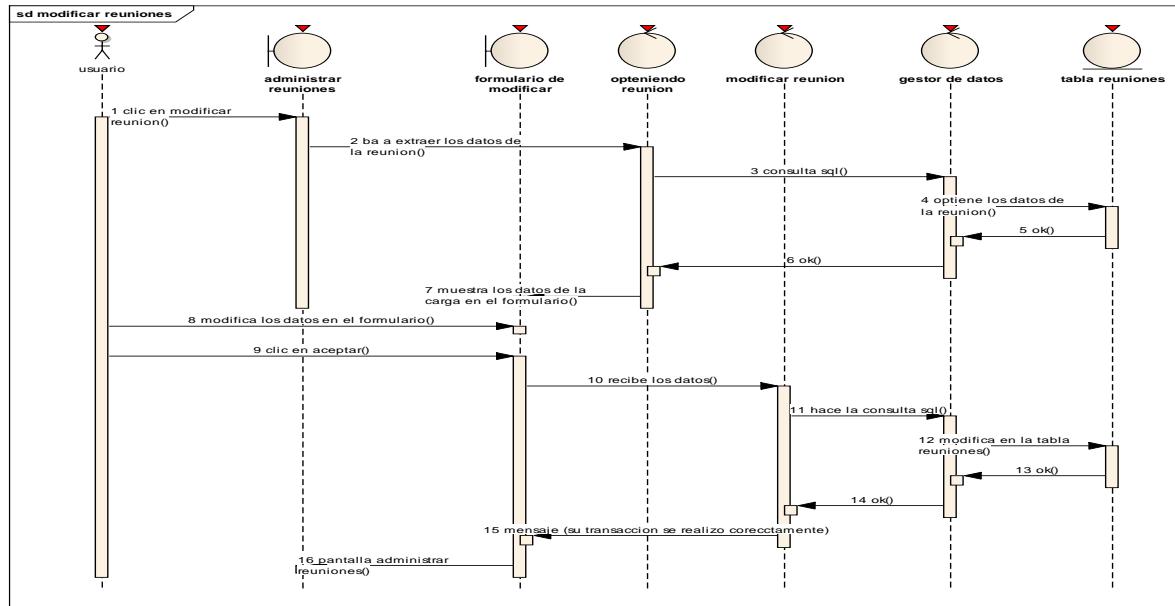


Figura 92 Modificar reuniones

### 1.5.1.7.9.4.57 Eliminar reunión

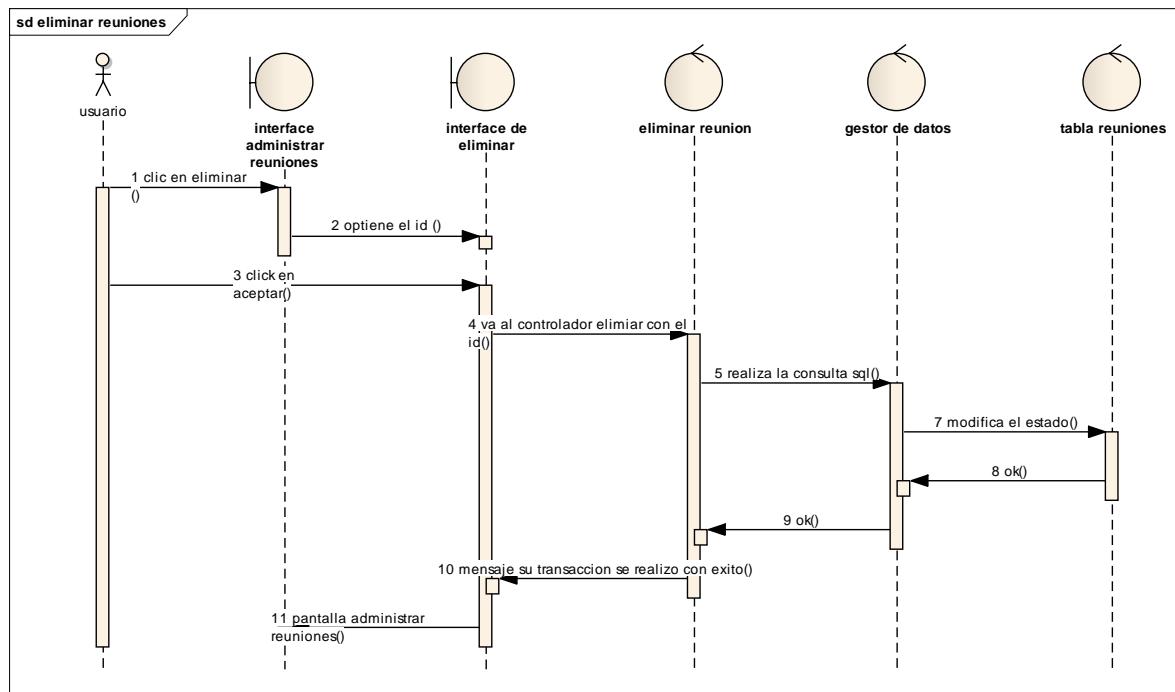


Figura 93 Eliminar reunión

### 1.5.1.7.9.4.58 Ver reunión

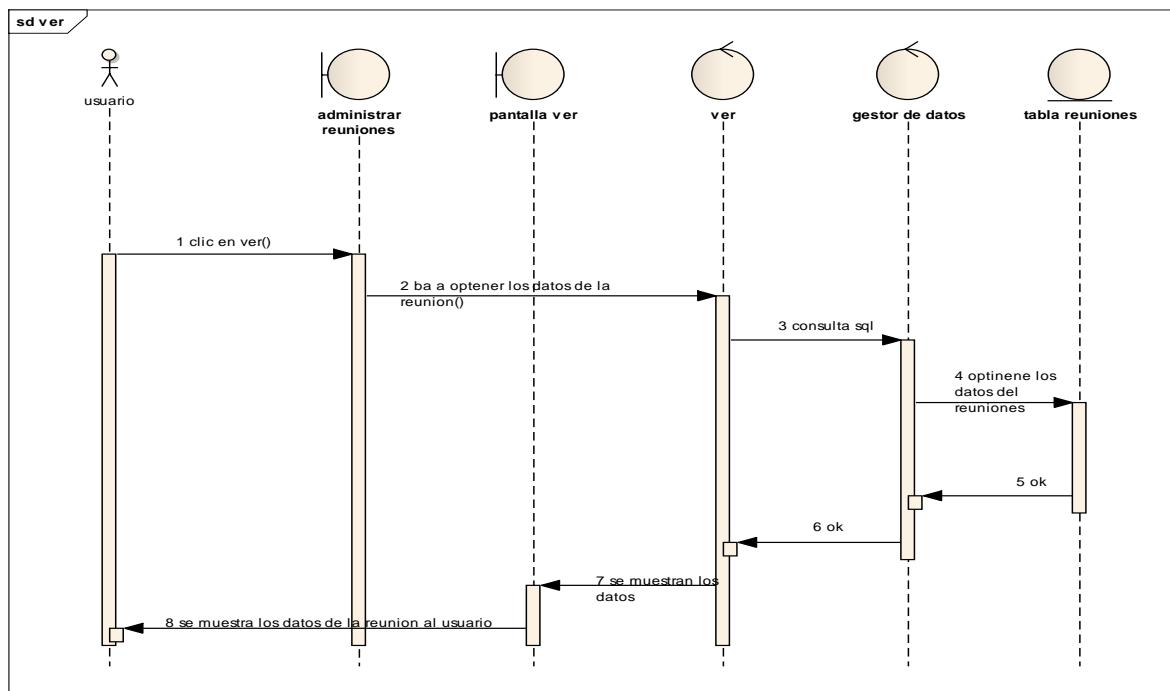


Figura 94 Ver reunión

### 1.5.1.7.9.4.59 Socios presentes

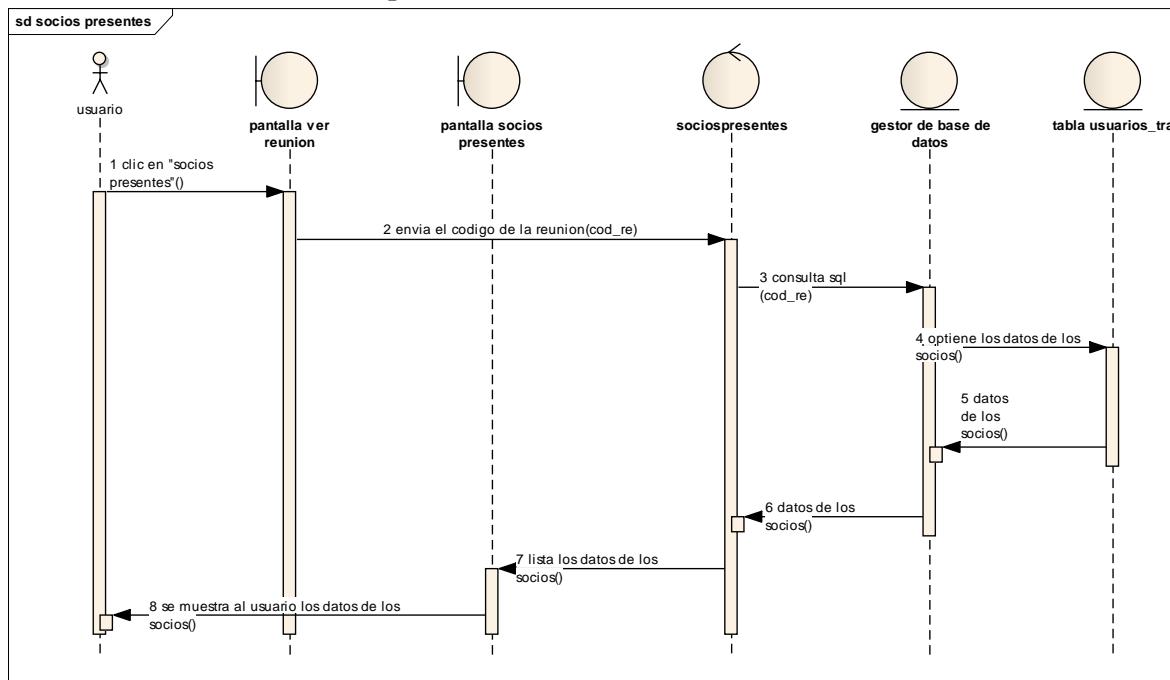


Figura 95 Socios presentes

### 1.5.1.7.9.4.60 Código qr

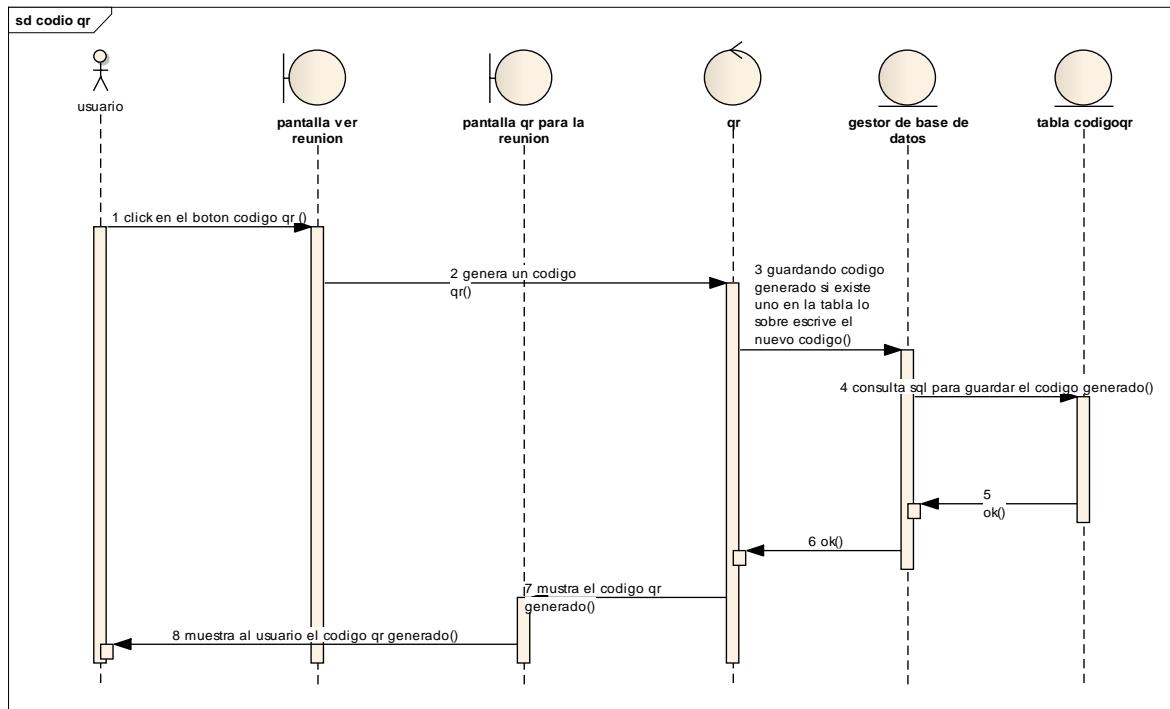


Figura 96 Código qr

### 1.5.1.7.9.4.61 Socios con falta

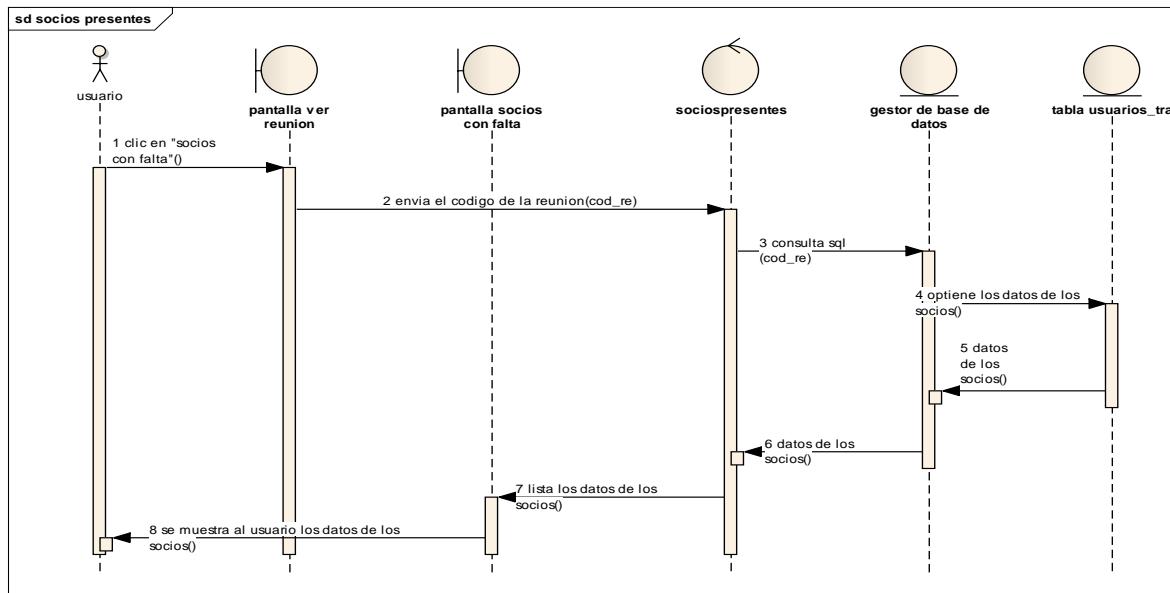


Figura 97 Socios con falta

### 1.5.1.7.9.4.62 Reporte reuniones

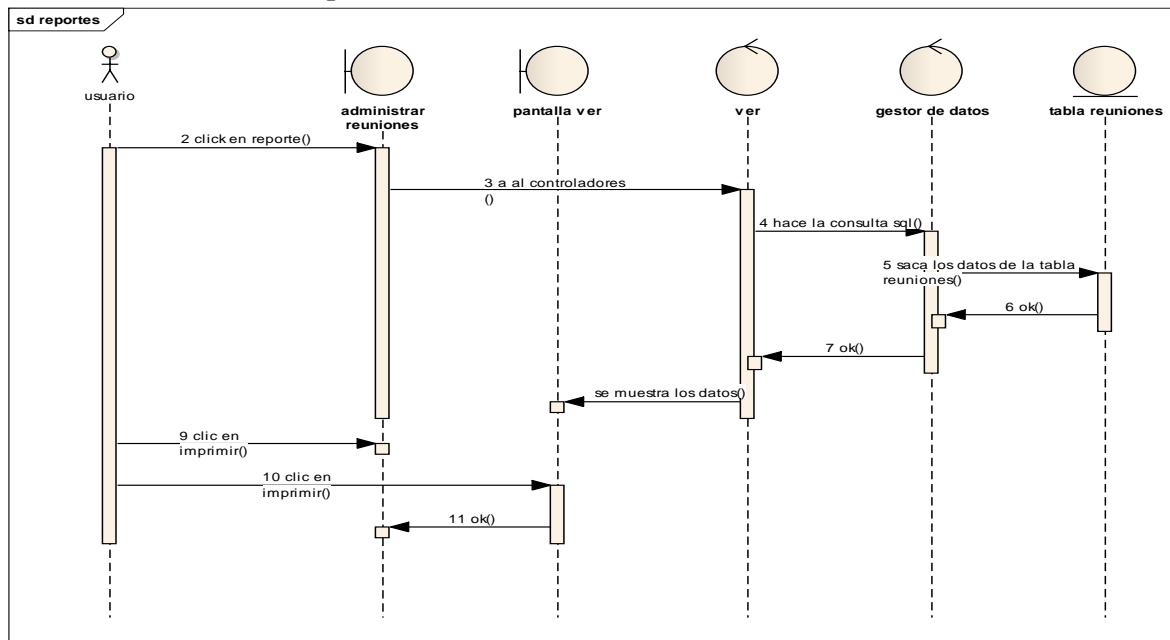


Figura 98 Reporte reuniones

### 1.5.1.7.10 Diagrama de despliegue

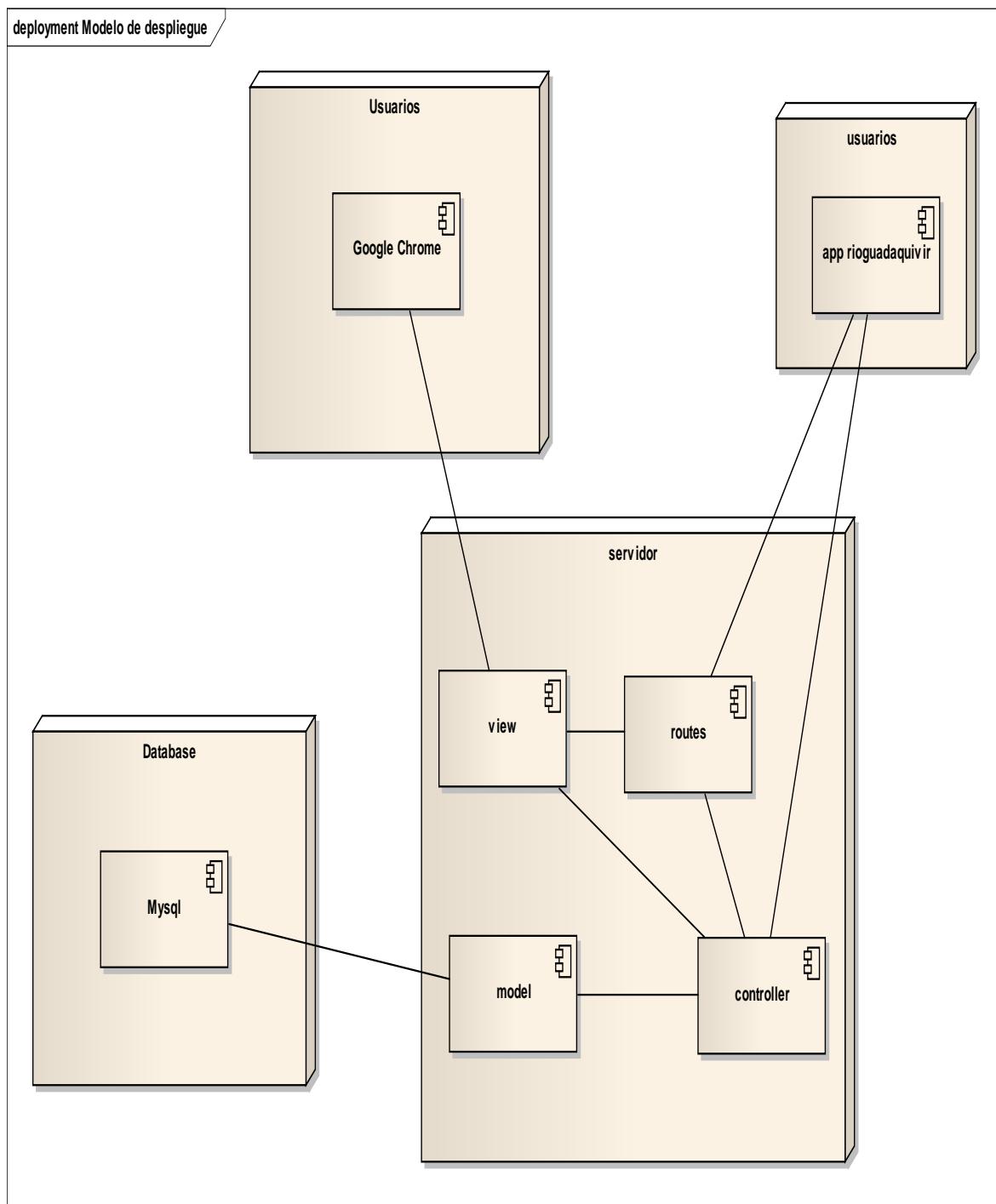


Figura 99 Diagrama de despliegue

### **1.5.1.7.11 Prototipo de interfaz del usuario**

#### **1.5.1.7.11.1 Introducción**

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de la interfaces que proveerá el sistema y así conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizan como: dibujos a mano en papel dibujos con alguna herramienta grafica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Solo los de ese último tipo serán entregados al final de la fase de elaboración, los otros serán desechados en la fase de construcción en la medida que el resultado de la iteraciones vaya desarrollando el producto final

