

Contenido

1. Introducción	4
2. Antecedentes	5
2.1. Uber	5
2.2. Cabify	6
2.3. Easy Taxi	6
3. Justificación	7
4. Descripción del problema	8
5. Formulación del problema	9
6. Objetivos	9
6.1. Objetivo general	9
6.2. Objetivos específicos	9
7. Alcance	10
7.1. Alcance web	10
7.2. Alcance móvil	12
8. Técnicas orientadas a mejorar la productividad	14
8.1. Uso de una metodología ágil de desarrollo de software	14
8.2. Uso de herramientas durante el ciclo de vida	15
8.3. Desarrollo de software basado en componentes	15

9. Actividades adicionales para asegurar la calidad.....	15
10. Innovación en el proyecto	16
10.1. Funcional	16
10.2. No funcional.....	17
10.2.1. Aplicación web progresiva	17
10.2.2. Aplicaciones móviles multiplataforma nativas	18
10.3. Herramienta	19
11. Arquitectura del software	19
12. Ingeniería inversa y re-ingeniería	21
12.1. Ingeniería inversa	21
12.2. Re-ingeniería	22
13. Descripción de herramientas de desarrollo	24
13.1. PhpStorm.....	24
13.2. Apache	24
14.3. Enterprise Architect	25
13.4. Git.....	26
13.5. Visual Studio Code	26
13.6. Postman.....	27
14. Proceso del software.....	28
14.1. Modelo de requisito	28

14.1.1. Identificar actores y casos de uso	28
14.1.2. Diagrama general de casos de Uso	29
14.1.3. Detalle de casos de uso.....	31
14.2. Modelo de arquitectura del software a nivel del código.....	43
14.3. Modelo de arquitectura del software a nivel de despliegue.....	44
14.4. Modelo de persistencia	45
14.4.1. Diagrama de clases.....	45
14.4.2. Tabla de volumen	46
14.4.3. Script SQL	48
15. Referencias	52

1. Introducción

Un factor importante para el crecimiento de cualquier institución académica es la movilidad, ya que resulta indispensable para los ciudadanos de la misma (plantel docente, estudiantil y administrativo) disminuir los tiempos de desplazamiento; es por ello que el uso de taxis, como servicio público, se ha incrementado en los últimos años de forma considerable. Esto debido a la comodidad y rapidez que ofrece el servicio hacia sus usuarios.

A medida que ha aumentado la demanda de este servicio, también se han incrementado el número de usuarios inconformes debido a el cobro elevado de la tarifa por parte de los taxistas, también por la poca seguridad ofrecida ya que en la ciudad existen muchos casos de robos, secuestros y maltrato durante el uso de este servicio.

Con el siguiente proyecto se propone la creación de un software para el control y uso de un servicio de transporte universitario realizado únicamente por y para los ciudadanos de la institución, pudiendo ellos ofrecer transporte a sus conciudadanos, haciendo un bien social y al mismo tiempo generando una pequeña fuente de ingreso o también pudiendo ellos recibir un servicio de transporte cómodo, seguro y eficiente.

2. Antecedentes

2.1. Uber



Uber es una plataforma tecnológica que pone en contacto a socios conductores con pasajeros gracias a una aplicación móvil.

Funciona de la siguiente manera. Primero, solicitas un viaje desde la aplicación en cualquier ciudad donde Uber esté activo. Tu solicitud se enviará a los socios conductores más cercanos. Cuando un conductor acepte tu solicitud de viaje, podrás consultar en la aplicación el tiempo estimado de llegada de tu conductor a la ubicación de recogida. También te llegará una notificación cuando el conductor esté a punto de llegar.

Al solicitar un viaje, la aplicación te proporciona el nombre del conductor, el tipo de vehículo y su número de matrícula para que puedas reconocerlo sin problemas. Cuando estés dentro del vehículo, comprueba que el destino que has introducido en la aplicación sea el correcto. Los conductores utilizan su aplicación Uber para confirmar los detalles del viaje. Si prefieres alguna ruta en concreto, díselo al conductor.

Cuando llegues a tu destino, el conductor finalizará el viaje. El precio del viaje se calcula automáticamente y se te cobra a través del método de pago vinculado a tu cuenta de Uber. En determinadas ciudades Uber te permitirá pagar con dinero en efectivo.

Inmediatamente después, la aplicación te pedirá que valores la experiencia de tu viaje. A los conductores también se les pide que valoren el viaje.

2.2. Cabify



Al igual que en Uber, el pago se realiza online y no hay necesidad de ningún intercambio de dinero físico entre el conductor y el pasajero. Estas son las pautas para disfrutar de sus servicios:

1. Darse de alta en la web de la plataforma y, muy recomendable, descargarse la aplicación en el móvil.
2. Rellenar los datos de registro: nombre, email y número de teléfono.
3. Insertar la dirección de destino: tras ello, el usuario conoce quién será su chófer, en cuánto tiempo le recoge y qué tipo de coche será.
4. Se puede reservar el vehículo (sin coste) en caso de requerir sus servicios para una determinada fecha y hora. La cancelación es gratuita siempre y cuando el coche no haya llegado a la zona acordada.
5. Elegir el tipo de vehículo. Lea las indicaciones bajo el apartado 'Tipos de coches de Cabify'.
6. Seleccionar un método de pago: o PayPal o tarjeta. Cabify cobra por distancia y no por tiempo.

2.3. Easy Taxi



Fundada en San Pablo (Brasil) en junio de 2011, llegó a la Argentina a comienzos del año pasado y empezó a operar unos meses después. Para usarla, se debe descargar la aplicación en forma gratuita y crear una cuenta por única vez con nombre y contraseña. De esta manera, el GPS del celular detectará automáticamente la calle en la que se encuentra el pasajero y éste sólo deberá ingresar la altura para facilitar su búsqueda.

