

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO FUNCIONAL DE UN SISTEMA DE COMUNICACIÓN APLICADO AL CARPOOLING EN LAS UNIVERSIDADES DEL VALLE DE ABURRÁ.

JUAN DIEGO PÉREZ MARTÍNEZ

HENRY ESTEBAN HERNÁNDEZ ARIAS

Trabajo de grado para optar al título de Ingeniero de Sistemas y Computación

JOHN EDGAR CONGOTE CALLE

Doctor en Ingeniería



**ESCUELA DE INGENIERÍA DE ANTIOQUIA
PROGRAMAS ACADÉMICOS EN LOS QUE SE ENMARCA
ENVIGADO
2015**

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

CONTENIDO

	pág.
INTRODUCCIÓN.....	ERROR! BOOKMARK NOT DEFINED.
1. PRELIMINARES.....	10
1.1 Planteamiento del problema	10
1.2 Objetivos del proyecto	10
1.2.1 Objetivo General.....	10
1.2.2 Objetivos Específicos	10
1.3 Marco de referencia.....	11
1.4 Hipótesis (opcional)	11
2. METODOLOGÍA.....	16
3. DESARROLLO DEL PROYECTO	16
4. DISCUSIÓN DE RESULTADOS.....	40
5. CONCLUSIONES Y CONSIDERACIONES FINALES	41
BIBLIOGRAFÍA.....	42
ANEXO 1.....	42

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

LISTA DE TABLAS

	pág.
Tabla 1. Ventajas y funcionalidades de la utilización de frameworks.	15
Tabla 2. Ficha registrar usuario.	20
Tabla 3. Ficha Autenticar usuario	21
Tabla 4. Ficha editar información de transporte.	21
Tabla 5. Ficha consultar información.	22

LISTA DE FIGURAS

	pág.
Figura 1. Diagrama de clases.	11
Figura 2. Diagrama de casos de uso.	14
Figura 3. Diagrama Casos de uso carpooling universitario.	19
Figura 4. Diagrama entidad relación.	23
Figura 5. Diagrama de clases.	24
Figura 6. Diagrama de secuencia - Registro Usuario.	25
Figura 7. Diagrama de secuencia - Actualizar Usuario.	26
Figura 8. Diagrama de secuencia - Eliminar Usuario.	27
Figura 9. Diagrama de secuencia - Insertar Ruta.	28
Figura 10. Diagrama de secuencia - Eliminar Ruta.	29
Figura 11. Diagrama de secuencia - Aplicar Ruta	30
Figura 12. Diagrama de flujo.	32
Figura 13. Formulario de registro	33
Figura 14. Login	34
Figura 15. Página principal del sistema	35
Figura 16. Administración de usuarios.	36
Figura 17. Creación de rutas.	37
Figura 18. Administración de rutas.	38
Figura 19. Detalle de ruta.	39

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

RESUMEN

El carpooling es una práctica que consiste en compartir automóviles entre personas del común que tienen trayectos similares, se aplica con el fin de reducir la congestión de tránsito en las vías. El objetivo de este proyecto es diseñar e implementar un sistema informático que permita mejorar el funcionamiento del carpooling en las universidades de del valle de Aburrá, el cual permitiría ofrecer un canal de comunicación mantenible y controlado, proveyendo una manera organizada de la aplicación de carpooling. Se seguirá una metodología scrum para la realización tanto de la documentación como de la implementación del sistema, la cual permitirá tener avances cortos, pero con valor funcional.

Se espera obtener una documentación conformada por el análisis y diseño del sistema, en el cual se definirá de forma clara el alcance, características y funcionamiento del mismo. También se espera implementar un prototipo funcional en un ambiente controlado, el cual proveerá un acercamiento de cómo funcionaría el sistema en un ambiente real.

Palabras clave: Carpooling, transporte, comunidad universitaria, tránsito.

ABSTRACT

Carpooling is a method used by the people to share vehicles for common destinies, and is aimed to reduce the road traffic. The goal of this Project is to design and implement computer system that allow to improve the performance of carpooling in universities from Aburra valley, providing a suitable communication channel. A scrum methodology will be used for the development and documentation of the project, with short progress but highly valuable.

It is expected to obtain a documentation conformed by analysis and design, with a restricted scope and functionalities, plus a functional prototype in a control environment, that will provide an approach of a real system performance.

Key words: Carpooling, transportation, university community, traffic.

INTRODUCCIÓN

Los sistemas de transporte son un elemento muy importante dentro de una sociedad, que dan solución a las necesidades de movilidad, pero que a su vez con el aumento en tamaño y complejidad de estos, se han presentado varias dificultades, lo cual ha hecho de este un tema crítico de estudio.

En las grandes ciudades alrededor del mundo uno de los problemas más significativos, gira entorno a la movilidad, ya que con la sobrepoblación de vehículos que se ha evidenciado en los últimos años se ha generado una tendencia a la inmovilidad, lo que ha convertido en una problemática lo que en un principio fue planteado como solución.

Uno de los factores que causan problemas de movilidad, es el hecho de que muchas personas no utilizan la capacidad máxima de pasajeros por vehículo, lo que incrementa la congestión en el sistema vial. El caso particular de las universidades del valle de aburra no es ajeno a esta situación, ya que gran cantidad de miembros de la comunidad educativa se movilizan para las actividades académicas, sin hacer uso completo de la capacidad de pasajeros de sus vehículos.

La disponibilidad de un medio de comunicación en el cual se permita compartir información de rutas y horarios, permitiría disminuir el uso ineficiente de vehículos en cuanto al volumen de tripulantes, lo cual contribuye de manera positiva a la reducción de esta situación, permitiendo utilizar el concepto de carpooling.

Este documento desarrollará el análisis, diseño e implementación de un sistema con las características previamente descritas.

1. PRELIMINARES

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad la mayoría de las universidades del valle de aburra no cuentan con un sistema de comunicación que proporcione la posibilidad de tener acceso a información relevante relacionada con el tema de transporte, tal como el camino que un individuo toma para llegar a las instalaciones universitarias, los horarios en que realiza los desplazamientos, la capacidad de personas que puede transportar, entre otros factores de interés. El desarrollo e implementación de una plataforma que provea funcionalidad de acceder a la información anteriormente mencionada, traería beneficios a la comunidad educativa.

Para una persona que no cuenta con un medio de transporte propio, se brindaría la posibilidad identificar una persona que tenga el potencial de prestar un servicio de transporte. Un individuo que utilice el sistema de transporte público podría acceder a un medio que se ajusta de manera más directa a sus necesidades de desplazamiento, reduciendo el tiempo de viaje. En el ámbito social se posibilitarían las interacciones entre la población universitaria, ayudando a crear relaciones interpersonales, que aportan al desarrollo académico y personal de cada individuo.

A medida que las personas compartan sus medios de transporte particulares, se reduciría el flujo vehicular en las vías de acceso y al interior de las instalaciones universitarias, lo que a su vez disminuye el tiempo de ingreso en horas de alta congestión vehicular. Otro aspecto importante, en el cual se beneficiaría la universidad, sería el aumento de disponibilidad de espacios de estacionamiento, debido a que el número de vehículos decrecería.

La medida del pico y placa en la ciudad de Medellín, limita que algunas personas puedan utilizar su vehículo todos los días, así la opción de utilizar carpooling es una alternativa de transporte para este tipo de personas.

Por los motivos mencionados anteriormente, se evidencia la importancia de realizar una investigación en la manera de analizar e implementar una plataforma de comunicación, con las características previamente expuestas.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Diseñar e implementar un sistema para almacenar, modificar y consultar información generada por miembros de la comunidad universitaria, correspondiente a datos de rutas

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

de desplazamiento, horarios y capacidad de transporte para generar oportunidades de transporte, utilizando la estrategia de carpooling.

1.2.2 Objetivos Específicos

- Identificar la información relevante para el desarrollo de un sistema de comunicación que aporte a la generación de oportunidades de transporte.
- Realizar el levantamiento de los requerimientos principales del sistema.
- Realizar el análisis y diseño del sistema de almacenamiento de información, de acuerdo con los requerimientos obtenidos a partir del segundo objetivo.
- Implementar un prototipo funcional del sistema, que proporcione un acercamiento práctico de su comportamiento.

1.3 MARCO DE REFERENCIA

Arquitectura de software

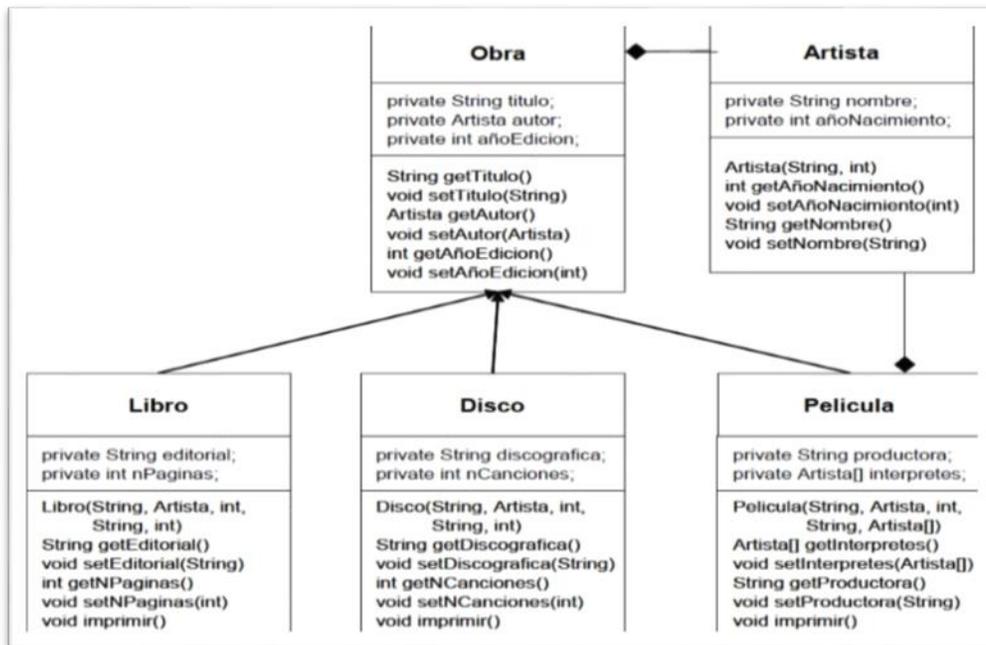
Existen múltiples definiciones del término arquitectura de software, y debido al gran número de estas no existe una definición que este respaldada unánimemente por el grupo de arquitectos de software en su totalidad. Por su parte Edgar Gómez Reyes (Desarrollador de Proyectos, Aspec System, Veracruz, Mexico) define la arquitectura de software como el equipamiento lógico y la arquitectura de los servicios que se desarrollan para ofrecer una vida más fácil y cómoda para la humanidad. Los desarrolladores "Trabajamos en lo difícil para que usted lo haga fácil" (Usuario, 2006).