#### **1.5.1.7.11.2 Propósito**

- Comprender la idea de cómo será el sistema más adelante
- Identificar posibles mejoras

#### **1.5.1.7.11.3 Alcance**

Mostrar los prototipos de pantallas, solamente el diseño que adoptaran

#### **1.5.1.7.11.4 Diseño de pantallas**

##### **1.5.1.7.11.4.1 Acceso al sistema**

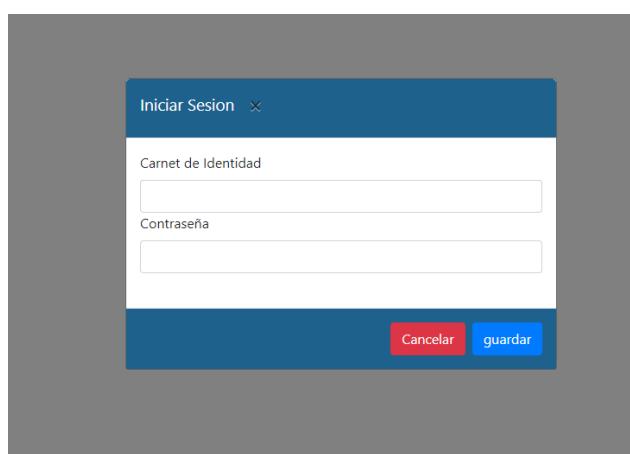


Figura 100 Acceso al sistema

### 1.5.1.7.11.4.2 Pantalla de inicio



Figura 101 Pantalla de inicio

### 1.5.1.7.11.4.3 Administrar socios

Imagen	Carnet de identidad	Socio	Telefono	Camion	Puede Recibir Carga
	7509740	Luis Miguel Veliz Cardozo	68708797	Con Camion	Puede Recibir Carga

Figura 102 Administrar socios

### 1.5.1.7.11.4.4 Nuevo socio

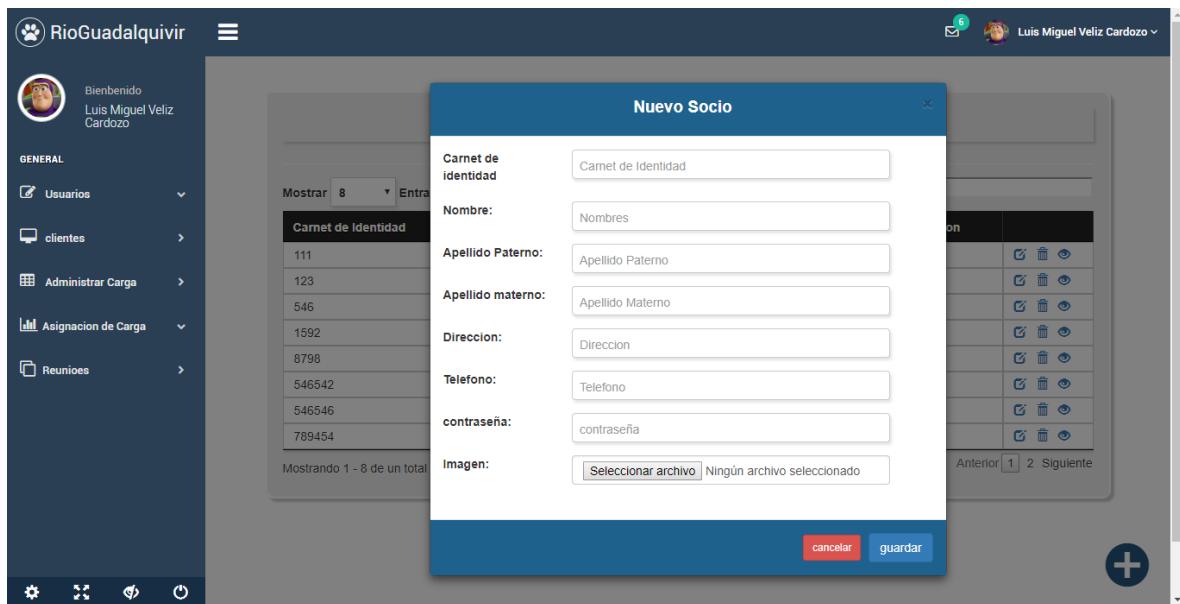


Figura 103 Nuevo socio

### 1.5.1.7.11.4.5 Modificar socio

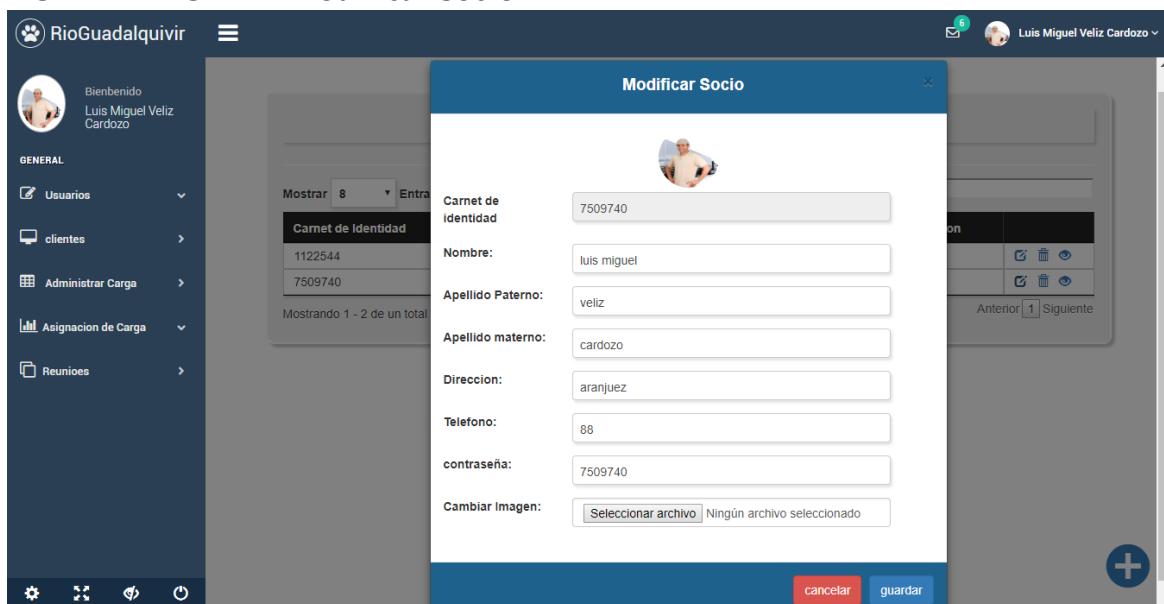


Figura 104 Modificar socio

### 1.5.1.7.11.4.6 Eliminar socio

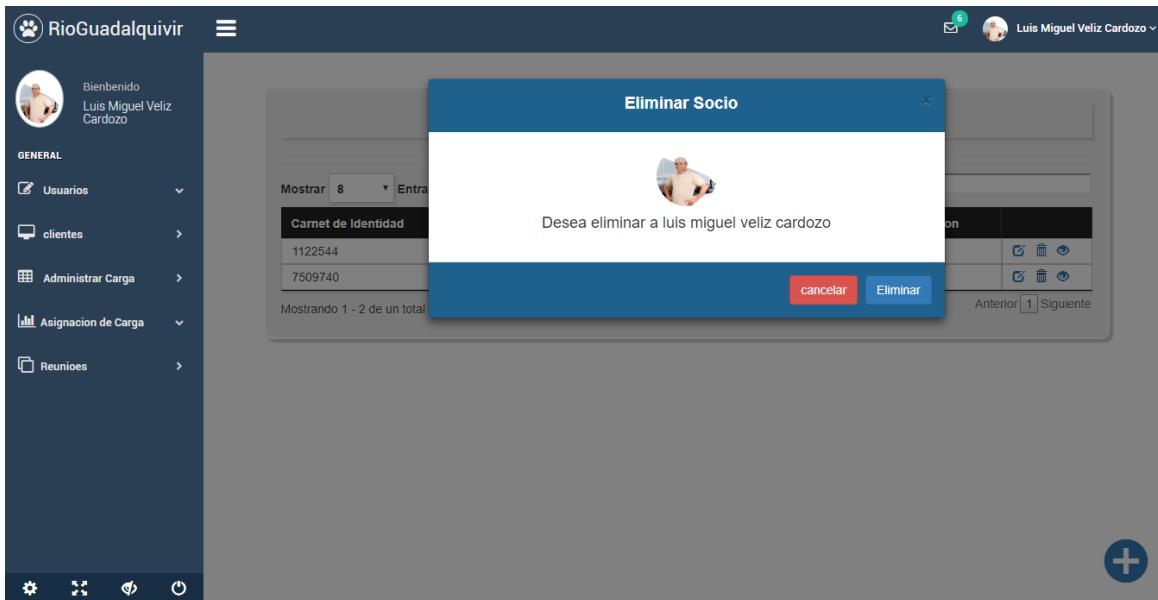


Figura 105 Eliminar socio

### 1.5.1.7.11.4.7 Ver socio

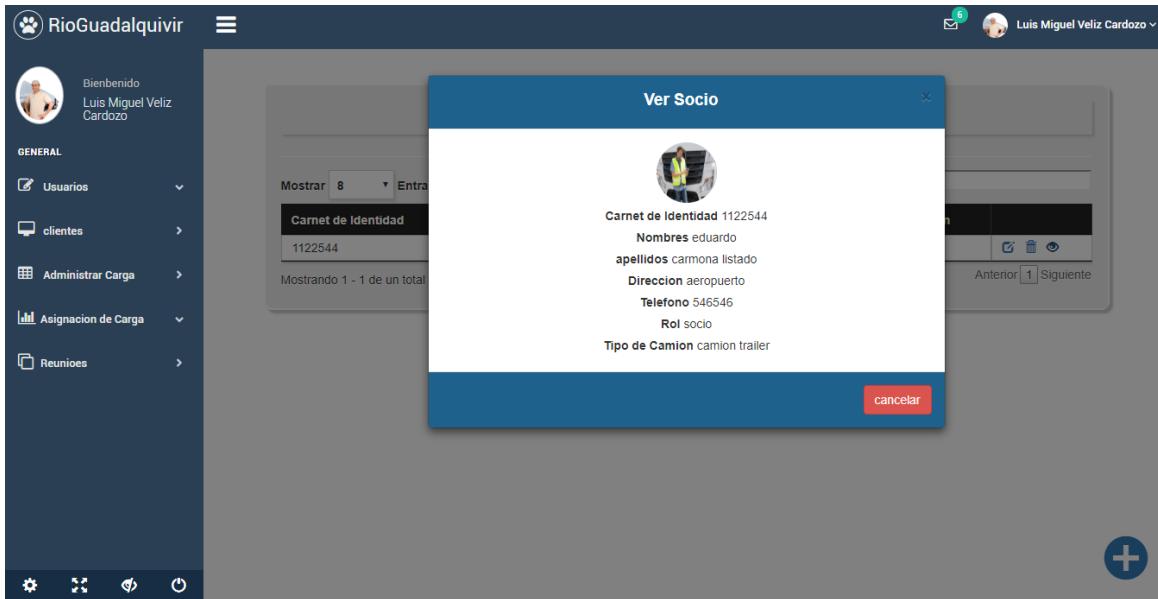


Figura 106 Ver socio

### 1.5.1.7.11.4.8 Reporte socios

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno	Tipo de Camion
111	luis miguel	veliz	cardozo	camion trailer
123	luis	rodriguez	peralta	camion trailer
546	hermeline	cardozo	zaraso	camion trailer
1592	carlo	carlo	carlo	camion trailer
8798	carlos	benites	alcoba	camion trailer
546542	hector	camacho	canciel	camion trailer
546546	gielermo	senteno	retamozo	camion trailer
789454	eduardo	rivera	condori	camion trailer

Figura 107 Reporte socios

### 1.5.1.7.11.4.9 Administrar trabajadores

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno	
11	kjhñlk	ñlkjhñ	lkj	
111	luis miguel	veliz	cardozo	
156	luis	cardozo	miranda	
789	hgoj	kjhñk	hikjh	
7987	jhnkj	lkjh	lkjh	
8779	gkjhg	jngk	jngk	
88888888	nose	jngl	kjgh	

Mostrando 1 - 7 de un total de 7 entradas

Anterior 1 Siguiente

Figura 108 Administrar trabajadores

### 1.5.1.7.11.4.10 Nuevo trabajador

**Nuevo Trabajador**

Carnet de Identidad: Carnet de Identidad

Nombre: Nombres

Apellido Paterno: Apellido Paterno

Apellido materno: Apellido Materno

Direccion: Direccion

Telefono: Telefono

contraseña: contraseña

Imagen: Seleccionar archivo Ningún archivo seleccionado

cancelar guardar

Figura 109 Nuevo trabajador

### 1.5.1.7.11.4.11 Modificar trabajador

**Modificar Trabajador**

Carnet de Identidad: 5789452

Nombre: carla

Apellido Paterno: orosco

Apellido materno: peñalosa

Direccion: barrio las pascuas

Telefono: 46576879

contraseña: carla

Cambiar Imagen: Seleccionar archivo Ningún archivo seleccionado

cancelar guardar

Figura 110 Modificar trabajador

### 1.5.1.7.11.4.12 Eliminar trabajador

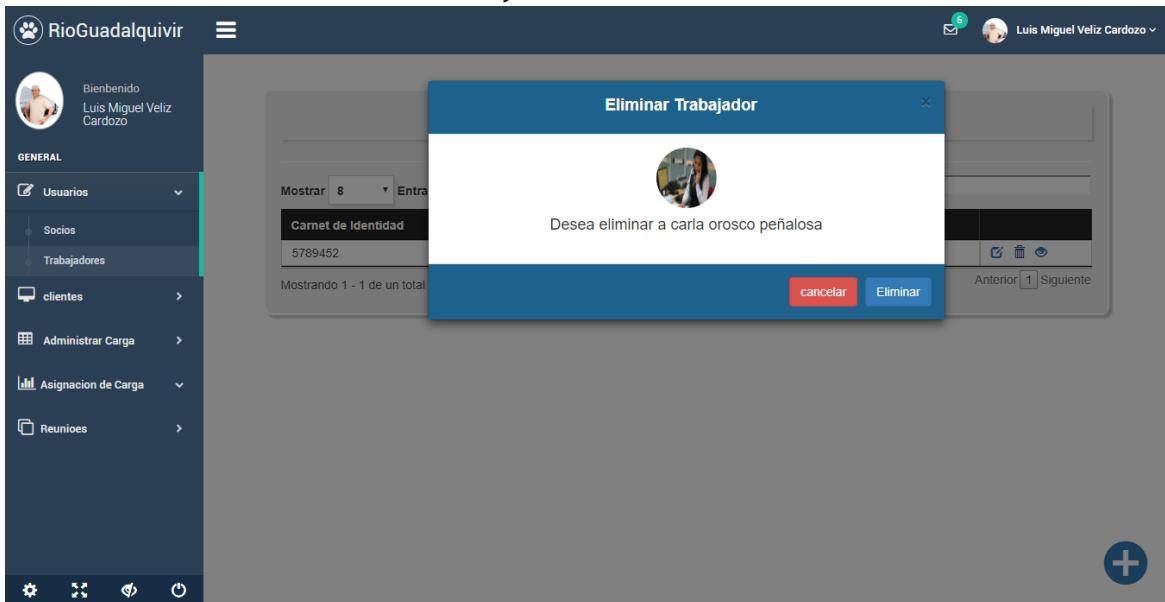


Figura 111 Eliminar trabajador

### 1.5.1.7.11.4.13 Ver trabajador

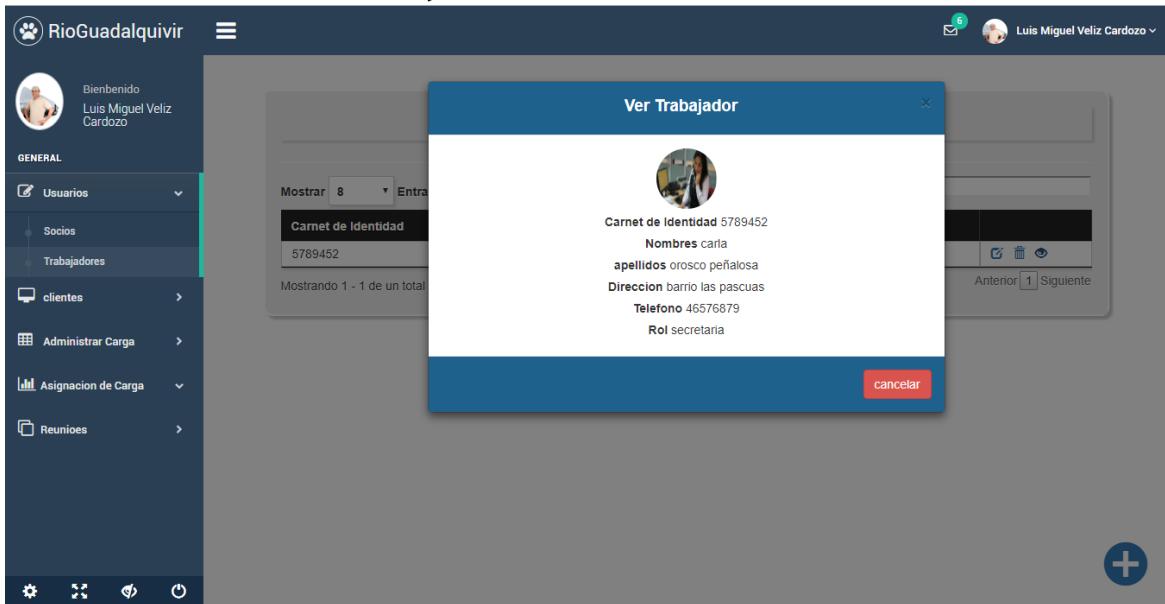


Figura 112 Ver trabajador

### 1.5.1.7.11.4.14 Reporte trabajadores

Rio Guadalquivir - Administrar Trabajadores

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno
11	kjhñlk	ñlkjhñ	lkj
111	luis miguel	veliz	cardozo
156	luis	cardozo	miranda
789	hgoj	kjhkj	hlkjh
7987	jhkij	lkjh	lkjh
8779	gkjhg	jhgk	jhgk
88888888	nose	jhgj	kjghk