Una vez ubicado exactamente en el mapa, se presiona el botón verde "Pedir taxi". A partir de ese momento, la aplicación realiza una búsqueda basándose en el taxista que está más cercano a ese punto. Apenas lo encuentra, muestra en qué lugar del mapa está con un símbolo de un vehículo. Si el taxista acepta el viaje, aparecerá una notificación en la pantalla y el usuario recibirá automáticamente toda la información del conductor.

Así, verá en su pantalla el nombre completo del conductor, foto, número de celular, modelo del auto y patente. Además, puede seguir en tiempo real el recorrido que el taxi hace hasta el domicilio desde el cual se pidió el servicio. También el taxista recibirá los datos del usuario, como su nombre y número de teléfono. En caso de querer suspender el viaje se puede optar por el botón "Cancelar" y elegir el motivo de cancelación.

3. Justificación

Debido al reciente cambio de la ciudad universitaria a lejanías de la ciudad la demanda de un servicio de transporte eficiente, eficaz y seguro ha incrementado de forma considerable. Ya que en la institución se cuenta con miembros de distintas clases sociales no se puede asumir que todos poseen un vehículo o los recursos necesarios para tomar un servicio público (taxi) diariamente, debido a sus altos costos, por lo que el transportarse de un lado a otro se ha vuelto un desafío financiero, además hay que recordar que en este servicio se puede sufrir todo tipos de peligros como ser robos, secuestros y abusos.

Por tal motivo la introducción de una aplicación capaz de poder ayudar a los miembros de la universidad a generar recursos y al mismo tiempo transportarse unos a otros de forma rápida, sencilla y segura viene a ser la mejor solución.

4. Descripción del problema

- Consumo excesivo de tiempo

Con la mudanza de los previos universitarios fuera de la ciudad el uso de transporte público (micros) ha sido una de las opciones como medio de desplazamiento para los ciudadanos universitarios. Pero como ellos tiene una ruta fija que abarca grandes recorridos dentro de la ciudad y un constante alojo y desalojo de los pasajeros ha vuelto que un viaje promedio dure hasta más de 45 minutos perjudicando de esta forma a todo aquel que use este servicio.

- Altos precios del pasaje en taxi

Como la distancia hasta las nuevas instalaciones de la universidad es considerablemente larga los precios de una carrera de taxi son demasiado altos superando fácilmente los 30 bs, por lo cual hacer uso de este servicio diariamente significaría un golpe económico muy grande en los recursos de cualquier miembro de la universidad.

- Inseguridad

Ya sea haciendo uso de un medio de transporte público masivo (microbuses) o de los taxis la población universitaria se ha envuelto en todo tipo de peligros siendo los principales los robos, asaltos, secuestros y los abusos.

- Falta de espacio de aparcamiento

Debido a la gran cantidad de miembros que posee la universidad seria casi impensable que todos y cada uno de ellos puedan disponer de un parqueo para sus propios autos particulares, por lo que al llevar uno su carro corre el riesgo de no encontrar un lugar donde aparcar o perder tiempo en el proceso.

5. Formulación del problema

Se pretende desarrollar una aplicación para el uso del servicio de transporte universitario, el cual permitirá a todos los miembros de la universidad registrarse para poder prestar servicio como conductores o recibir el servicio como pasajero. Dicha aplicación estará separada en una versión web y otra móvil, siendo la versión web para el registro del usuario, sus vehículos, las rutas que realizará y la visualización de los movimientos realizados de su cuenta monetaria. Y la parte móvil para la solicitud del servicio, señalando el punto de encuentro dentro de la ruta especificada por el conductor, recibiendo notificaciones según la proximidad de ambos, además dicha aplicación móvil debe adaptarse a distintos sistemas operativos siendo los más comunes Android y IOS.

6. Objetivos

6.1. Objetivo general

Desarrollar una aplicación web y móvil para el servicio de transporte universitario, el cual será ofrecido únicamente por y para la población universitaria, siendo estos los estudiantes, docentes y personal administrativo.

6.2. Objetivos específicos

- Diseñar e implementar una aplicación web progresiva, la cual permita a los usuarios poder registrarse y hacer uso del servicio.
- Diseñar e implementar una aplicación móvil multiplataforma basados en sistemas Android y IOS que permita solicitar el servicio, y además muestre el progreso del mismo a través de la geolocalización.

- Ofrecer a la población universitaria una herramienta el cual proporcionará una alternativa económica, rápida, sencilla y segura para su movilización hacia los previos de la Universidad.
- Liberar el congestionamiento automovilístico en los aparcamientos de la universidad.
- Ofrecer una pequeña fuente de ingreso a aquellos que deseen prestar servicio como conductor para el recojo de pasajeros mientras se encuentra camino hacia la universidad.

7. Alcance

7.1. Alcance web

- Todos los miembros activos de la universidad podrán registrarse gratuitamente a la aplicación a través de la página web proporcionando los siguientes datos: nombres, apellidos, código universitario, número de carnet de identidad, teléfono celular, dirección, fotografía tipo carnet, correo electrónico y una contraseña. Luego de realizarse el registro automáticamente se iniciará sesión, y se mostrará la pantalla principal por la cual el podrá tener acceso a todas las funcionalidades que ofrece la aplicación. Si el usuario ya se encuentra registrado este simplemente podrá iniciar sesión proporcionando su email y contraseña.
- Una vez que el usuario haya iniciado sesión este podrá actualizar la información básica de su perfil, como ser sus nombres, apellidos, teléfono celular, dirección, fotografía, correo electrónico y contraseña. El no podrá modificar su número de carnet de identidad ni el código de la universidad.