Una definición es: "La Arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones".(Reynoso, 2004).

Modelamiento y requisitos

El levantamiento de requisitos es una etapa de gran importancia dentro de un proyecto de implementación de software, ya que en este proceso se determina la funcionalidad que el software debe tener el sistema que responda satisfactoriamente a los requerimientos del cliente y del entorno en el cual se realizará la implementación y posterior despliegue del software. Para realizar este proceso se han creado diferentes artefactos los cuales aportan a la identificación de los requisitos y permiten un modelamiento de los mismos (Usuario, 2006) . Entre estos se encuentran:

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.



- Diagrama de clases:** Antes de definir que es un diagrama de clases, es conveniente definir que es una clase, la cual tiene por definición el conjunto de objetos que tienen en común características, atributos, métodos, operaciones y comportamientos; al crear una clase se hace una abstracción, en el que se agrupan elementos similares entre sí, lo que hace que su posterior utilización dentro del sistema se más modular, y conserve sus principales características al encapsular sus atributos (Aguilar, n.d.).

Al tener claridad sobre lo que es una clase, se puede decir que Los diagramas de clase describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. Los diagramas de clase se convierten así en la técnica más potente para el modelado conceptual de un sistema software, la cual suele recoger los conceptos clave del modelo de objetos subyacente al método orientado a objetos que la incorpora. El propósito de un diagrama de clase es describir las clases que conforman el modelo de un determinado sistema. Dado el carácter de refinamiento iterativo que caracteriza un desarrollo orientado a objetos, el diagrama de clase va a ser creado y refinado durante las fases de análisis y diseño, estando presente como guía en la implementación del sistema (Aguilar, n.d.).

El siguiente es un ejemplo de diagrama de clases:

Figura 1. Diagrama de clases.

- **Diagrama de casos de uso:** Un caso de uso representa una unidad funcional coherente de un sistema, subsistema o clase. En un caso de uso uno o más actores interactúan con el sistema que realiza algunas acciones (Aguilar, n.d.).

Los elementos de un modelo de casos de uso son:

- Actores: Es cualquier agente externo que interactúa con el sistema, y realiza un intercambio de datos; suelen tener roles dentro del funcionamiento del sistema, los cuales son importantes definirlos para tener claridad de responsabilidades (Aguilar, n.d.).
- Casos de uso: Es cada uno de los escenarios de funcionalidad que provee el sistema, en su interacción ya sea con el cliente o con un sistema externo. Se suelen definir elementos diferenciadores de cada caso de uso, tales como, descripción, escenario, requerimientos, restricciones, y excepciones (Aguilar, n.d.).
- Relaciones: En este componente se define el tipo de comunicación que se da entre el sistema y/o un agente externo, además de las responsabilidades de cada actor dentro del caso de uso (Aguilar, n.d.).

A continuación un ejemplo de un diagrama de casos de uso:

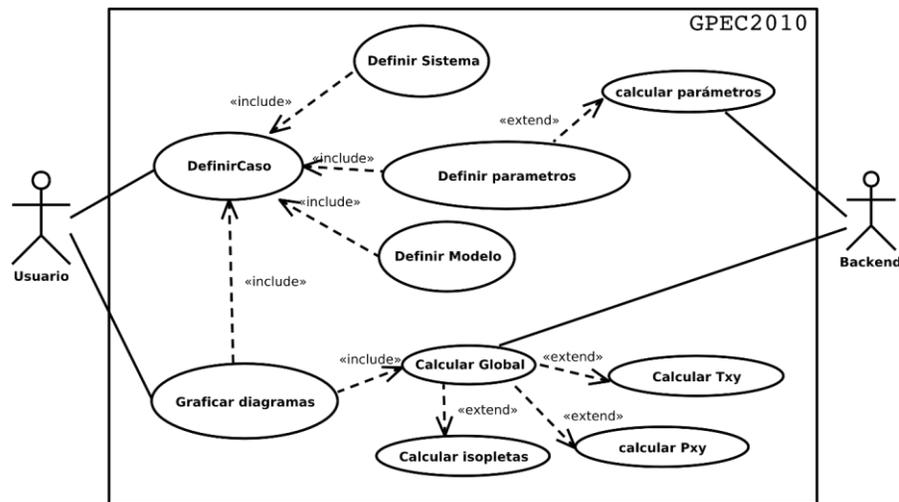


Figura 2. Diagrama de casos de uso.

Framework

El concepto de *framework* es muy utilizado en las implementaciones de software, no únicamente en desarrollos que involucran la web. Los *frameworks* emplean una estructura robusta en la que ya se tiene una funcionalidad predeterminada, y pueden ser adecuados a una situación específica (Gutiérrez, 2006)..

En general, el término *framework*, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación, este también se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Las principales ventajas que ofrecen los *frameworks* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un *framework* Web, se puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. (Gutiérrez, 2006).

Uno de los *frameworks* más importantes del desarrollo web es el *framework* MVC (*Model, View, Controller*) el cual se definirá a continuación.

Modelo MVC

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema (Gutiérrez, 2006).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

El modelo MVC es un patrón de arquitectura de las aplicaciones software que separa la lógica de negocio de la interfaz de usuario, facilitando así la evolución por separado de ambos aspectos y a su vez incrementa reutilización y flexibilidad (Mvc, 2008).

- **Modelo:** El modelo se refiere a las reglas del negocio, conoce todos los detalles acerca de la organización. También tiene información acerca de la base de datos (Mvc, 2008).
- **Vista:** La vista se refiere a todo lo relacionado, con las páginas html del sistema, es la parte que es visible para el cliente, no tiene nada que ver con las reglas del negocio, simplemente se encarga de recibir y solicitar información, para luego permitir que sea visualizada por el cliente, en una interfaz amigable (Mvc, 2008).
- **Controlador:** Funciona como intermediario entre el modelo y la vista, es un puente de comunicación en la cual los elementos de *Frontend* pueden interactuar con los de *Backend*, sin que se violen los protocolos de responsabilidades. Tiene información acerca de la organización y sabe en detalle como es la distribución de la información (Mvc, 2008).

A continuación se presenta una lista de las funcionales principales que ofrece un Framework en la implementación web:

<i>Abstracción de URLs y sesiones.</i>	No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
<i>Acceso a datos.</i>	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc..
<i>Controladores.</i>	La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
<i>Autenticación y control</i>	Incluyen mecanismos para la identificación de usuarios

Tabla 1. Ventajas y funcionalidades de la utilización de *frameworks*.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

2. METODOLOGÍA

Para la realización de este proyecto se va a emplear SCRUM, que es un marco de trabajo para la gestión y desarrollo de proyectos. Una de las características de este marco es que define la realización de las actividades en un plazo máximo de 2 a 4 semanas.

A continuación se muestra el procedimiento que se llevara a cabo para cumplir los objetivos específicos:

Objetivo 1: Se estudiarán diferentes aplicaciones de *carpooling* y se analizarán los elementos que tengan en común. Posteriormente a este análisis se seleccionará y se documentará la información hallada que sea relevante para el sistema.

Objetivo 2: Se utilizarán herramientas de modelación de requisitos, bajo el estándar UML, para realizar un diagrama de casos de uso, el cual permitirá identificar la funcionalidad del sistema, sus actores y escenarios.

Objetivo 3: Para la realización de los diagramas de entidad relación de clases, de secuencia y flujo de información se utilizará el lenguaje unificado de modelado, utilizando software de diseño especializado, que ofrecen todas las funcionales necesarias para la correcta implementación de tales diagramas.

Objetivo 4: Para la realización del prototipo funcional se utilizará el *framework* de Php Laravel, el cual provee una estructuración de modelo-vista-controlador, que permite tener un manejo modular del sistema, lo que aporta a la escalabilidad y mantenibilidad del mismo.

Resultados esperados: Se espera tener una documentación del sistema, conformada por el análisis y diseño de éste, con sus respectivos artefactos y diagramas, que dan claridad del funcionamiento de la aplicación. También se espera implementar un prototipo funcional, que proveerá un acercamiento de cómo funcionaría el sistema en un escenario real.

3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 IDENTIFICACIÓN DE LA INFORMACIÓN RELEVANTE A PARTIR DE ANTECEDENTES

En ésta sección se realiza un estudio de las diferentes aplicaciones de carpooling existentes en la región, para identificar la información que aporte valor al sistema. Primero se hace una descripción de algunas de estas aplicaciones para el posterior análisis.