Figura 113 Reporte trabajadores

### 1.5.1.7.11.4.15 Administrar clientes

Administrador Clientes

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno	De la Empresa
548	emilio	palala	cangrajas	metales oriente
1565	carmelo	aguirre	miranda	cemento el puente
2487	luis	cespedes	gramoneda	el arriero
5246	carlos favian	armella	mamani	constructora la casa
5468	ramiro	ortis	molo	constructora los amigos
51468	carelos	romero	miranda	metales el oriente
54654	roberto	sulca	carcara	faboce
4407654	carlos arturo	farfan	callejas	constructora

Mostrando 1 - 8 de un total de 10 entradas

Figura 114 Administrar clientes

### 1.5.1.7.11.4.16 Nuevo cliente

**Nuevo Cliente**

Carnet de identidad: Carnet de Identidad

Nombre: Nombres

Apellido Paterno: Apellido Paterno

Apellido materno: Apellido Materno

Direccion: Direccion

Telefono: Telefono

De la Empresa: De la Empresa

Imagen: Seleccionar archivo Ningún archivo seleccionado

cancelar guardar

Figura 115 Nuevo cliente

### 1.5.1.7.11.4.17 Modificar cliente

**Modificar Cliente**

Carnet de identidad: 4487654

Nombre: carlos arturo

Apellido Paterno: farfan

Apellido materno: callejas

Direccion: barrio petrolero

Telefono: 4254145

De la Empresa: constructora

Cambiar Imagen: Seleccionar archivo Ningún archivo seleccionado

cancelar guardar

Figura 116 Modificar cliente

### 1.5.1.7.11.4.18 Eliminar cliente

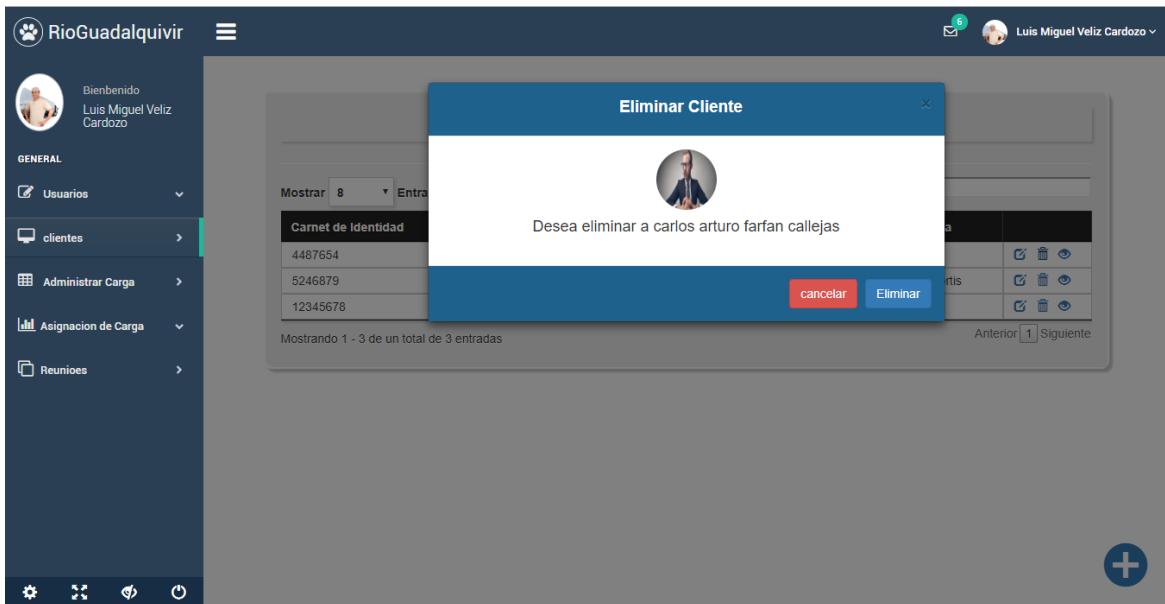


Figura 117 Eliminar cliente

### 1.5.1.7.11.4.19 Ver cliente

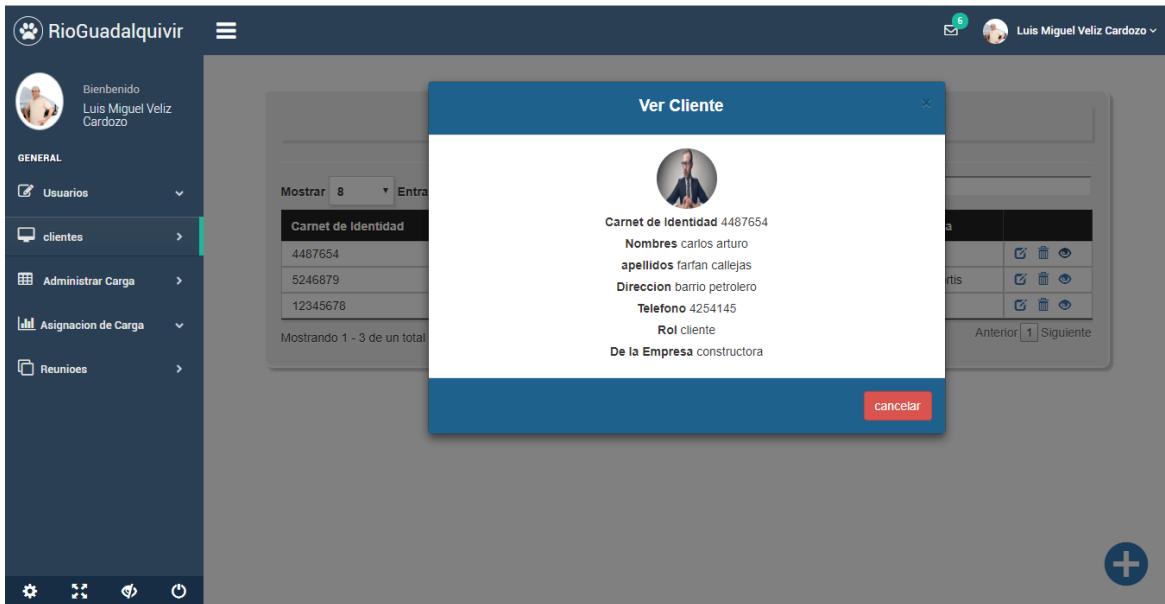


Figura 118 Ver cliente

### 1.5.1.7.11.4.20 Reporte clientes

Rio Guadalquivir - Administrar Socios				
Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno	De la Empresa
548	emilio	palala	cangrejas	metales oriente
1565	carmelo	aguire	miranda	cemento el puente
2487	luis	cespedes	gramoneda	el arriero
5246	carlos favian	armella	mamani	constructora la casa
5468	ramir	ortis	mollo	constructora los amigos
51468	carelos	romero	miranda	metales el oriente
4487654	carlos arturo	farfan	callejas	constructora
5246879	roberto	carmona	lizarazo	constructora ortis

Figura 119 Reporte clientes

### 1.5.1.7.11.4.21 Administrar empresa

The screenshot shows the RioGuadalquivir application's main menu on the left, with 'GENERAL' selected. Under 'GENERAL', 'Cuentas' is expanded, showing 'Usuarios', 'Clientes', 'Cuentas', and 'Choferes'. 'Choferes' is currently selected. On the right, the 'Administrar Empresas' page is displayed. It has a title 'Administrar Empresas' and a sub-section 'Empresas Activas'. A red button labeled 'Click para Empresas Inactivas' is visible. Below these are search and filter controls: 'Mostrar 10 Entradas' and 'Buscar:'. A table lists five active companies with columns for Nombre de la Empresa, Telefono, Direccion, and estado (Active). Each row includes edit and delete icons. At the bottom, it says 'Mostrando 1 - 5 de un total de 5 entradas'. A large blue '+' button is located at the bottom right.

Nombre de la Empresa	Telefono	Direccion	estado
aaaaa	4534553	<script >alert('hola')</script>	Activo
FABOCE	6666666	salida a bermejo kilometra 7	Activo
Inserpaz	6666666	santa anaaaa	Activo
Los Olivos	456132	Los Olivos	Activo
Metales del Oriente	545465	barrio contructor	Activo

Figura 120 Administrar empresas

### 1.5.1.7.11.4.22 Nueva empresa

Reporte

**Nueva Empresa**

!!Atencion!! Los campos que contengan (\*) deben ser llenados obligatoriamente.

Nombre (\*)

Direccion (\*)

Telefono (\*)

Nombre de la Empresa	Direccion	Telefono	Estado
aaaa			Activo
FABOCE			Activo
Inserpaz			Activo
Los Olivos	456132	Los Olivos	Activo
Metales del Oriente	545465	barrio contructor	Activo

Mostrando 1 - 5 de un total de 5 entradas

Anterior 1 Siguiente

+ + +

Figura 121 Nueva empresa

### 1.5.1.7.11.4.23 Modificar empresa

Reporte

**Nueva Empresa**

!!Atencion!! Los campos que contengan (\*) deben ser llenados obligatoriamente.

Nombre (\*)

Direccion (\*)

Telefono (\*)

Nombre de la Empresa	Direccion	Telefono	Estado
aaaa	&lt;script &gt;alert(&#039;holo&#039;);&lt;/script&gt;	4534553	Activo
FABOCE			Activo
Inserpaz			Activo
Los Olivos	456132	Los Olivos	Activo
Metales del Oriente	545465	barrio contructor	Activo

Mostrando 1 - 5 de un total de 5 entradas

Anterior 1 Siguiente

+ + +

Figura 122 Modificar empresa

### 1.5.1.7.11.4.24 Eliminar empresa

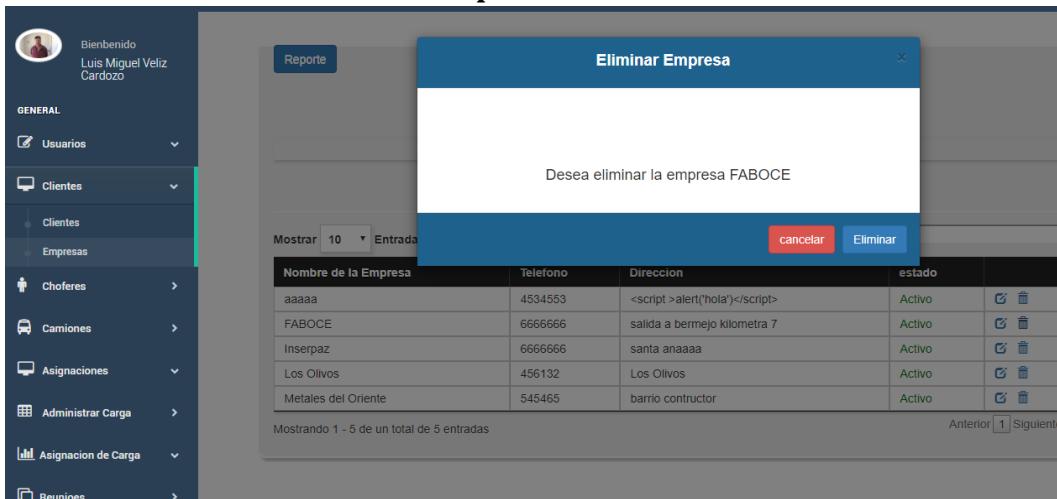


Figura 123 Eliminar empresa

### 1.5.1.7.11.4.25 Empresas inactivas

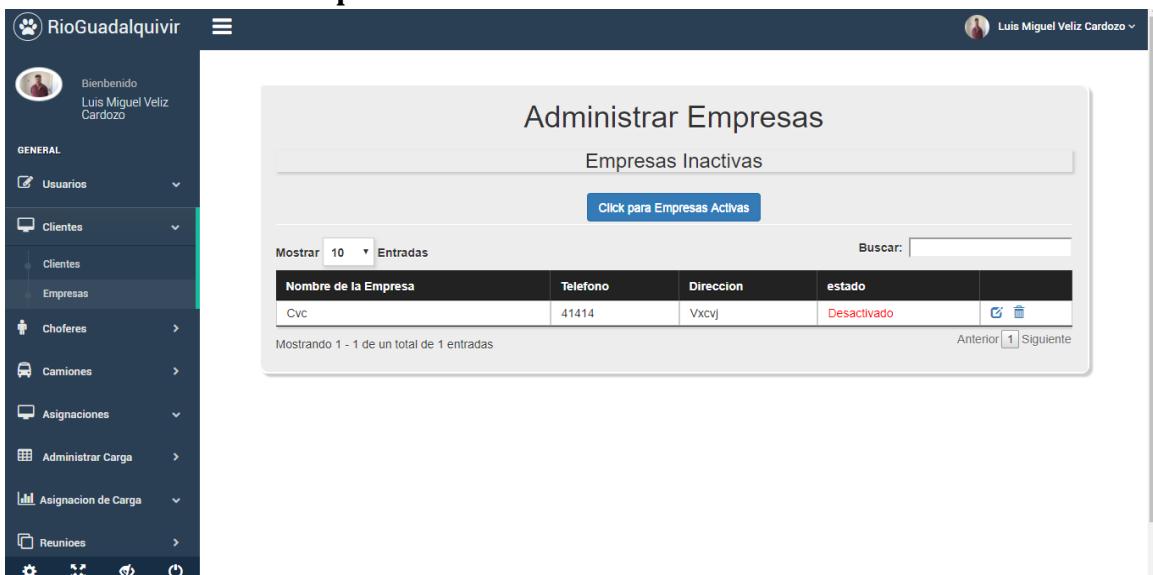


Figura 124 Empresas inactivas

### 1.5.1.7.11.4.26 Habilitar empresa

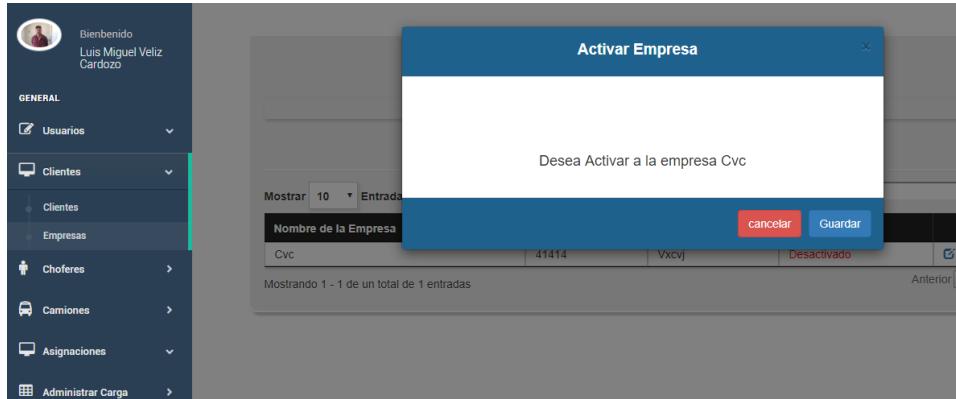


Figura 125 Habilitar empresa

### 1.5.1.7.11.4.27 Administrar carga

Administrador Cargas					
Mostrar 8 Entradas		Copy CSV Excel PDF Print		Buscar:	
Código de Carga	Nombre de la Carga	Tipo de Camion	Fecha	Agregado por	
17	madera	camion trailer	2018-11-08 00:22:45	veliz	
18	ferro	camion trailer	2018-11-08 09:23:42	veliz	
19	ghfgh	camion trailer	2018-11-08 09:29:03	veliz	
20	nose	camion trailer	2018-11-14 11:23:37	veliz	
21	luis miguel veliz	camion	2018-11-14 11:24:50	veliz	
22	carlo caro	camion	2018-11-18 18:35:54	veliz	
23	dhgfskj	camion trailer	2018-11-19 00:50:56	veliz	
24	gfsd	camion con acople	2018-11-19 00:51:17	veliz	