- Aquellos usuarios que deseen proporcionar el servicio de transporte como conductores deberán registrar su o sus vehículos que serán utilizados para dar dicho servicio. El registro solicitará la siguiente información: placa, marca del vehículo, modelo, año de producción, color y capacidad de pasajeros que soporta, además también de subir una fotografía actual del coche. Todos los datos nombrados anteriormente son obligatorios y podrán ser editados si fuera necesario por el mismo usuario luego de ser registrado.
- Aparte de gestionar sus vehículos los usuarios que deseen prestar servicio como conductores deberán también crear una ruta de viaje la cual será señalada al momento de crear un nuevo viaje. Para la creación de la ruta primeramente se desplegará un mapa de la ciudad sobre la pantalla, luego el usuario podrá ir señalando con “clics” sobre el mapa cómo será la trayectoria que el tomará, a medida que él vaya pulsando sobre el mapa dicha ruta se irá dibujando de forma automática, el seguirá señalando así hasta llegar a su destino. En caso de que el cometiera algún error mientras dibuja su ruta el dispondrá de dos botones en la parte inferior que le permitirán retroceder hacia un punto anterior o si él lo desea podrá eliminar la ruta completa para empezar desde cero. Luego de dibujar el mapa el usuario deberá darle un nombre y una descripción a la ruta creada.
- El usuario podrá revisar una bandeja de entrada de mensajes donde se le enviará notificaciones relacionadas con la aplicación como ser las multas por una falta en el uso del servicio.
- El usuario podrá visualizar el estado de su cuenta bancaria de la universidad donde se le mostrará el monto de dinero que tiene disponible, y todos los movimientos que se realizaron con la fecha, hora y la descripción del movimiento.

- Una función especial solo para los usuarios tipo administrativos, ellos podrán realizar depósitos a la cuenta de cualquier miembro de la población universitaria necesitando solo el código universitario y el monto a depositar.
- Como una de las funciones principales para los usuarios que deseen dar el servicio ellos luego de tener ya rutas registradas y sus vehículos ellos podrán crear un nuevo servicio o viaje, en el cual se le pedirá que seleccione el vehículo que utilizará y la ruta que realizará, además también de señalar la cantidad de pasajeros que llevará, el costo del pasaje y si este viaje es de ida a la universidad o de retorno.

7.2. Alcance móvil

- Todos los miembros activos de la universidad podrán registrarse gratuitamente a la aplicación a través de la página web proporcionando los siguientes datos: nombres, apellidos, código universitario, número de carnet de identidad, teléfono celular, dirección, fotografía tipo carnet, correo electrónico y una contraseña. Luego de realizarse el registro automáticamente se iniciará sesión, y se mostrará una pantalla donde el seleccionará el rol que desempeñará en ese momento ya sea un “conductor” o un “cliente”. Según su elección se le mostrará una pantalla principal respectiva. Si él lo desea puede cambiar de rol en cualquier instante. Si el usuario ya se encuentra registrado este simplemente podrá iniciar sesión proporcionando su email y contraseña.
- Si la elección del usuario fue como conductor se le mostrará en la pantalla principal una lista donde se mostrarán todos sus viajes pendientes por realizarse ordenándolos desde el más próximo hasta el más lejano. Si él lo desea también podrá ver toda la información de ese viaje, desde la ruta, el vehículo y toda la descripción que ingresó al momento de crear el servicio.

También el usuario podrá seleccionar el viaje que le corresponde realizar y iniciarlo (siempre y cuando sea ya la hora establecida y también que no hayan pasado más de 5 minutos), al momento de iniciar el viaje se le mostrará un mapa con la ruta y se le indicará con marcadores las posiciones donde los pasajeros quedaron en esperarle. Cada vez que el conductor se esté acercando a uno de estos marcadores tanto el cliente como el conductor recibirán una notificación informándole de la proximidad que tienen.

- Si la elección del usuario fue como cliente la aplicación mostrará una lista con todos los servicios que ha solicitado ordenándolo desde el más próximo hasta el más lejano, podrá visualizar toda la información correspondiente a dicho servicio, como ser la ruta, el conductor, el auto precio, etc. Si este no ha solicitado ningún servicio podrá hacerlo de la siguiente manera, en la pantalla principal habrá un botón que le permita hacer dicha función, este botón le redirigirá a un mapa donde el podrá señalar con un marcador la zona de búsqueda de servicios cercanos. Luego de esto le aparecerá una lista con todos los servicios disponibles, empezando desde los más cercanos, el podrá ver la información completa de cada uno de los servicios mostrados en esa lista. Cuando el usuario encuentre el adecuado para él, este deberá presionar el botón “Solicitar” con lo cual se desplegará un mapa con la ruta trazada sobre él, donde el usuario con un marcador señalará el lugar donde lo esperará al conductor, se mostrará una ventana de confirmación y una vez aceptada esta, el usuario ya habrá hecho una reserva la que al mismo tiempo le estará descontando el pasaje desde su cuenta bancaria.
- Como medida de seguridad se debe verificar la identidad tanto del conductor como del pasajero, para lo cual se empleará reconocimiento facial verificado por ambos al momento de abordar el vehículo, previendo así que personas ajenas realicen el servicio ocasionando todo tipo de irregularidades.

- Cuando el vehículo se encuentre alrededor del punto de destino la aplicación automáticamente desplegará un pequeño formulario de verificación en el móvil del conductor, donde el dará un informe del viaje como ser, cuantos pasajeros recogió, si no recogió alguno debe indicar el motivo (“No se encontraba en el punto de encuentro”, “No lo quise recoger”).
- Todos aquellos pasajeros que no fueron recogidos por el conductor a pesar pueden realizar una denuncia la cual si fuera cierta multará al conductor con 3 meses de suspensión como conductor.
- Todos aquellos pasajeros que no se encontraban en el punto de encuentro y por ende no fueron recogidos se les cobrará automáticamente el 50% del costo del pasaje como multa.

8. Técnicas orientadas a mejorar la productividad

8.1. Uso de una metodología ágil de desarrollo de software

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

8.2. Uso de herramientas durante el ciclo de vida

Hoy en día se encuentra una vasta cantidad de tecnologías y herramientas que facilitan y agilizan el desarrollo de software, ayudando en todo el ciclo de vida de un proyecto, desde el diseño y modelado hasta la implementación, puesta en marcha y prueba. Ayudando de manera significativa a la reducción de tiempo y proporcionando agilidad en el proceso de desarrollo.