3.1.1 ANTECEDENTES

- Uber es una aplicación móvil en la que se puede conseguir una persona que brinde un servicio de transporte hasta un lugar a partir de una dirección dada. Al ingresar a la aplicación, ésta da la posibilidad de registrarse como “conductor” o como usuario que puede solicitar transporte. Esta aplicación está disponible para Iphone o para Android y se encuentra disponible en 55 países dentro de los cuales está Colombia, en ciudades como Cali, Bogotá y Medellín, prestando servicio para conseguir transporte vía taxi. Vale la pena aclarar que Uber no es un proveedor de transporte, sino una plataforma que permite a taxistas y usuarios de taxi contactarse entre sí. En Medellín solo está disponible para usuarios de taxi y tiene unas tarifas básicas (fijas) ya establecidas (UBER, 2015). Los datos que solicita para registrarse son: ciudad, nombre, apellido, correo electrónico, celular, contraseña, y opcional un código de invitación.
- EASYTAXI es otra aplicación Móvil que permite como CITYTAXI acceder a un servicio de taxi desde el celular. Esta aplicación fue creada en Brasil pero ahora funciona en toda Latinoamérica (EASYTAXI, 2015). No requiere registro o es posible registrarse con facebook o la dirección de correo.
- Fuimonos es una aplicación que permite establecer un contacto entre conductores y pasajeros, como ellos les llaman, para que compartan el carro. Solo funciona para comunidades cerradas en Bogotá (FUIMONOS, 2015). La información que solicita para registrarse es: empresa, nombre, teléfono, correo y contraseña.
- Tripda es una aplicación que permite a usuarios que tienen auto particular, comunicarse con personas que necesitan viajar al mismo destino y que no tienen un vehículo disponible para hacerlo. Es una aplicación que conecta a la gente y permite que estas compartan información de ruta y destinos y que puedan llegar a un acuerdo por medio de una aplicación web, o también disponible en aplicación móvil para Android y IOS.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.1.2 SELECCIÓN DE INFORMACIÓN RELEVANTE

De acuerdo al análisis realizado de las aplicaciones anteriores y observando la información que estas requieren de los usuarios, se llega a la conclusión de que los datos relevantes, se dividen en dos grupos. La información necesaria para el registro de usuario común y otro que requiere datos extra en el caso de usuarios conductores.

- **Registro usuario común:** Son los usuarios que desean acceder a las funcionalidades básicas del sistema, como solicitudes y consultas. Los datos necesarios para este registro son:
 - ◆ Nombre.
 - ◆ Documento de Identidad.
 - ◆ Correo.
 - ◆ Universidad a la que pertenece.
 - ◆ Celular.
 - ◆ Contraseña.

- **Registro de conductores:** A demás de la información anterior, cuando un usuario desea publicar rutas y ofrecer viajes, se requiere la siguiente información adicional:
 - ◆ Placa del vehículo.
 - ◆ Ruta (La cuál será trazada en un mapa).
 - ◆ Capacidad en el vehículo.
 - ◆ Horario de ruta.
 - ◆ Ocupación.
 - ◆ Carrera.
 - ◆ Semestre (los dos campos anteriores, aplica únicamente para usuarios que son estudiantes).

- **Identificación de Zonas:** Se evaluaron diferentes opciones para la creación de rutas dentro del sistema:
 1. La creación de rutas mediante el posicionamiento de dos puntos en el mapa, origen y destino. Esta opción aunque es ágil y eficiente de desarrollar, no proporciona información valiosa al usuario del recorrido que la ruta tendrá, por lo cual esta opción fue descartada.
 2. En la segunda opción se resuelve agregar el trazo de la ruta mostrando todo el recorrido desde el punto inicial, hasta el punto de destino. La observación que se hizo en este punto fue la necesidad de crear una ruta flexible, la cual está sujeta a muchos factores, como disponibilidad de tramos de carretera, congestión, entre otros, por lo cual se requiere de un modelo que permita realizar desviaciones y variaciones con respecto a la ruta estándar.
 3. Luego de analizar las dos opciones anteriores, se decidió implementar la inclusión de zonas dentro de la creación de una ruta. Así un usuario al crear una ruta, puede especificar las zonas por las cuales transitará, según sus necesidades, preferencias y conocimiento de vías. Esto permite delimitar los tramos que un conductor está dispuesto a desviarse para recoger a un posible pasajero. Por lo anterior se decidió seguir este modelo para la creación de rutas.

3.2 LEVANTAMIENTO DE REQUISITOS DEL SISTEMA

Para el desarrollo de un sistema computacional, es importante definir las funcionalidades del sistema, los escenarios que se puedan presentar y los actores involucrados, con anterioridad, para tener claridad en la etapa de implementación. Uno de los artefactos más utilizados para realizar el levantamiento de requisitos es el diagrama de casos de uso (el cuál ha sido descrito en el capítulo 1.3 del marco de referencia).

A continuación se muestra el diagrama de casos de uso, con sus respectivas fichas técnicas:

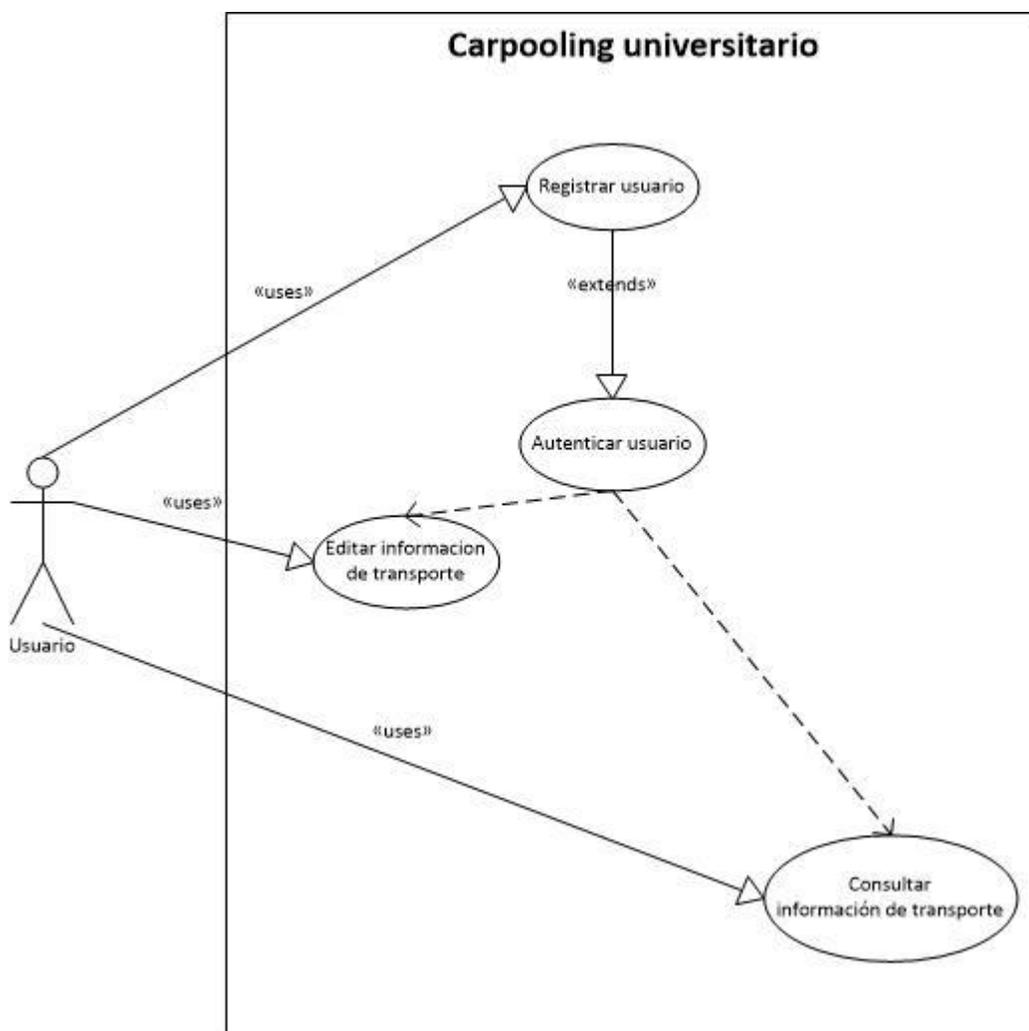


Figura 3. Diagrama Casos de uso carpooling universitario.

Nombre id	Autenticar usuario
Nombre Id	Autenticar Usuario
Descripción	El usuario desea Ingresar dentro de la plataforma web.
Requerimientos	El usuario debe haber ingresado al sitio web de la plataforma, debe suministrar información personal, como: Nombre usuario Nombre completo, correo electrónico, entre otros.
Restricciones	El usuario debe utilizar un nombre de usuario disponibles, dirección de correo electrónico habilitada, contraseña segura (mayúscula, al menos un número, al menos un carácter especial).
Escenarios	<ol style="list-style-type: none"> 1. El usuario escoge la opción "Registrarse". 2. La plataforma muestra el formulario de registro de usuarios. 3. Después de diligenciar correctamente el formulario, el usuario escoge la opción "Enviar". 4. La aplicación notifica que el proceso ha sido exitoso.
Excepciones	El usuario no diligencia correctamente el formulario. El nombre de usuario ya está en uso. El correo electrónico ya ha sido registrado.
Información adicional	Se envía un correo electrónico al usuario, notificando que el registro ha sido exitoso, con una petición de rectificación de cuenta.

Tabla 2. Ficha registrar usuario.

	plataforma web.
Requerimientos	El usuario se debe haber registrado correctamente dentro de la plataforma y haber rectificado su cuenta.
Restricciones	El usuario debe ingresar el nombre de usuario y su contraseña correctamente.
Escenarios	<ol style="list-style-type: none"> 1. El usuario escoge la opción "Ingresar". 2. La plataforma muestra la página de Login del sistema. 3. El usuario escoge la opción "Enviar". 4. La plataforma muestra la página principal del sistema.
Excepciones	El usuario ingresa el nombre de usuario o contraseña incorrectamente.
Información adicional	Se ofrecen opciones de recuperar contraseña y nombre de usuario.

Tabla 3. Ficha Autenticar usuario.

Tabla 4. Ficha editar información de transporte.