Mostrando 1 - 8 de un total de 13 entradas

Anterior 1 2 Siguiente

Figura 126 Administrar carga

### 1.5.1.7.11.4.28 Nueva carga

Nueva Carga

Agrado por C.I.: 111

Nombre de Carga: Nombres

Detalle: Escriba el detalle

Tipo de Camion: seleccione un tipo

cancelar guardar

Mostrando 1 - 8 de un total de 13 entradas

Anterior 1 2 Siguiente

Figura 127 Nueva carga

### 1.5.1.7.11.4.29 Modificar carga

Modificar Carga

Agrado por C.I.: 111

Nombre De Carga: madera

Detalle: madera de santa cruz a tarja  
ubicacion : de el barrio el teja

Tipo de Camion: camion trailer

cancelar guardar

Mostrando 1 - 8 de un total de 13 entradas

Anterior 1 2 Siguiente

Figura 128 Modificar carga

### 1.5.1.7.11.4.30 Eliminar carga

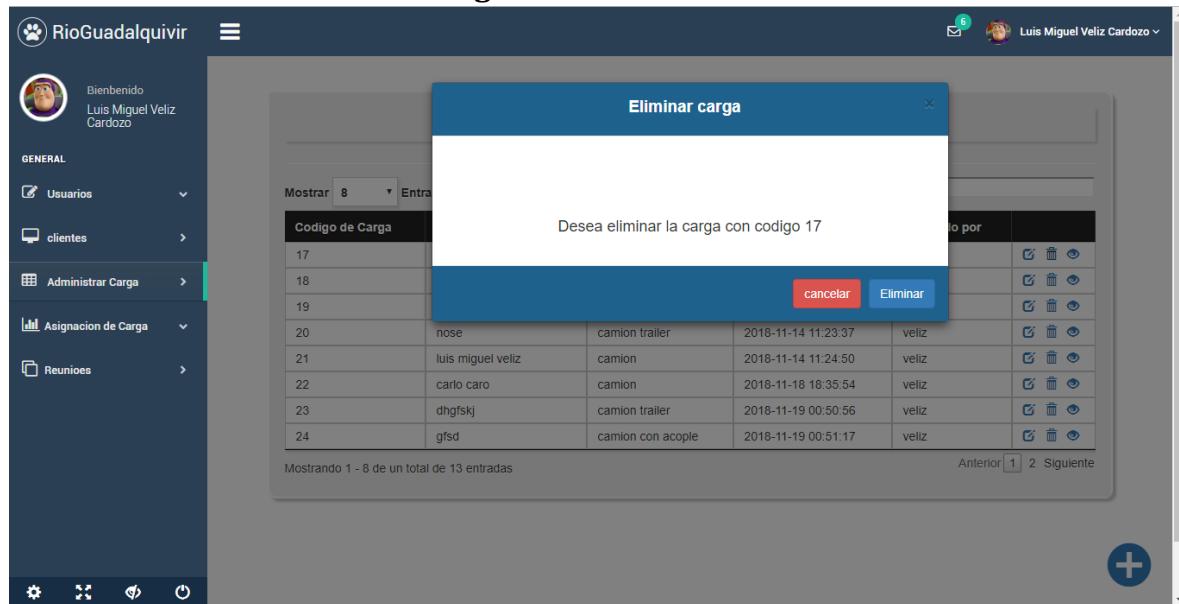


Figura 129 Eliminar carga

### 1.5.1.7.11.4.31 Ver carga

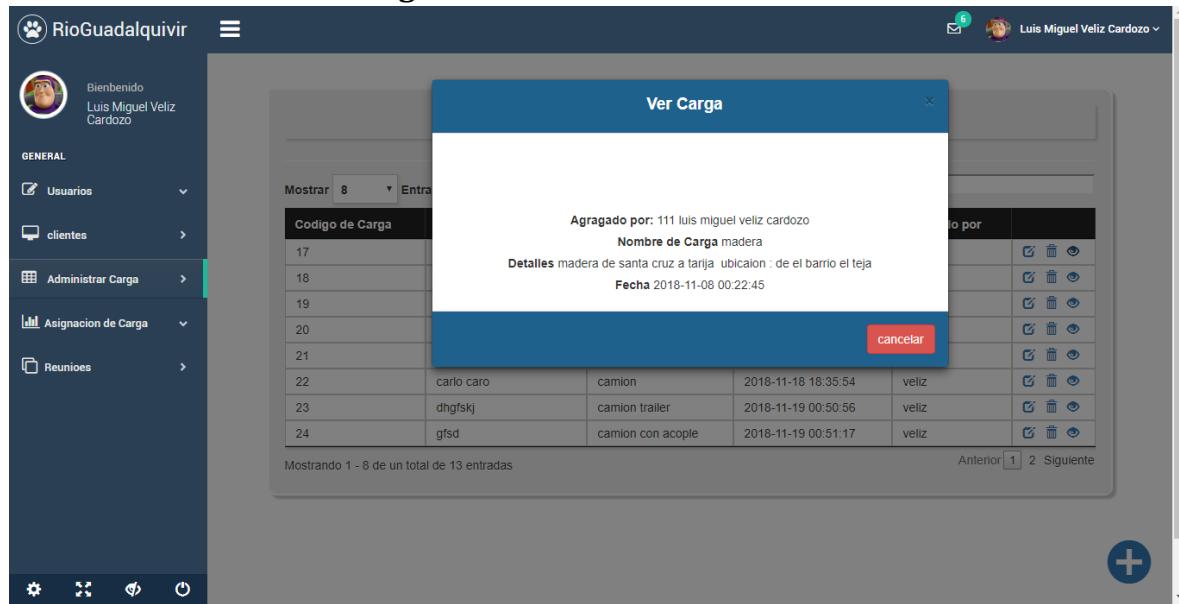


Figura 130 Ver carga

### 1.5.1.7.11.4.32 Reporte carga

Código de Carga	Nombre de la Carga	Tipo de Camion	Fecha	Agregado por
17	madera	camion trailer	2018-11-08 00:22:45	veliz
18	fierro	camion trailer	2018-11-08 09:23:42	veliz
19	ghfgh	camion trailer	2018-11-08 09:29:03	veliz
20	nose	camion trailer	2018-11-14 11:23:37	veliz
21	luis miguel veliz	camion	2018-11-14 11:24:50	veliz
22	carlo caro	camion	2018-11-18 18:35:54	veliz
23	dhgfksj	camion trailer	2018-11-19 00:50:56	veliz
24	gfsd	camion con acople	2018-11-19 00:51:17	veliz

Figura 131 Reporte carga

### 1.5.1.7.11.4.33 Rol de espera

num	Nombres	Apellido Paterno	Apellido Materno	activo	Fecha Ultima de Carga
1	eduardo	carmona	listado	1	2018-11-08 00:22:45
2	luis miguel	veliz	cardozo	1	2018-11-14 11:24:50
3	carlo	carlo	carlo	1	2018-11-18 18:35:54
4	luis	rodriguez	peralta	1	2018-11-19 00:50:56
5	carlos	benites	alcoba	1	2018-11-19 00:51:17
6	gielmermo	senteno	retamozo	1	2018-11-19 00:51:36
7	hector	camacho	carnicel	1	2018-11-19 00:52:22
8	hermelindo	cardozo	zaraso	1	2018-11-19 00:54:35

Figura 132 Rol de espera

### 1.5.1.7.11.4.34 Cambiar estado

**Cambiar Estado**

Su camion esta en mantenimiento??  
Señor eduardo carmona listado

num	Nombres	Carga
1	eduardo	2:45
2	luis miguel	4:50
3	carlo	5:54
4	luis	0:56
5	carlos benites	alcoba
6	giellermo senteno	retamozo
7	hector camacho	carnicel
8	hermelindo cardozo	zaraso

No Si

Mostrando 1 - 8 de un total de 10 entradas

Figura 133 Cambiar estado

### 1.5.1.7.11.4.35 Programación

**programacion**

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno
111	luis miguel	veliz	cardozo
123	luis	rodriguez	peralta
546	hermelindo	cardozo	zaraso
1592	carlo	carlo	carlo
8798	carlos	benites	alcoba
546542	hector	camacho	carnicel
546546	giellermo	senteno	retamozo
789454	eduardo	rivera	condori

Mostrando 1 - 8 de un total de 11 entradas

Figura 134 Programación

### 1.5.1.7.11.4.36 Ver programación

Figura 135 Ver programación

### 1.5.1.7.11.4.37 Administrar reuniones

Código de Reunión	Título	Fecha	Agregado por
1	reunion	2018-11-05 21:30:52	veliz
2	reunion de emergencia	2018-11-05 21:17:45	veliz
3	dfgsd	2018-11-05 21:18:33	veliz
4	gfhdfh	2018-11-05 21:18:47	veliz
5	tfgs	2018-11-05 21:18:55	veliz
6	fdgdf	2018-11-05 21:19:03	veliz
8	hjg	2018-11-05 21:19:18	veliz
9	gdfg	2018-11-05 21:19:37	veliz

Figura 136 Administrar reuniones

### 1.5.1.7.11.4.38 Nueva reunión

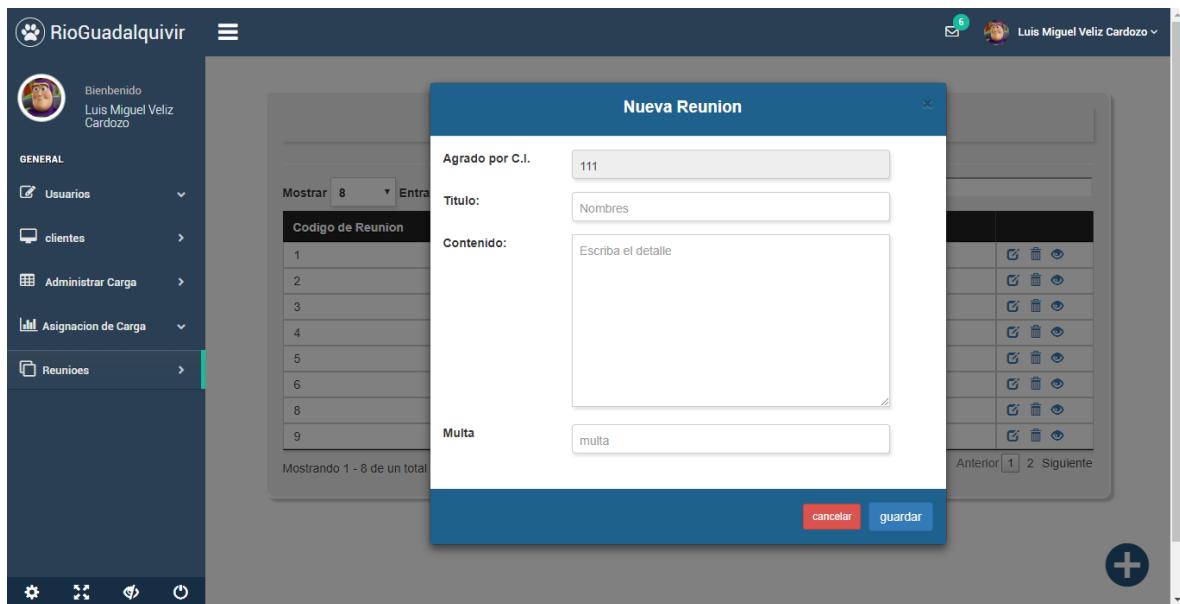


Figura 137 Nueva reunión

### 1.5.1.7.11.4.39 Modificar reunión

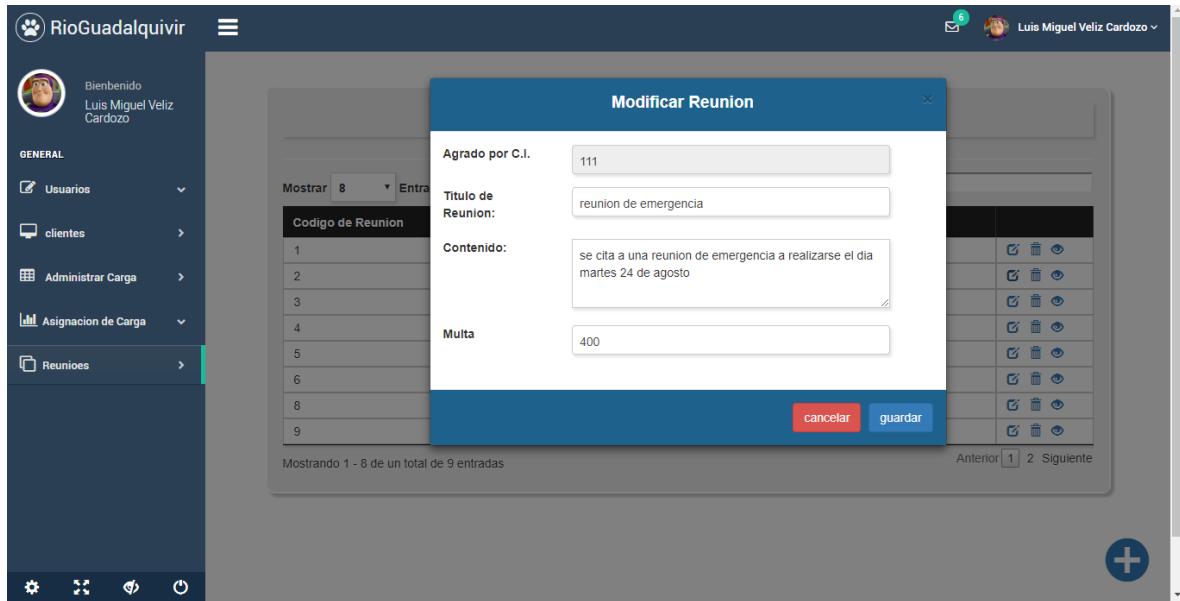


Figura 138 Modificar reunión

### 1.5.1.7.11.4.40 Eliminar reunión

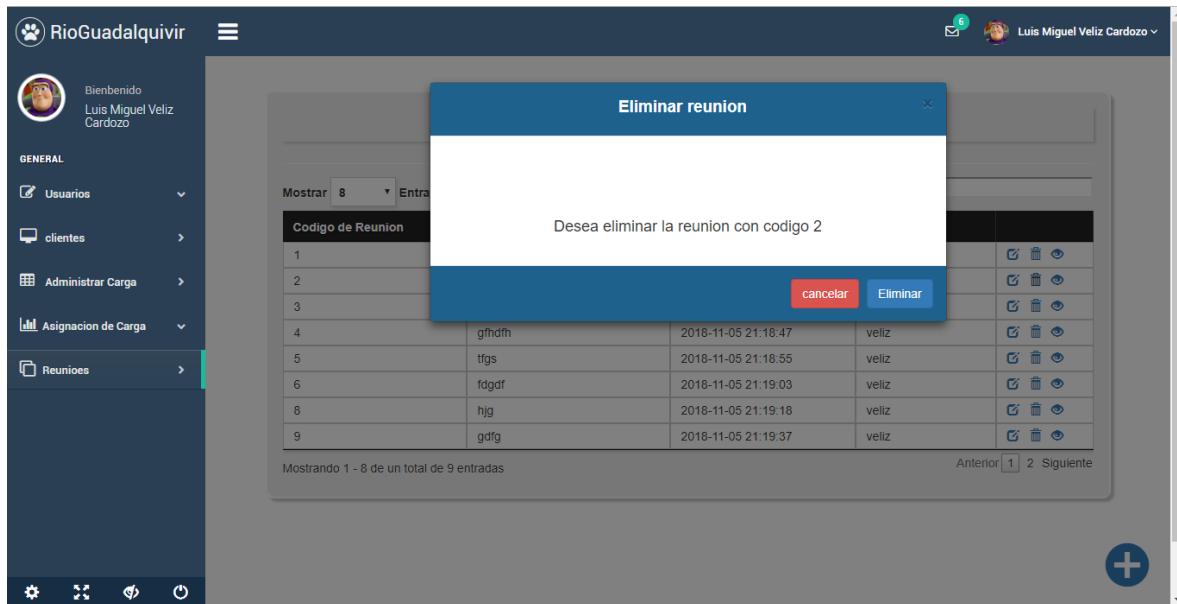


Figura 139 Eliminar reunión

### 1.5.1.7.11.4.41 Ver reunión

Reunion 1

Agragado por luis miguel cardozo

contenido

reunion para elejir nueva directiva urgente  
fecha 2018-11-28 01:48:06 hace 10 horas

Codigo qr    socios presentes    socios con falta

Figura 140 Ver reunión

### 1.5.1.7.11.4.42 Socios presentes

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno
1122544	eduardo	carmona	listado
7509740	luis miguel	veliz	cardozo

Mostrando 1 - 2 de un total de 2 entradas

Anterior 1 Siguiente

Figura 141 Socios presentes

### 1.5.1.7.11.4.43 Código qr

Figura 142 Código qr

### 1.5.1.7.11.4.44 Socios con falta

Carnet de Identidad	Nombre	Apellido paterno	Apellido Materno
1122544	eduardo	carmona	listado
7509740	luis miguel	veliz	cardozo

Mostrando 1 - 2 de un total de 2 entradas

Anterior [1] Siguiente

Figura 143 Socios con falta

### 1.5.1.7.11.4.45 Reporte reuniones

Código de Reunión	Título	Fecha	Agregado por
1	reunion	2018-11-05 21:30:52	veliz
3	dfgsd	2018-11-05 21:18:33	veliz
4	gfhdfh	2018-11-05 21:18:47	veliz
5	tfgs	2018-11-05 21:18:55	veliz
6	fdgdf	2018-11-05 21:19:03	veliz
8	hjg	2018-11-05 21:19:18	veliz
9	gdfg	2018-11-05 21:19:37	veliz
10	kjhjk	2018-11-05 21:19:49	veliz

Figura 144 Reporte reuniones

### **1.5.1.7.12 Pruebas del software**

#### **1.5.1.7.12.1 Objetivos**

- Verificar que el comportamiento externo del sistema software satisface los requisitos establecidos por los futuros usuarios del mismo
- Comprobar la facilidad de uso de la aplicación(usabilidad)
- Comprobar la visualización de las pantallas en los entornos mostrados en los requisitos

#### **1.5.1.7.12.2 Alcance**

Se realizaran pruebas de caja negra mediante el método de partición equivalente sobre 2 accesos de datos clave, estos poseen todos los campos y tipos de datos del sistema:

**Formulario de socios.** - puesto que es más especializado entre los registros de socios contiene todos los campos de prueba .

**Formulario de cargas.** - puesto que es el más completo e importante dentro de todos los registros.

#### **1.5.1.7.12.3 Enfoque (estrategia)**

##### **Pruebas de validación**

Es un tipo de prueba orientado a caja negra que demuestra conformidad con los requisitos (funcionales, rendimiento, documentación)

##### **Metodología de partición equivalente**

Es un método de prueba de caja negra que divide el dominio de la entrada de un programa en clases de datos, para diseñar pruebas que se apliquen sobre el sistema sin necesidad de conocer con esta construido por dentro. Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si el sistema bajo prueba está desempeñando correctamente.

Las pruebas estarán enfocadas a la validación de datos ejecutando pruebas de caja negra en las interfaces clave que dispone el sistema, se realizaran las pruebas mediante el método de partición equivalente.

Esta prueba consiste en:

- Identificar las entradas usuarios
- Selección de tipo de datos (rango, valor específico, lógico y miembro de un conjunto)
- Identificar las clases de equivalencia válidas y no validas
- Diseñar los casos de prueba para todas las clases equivalencia

#### 1.5.1.7.12.4 Diseño de pruebas

Socios:

Clases de equivalencia válidas y no validas

CONDICION DE ENTRADA	TIPO	CLASES DE EQUIVALENCIAS VALIDA	CLASES DE EQUIVALENCIA NO VALIDA
Carnet_de_Identidad	rango	1. 10000000<= Carnet_de_Identidad<=99999999	2 Carnet_de_Identidad <10000000 3 Carnet_de_Identidad >99999999 4 En blanco 5 Valor no numérico
Nombres	valor	6 cadena alfabética	6 en blanco 7 cadena más de 50 caracteres
Apellido_Paterno	lógico valor	8 Cadena alfabética 9 En blanco	10 cadena no alfabética 11 cadena más de 50 caracteres
Apellido_Materno	lógico valor	12 cadena alfabética 13 en blanco	14 cadena no alfabética 15 cadena más de 50 caracteres
Direccion	Valor	16 cadena alfabética	16 cadena no alfabética 17 cadena más de 100 caracteres
Telefono	rango	18 10000000<= Telefono<=99999999	19 en blanco 20 valor no numerico
Rol	Rango	21 1<= rol<=n	22 en blanco 23 Valor no numérico
activo	lógico	24 activo =1 or 0	25 otro valor
imagen	archivo	26 cadena alfabética	27 en blanco 28 otro tipo de archivo

			<b>que no sea imagen</b> <b>29 cadena más de 50</b> <b>caracteres</b>
<b>Created_at</b>	<b>fecha</b>	<b>30 fecha de la forma(año-mes-dia-h-m-s)</b>	<b>31 otro valor</b> <b>32 fecha retrasada o adelantada del dia</b>
<b>Updated_at</b>	<b>fecha</b>	<b>33 fecha de la forma(año-mes-dia-h-m-s)</b>	<b>34 otro valor</b> <b>35 fecha retrasada o adelantada del dia</b>
<b>Tipo</b>	<b>rang o</b>	<b>36 1&lt;= rol&lt;=9</b>	<b>37 en blanco</b> <b>38 Valor no numérico</b>

Tabla 113 Diseño de pruebas

### 1.5.1.7.13 Medios de verificación

#### 1.5.1.7.13.1 Informe de aprobación de especificaciones de requerimientos de software

##### COOPERATIVA DE TRANSPORTE RIO GUADALQUIVIR

Tarija 17 de noviembre de 2018

Señor:

Ing. Succi Aguirre Clovis Gustavo  
(docente de la materia de taller III)

Ref. Proyecto: Mejoramiento en el control y administración de la información en la cooperativa de transporte rio Guadalquivir

El motivo de la presente, es para darle a conocer que doy por aprobado el documento de especificaciones de requerimiento del software que es parte del proyecto “Mejoramiento en el control y administración de la información en la cooperativa de transporte rio Guadalquivir” que es realizado por el universitario Luis Miguel Veliz Cardozo

Expresando así nuestra conformidad le comunico que el documento es aprobado y cumple con todos los requisitos solicitados por la cooperativa rio guadalquivir.