8.3. Desarrollo de software basado en componentes

La complejidad de los sistemas computacionales actuales nos ha llevado a buscar la reutilización del software existente. El desarrollo de software basado en componentes permite reutilizar piezas de código pre-elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.

El uso de este paradigma posee algunas ventajas:

- Reutilización del software.
- Simplifica las pruebas.
- Simplifica el mantenimiento del sistema.
- Mayor calidad

9. Actividades adicionales para asegurar la calidad

Se realizó una gran cantidad de distintas pruebas verificando en todos los posibles ambientes imaginados verificando de esta manera que se están cumpliendo con todos los requisitos solicitados.

Se cuenta con un servicio de hosting estable y seguro, siendo que nuestra aplicación web y móvil funcionará únicamente con conexión a internet es muy necesario que esta se encuentre alojada en un lugar confiable, seguro y libre de

caídas, ya que un problema en el servidor ocasionaría fallas al funcionamiento de la aplicación.

Se trabajó con una considerable población de datos para la verificación de la velocidad de respuesta del servicio.

10. Innovación en el proyecto

10.1. Funcional

Se agregó una capa de seguridad a través del reconocimiento facial. Utilizando el api de Kairos. El proceso consta de cuatro módulos principales:

1. Detección de la cara: detecta que hay una cara en la imagen, sin identificarla. Si se trata de un video, también podemos hacer un seguimiento de la cara. Proporciona la localización y la escala a la que encontramos la cara.
2. Alineación de la cara: localiza las componentes de la cara y, mediante transformaciones geométricas, la normaliza respecto propiedades geométricas, como el tamaño y la pose, y fotométricas, como la iluminación. Para normalizar las imágenes de caras, se pueden seguir diferentes reglas, como la distancia entre las pupilas, la posición de la nariz, o la distancia entre las comisuras de los labios. También se debe definir el tamaño de las imágenes y la gama de colores. Normalmente, para disminuir la carga computacional del sistema, se acostumbra a utilizar imágenes pequeñas en escala de grises. A veces también se realiza una ecualización del histograma.
3. Extracción de características: proporciona información para distinguir entre las caras de diferentes personas según variaciones geométricas o fotométricas.
4. Reconocimiento: el vector de características extraído se compara con los vectores de características extraídos de las caras de la base de datos. Si encuentra uno con un porcentaje elevado de similitud, nos devuelve la identidad de la cara; si no, nos indica que es una cara desconocida.

10.2. No funcional

10.2.1. Aplicación web progresiva

Las Progressive Web Apps son experiencias que combinan lo mejor de la Web y lo mejor de las apps. Están disponibles para los usuarios a partir de la primera visita en una pestaña del navegador y no requieren instalación. A medida que el usuario compila progresivamente una relación con la app con el paso del tiempo, se hace más y más poderosa. Se carga rápidamente, incluso con redes débiles, envía notificaciones push relevantes, tiene un ícono en la pantalla principal y se carga como experiencia de pantalla completa y de primer nivel.

¿Qué es una Progressive Web App?

Una Progressive Web App es:

- Progresiva: funciona para todos los usuarios, sin importar la elección de navegador, porque está construida con mejora progresiva como principio central.
- Adaptable: se adapta a cualquier factor de formulario, sea escritorio, móvil, tablet o lo que venga en el futuro.
- Independiente de la conectividad: mejorada con service workers para trabajar sin conexión o con redes de mala calidad.
- Estilo app: al usuario le parece una app con interacciones y navegación estilo app, porque está construida con modelo de shell de app.
- Fresca: siempre actualizada gracias al proceso de actualización de service worker.
- Segura: emitida vía HTTPS para evitar intromisiones y para garantizar que el contenido no se haya manipulado.
- Descubrible: se puede identificar como "app" gracias al manifiesto W3C y al alcance de registro de service worker, lo que permite que los motores de búsqueda la encuentren.

- Posibilidad de volver a interactuar: facilita la posibilidad de volver a interactuar a través de funciones como notificaciones push.
- Instalable: les permite a los usuarios "conservar" las apps que les resultan más útiles en su pantalla principal sin la molestia de una tienda de app.
- Vinculable: se puede compartir fácilmente vía URL, no requiere instalación compleja.

10.2.2. Aplicaciones móviles multiplataforma nativas

Los desarrolladores front normalmente trabajamos dentro del ecosistema de los navegadores web. Por lo general, el desarrollo de aplicaciones móviles nativas ha sido un mundo aparte para nosotros.

Pero parece que los tiempos están cambiando y, con la promesa de poder programar aplicaciones tanto para iOS como para Android con un solo lenguaje de programación (en este caso Javascript), han nacido los llamados Cross-Platforms frameworks, como Ionic, NativeScript y React Native, de modo que los frontend que queremos matar el gusanillo de hacer apps nativas con JS ya lo tenemos más fácil.

Los marcos multiplataforma proporcionan a los desarrolladores un conjunto completo de herramientas diseñadas para aumentar la productividad al resolver problemas comunes. La pregunta es qué marco es el mejor para usted. Para ayudarlo a responder esta pregunta, hemos preparado esta lista de marcos multiplataforma para el desarrollo móvil.

Con casi 3 millones de aplicaciones en Google Play, el sistema operativo Android domina el panorama móvil. Las personas, las pequeñas empresas y las grandes empresas trabajan arduamente para establecer una fuerte presencia móvil y obtener su parte del mercado. Sin embargo, no todos tienen la experiencia y los recursos necesarios para crear una aplicación desde cero utilizando herramientas nativas.

10.3. Herramienta

Uso de React Native. React Native es un framework que permite a los desarrolladores implementar apps nativas para dispositivos móviles utilizando Javascript. De momento los sistemas operativos mínimos soportados son: Android 4.1 (API 16) y \geq iOS 8.0.

Las RN apps tienen acceso directo a todas las APIs y views nativas que ofrecen los sistemas operativos nativos. De forma que la experiencia de usuario y el rendimiento es el mismo de una aplicación nativa.