Nombre id	Editar información de transporte
Descripción	El usuario desea suministrar la información concerniente al transporte que puede ofrecer.
Requerimientos	El usuario se debe haber autenticado correctamente dentro de la plataforma.
Restricciones	El usuario debe ingresar la información correctamente en el formato requerido.
Escenarios	<ol style="list-style-type: none"> 1. El usuario escoge la opción "Editar información de transporte". 2. La plataforma muestra el formulario de información de transporte. 3. Después de diligenciar correctamente el formulario, el usuario escoge la opción "Enviar". 4. La aplicación notifica que el proceso ha sido exitoso.
Excepciones	El usuario no diligencia correctamente el formulario. Existen algunos campos obligatorios que quedan vacíos.
Información adicional	Se realizan validaciones de cada uno de los campos del formulario.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

Tabla 5. Ficha consultar información.

Nombre id	Consultar información de transporte
Descripción	El usuario desea consultar la información de otros usuarios con el fin de encontrar una oportunidad de transporte.
Requerimientos	El usuario se debe haber autenticado correctamente dentro de la plataforma.
Restricciones	El usuario debe tener permisos de consulta de información de transporte.
Escenarios	<ol style="list-style-type: none"> 1. El usuario escoge la opción “Consultar información de transporte”. 2. La plataforma muestra la página de consulta de información. 3. Por medio de filtros el usuario determina la información que necesita. 4. Se muestran los resultados de la búsqueda.
Excepciones	No existe información disponible, que esté relacionada con los criterios de búsqueda.
Información adicional	Algunos de los datos importantes son: Horarios, rutas y capacidad disponible de transporte del vehículo.

3.3 DISEÑO CONCEPTUAL DEL SISTEMA

Ya definido cada una de las funcionalidades del sistema, con alcances determinados y los diferentes escenarios que se pueden prestar, así como los actores involucrados y las responsabilidades de respuesta, es necesario expresar cómo el sistema proporcionará tales funcionalidades. Para este proceso existen diferentes artefactos comprendidos dentro del estándar UML, entre los cuales se destacan, diagrama de entidad relación, diagrama de clases, diagrama de secuencia y diagrama de flujo entre otros.

A continuación se desarrollarán los diagramas anteriormente enunciados, para definir el diseño del sistema.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.1 DIAGRAMA ENTIDAD RELACIÓN

El diagrama entidad relación describe la estructura básica de los datos, sus características y relaciones.

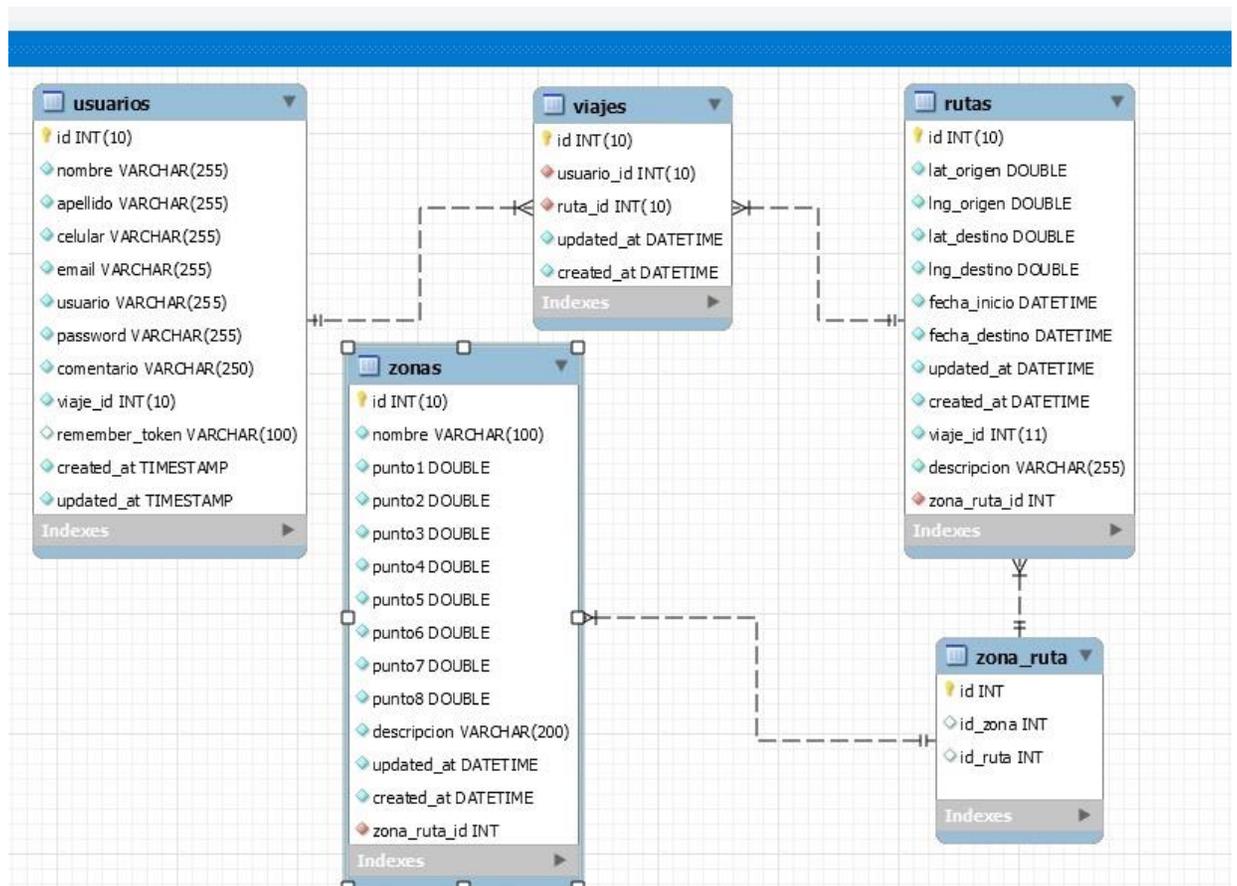


Figura 4. Diagrama entidad relación.

3.3.2 DIAGRAMA DE CLASES

El diagrama de clases es un artefacto en el que se definen los atributos y métodos de cada una de las clases, así como las relaciones entre ellas.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

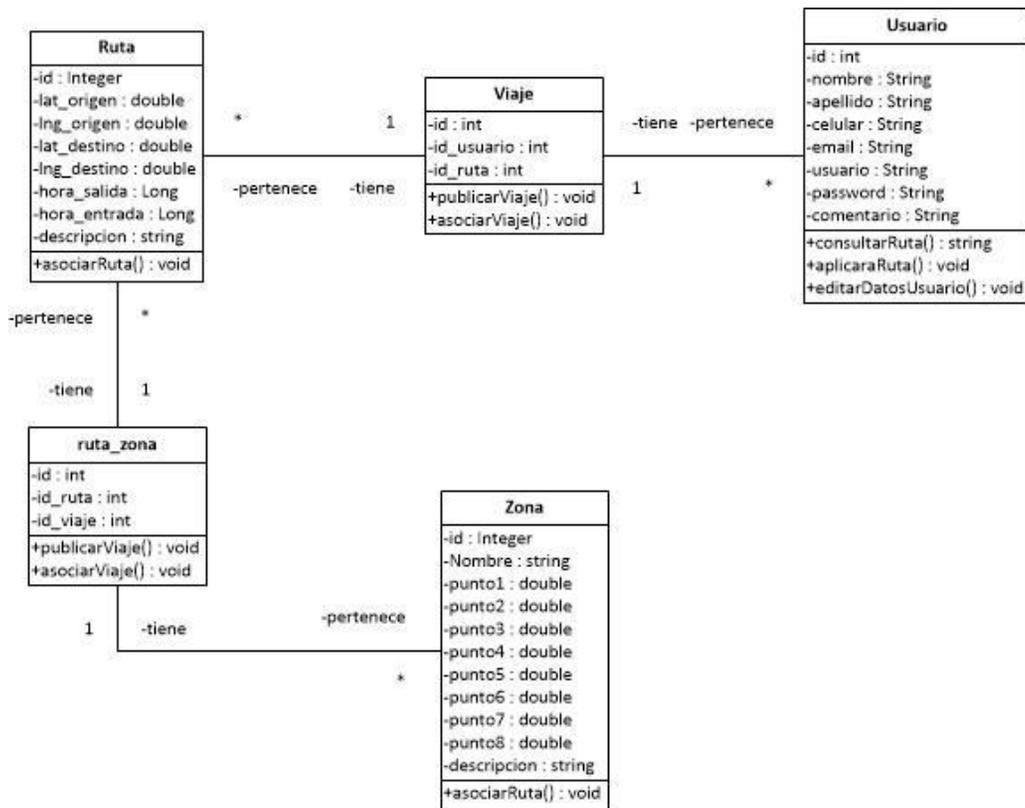


Figura 5. Diagrama de clases.