Sin otro particular me despido haciéndole llegar un saludo cordial

Atentamente:



Señor francisco Carmona

Presidente de la cooperativa

**1.5.1.7.13.1      Informe de aprobación de análisis y diseño del sistema****COOPERATIVA DE TRANSPORTE RIO GUADALQUIVIR**

Tarija 17 de noviembre de 2018

Señor:

Ing. Succi Aguirre Clovis Gustavo  
(docente de la materia de taller III)

**Ref. Proyecto: Mejoramiento en el control y administración de la información en la cooperativa de transporte rio Guadalquivir**

El motivo de la presente, es para darle a conocer que doy por aprobado la fase de análisis y diseño de software del proyecto “**Mejoramiento en el control y administración de la información en la cooperativa de transporte rio Guadalquivir**” que es realizado por el universitario Luis Miguel Veliz Cardozo

Expresando así nuestra conformidad le comunico que el documento es aprobado y cumple con todos los requisitos solicitados por la cooperativa rio guadalquivir.

Sin otro particular me despido haciéndole llegar un saludo cordial

Atentamente:



Señor francisco Carmona

Presidente de la cooperativa

## **COMPONENTE II**

**1.6 Componente II:** Desarrollar una aplicación móvil para el control y administración de la información en la cooperativa de transporte rio Guadalquivir

### **1.6.1 Propósito**

Contribuir de manera eficiente y eficaz en la mejora y control de la administración de la información en la cooperativa rio Guadalquivir que permitirá otorgar información rápida y confiable a sus usuarios

### **1.6.2 Alcance**

La aplicación desarrollada otorgara una interfaz amigable para que este pueda tener información actualizada y confiable.

Mediante la aplicación se podrá ver los datos de los socios, clientes, trabajadores, asignación de carga a camiones, mostrara el rol de espera para cargar, aviso de reuniones, además se podrá deshabilitar y habilitar desde la aplicación móvil.

### **1.6.3 Objetivos**

#### **1.6.3.1 Objetivo general**

Desarrollar aplicación móvil para el control y administración de la información en la cooperativa de transporte rio Guadalquivir.

#### **1.6.3.2 Objetivo específico**

- Recolectar de mejor manera la información para poder entender el funcionamiento de la cooperativa.
- Analizar la información recolectada para poder analizarla con el personal involucrado de la cooperativa, los requerimientos que cubrirá el sistema.
- Diseñar y estructurar de la aplicación de forma que cumpla con los requerimientos identificados y además pueda soportar futuros cambios .

- Elaborar la aplicación tomando en cuenta las tecnologías que mejor se adapten a su propósito.
- Ejecutar pruebas que garanticen la calidad del de la aplicación e identifiquen los errores a conseguir antes de su implementación.
- Instrumentar la aplicación con las siguientes tecnologías
  - Ionic
  - Html 5y css3
  - angular
  - TypeScript
- Aplicar el lenguaje de modelado unificado(UML) para el desarrollo de la aplicación.
- Aplicar la metodología RUP
- Diseñar una interfaz gráfica interactiva, amigable y de fácil uso para el manejo del usuario.

#### **1.6.4 Herramientas**

##### **1.6.4.1 Visual Studio Code**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS. 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,<sup>12</sup> aunque la descarga oficial está bajo software propietario.<sup>3</sup>

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar aplicaciones Node.js para el escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor ("Monaco") utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online)

Visual Studio Code es un editor de código fuente. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles

para un idioma dado, como se muestra en la siguiente tabla. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos.json (por ejemplo,

preferencias del usuario). La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo que está fuera de él. Esto también se aplica a los comandos que requieren mucho tiempo. Cuando esto sucede, el comando en progreso se cancela.

En el rol de editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una opción entre LF y CRLF) y el lenguaje de programación del documento activo

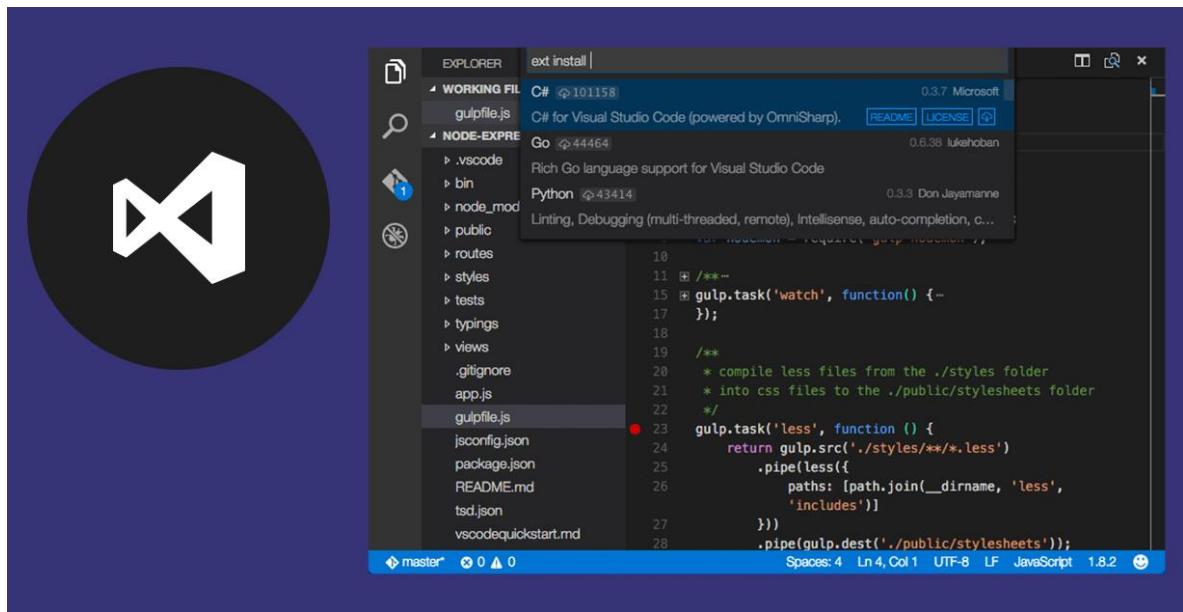


Figura 145 Visual Studio Code

<https://code.visualstudio.com/>

#### 1.6.4.2 Npm y Nodejs

**npm** es el manejador de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.

Desde la versión 0.6.3 de Node.js npm es instalado automáticamente con el entorno. npm se ejecuta desde la línea de comandos y maneja las dependencias para una aplicación. Además, permite a los usuarios instalar aplicaciones Node.js que se encuentran en el repositorio. npm está escrito enteramente en JavaScript y fue desarrollado por Isaac Z. Schlueter a raíz de la frustración que experimentó mientras trabajando con CommonJS y considerando las deficiencias de otros proyectos similares como PHP (PEAR) y Perl (CPAN).

**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 ECMAScript, asíncrono, con I/O 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.<sup>4</sup> Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla

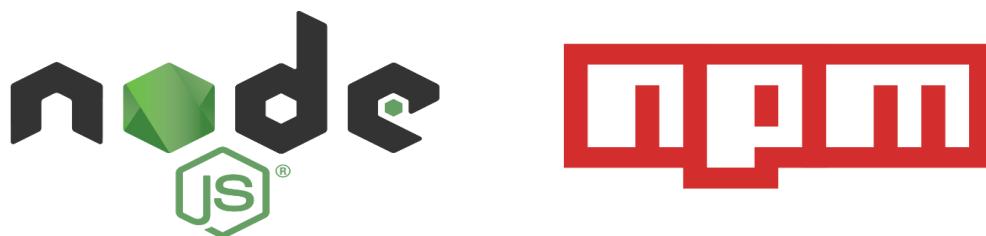


Figura 146 Npm y Nodejs

<https://nodejs.org/es/>

<https://www.npmjs.com/>

### 1.6.4.3 Ionic

Ionic es un completo SDK de código abierto para el desarrollo de aplicaciones móviles híbridas creado por Max Lynch, Ben Sperry y Adam Bradley de Drifty Co. en 2013. La versión original se lanzó en 2013 y se construyó sobre AngularJS y Apache Cordova . Los lanzamientos más recientes, conocidos como Ionic 3 o simplemente "Ionic", se basan en Angular . Ionic proporciona herramientas y servicios para desarrollar aplicaciones móviles híbridas utilizando tecnologías web como CSS , HTML5 y Sass . Las aplicaciones pueden construirse con estas tecnologías web y luego distribuirse a través de tiendas de aplicaciones nativasPara ser instalado en dispositivos aprovechando Cordova.

Ionic utiliza complementos de Cordova para obtener acceso a funciones de sistemas operativos host como cámara, GPS, linterna, etc. Los usuarios pueden crear sus aplicaciones, y luego pueden personalizarse para Android , iOS , Windows , o los navegadores modernos. Ionic te permite crear e implementar tus aplicaciones envolviendo la herramienta de compilación Cordova con una herramienta de línea de comandos 'iónica' simplificada.

Ionic incluye componentes móviles, tipografía, paradigmas interactivos y un tema base extensible.

Usando Angular, Ionic proporciona componentes personalizados y métodos para interactuar con ellos. Uno de esos componentes, la repetición de la colección, permite a los usuarios desplazarse por una lista de miles de elementos sin ningún impacto en el rendimiento. Otro componente, scroll-view, crea un contenedor desplazable con el que los usuarios pueden interactuar utilizando un sistema de delegado con influencia nativa



Figura 147 Ionic

<https://ionicframework.com/>

#### 1.6.4.4 TypeScript

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases. Anders Hejlsberg, diseñador de C# y creador de Delphi y Turbo Pascal, ha trabajado en el desarrollo de TypeScript. TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js).

TypeScript extiende la sintaxis de JavaScript, por tanto cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original.

TypeScript soporta ficheros de definición que contengan información sobre los tipos de librerías JavaScript existentes, similares a los ficheros de cabeceras de C/C++ que describen la estructura de ficheros de objetos existentes. Esto permite a otros programas usar los valores definidos en los ficheros como si fueran entidades TypeScript de tipado estático. Existen cabeceras para librerías populares como jQuery, MongoDB y D3.js, y los módulos básicos de Node.js.

El compilador de TypeScript está escrito asimismo en TypeScript, compilado a JavaScript y con Licencia Apache 2.

TypeScript se incluye como lenguaje de programación de primer nivel en Microsoft Visual Studio 2013 Update 2 y posteriores, junto a C# y otros lenguajes de Microsoft. Una extensión oficial permite a Visual Studio 2012 soportar también TypeScript.



Figura 148 TypeScript

<https://www.typescriptlang.org/>

#### 1.6.4.5 AngularJS

Angular (comúnmente llamado "Angular 2+" o "Angular 2"), es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o ser recuperados de recursos JSON estáticos o dinámicos.



Figura 149 AngularJS

<https://angular.io/>

## 1.6.5 Captura de requisitos

### 1.6.5.1 Modelo de casos de uso

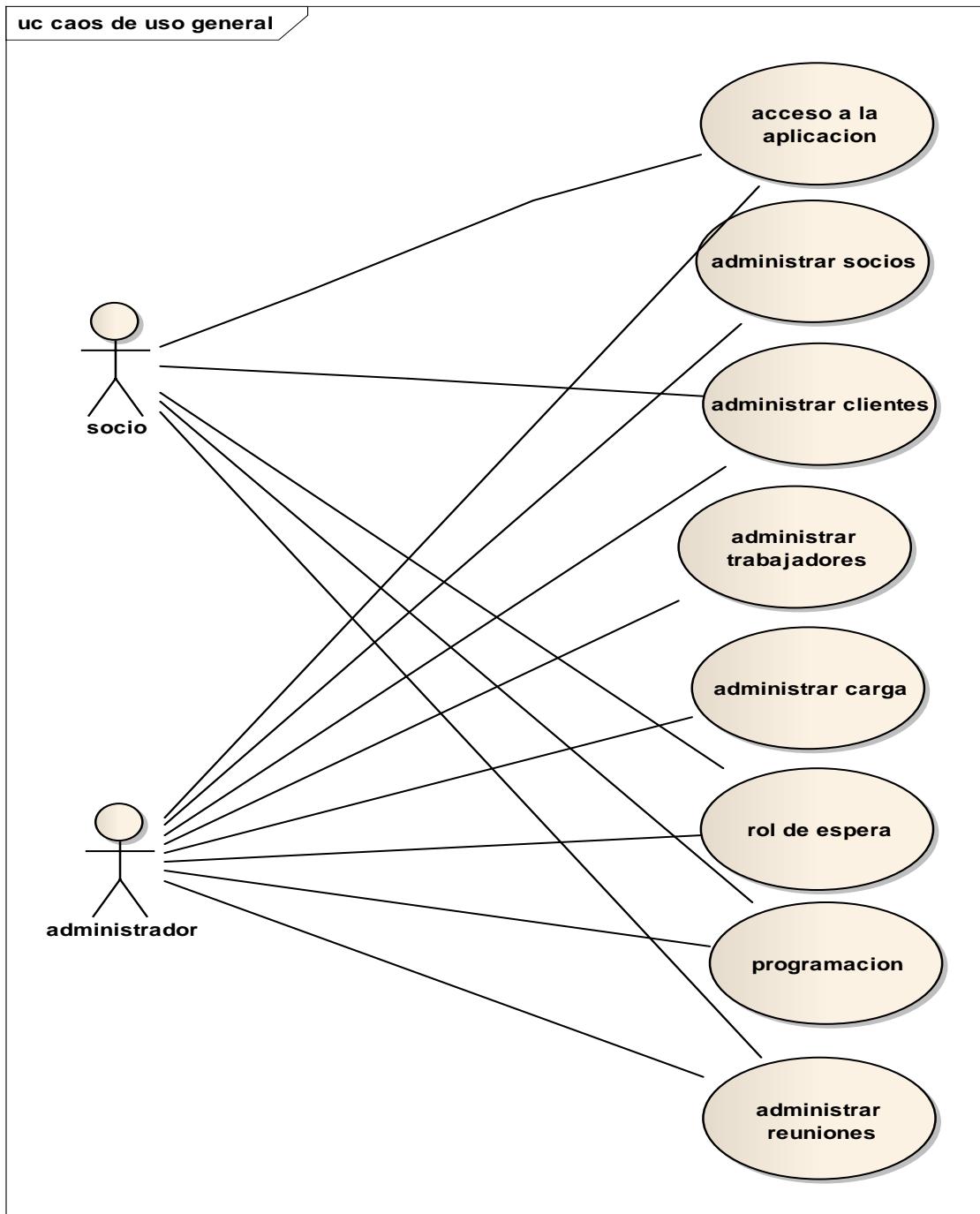


Figura 150 Modelo de casos de uso

### 1.6.5.2 Explosión de casos de uso

#### 1.6.5.2.1 Caso de uso acceso a la aplicación

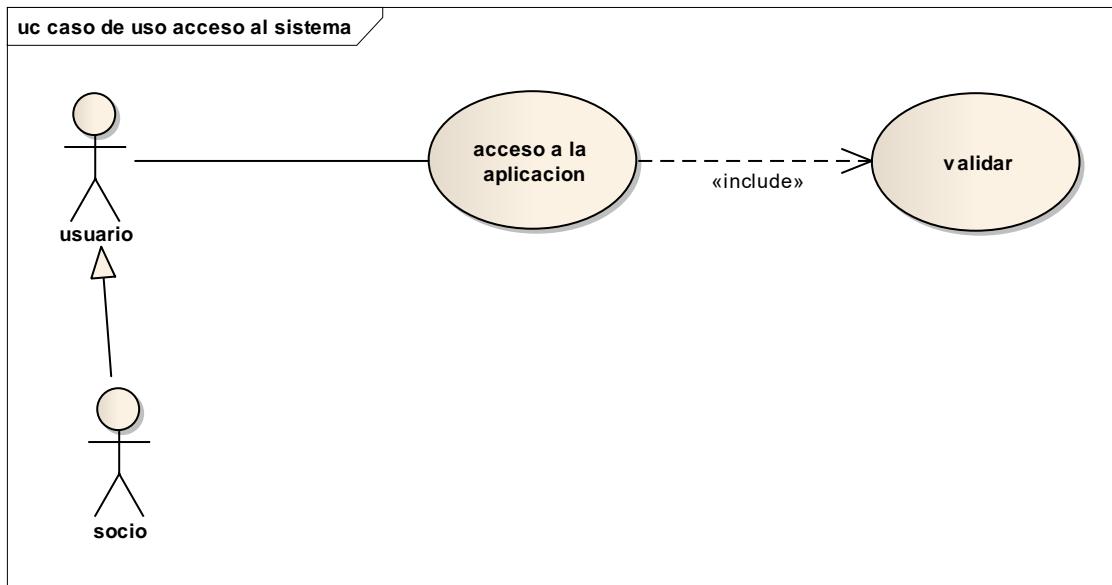


Figura 151 Caso de uso acceso a la aplicación

Caso de uso	Acceso a la aplicación
Actores	Administrador socio
Propósito	Controlar el acceso del personal autorizado para acceder ala aplicación
Resumen	Este caso de uso hace referencia al control de acceso del usuario autorizado al manejo de la aplicación, teniendo que introducir el usuario y clave de acceso, cada usuario que está autorizado para acceder a la aplicación tiene su respectivo rol, que puede ser socio de la aplicación y de acuerdo a este se habilitarán o deshabilitarán algunas opciones.
Precondiciones	Se despliega la pantalla de “ingreso a la aplicación” y tener usuario y contraseña para poder ingresar a la aplicación

Flujo Normal	<p>Se presenta al usuario la pantalla iniciar sesión</p> <p>1.- El usuario introduce nombre del usuario y contraseña.</p> <p>2.- Selecciona la opción “Ingresar”.</p>
Sub flujo	Validar datos

Tabla 114 Caso de uso acceso a la aplicación

**1.6.5.2.2 Caso de uso validar datos**

Caso de uso	Validar datos
Actores	Administrador, socios
Propósito	Permite validar usuario y contraseña si existen o no
Resumen	Este caso de uso hace referencia a validar los datos para de usuario y contraseña que sean escritos correctamente y que solo se ingresen números y letras nada mas
Precondiciones	El usuario debe haber llenado el formulario de ingreso a la aplicación ingresando su usuario y contraseña
Flujo Principal	<p>1.- Se validan los datos del usuario y la contraseña.</p> <p>2.- Si existe, se muestra la pantalla principal de la aplicación</p>
Flujo alternativo	El sistema no encuentra al usuario y Se muestra mensaje “usuario o contraseña incorrectos” en la pantalla iniciar sesión

Tabla 115 Caso de uso validar datos

### 1.6.5.2.3 Caso de uso administrar socios

Caso de uso	Administrar socios
Actores	Administrador socios
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los socios
Resumen	Este caso de uso hace referencia a la administración de los socios.
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar los socios y haber seleccionado del menú de la pantalla principal la opción administrar socios.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar socios con la lista de los socios registrados.</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver.
Excepciones	Ninguna.