Lo primero que se podría suponer es que React Native compila código JS en el correspondiente código nativo directamente. Pero esa empresa es bastante dura de realizar, ya que Java y Objective C/Swift son lenguajes fuertemente tipados mientras que Javascript no lo es.

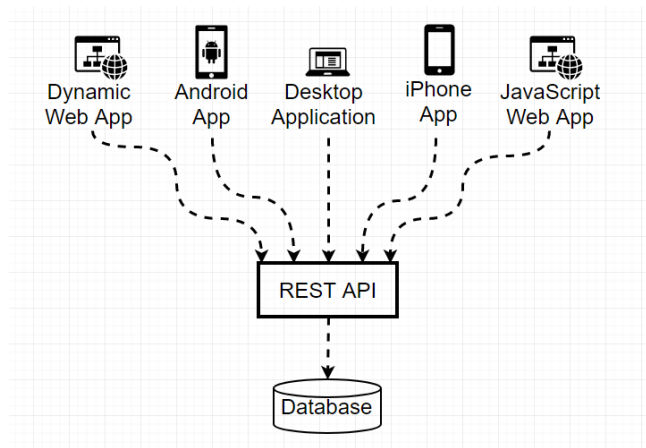
En lugar de eso, RN hace algo más inteligente: React Native es, en esencia, un conjunto de componentes React, donde cada uno de ellos tiene su correspondiente equivalente en views y componentes nativos.

11. Arquitectura del software

Arquitectura de software. La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

La arquitectura de software forma la columna vertebral para construir un sistema de software, es en gran medida responsable de permitir o no ciertos atributos de calidad del sistema entre los que se destacan la confiabilidad y el rendimiento del

software. Además, es un modelo abstracto reutilizable que puede transferirse de un sistema a otro y que representa un medio de comunicación y discusión entre participantes del proyecto, permitiendo así la interacción e intercambio entre los desarrolladores con el objetivo final de establecer el intercambio de conocimientos y puntos de vista entre ellos.



El término REST (Representational State Transfer) un servicio REST es un conjunto de restricciones con las que podemos crear un estilo de arquitectura software, la cual podremos usar para crear aplicaciones web respetando HTTP.

Hoy en día la mayoría de las empresas utilizan API REST para crear servicios. Esto se debe a que es un estándar lógico y eficiente para la creación de servicios web.

Según Fielding las restricciones que definen a un sistema RESTful serían:

- Cliente-servidor: esta restricción mantiene al cliente y al servidor débilmente acoplados. Esto quiere decir que el cliente no necesita conocer los detalles de implementación del servidor y el servidor se “despreocupa” de cómo son usados los datos que envía al cliente.
- Sin estado: aquí decimos que cada petición que recibe el servidor debería ser independiente, es decir, no es necesario mantener sesiones.
- Cacheable: debe admitir un sistema de almacenamiento en caché. La infraestructura de red debe soportar una caché de varios niveles. Este

almacenamiento evitará repetir varias conexiones entre el servidor y el cliente para recuperar un mismo recurso.

- Interfaz uniforme: define una interfaz genérica para administrar cada interacción que se produzca entre el cliente y el servidor de manera uniforme, lo cual simplifica y separa la arquitectura. Esta restricción indica que cada recurso del servicio REST debe tener una única dirección, “URI”.
- Sistema de capas: el servidor puede disponer de varias capas para su implementación. Esto ayuda a mejorar la escalabilidad, el rendimiento y la seguridad.

12. Ingeniería inversa y re-ingeniería

12.1. Ingeniería inversa

La ingeniería inversa es el proceso llevado a cabo con el objetivo de obtener información o un diseño a partir de un producto, con el fin de determinar cuáles son sus componentes y de qué manera interactúan entre sí y cuál fue el proceso de fabricación.

El método se denomina así porque avanza en dirección opuesta a las tareas habituales de ingeniería, que consisten en utilizar datos técnicos para elaborar un producto determinado. En general, si el producto u otro material que fue sometido a la ingeniería inversa fue obtenido en forma apropiada, entonces el proceso es legítimo y legal. De la misma forma, pueden fabricarse y distribuirse, legalmente, los productos genéricos creados a partir de la información obtenida de la ingeniería inversa.

La ingeniería inversa es un método de resolución. Aplicar ingeniería inversa a algo supone profundizar en el estudio de su funcionamiento, hasta el punto de que podamos llegar a entender, modificar y mejorar dicho modo de funcionamiento.

La aplicación de ingeniería inversa nunca cambia la funcionalidad del producto objeto de la aplicación, sino que permite obtener productos que indican cómo se ha construido el mismo. Su realización permite obtener los siguientes beneficios:

- Reducir la complejidad del sistema: al intentar comprender el sistema se facilita su mantenimiento y la complejidad existente disminuye.
- Generar diferentes alternativas: del punto de partida del proceso, principalmente código fuente, se generan representaciones gráficas lo que facilita su comprensión.
- Recuperar y/o actualizar la información.
- Detectar efectos laterales: los cambios que se puedan realizar en un sistema puede conducirnos a que surjan efectos no deseados, esta serie de anomalías puede ser detectados por la ingeniería inversa.
- Facilitar la reutilización: por medio de la ingeniería inversa se pueden detectar componentes de posible reutilización de sistemas existentes, pudiendo aumentar la productividad, reducir los costes y los riesgos de mantenimiento.