3.3.3 DIAGRAMA DE SECUENCIA

Define la interacción de los diferentes objetos del sistema a través del tiempo, determinando sus responsabilidades y respuestas. Se realiza un diagrama por cada funcionalidad del sistema. (En este caso se realizaron las funcionalidades más básicas).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.3.1 REGISTRAR USUARIO

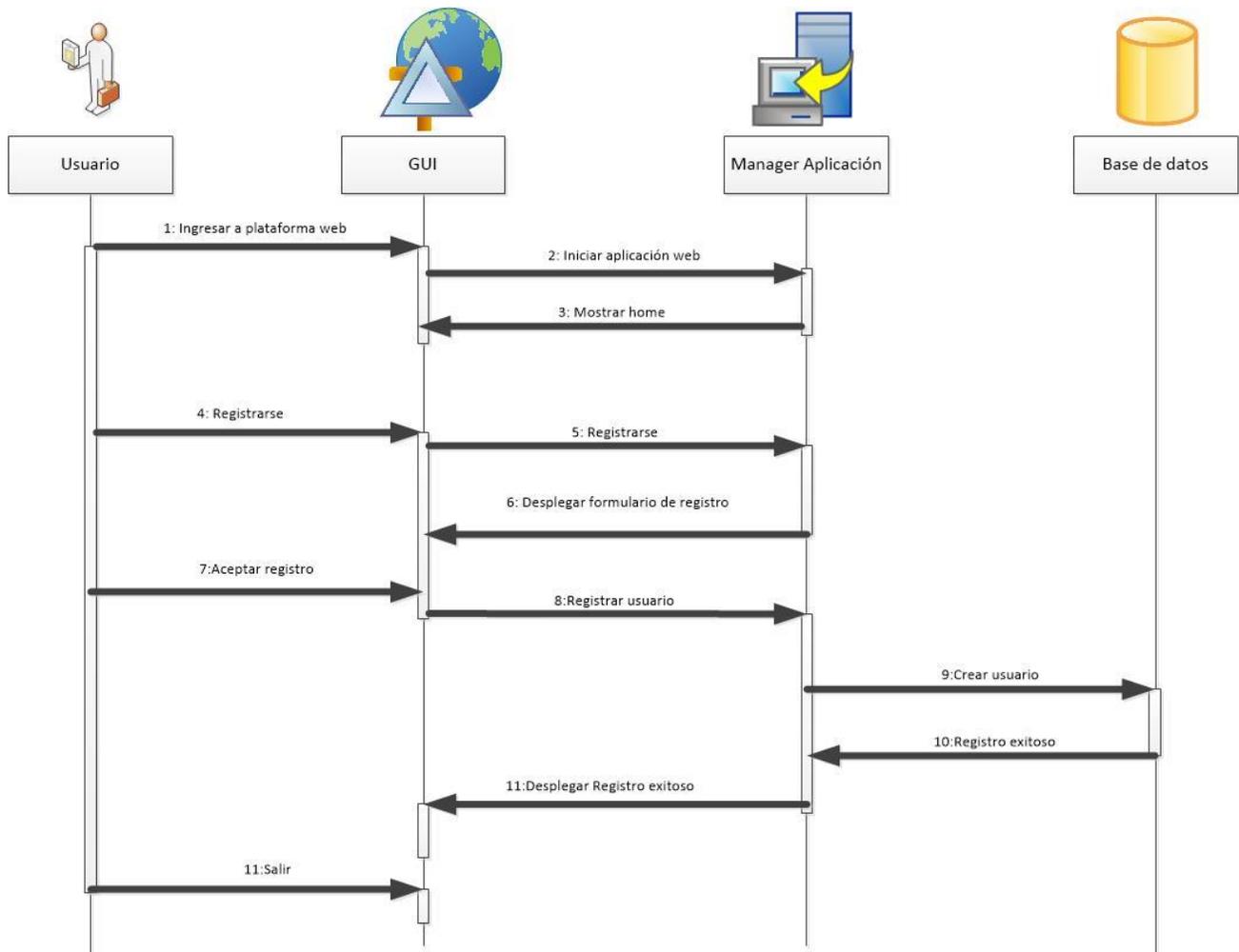


Figura 6. Diagrama de secuencia - Registro Usuario.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.3.2 ACUALIZAR USUARIO

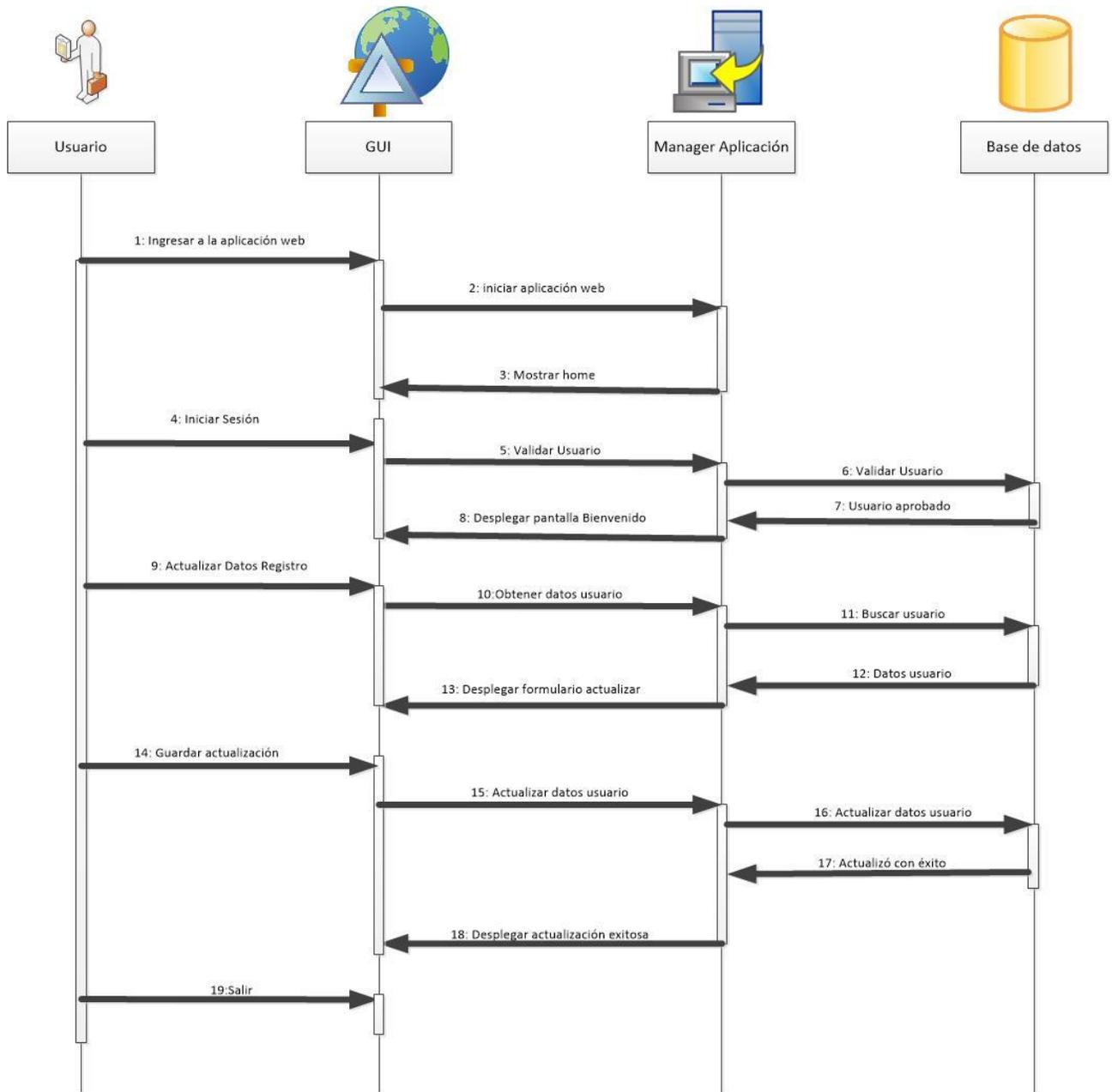


Figura 7. Diagrama de secuencia - Actualizar Usuario.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.3.3 ELIMINAR USUARIO

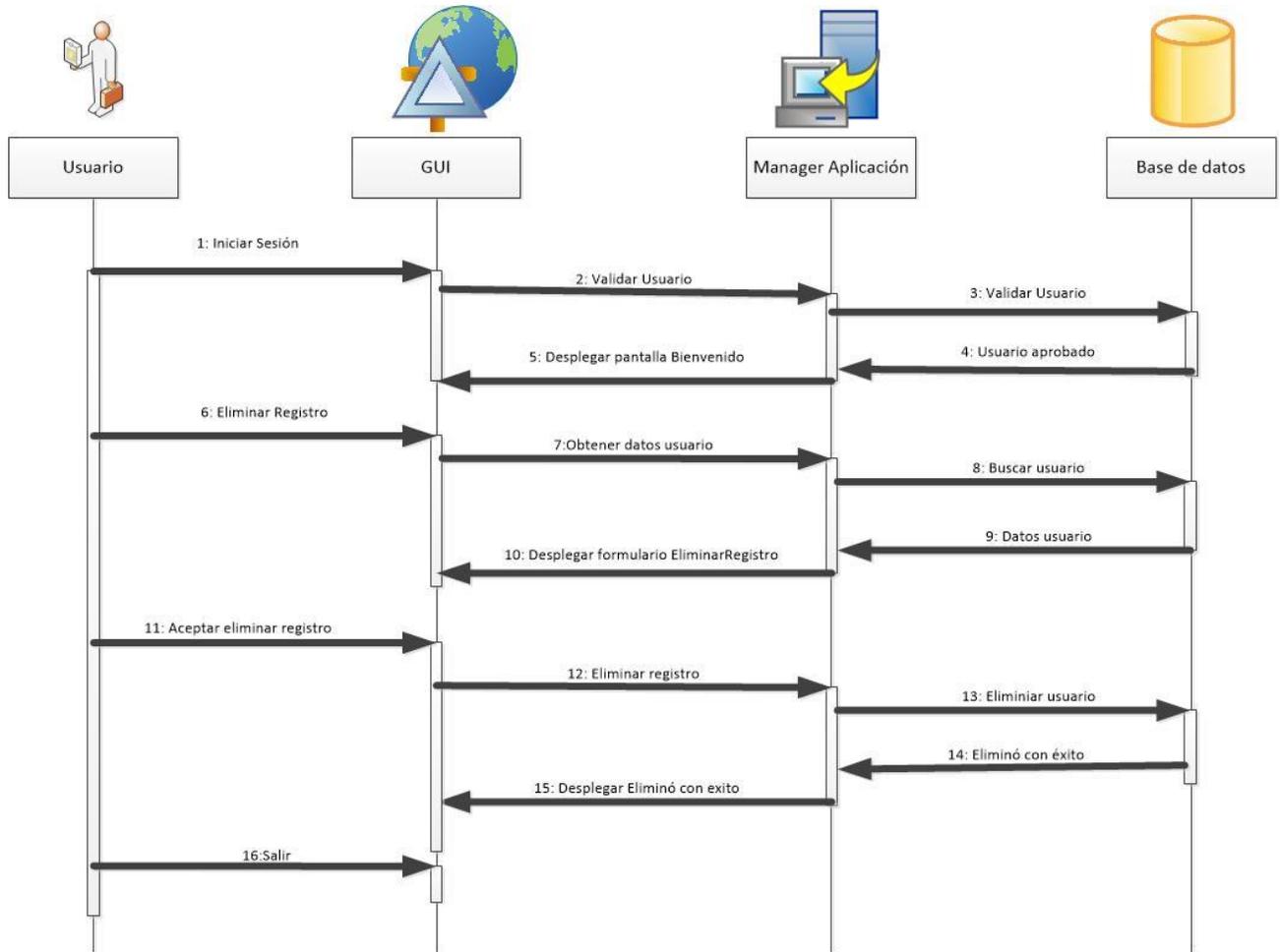


Figura 8. Diagrama de secuencia - Eliminar Usuario.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.3.4 INSERTAR RUTA