Tabla 116 Caso de uso administrar socios

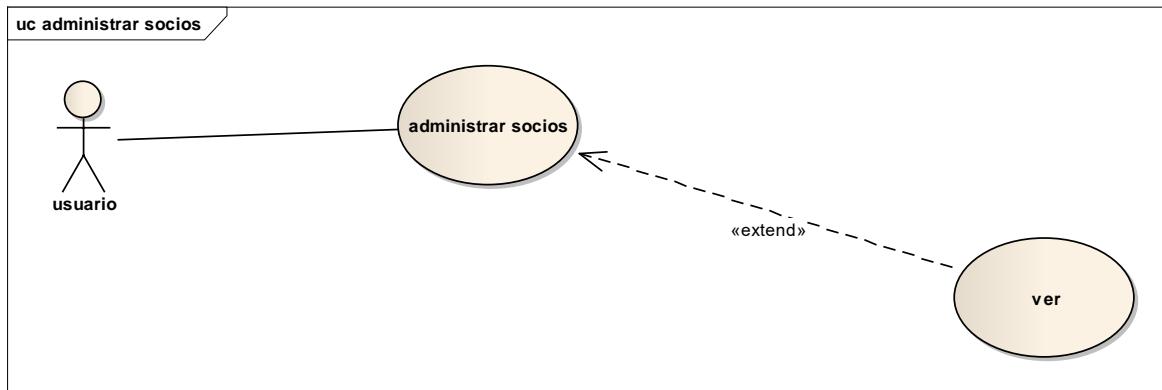


Figura 152 Caso de uso administrar socios

#### 1.6.5.2.4 Casos de uso ver

Caso de uso	Ver
Actores	Administrador socios
Propósito	Permite ver todos los datos del socio
Resumen	Este caso de uso hace referencia a ver los datos del socio con todos sus detalles
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar socios, y seleccionar administrar socios y seleccionar un registro existente de la lista
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles de los socios
Subflujos	Ninguna.

Tabla 117 Casos de uso ver

#### 1.6.5.2.5 Casos de uso administrar clientes

Caso de uso	Administrar clientes
Actores	socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los clientes.
Resumen	Este caso de uso hace referencia a la administración de los clientes.
Precondiciones	El usuario debe haber ingresado a la aplicación , estar autorizado para administrar los clientes y haber seleccionado del menú de la pantalla principal la opción administrar clientes.
Flujo Principal	Se muestra al usuario la pantalla de administrar clientes con la lista de los clientes registrados

	Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.
Subflujos	Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver
Excepciones	Ninguna.

Tabla 118 Casos de uso administrar clientes

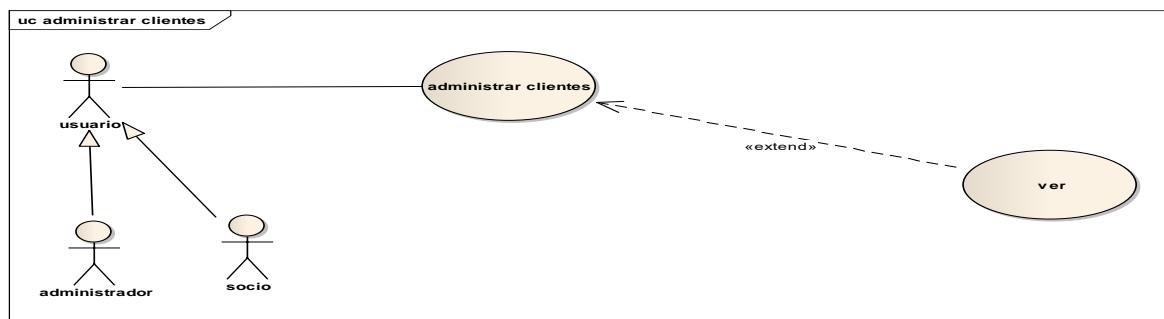


Figura 153 Casos de uso administrar clientes

#### 1.6.5.2.6 Casos de uso ver

Caso de uso	Ver
Actores	Administrador , socios
Propósito	Permite ver todos los datos de los clientes
Resumen	Este caso de uso hace referencia a ver los datos del cliente con todos sus detalles
Precondiciones	El usuario debe haber ingresado a aplicación, estar autorizado para administrar cliente, y seleccionar administrar clientes y seleccionar un registro existente de la lista
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles del cliente

Subflujos	Ninguna.
-----------	----------

Tabla 119 Casos de uso ver

**1.6.5.2.7 Casos de uso administrar trabajadores**

Caso de uso	Administrar trabajadores
Actores	Socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los trabajadores.
Resumen	Este caso de uso hace referencia a la administración de los trabajadores de la cooperativa rio Guadalquivir
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar trabajadores y haber seleccionado del menú de la pantalla principal la opción administrar trabajadores.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar trabajadores con la lista de los trabajadores registrados</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver
Excepciones	Ninguna.

Tabla 120 Casos de uso administrar trabajadores

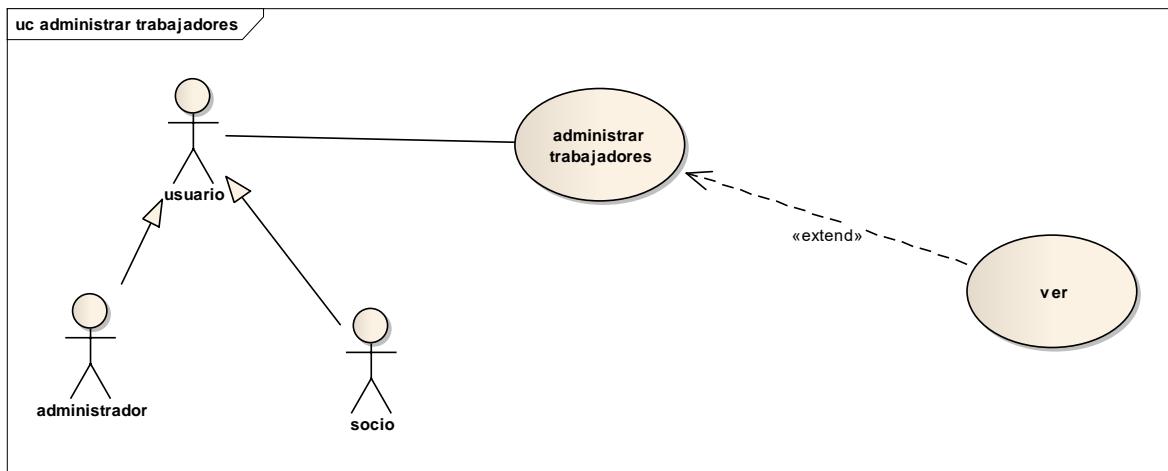


Figura 154 Casos de uso administrar trabajadores

#### 1.6.5.2.8 Caso de uso ver

Caso de uso	Ver
Actores	Administrador , socio
Propósito	Permite ver todos los datos de los trabajadores
Resumen	Este caso de uso hace referencia a ver los datos del trabajador con todos sus detalles
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar trabajadores, y seleccionar administrar trabajadores y seleccionar un registro existente de la lista
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles del trabajador
Subflujos	Ninguna.

Tabla 121 Caso de uso ver

### 1.6.5.2.9 Caso de uso administrar carga

Caso de uso	Administrar carga
Actores	Socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de las cargas.
Resumen	Este caso de uso hace referencia a la administración de carga de la cooperativa.
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar carga y haber seleccionado del menú de la pantalla principal la opción administrar carga.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar carga con la lista de los cargas registrados en la aplicación</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver.
Excepciones	Ninguna.

Tabla 122 Caso de uso administrar carga

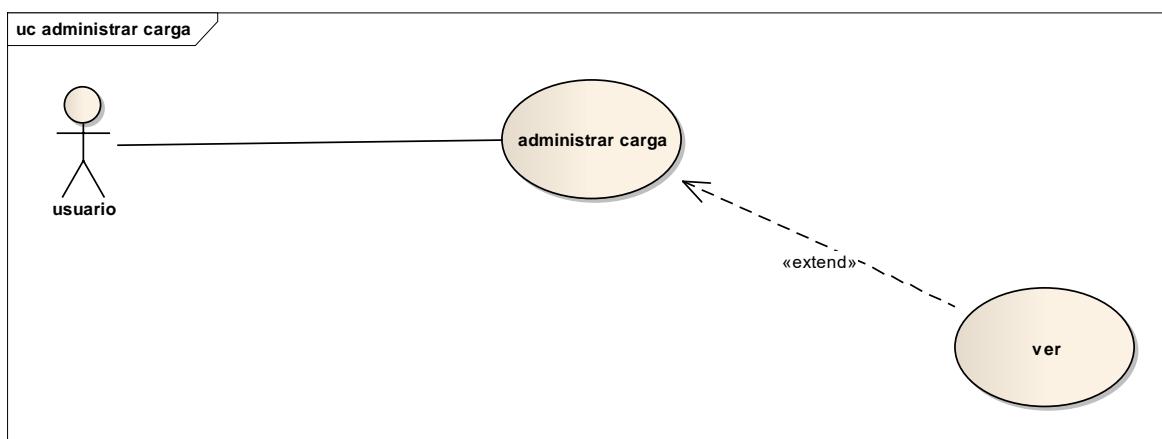


Figura 155 Caso de uso administrar carga

### 1.6.5.2.10 Caso de uso ver

Caso de uso	Ver
Actores	Administrador , socio
Propósito	Permite ver todos los datos de las cargas disponibles para los camiones de los socios de la cooperativa
Resumen	Este caso de uso hace referencia a ver los datos del cargas disponibles para los camiones de los socios de la cooperativa
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar cargas, y seleccionar administrar cargas y seleccionar un registro existente de la lista
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles de la carga
Subflujos	Ninguna.

Tabla 123 Caso de uso ver

### 1.6.5.2.11 Caso de uso rol de espera

Caso de uso	Rol de espera
Actores	socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de socios que están haciendo fila para recibir carga.
Resumen	Este caso de uso hace referencia a una lista donde los socios están esperando carga de forma ordenada
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para rol de espera y haber seleccionado del menú de la pantalla principal la

	opción rol de espera.
Flujo Principal	<p>Se muestra al usuario la pantalla de rol de espera con la lista de los socios registrados</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “cambio de estado” se ejecuta el subflujo cambio de estado.</p>
Excepciones	Ninguna.

Tabla 124 Caso de uso rol de espera

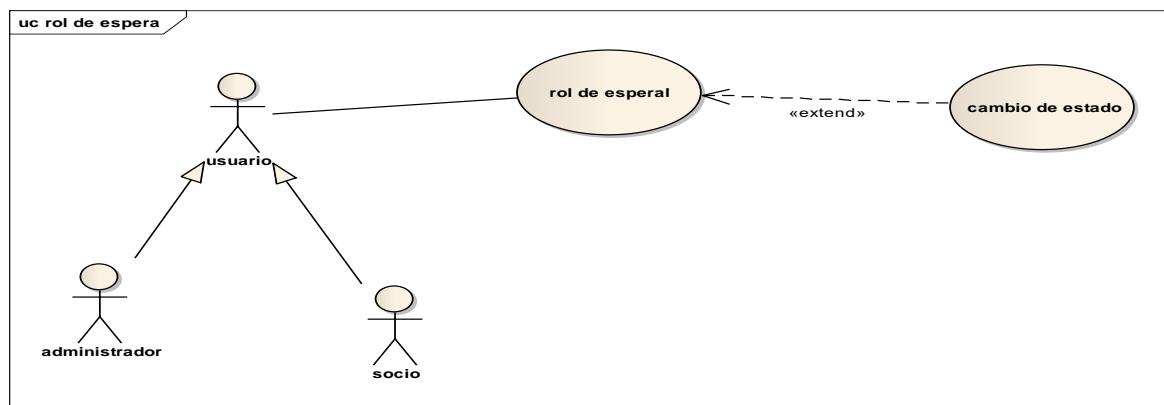


Figura 156 Caso de uso rol de espera

### 1.6.5.2.12 Cambio de estado

Caso de uso	Cambio de estado
Actores	administrador, socio
Propósito	Permite cambiar de estado.
Resumen	<p>Este caso de uso hace referencia a cambiar de estado.</p> <p>Cuando su camión este fregado o en mantenimiento el socio puede cambiar su estado para no recibir una carga y puede activar su estado</p>

	cuento su camión este de listo para transportar una carga
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para rol de espera, seleccionar su nombre para poder cambiar de estado.
Flujo Principal	<p>Se muestra al usuario la pantalla con la lista de los socios haciendo fila y seleccionar el botón cambio de estado</p> <p>El usuario puede seleccionar las siguientes opciones:</p> <p>“cambio de estado” se hace el cambio de estado de 1 a 0 si estado está en 0 se cambia 1 El registro se guardara en la tabla usuarios_tra de la base de datos y muestra mensaje de “su transacción se realizó correctamente” y se retorna a la pantalla “programación”.</p>

Tabla 125 Cambio de estado

#### 1.6.5.2.13 Caso de uso programación

Caso de uso	programación
Actores	socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en una lista de los socios ya con su carga asignada
Resumen	Este caso de uso hace referencia a una lista donde ya se asignó la carga a cada socio y el socio puede ver que carga le toco transportar
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado programación y haber seleccionado del menú de la pantalla principal la opción programación.
Flujo Principal	<p>Se muestra al usuario la pantalla de programación con la lista de los socios registrados y asignado una carga</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará</p>

	con los diversos subflujos de este caso de uso.
Subflujos	El usuario puede seleccionar entre las siguientes opciones: Si el usuario selecciona la opción “ver” se ejecuta el subflujo ver
Excepciones	Ninguna.

Tabla 126 Caso de uso programación

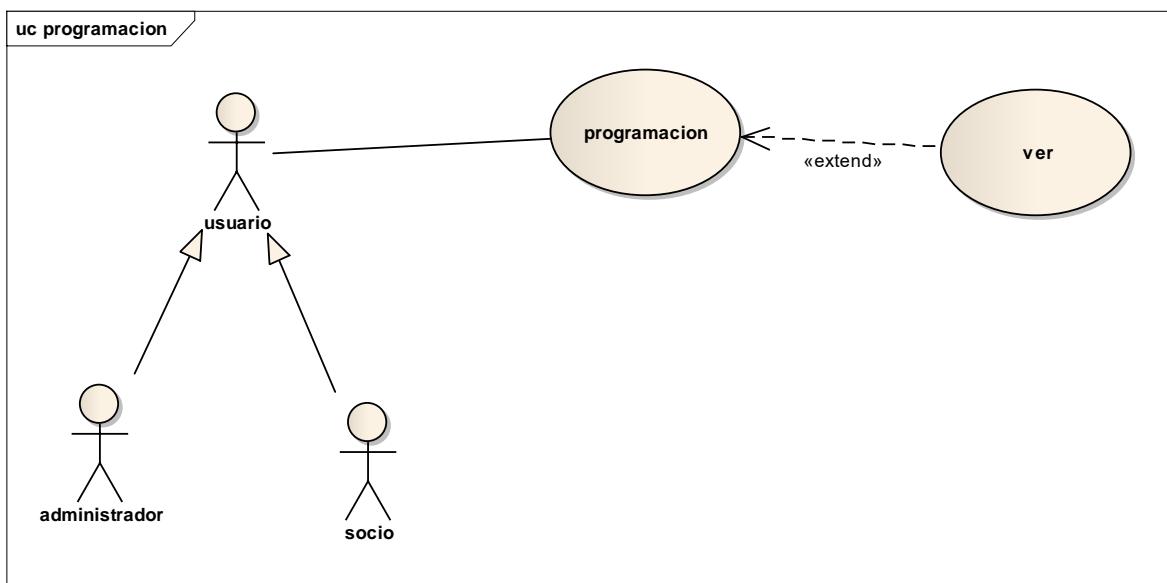


Figura 157 Caso de uso programación

#### 1.6.5.2.14 Casos de uso ver

Caso de uso	Ver
Actores	Administrador , socio
Propósito	Permite ver todos los datos de la programación
Resumen	Este caso de uso hace referencia a ver los datos de la carga que le toco a un determinado socio

Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para programación, y seleccionar programación y seleccionar un registro existente de la lista
Flujo Principal	Se muestra al usuario la pantalla ver con todos sus detalles de la carga que le toco
Subflujos	Ninguna.

Tabla 127 Casos de uso ver

#### 1.6.5.2.15 Caso de uso administrar reuniones

Caso de uso	administrar reuniones
Actores	socio, administrador
Propósito	Permite visualizar y realizar diferentes acciones en la última reunión registrada
Resumen	Este caso de uso hace referencia a la administración de reuniones de la cooperativa.
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar reuniones y haber seleccionado del menú de la pantalla principal la opción administrar reuniones.
Flujo Principal	<p>Se muestra al usuario la pantalla de administrar reuniones con la reunión activa registrada</p> <p>Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos subflujos de este caso de uso.</p>
Subflujos	<p>El usuario puede seleccionar entre las siguientes opciones:</p> <p>Si el usuario selecciona la opción “scanner código qr” se ejecuta el subflujo scanner código qr</p>

Excepciones	Ninguna.

Tabla 128 Caso de uso administrar reuniones

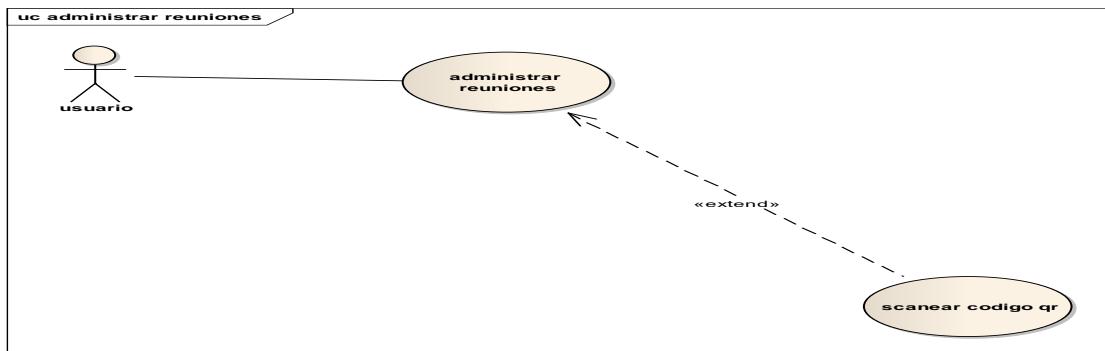


Figura 158 Caso de uso administrar reuniones

#### 1.6.5.2.16 Caso de uso scanear código qr

Caso de uso	Scanear código qr
Actores	Administrador ,socio
Propósito	permite scanear el código qr generado por el sistema para poner estar registrado como presente dicho sistema
Resumen	Este caso de uso hace referencia a escanear el código qr generado por el sistema
Precondiciones	El usuario debe haber ingresado a la aplicación, estar autorizado para administrar reuniones, y seleccionar administrar reuniones y seleccionar scanear código qr
Flujo Principal	Se muestra al usuario la pantalla para escanear el código qr si se registra un error abra un sonido diciendo error si todo se realizó con éxito un sonido correcto
Subflujos	Ninguna.