12.2. Re-ingeniería

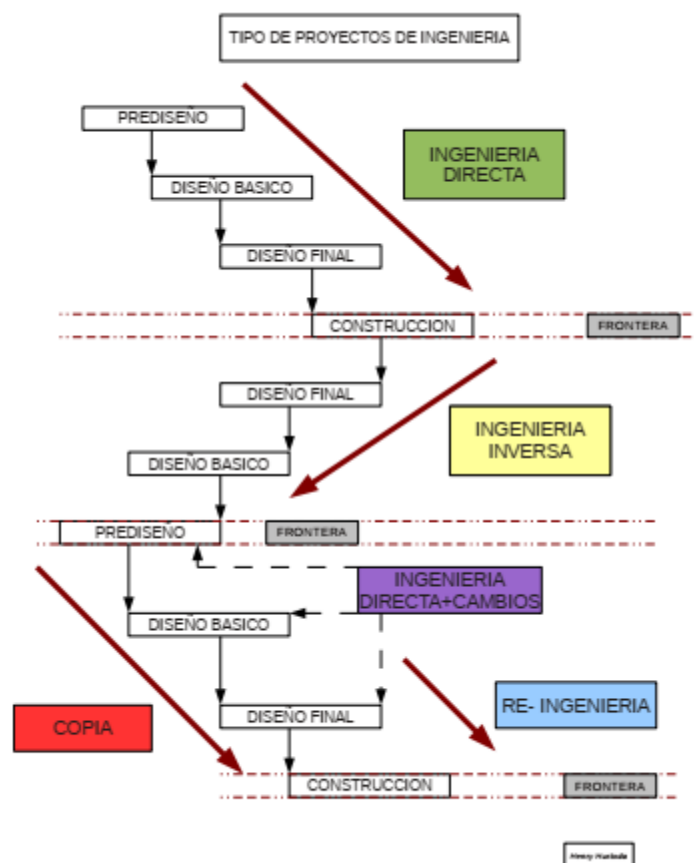
Reingeniería en un concepto simple es el rediseño de un proceso en un negocio o un cambio drástico de un proceso. A pesar que este concepto resume la idea principal de la reingeniería esta frase no envuelve todo lo que implica la reingeniería. Reingeniería es comenzar de cero, es un cambio de todo o nada, además ordena la empresa alrededor de los procesos. La reingeniería requiere que los procesos fundamentales de los negocios sean observados desde una perspectiva transfuncional y en base a la satisfacción del cliente.

Una definición rápida de reingeniería es "comenzar de nuevo". Reingeniería también significa el abandono de viejos procedimientos y la búsqueda de trabajo que agregue valor hacia el consumidor.

La definición más aceptada actualmente es la siguiente "La Reingeniería es el replanteamiento fundamental y el rediseño radical de los procesos del negocio para lograr mejoras dramáticas dentro de medidas críticas y contemporáneas de desempeño, tales como costo, calidad, servicio y rapidez". (Hammer 1994)

La implementación paulatina de un enfoque basado en procesos permite a una organización:

- Establecer indicadores de gestión para los procesos básicos de la organización e indicadores de resultados
- Eliminar actividades sin valor agregado
- Mejorar los flujos de información
- Reducir tiempos de operación
- Mantener los procesos focalizados en el cliente
- Mejorar la calidad del servicio



13. Descripción de herramientas de desarrollo

13.1. PhpStorm



PhpStorm de JetBrains es un PHP IDE inteligente y profesional que proporciona a los desarrolladores una mezcla de potentes herramientas inteligentes, hábitos útiles y mejores prácticas para el desarrollo de PHP, todos orientados hacia sacar provecho a la productividad del desarrollador.

- Asistencia de codificación inteligente
- Análisis de Calidad del código
- Entorno de desarrollo
- HTML/CSS/JavaScript Editor
- Fácil depuración y pruebas
- Refactorización rápida y segura

13.2. Apache



El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico, pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de Estados Unidos, y en esos momentos la

preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además, Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte)

14.3. Enterprise Architect



Sparx Systems Enterprise Architect es una herramienta de modelado y diseño visual basada en OMG UML. La plataforma admite: el diseño y la construcción de sistemas de software; modelar procesos comerciales; y modelado de dominios basados en la industria. Las empresas y organizaciones lo usan no solo para modelar la arquitectura de sus sistemas, sino también para procesar la implementación de estos modelos en todo el ciclo de vida de desarrollo de aplicaciones.

- Modelado y análisis de negocios y datos
- Gestión de pruebas y simulación
- Análisis de ejecución visual
- Gestión del proyecto
- Integración con otras herramientas

13.4. Git



Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

- Compartición selectiva
- Velocidad y seguridad
- Ramificación
- Convergencia

13.5. 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, aunque la descarga oficial está bajo software propietario.

- IntelliSense, una función que nos permite desde resaltar la sintaxis de nuestros proyectos.
- Cuenta con un avanzado depurador de código.

- Se integra en Git. Nos permite versionar nuestros trabajos fácilmente desde esta plataforma.
- Gracias a sus módulos, es compatible con prácticamente cualquier lenguaje de programación.

13.6. Postman



Postman surgió originariamente como una extensión para el navegador Google Chrome. A día de hoy dispone de aplicaciones nativas para MAC y Windows y están trabajando en una aplicación nativa para Linux (disponible en versión beta).

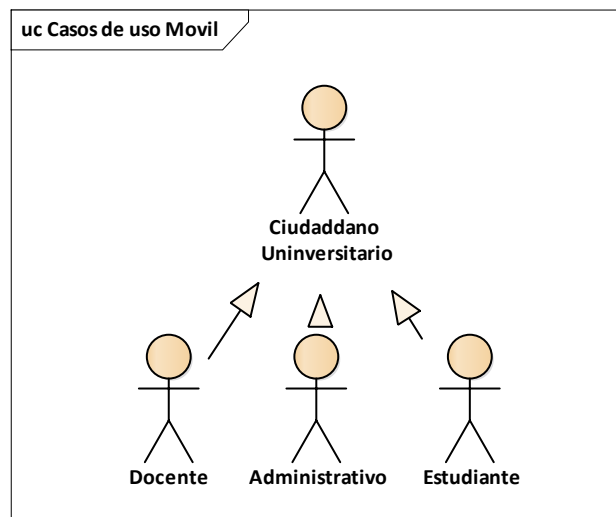
Está compuesto por diferentes herramientas y utilidades gratuitas (en la versión free) que permiten realizar tareas diferentes dentro del mundo API REST: creación de peticiones a APIs internas o de terceros, elaboración de tests para validar el comportamiento de APIs, posibilidad de crear entornos de trabajo diferentes (con variables globales y locales), y todo ello con la posibilidad de ser compartido con otros compañeros del equipo de manera gratuita (exportación de toda esta información mediante URL en formato JSON).