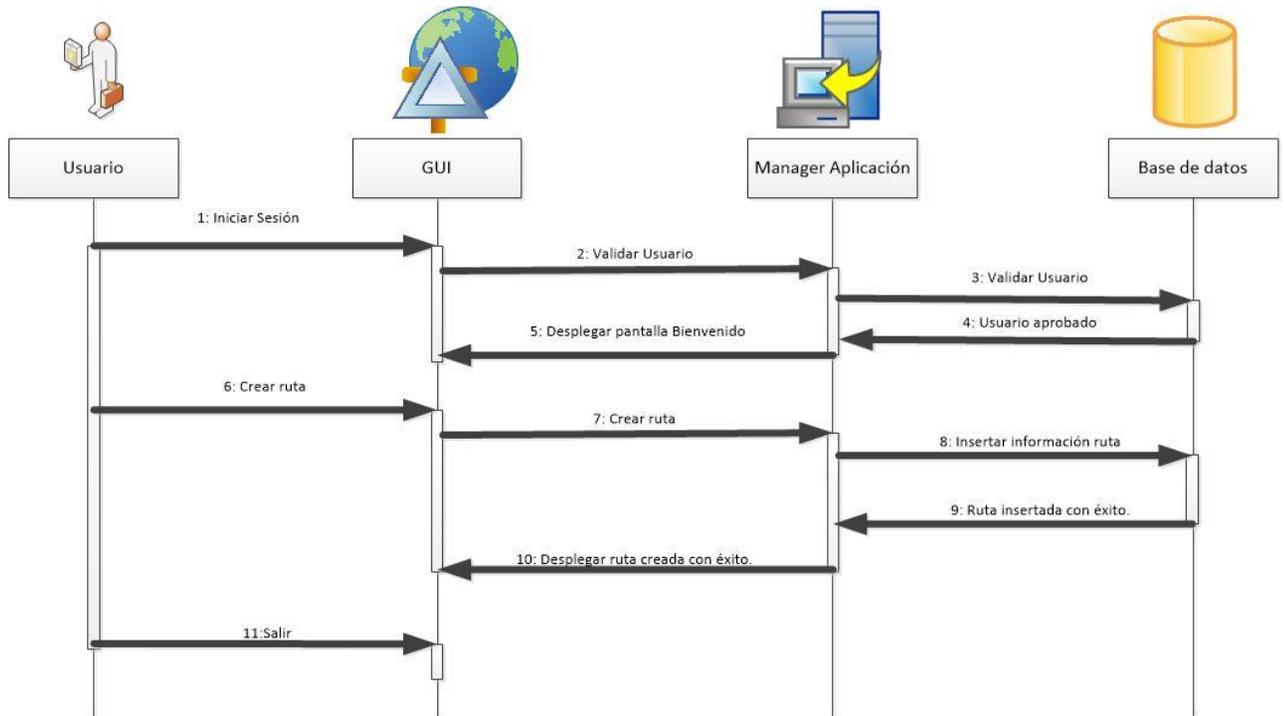


Figura 9. Diagrama de secuencia - Insertar Ruta.

3.3.3.5 ELIMINAR RUTA

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

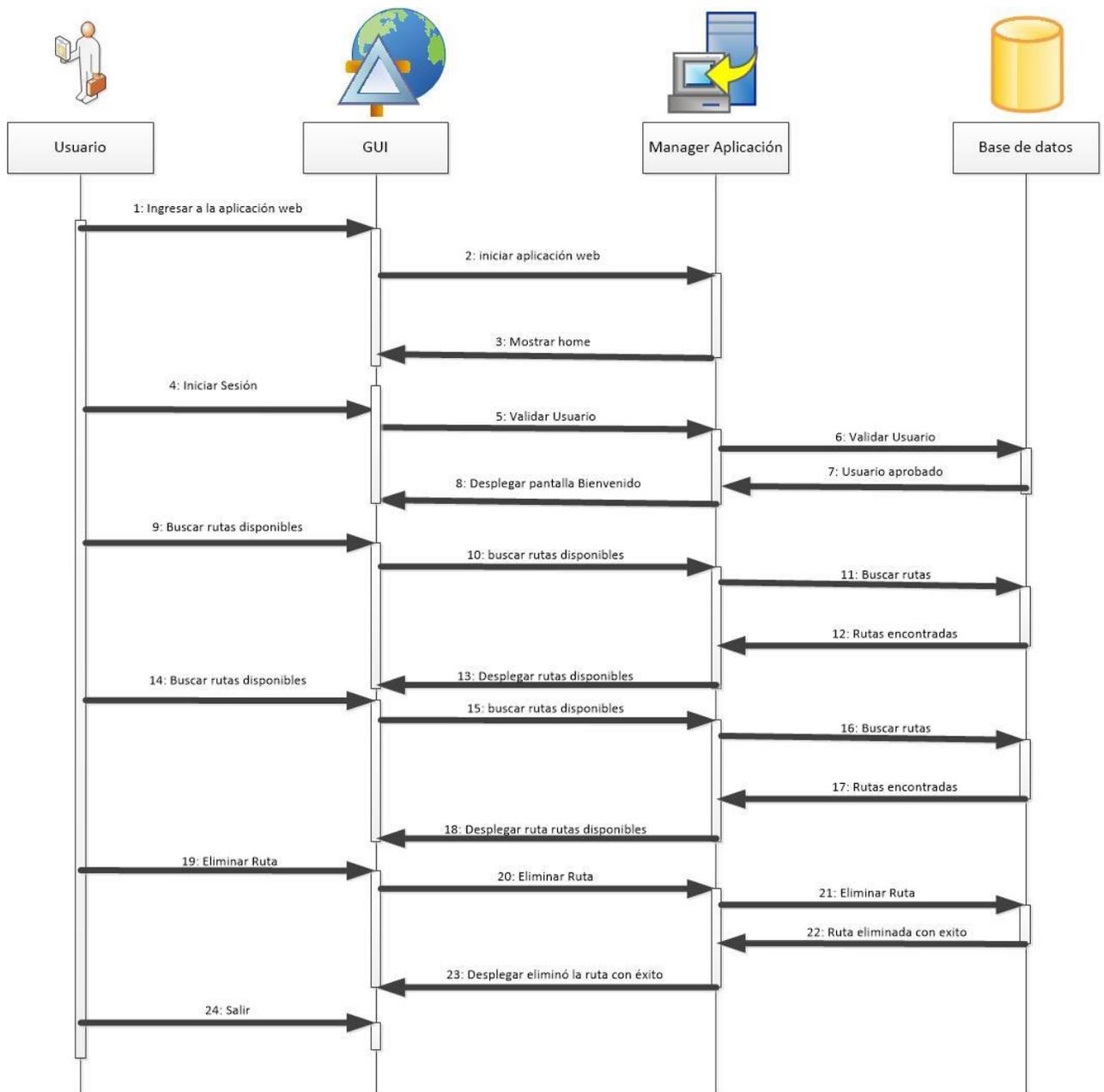


Figura 10. Diagrama de secuencia - Eliminar Ruta.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.3.6 APLICAR A RUTA