Tabla 129 Caso de uso ver

### 1.6.5.3 Diagrama de secuencia

#### 1.6.5.3.1 Acceso al sistema

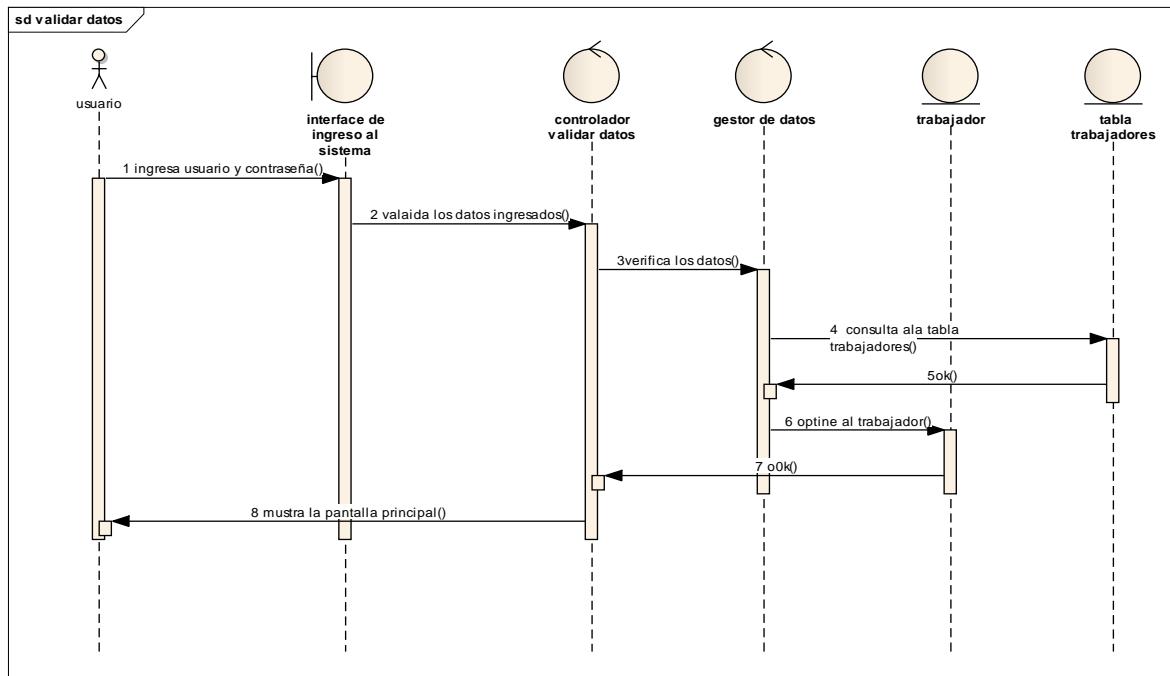


Figura 159 Acceso al sistema

#### 1.6.5.3.2 Ver socio

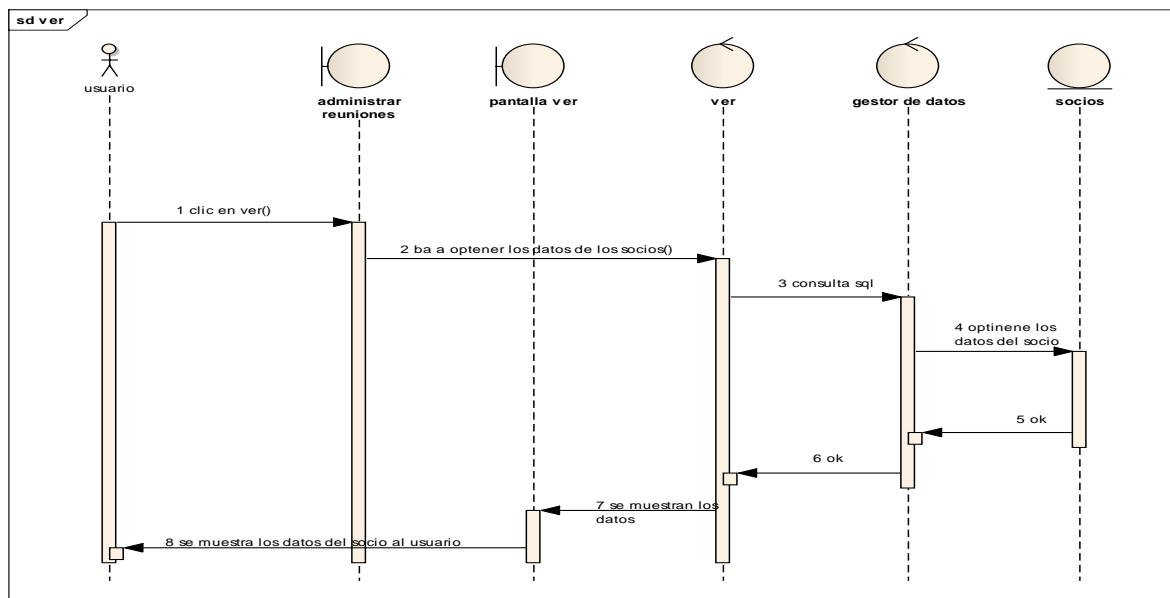


Figura 160 Ver socio

### 1.6.5.3.3 Ver cliente

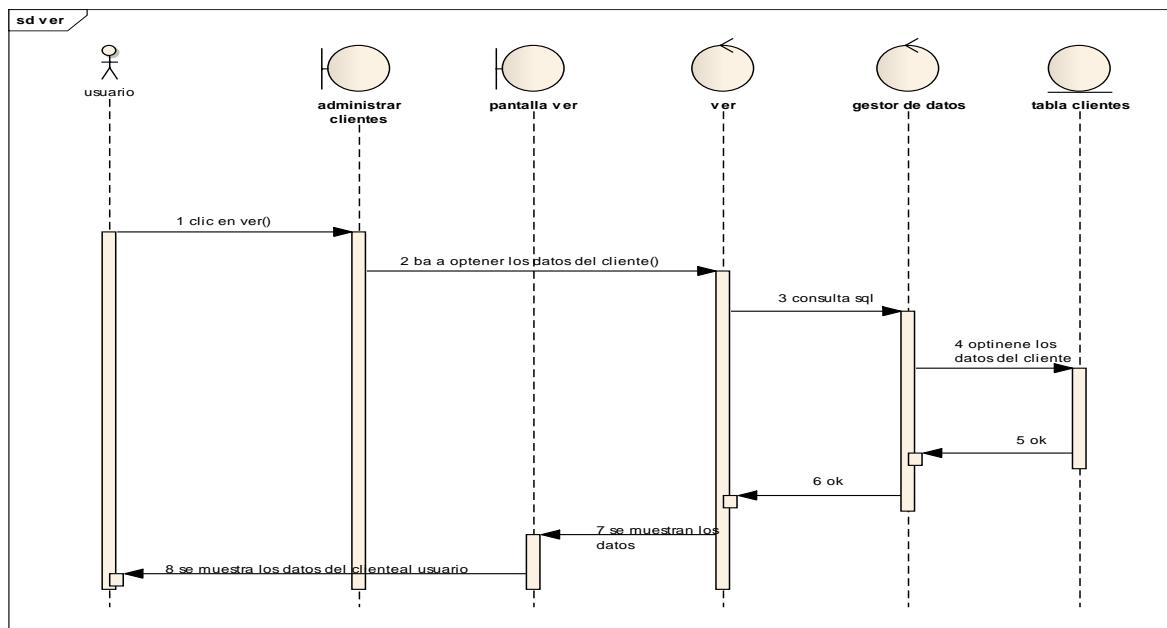


Figura 161 Ver cliente

### 1.6.5.3.4 Ver trabajador

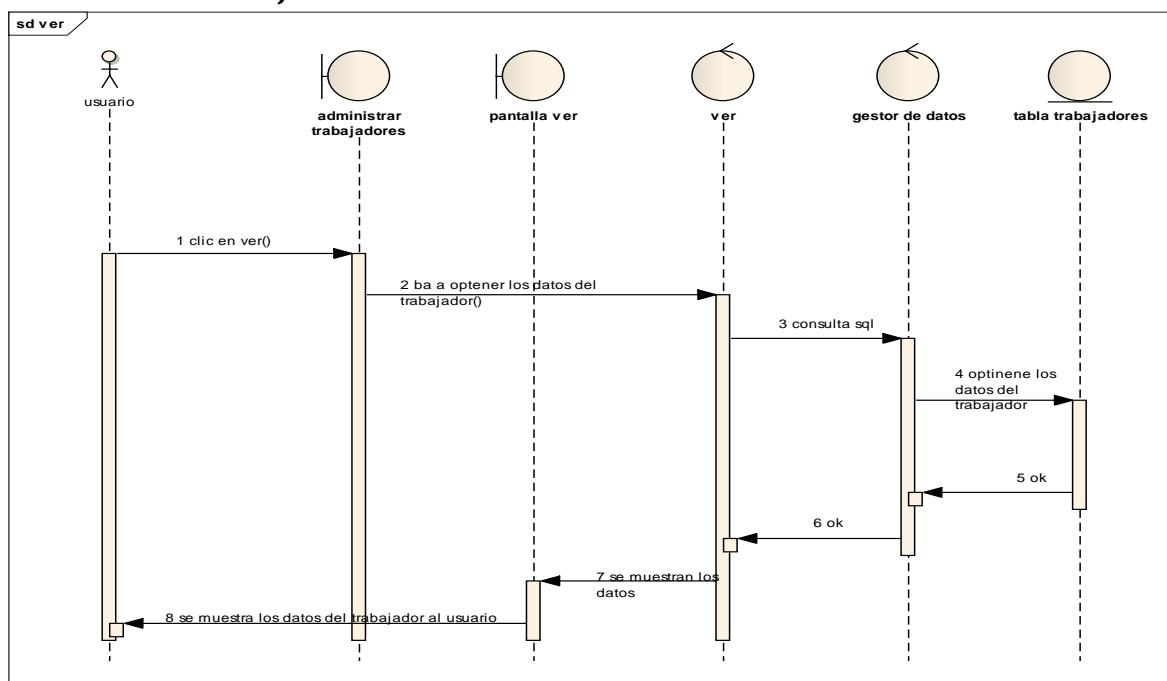


Figura 162 Ver trabajador

### 1.6.5.3.5 Ver carga

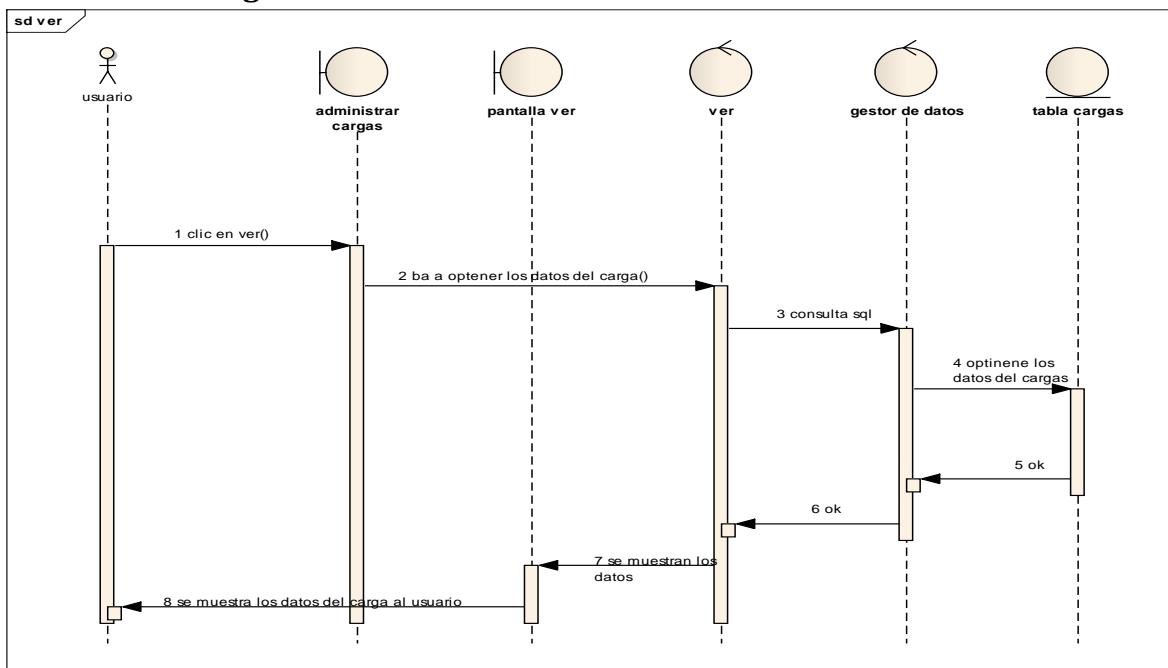


Figura 163 Ver carga

### 1.6.5.3.6 Ver reunión

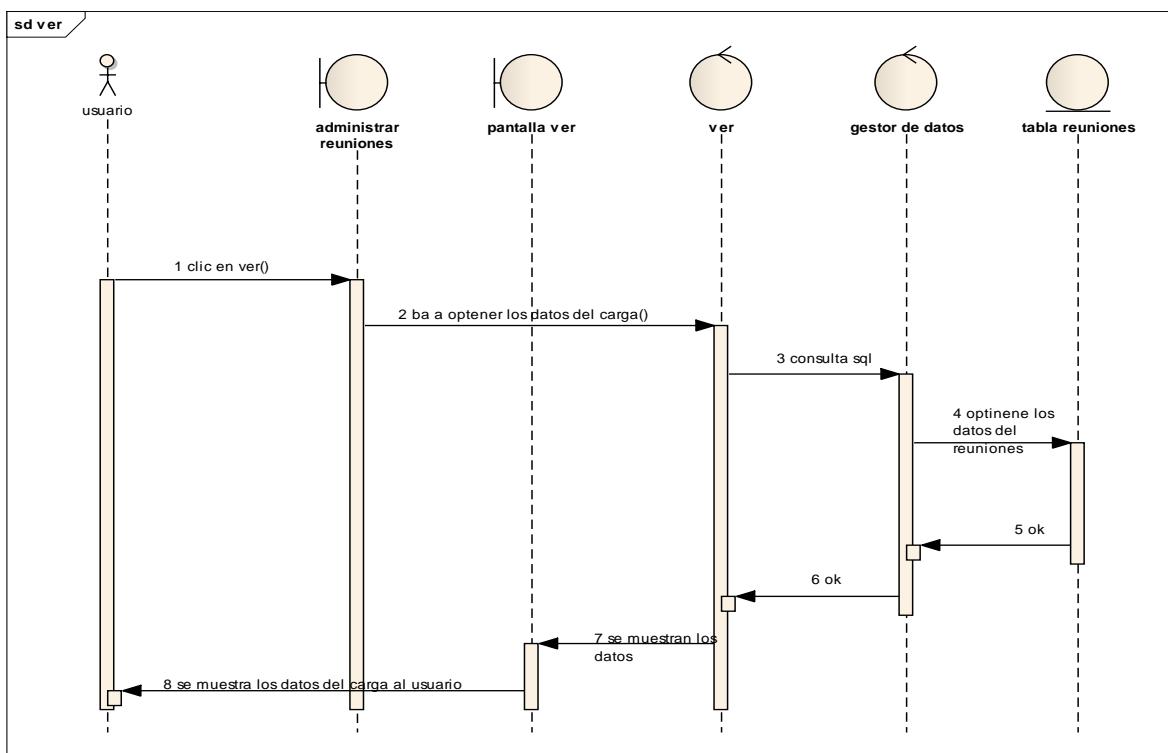


Figura 164 Ver reunión

#### 1.6.5.4 Prototipo de interfaces

##### 1.6.5.4.1 Iniciar sesión



Figura 165 Iniciar sesión

#### 1.6.5.4.2 Menú



Figura 166 Menú

#### 1.6.5.4.3 Socios



Figura 167 Socios

#### 1.6.5.4.4 Ver



Figura 168 Ver

#### 1.6.5.4.5 Clientes

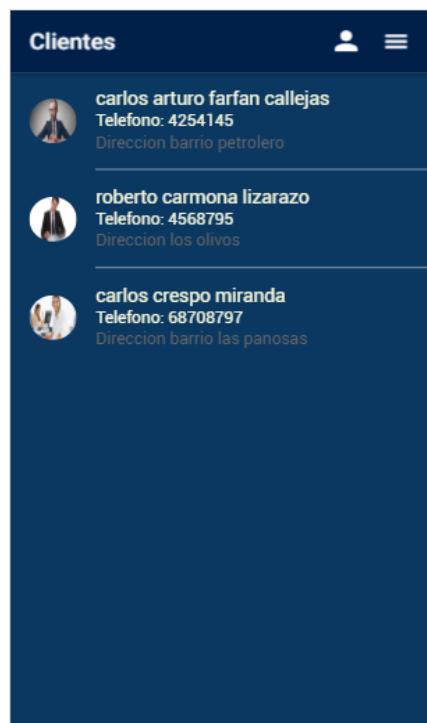


Figura 169 Clientes

#### 1.6.5.4.6 Ver cliente



Figura 170 Ver cliente

#### 1.6.5.4.7 Trabajadores

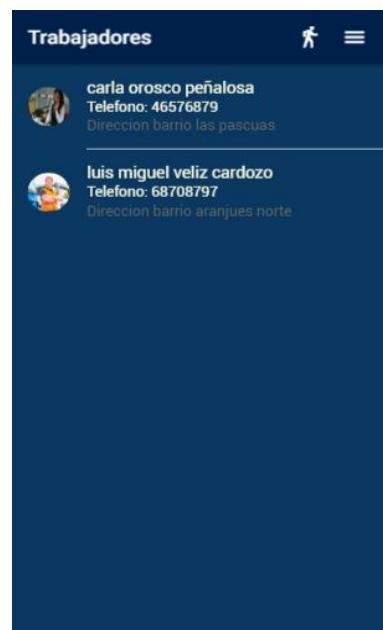


Figura 171 Trabajadores

#### 1.6.5.4.8 Ver trabajador



Figura 172 Ver trabajador

#### 1.6.5.4.9 Cargas



Figura 173 Cargas

#### 1.6.5.4.10 Ver carga



Figura 174 Ver carga

#### 1.6.5.4.11 Rol de espera



Figura 175 Rol de espera

#### 1.6.5.4.12 Cambio de estado



Figura 176 Cambio de estado

#### 1.6.5.4.13 Programación

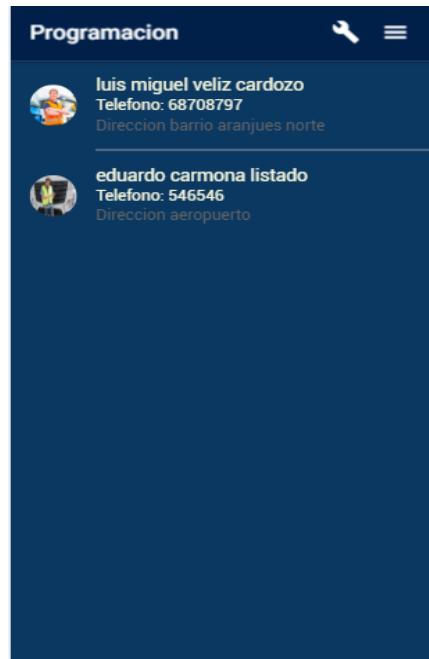


Figura 177 Programación

#### 1.6.5.4.14 Ver programación



Figura 178 Ver programación

#### 1.6.5.4.15 Reunión activa



Figura 179 Reunión activa

#### 1.6.5.4.16 Escanear código qr



Figura 180 Escanear código qr

#### 1.6.5.5 Arquitectura

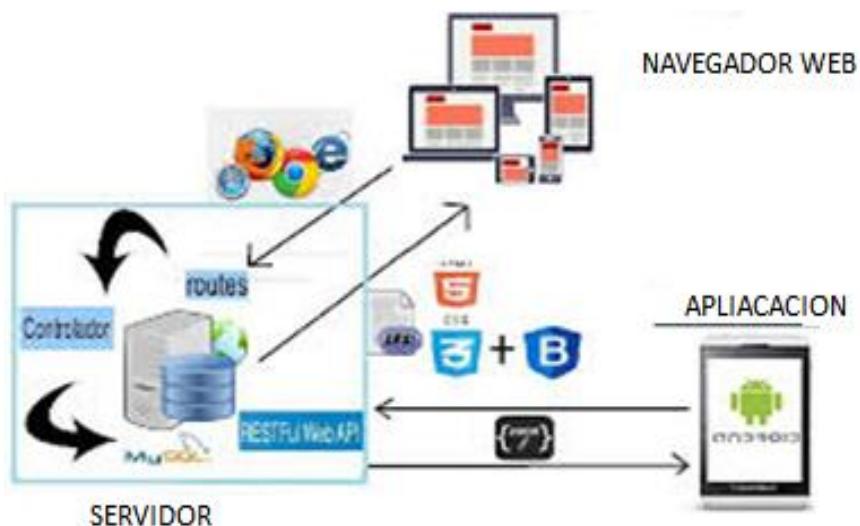


Figura 181 Arquitectura

### **1.6.5.6 Desarrollo**

**Aplicación para Android** se trata de una aplicación donde los socios y el administrador pueden entrar la aplicación proveerá la información de todos los socios clientes trabajadores y además los socios ya no tendrán que acudir a la oficina de la cooperativa para saber cuándo le toca carga o para poder asignarle carga cosa que el socio lo puede hacer desde la aplicación sin acudir a la oficina de la cooperativa