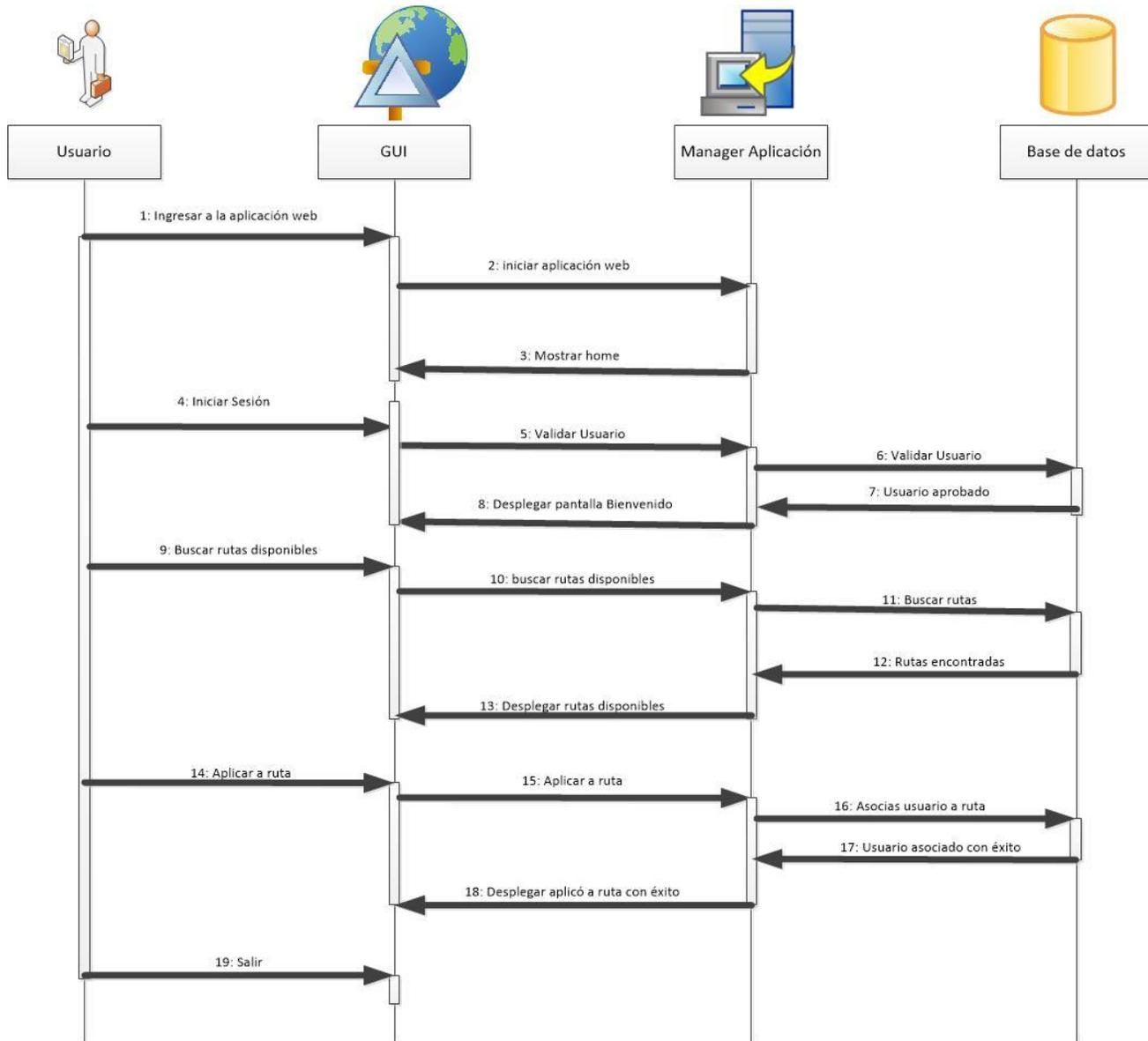


Figura 11. Diagrama de secuencia - Aplicar Ruta

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.3.4 DIAGRAMA DE FLUJO

Es la representación gráfica del proceso del sistema, se puede aceptar el flujo de los datos en el tiempo.

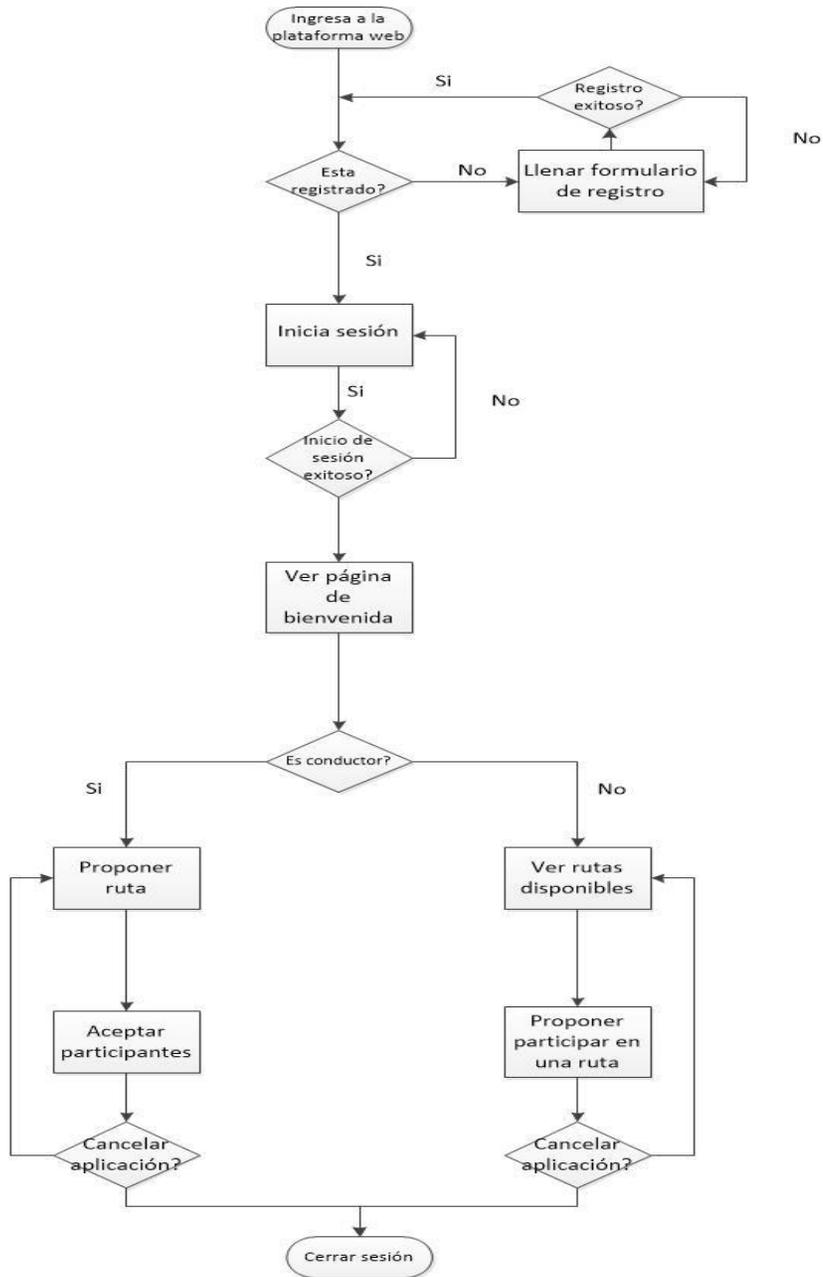


Figura 13. Diagrama de flujo.

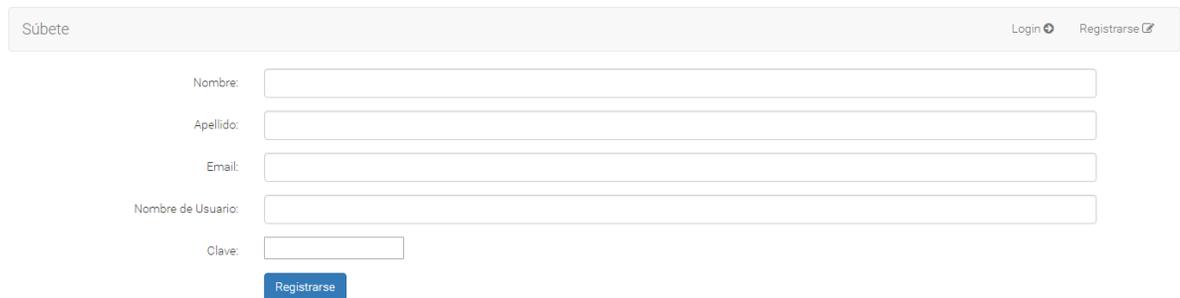
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.4 PROTOTIPO FUNCIONAL DEL SISTEMA DE CARPOOLING

En esta sección se presentan los escenarios de navegación y el recorrido de los módulos que componen el sistema de carpooling. Para mayor comprensión se mostrarán las imágenes de cada funcionalidad.

3.4.1 Registro de Usuario

Para el registro de un usuario en el sistema se ingresa la información básica, la cual incluye el correo electrónico y la contraseña, los cuales serán las credenciales del usuario para ingresos posteriores. A continuación se muestra una imagen del formulario de registro.



The image shows a user registration form. At the top left, there is a link labeled "Súbete". At the top right, there are links for "Login" and "Registrarse". The form contains five input fields: "Nombre:", "Apellido:", "Email:", "Nombre de Usuario:", and "Clave:". Below the "Clave:" field is a blue button labeled "Registrarse".

Figura 13. Formulario de registro

3.4.2 Ingreso al sistema

Una vez se realiza el proceso de registro, es posible ingresar al sistema utilizando el correo electrónico y la contraseña como credenciales. Se muestra una imagen del login.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

Súbete

Login Registrarse

Login

E-Mail Address esteban@saion.com

Password

Remember Me

Login

Figura 14. Login

3.4.3 Home del sistema

Al iniciar sesión en el sistema se redirige a la página principal desde donde se puede acceder a cada una de las funcionalidades de administración de usuarios. La siguiente es la página principal del sistema.

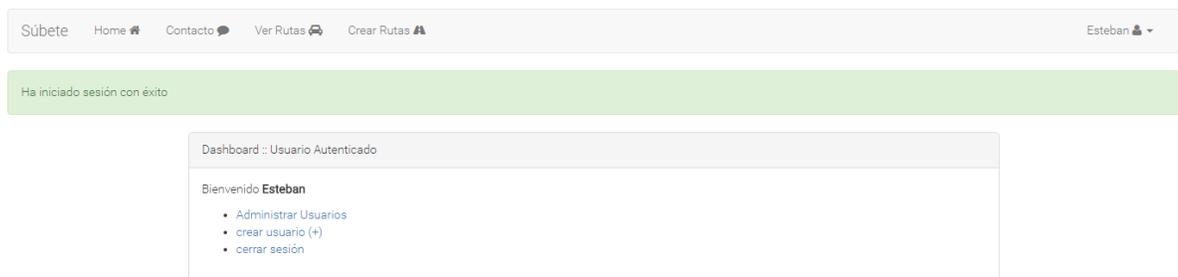


Figura 15. Página principal del sistema

3.4.4 Administración de Usuarios

Para la gestión de usuarios se ofrece muestra una lista de todos los usuarios registrados, y permite ver su información detallada, editar y eliminar. La siguiente imagen permite ver las opciones de administración de usuarios.

Súbete Home Contacto Ver Rutas Crear Rutas Esteban

Usuarios

crear (+)

Id	Nombre	Apellido	Email	Username	Creado en	Editar	Ver detalle	Eliminar
1	Esteban	Hernandez	esteban@saion.com	esteban	2016-05-06 21:50:40	editar	ver detalle	eliminar
2	Juan	Perez	Juan@Perez.com	Jenacho	2016-05-06 21:56:11	editar	ver detalle	eliminar
3	John	Orozco	jhornorozcoordona@gmail.com	JohnOrozco	2016-05-06 23:52:47	editar	ver detalle	eliminar

Figura 16. Administración de usuarios.

3.4.5 Creación de rutas

Para la publicación de rutas se cuenta con un módulo en él se ingresa la información respectiva, con la posibilidad de visualización en el mapa. La siguiente imagen ilustra en detalle los campos requeridos para la creación de una ruta.

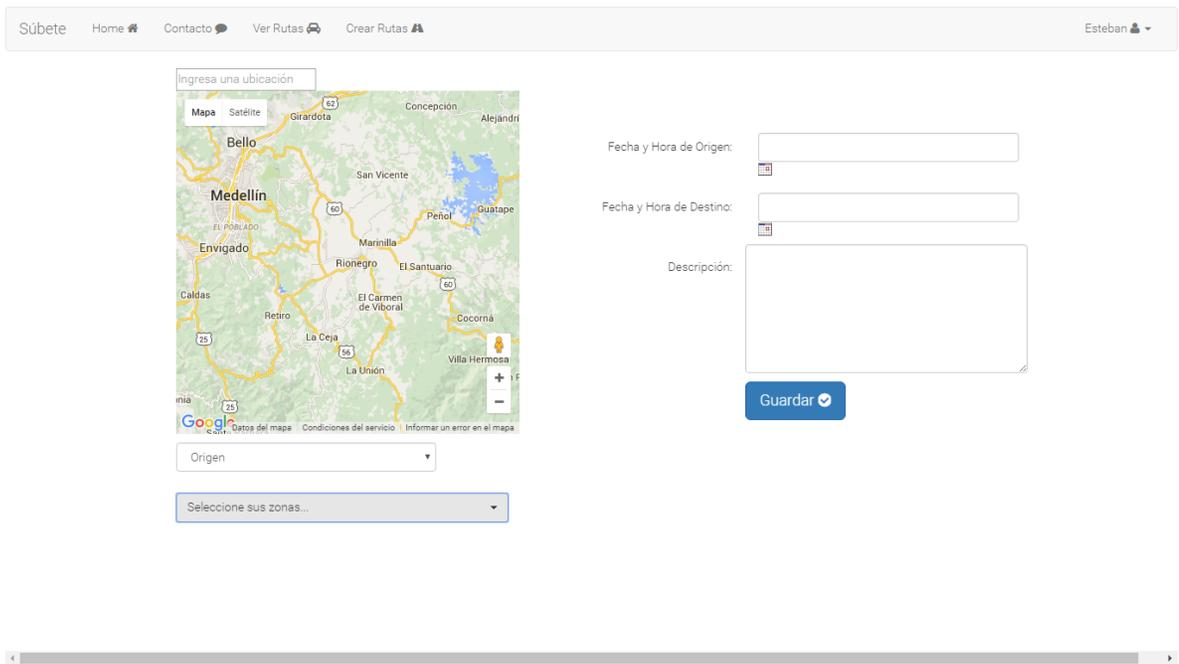


Figura 17. Creación de rutas.

3.4.6 Administración de rutas

Para la visualización y administración de las rutas que ya han sido publicadas se ofrece la funcionalidad del módulo de gestión rutas, en el cuál se permite ver en detalle una ruta, así como editarla o eliminarla. Se muestra una imagen de este módulo.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

Súbete Home Contacto Ver Rutas Crear Rutas Esteban

Rutas

Crear

Id	Latitud Origen	Longitud Origen	Latitud Destino	Longitud Destino	Fecha de Inicio	Fecha de Destino	Descripción			
1	6.1842461612806	-75.548708438873	6.1746889797481	-75.250704288483	2016-05-12 19:51:00	2016-05-20 19:52:00	Mi ruta	editar	ver detalle	eliminar
2	6.2320294726382	-75.470430850983	6.0709142484521	-75.246584415436	2016-05-17 19:54:00	2016-05-26 19:54:00	Ruta del camino Real	editar	ver detalle	eliminar
4	6.2333946463249	-75.573427677155	6.1432855989092	-75.359194278717	2016-05-07 06:56:00	2016-05-31 09:56:00	Viaje Casero	editar	ver detalle	eliminar

Figura 18. Administración de rutas.

3.4.7 Detalle de ruta – Unirse a ruta

La funcionalidad de ver en detalle una ruta, provee la información relacionada con dicha ruta. Además ofrece la opción de unirse a una ruta que haya sido creada anteriormente por otro usuario. Se visualiza una imagen del detalle de una ruta y la opción de unirse.

Súbete Home Contacto Ver Rutas Crear Rutas Esteban



Fecha y Hora de Salida
2016-05-12 19:51:00

Fecha y Hora de Llegada
2016-05-20 19:52:00

Descripción
Mi ruta

Unirse a la ruta Regresar

Figura 19. Detalle de ruta.

4. DISCUSIÓN DE RESULTADOS

Los resultados obtenidos del presente Trabajo de Grado, se evidencian en el desarrollo del proyecto. En esta primera etapa del proyecto contiene la documentación general del sistema, en la que se determina el análisis y diseño del mismo.

Se obtiene un análisis de las diferentes aplicaciones existentes y sus funcionalidades, de las cuales se identifica la información relevante para el desarrollo del sistema. Para la identificación del funcionamiento del sistema se obtiene el respectivo diagrama de casos de uso, el cual permite definir los requerimientos del sistema. Como parte del diseño, se obtienen diferentes artefactos como: Diagrama de entidad relación, Diagrama de clases, Diagramas de secuencia y Diagrama de flujo, los cuales aportan la estructura básica del sistema y la manera en que se realizan las funcionalidades.

Es importante mencionar, que el análisis y diseño de un sistema computacional puede ser susceptible a cambios y mejoras, durante el proceso de implementación, de acuerdo a las necesidades y dificultades que se presenten.

Cabe resaltar que se modificaron algunos diagramas de la etapa de análisis y diseño, debido a las necesidades que surgieron a la hora de la implementación del prototipo funcional.

Cada uno de los resultados se ven reflejados en los objetivos específicos previamente definidos.

5. CONCLUSIONES Y CONSIDERACIONES FINALES

El análisis y diseño de un sistema computacional es una parte muy importante del proceso de desarrollo de Software, por lo que permite determinar la estructura básica del sistema con todas sus características y funcionalidades, de manera lógica y organizada, lo cual es primordial para tener un proceso de implementación exitoso. Permite anticipar disfuncionalidades en el sistema en una etapa temprana, proporcionando la posibilidad de ahorrar en recursos de tiempo y trabajo en comparación a un escenario de identificación de fallas en una etapa más avanzada, como puede ser la etapa de implementación.

El estándar UML es una herramienta útil para el proceso de análisis y diseño de un sistema computacional, ya que han sido propuestos por expertos en ingeniería de software y a través del tiempo ha sido probado y mejorado con resultados satisfactorios.

La inclusión de zonas al módulo de creación de rutas agrega valor al sistema ya que permite tener flexibilidad en los tramos recorridos, y aumenta las oportunidades de encontrar una ruta que pueda satisfacer las necesidades de transporte, al proveer la posibilidad de desviarse del recorrido estándar.

BIBLIOGRAFÍA

Aguilar, C. P. (n.d.). Clase Clase Atributos Operaciones s.

Gutiérrez, J. (2006). ¿ Qué es un framework web?, 1–4. Retrieved from http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Mvc, E. M. (2008). El patrón MVC El patrón MVC El patrón MVC, (Mvc).

Usuario, M. De. (2006). Manual de usuario. *Media*, 806(Enero), 1–92. Retrieved from <http://www.emisarios.unican.es/herramientas.htm>

Reynoso, C. (2004). Introducción a la Arquitectura de Software. *Universidad de Buenos Aires*. Retrieved from <http://carlosreynoso.com.ar/archivos/arquitectura/Arquitectura-software.pdf>

Ruiz,(2009). *Monográfico Java* [Diagrama de clases]. Recuperado de <http://recursostic.educacion.es/observatorio/version/v2/es/software/programacion/911-monografico-java>

Proyecto sistema de registro (2008) [Diagrama de casos de uso]. Recuperado de <HTTP://USERS.DCC.UCHILE.CL/~JBARRIOS/J2EE/NODE105.HTML>

Gutiérrez, (2008) [¿Qué es un Framework web?]. Recuperado de: <http://www.nubelo.com/blog/ventajas-del-framework-moda-laravel/>

Uber. (n.d.). UberMedellinn. Retrieved from <https://www.uber.com/cities/medellin>

Taxis.com. (n.d.). Retrieved from <http://taxis.com.co/portal-personal/>

Citytaxi. (n.d.). Retrieved from <http://www.citytaxiapp.com/newPage/View/>

Easytaxi. (n.d.). Retrieved from <http://www.easytaxi.com/co>

Tripda. (n.d.). Retrieved from https://www.tripda.com.co/?gclid=Cj0KEQjwYqoBRDajuaTvsyq1PQBEiQAEhSjnGMDaWiK0XGtjY CZymQaPqfyQ4MgySGu0dr_dH4tQrMaAqj28P8HAQ

Miaguila. (n.d.). Retrieved from <http://www.miaguila.com/>

Fuimonos. (n.d.). Retrieved from <https://www.fuimonos.co/new.php>

Easyway. (n.d.). Retrieved from <https://www.easyway.com.co/>.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.