Y también el socio puede deshabilitarse desde la aplicación para no recibir ninguna carga

### **1.6.5.7 Verificación y evaluación**

#### **1.6.5.7.1 Pruebas del software**

##### **1.6.5.7.1.1 Pruebas de aceptación**

Durante este componente las pruebas que se han llevado a cabo son pruebas de aceptación. Estas pruebas han sido llevadas a cabo en distintas fases:

###### **Fase alfa**

Durante esta fase solo el desarrollador ha sido el encargado de probar el programa, las pruebas se iban sucediendo en base al trabajo finalizado en cada momento, es decir, cada parte que se iba desarrollando se iba probando. Durante esta fase eran muy comunes las reescrituras de código por problemas de programación

###### **Fase beta I**

En esta fase se eligieron sujetos con conocimientos avanzados del uso de Android para que probaran la aplicación. Una vez se encontraba algún fallo el desarrollador lo reparaba para la siguiente versión de la beta

###### **Fase beta II**

En esta versión los sujetos elegidos para la prueba fueron ampliados y elegidos con un menor nivel uso del sistema Android. De esta manera tenia experiencias, tanto de usuarios avanzados como de usuarios principiantes. La reparación de los errores encontrados se sucedió de la misma forma que en la fase beta I

### 1.6.5.7.2 Documentación de pruebas

Debido a la gran cantidad de pruebas realizadas para probar el correcto funcionamiento de las aplicaciones del sistema a continuación, se citan algunas de la pruebas realizadas en la fase alfa de las pruebas que es cuando más problemas surgieron a la hora de probar que todo funciona

Descripción	Resultado esperado	Resultado obtenido
Instalación de la aplicación por el usuario	La instalación se realizó con éxito	Se necesita permisos para instalar esta aplicación
Iniciar aplicación	La aplicación se muestra sin ningún error	Problemas de compatibilidad por tener una versión de Android antigua
Iniciar sesión	Usuario registrado con acceso al menú de la aplicación	Mensaje de aviso que la aplicación se ha detenido
Obteniendo datos de los socios	Datos obtenidos con éxito	No se pudo obtener ningún dato
Cambio de estado del socio	Cambio de estado con éxito	No se realizó ningún cambio de estado en la base de datos
Obteniendo imágenes de los socios	Imágenes obtenidas con éxito	Angular tiene una protección llamada Sanitization para url del servidor

Tabla 130 Documentación de pruebas

## **COMPONENTE III**

## **1.7 Componente III:** capacitación sobre el sistema desarrollado

### **1.7.1 Introducción**

El objetivo de este componente es capacitar a los usuarios para poder utilizar el sistema desarrollado y la aplicación, utilizando métodos y medios de enseñanza-aprendizaje adecuado. Con lo siguiente se pretende evitar errores y riesgo en el manejo del sistema y así poder sacar el máximo beneficio a dicho sistema y aplicación.

La capacitación en el uso del sistema informático al personal afectado por el proyecto se convierte en un componente fundamental para el logro del propósito que tiene el proyecto.

El componente de capacitación se encamina hacia el siguiente objetivo: usar adecuadamente el sistema por los usuarios de la cooperativa de transporte río Guadalquivir.

### **1.7.2 Propósito**

El propósito de esta capacitación es instruir el un manejo exitoso del sistema desarrollado y la aplicación a fin de explotar dicho sistema y aplicación de forma correcta y eficiente.

### **1.7.3 Alcances y limitaciones**

#### **1.7.3.1 Alcances**

Se realizará exposiciones apoyadas en el uso del sistema

La capacitación será práctica

#### **1.7.3.2 Limitaciones**

La capacitación será dos días y 3 horas al día.

#### **1.7.4 La técnica de capacitación utilizada**

La técnica de capacitación que se utilizará es la expositiva que tiene el objetivo de hacer capaz o hábil a una persona en el manejo de un sistema, implica explicarle adecuadamente el procedimiento y favorecer su práctica.

Se seguirán los siguientes pasos para capacitar a los usuarios del sistema

- Primero se realizarán exposiciones, en las que se introducirá el y a su forma general de uso
- Se darán explicaciones sobre los procedimientos básicos a seguir en el manejo del sistema
- Los participantes luego podrán en la práctica con la supervisión del capacitor, pudiendo consultar a este todas sus dudas y expresar sus inquietudes, lo que se requiere lograr es que el sistema desarrollado y la aplicación sean bastante intuitivos y los usuarios se acostumbren a él rápidamente
- Se realizarán prácticas de procedimientos más usuales con cada usuario, esto también ayudará a la administración para certificar la conformidad con el software

#### **1.7.5 Encargado de la capacitación**

En encargado en realizar las capacitaciones será el universitario que desarrolló el proyecto Luis Miguel Veliz Cardozo

#### **1.7.6 Definición de público**

El público objetivo de la capacitación será :

- Todo el socio de la cooperativa de transporte río Guadalquivir
- Los trabajadores

#### **1.7.7 Lugar de capacitación**

La capacitación se desarrollará en la instalación de la oficina de la cooperativa de transporte río Guadalquivir ubicada en la avenida panamericana a dos cuadras pasando el mástil

### **1.7.8 Material para la capacitación**

- Una computadora en la cual implementamos el servidor del sistema
- Computadora adicional para que los asistentes realicen las prácticas
- Impresora
- Teléfonos celulares para la aplicación

### **1.7.9 Contenido**

El contenido de la capacitación abarca la forma rápida y concisa toda la funcionalidad del sistema , se estructura el contenido de la misma forma que está estructurado el sistema

Se capacitará a los socios y a los trabajadores de la cooperativa

- Introducción
- Acceso al sistema
- Administrar clientes
- Administrar socios
- Administrar trabajadores
- Administrar cargas
- Rol de espera
- Programación
- Reuniones
- Reportes
- Manejo de la aplicación

### 1.7.10 Proceso de la capacitación



Figura 182 Proceso de la capacitación

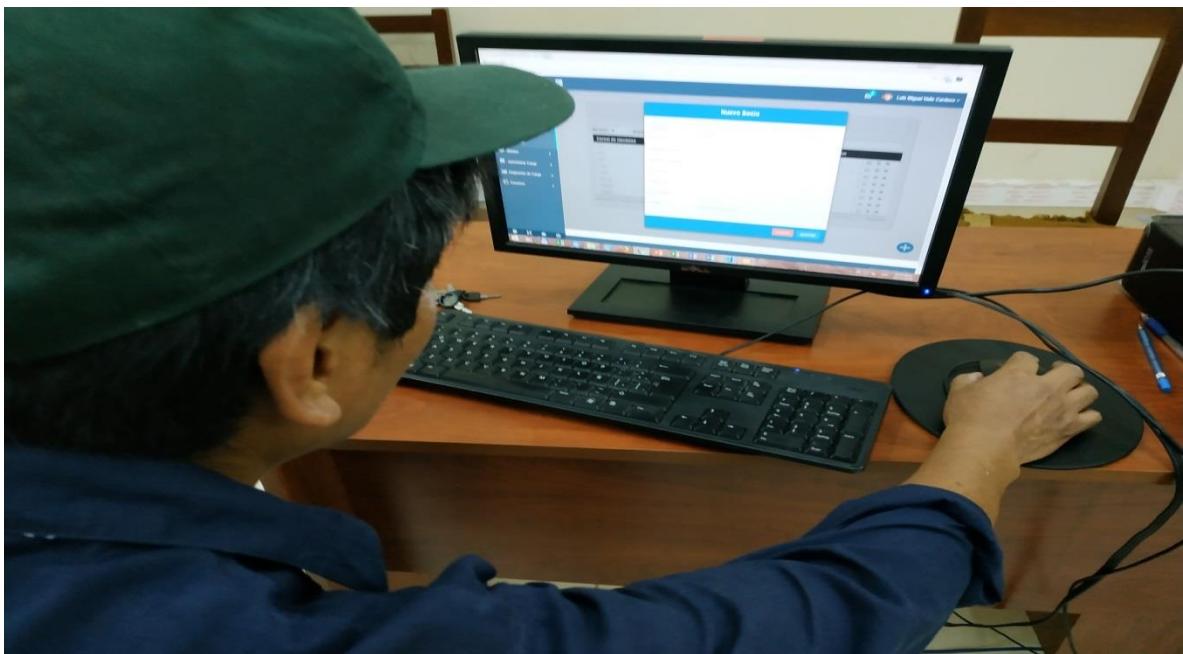


Figura 183 Proceso de la capacitación

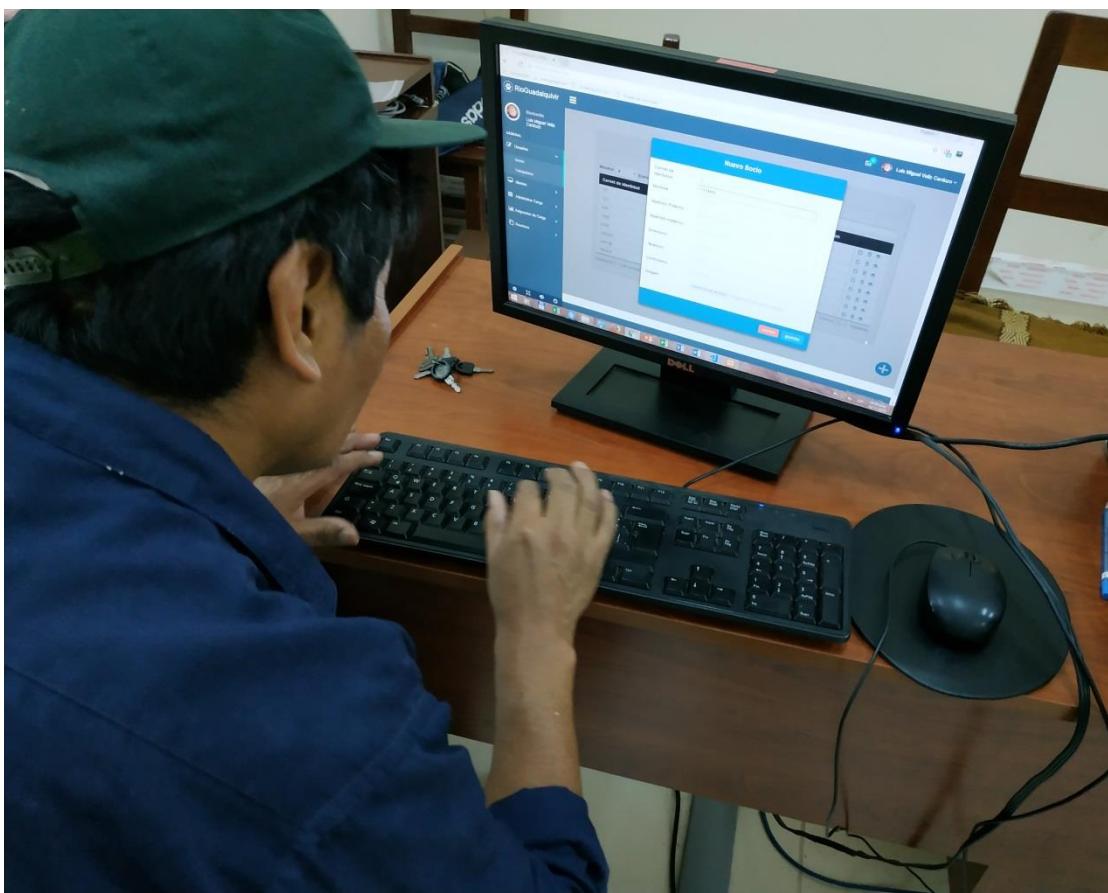


Figura 184 Proceso de la capacitación

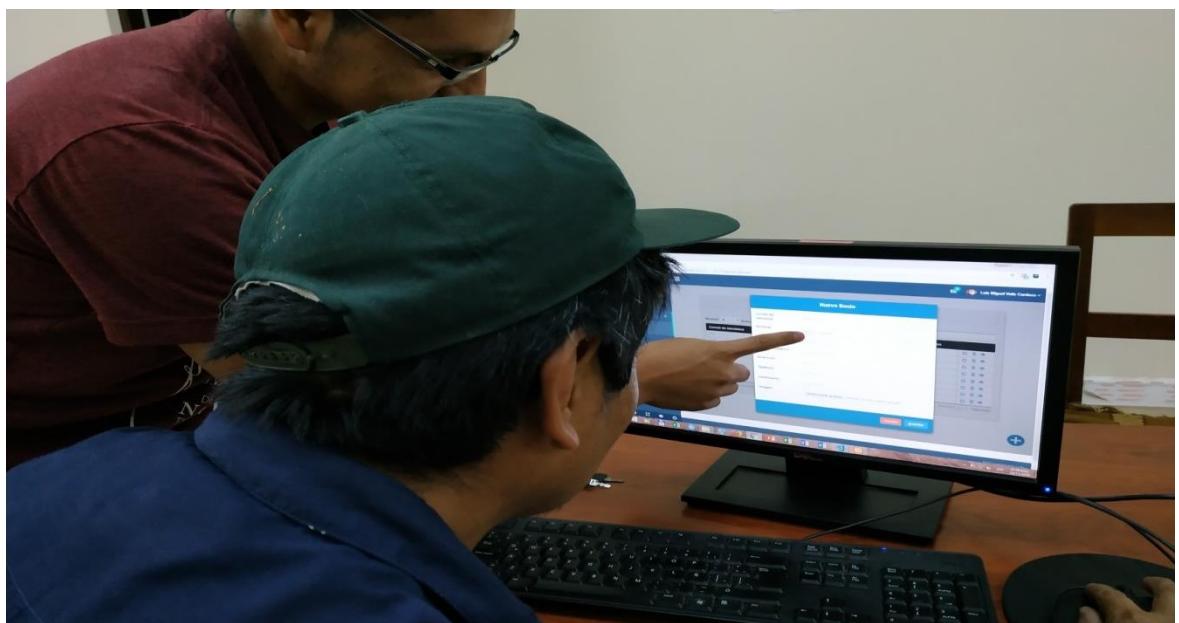


Figura 185 Proceso de la capacitación

## 1.7.11 Medios de verificación

### 1.7.11.1 Informe de realización de la capacitación



## **CAPÍTULO III**

## Capítulo III Conclusiones y recomendaciones

### 1. Conclusiones

La necesidad de incorporar el uso de sistemas y aplicaciones móviles en las instituciones y organizaciones es un hecho que no puede pasar desapercibido, así mismo el de realizar esta inserción, aplicación de manera responsable y siguiendo las normas propias a cada caso.

Siguiendo este concepto de responsabilidad informática, se ha detectado los siguientes aspectos importantes para este proyecto, primer aspecto, la intención de informatizar y realizar el “MEJORAMIENTO EN EL CONTROL Y ADMINISTRACION DE LA INFORMACION EN LA COOPERATIVA DE TRANSPORTE RIO GUADALQUIVIR.”, este propósito se logra a través de un sistema informático, aplicación móvil y un programa de capacitación efectivo. Segundo aspecto, tiene relación con la responsabilidad de relacionar las necesidades de los usuarios con la funcionalidad del sistema,. El tercer aspecto relevante en el proyecto, tiene que ver con el uso de lenguaje gráfico, metodologías, herramientas de desarrollo, los cuales ayudaron al desarrollo del sistema.

Luego de analizar la situación de la cooperativa de transporte Rio Guadalquivir se determinó

- El Sistema Informático ha sido desarrollado siguiendo la metodología de desarrollo RUP.
- El Manejo de una aplicación móvil brinda una información de los datos de la cooperativa dado que los socios evidenciaron mejoras.
- El sistema cumple con todas las características, dado que los socios de la cooperativa evidenciaron las mejoras

En conclusión el Sistema y la aplicación móvil cumplió las expectativas, del desarrollo del proyecto para “MEJORAMIENTO EN EL CONTROL Y ADMINISTRACION DE LA INFORMACION EN LA COOPERATIVA DE TRANSPORTE RIO GUADALQUIVIR.

## **2. Recomendaciones**

A la culminación del presente proyecto se presentan las siguientes recomendaciones

- Realizar mantenimiento al sistema informático.
- Utilizar nuevas tecnologías de desarrollo de Sistemas Web ya que hacen más rápido la realización de dicho sistema.
- Utilizar el manejo de control de versiones para mantener el respaldo del software.
- Implementar nuevos módulos que esto ayudara a que el control y administración de la información de la cooperativa fortalezca
- Establecer estrategias de seguridad que acompañen al sistema desarrollado.
- Finalmente debe acompañar siempre al desarrollo de software con una capacitación a sus futuros socios trabajadores, el restar importancia o no aplicar la capacitación hará que los socios trabajadores no puedan obtener información adecuada