14. Proceso del software

14.1. Modelo de requisito

14.1.1. Identificar actores y casos de uso

Actores



Ciudadano Universitario: Este usuario sale como el resultado de una generalización de todos los ciudadanos universitarios es decir (estudiantes, docentes y plantel administrativo) ya que la aplicación está dirigida únicamente para ellos, pero como su uso no varía dentro de la aplicación se decidió generalizarlo.

Lista de casos de uso

Aplicación Web:

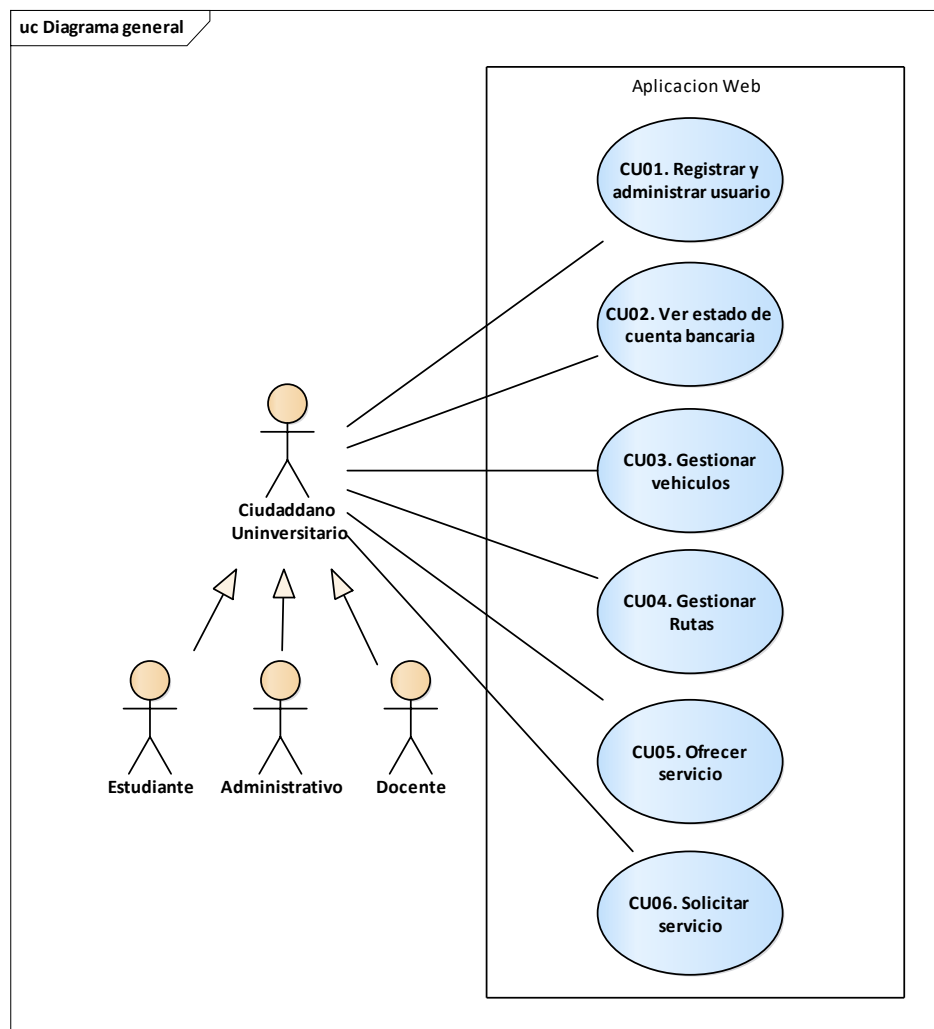
- CU01. Registrar y administrar usuario
- CU02. Ver estado de cuenta bancaria
- CU03. Gestionar vehículos
- CU04. Gestionar Rutas
- CU05. Ofrecer servicio
- CU06. Solicitar servicio

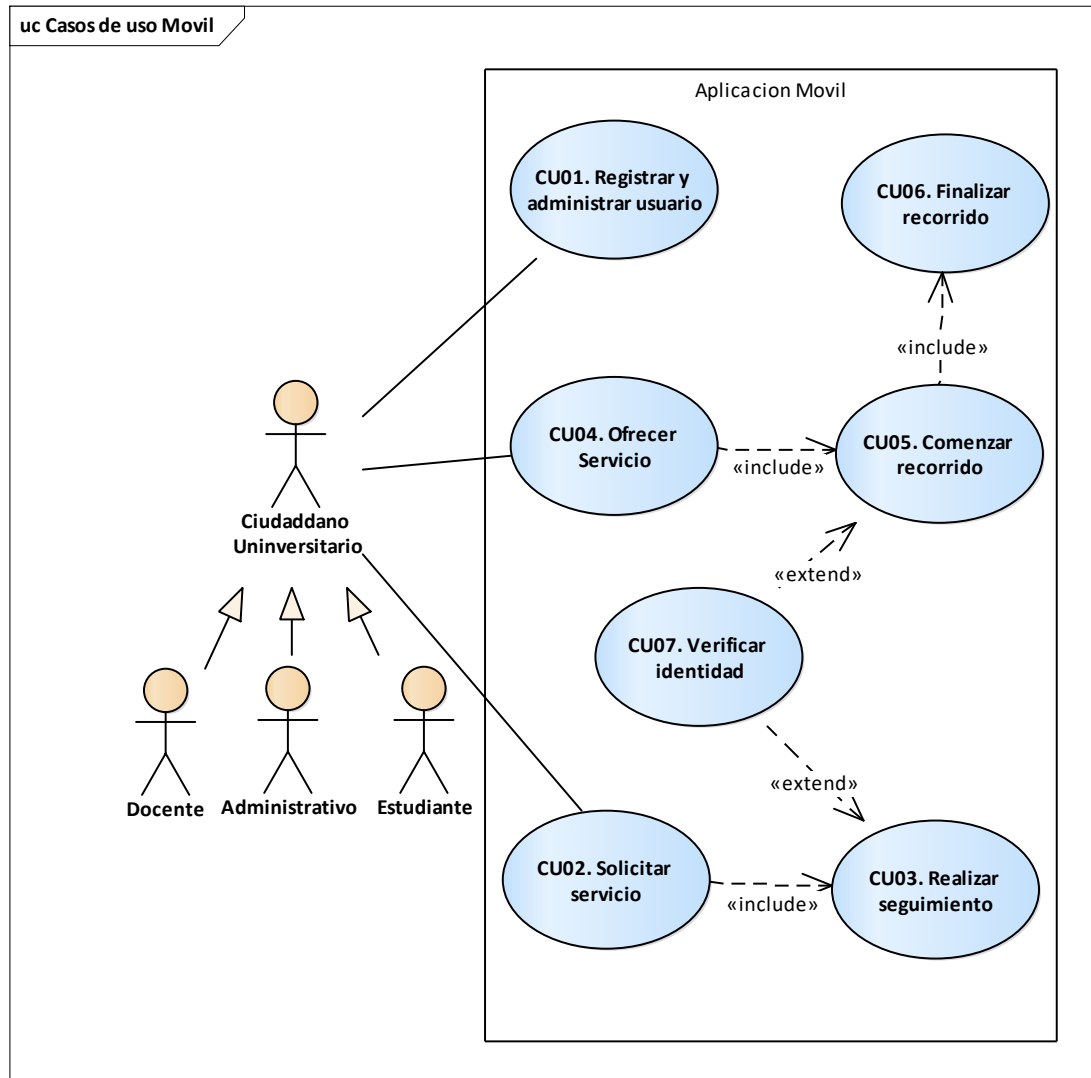
Aplicación Móvil:

- CU01. Registrar y administrar usuario
- CU02. Solicitar servicio
- CU03. Realizar seguimiento
- CU04. Ofrecer Servicio
- CU05. Comenzar recorrido
- CU06. Finalizar recorrido
- CU07. Verificar identidad

14.1.2. Diagrama general de casos de Uso

Web





14.1.3. Detalle de casos de uso

Web

Caso de uso	CU01. Registrar y administrar usuario
Propósito	Registrar a los usuarios que usarán la aplicación, y permitirles modificar sus datos si fuere necesario.
Descripción	Todos los usuarios que deseen utilizar la aplicación deberán registrarse depositando algunos datos dentro de un formulario, se le pedirá su código universitario, carnet de identidad, nombre completo, teléfono celular, una fotografía, dirección, email y una contraseña. Si el usuario lo deseara podrá actualizar sus datos con excepción a su carnet de identidad y código universitario.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Pertenecer oficialmente a la universidad.
Flujo principal	<ol style="list-style-type: none">1. Registro:<ol style="list-style-type: none">a. Se le despliega un formulario de registro a todos los usuarios no registradosb. El formulario pedirá los siguientes datos: código universitario, carnet de identidad, nombre completo, teléfono celular, una fotografía, dirección, email y una contraseña.c. El usuario rellena los campos y selecciona "Registrar"d. Se verificará que los campos hayan sido rellenos correctamente y se guardan los datos.2. Editar perfil:<ol style="list-style-type: none">a. Se desplegará un formulario donde se cargará toda la información del usuario, juntamente con su foto actual.b. El usuario modifica los campos que él desea y sea permitidoc. Se verificará que los campos hayan sido rellenos correctamente y se actualizan los datos.
Post-condición	Ninguna
Excepciones	<ol style="list-style-type: none">1.c. Debe rellanar los campos obligatorios1.c. El correo ingresado ya se encuentra registrado2.c. Debe rellanar los campos obligatorios

Caso de uso	CU03. Gestionar vehículos
Propósito	Registrar, editar y eliminar los vehículos que se usarán durante el servicio de transporte universitario.
Descripción	El usuario que desee realizar el servicio como conductor, primeramente deberá registrar su vehículo, para lo cual deberá proporcionar los siguientes datos: placa, marca, modelo, año, color, capacidad de pasajeros y una fotografía actual, si él también lo deseara podrá editar dichos datos o quitar de la aplicación a cualquiera de sus vehículos.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Estar registrado en la aplicación.
Flujo principal	<ol style="list-style-type: none"> 1. Listar vehículos: <ol style="list-style-type: none"> a. Se obtendrán todos los vehículos registrados por el usuario y serán mostrados en una tabla detallando la placa, marca y una foto. 2. Registrar vehículo: <ol style="list-style-type: none"> a. En la vista principal del caso de uso el usuario seleccionará "Nuevo" b. El sistema desplegará un formulario donde se solicitará la siguiente información: placa, marca, modelo, año, color, capacidad de pasajeros y una fotografía actual. c. El usuario rellena todos los campos y selecciona "guardar". d. Se verifican los datos enviados por el usuario y se guarda. 3. Editar vehículo: <ol style="list-style-type: none"> a. En la vista principal del caso de uso el usuario seleccionará "Editar" sobre el vehículo que desee modificar b. El sistema desplegará un formulario donde se mostrará toda la información del vehículo registrada por el usuario. c. El usuario modificará todos los campos que le sean convenientes y seleccionará "guardar". d. El sistema actualizará la información de ese vehículo. 4. Eliminar vehículo: <ol style="list-style-type: none"> a. Dentro de la vista principal del caso de uso el usuario selecciona "Eliminar" en la fila del vehículo que desee quitar. b. Aparecerá una ventana emergente pidiendo una confirmación. c. El usuario acepta. d. Se ubica el registro del vehículo y este es quitado.
Post-condición	Ninguna
Excepciones	2.d. Debe llenar todos los campos que son obligatorios.

Caso de uso	CU04. Gestionar Rutas
Propósito	Registrar de manera interactiva las rutas que los conductores tomarán al momento de ofrecer el servicio de transporte.
Descripción	El conductor que desee ofrecer el servicio de transporte deberá primeramente registrar sus rutas que realizará. Para ello en el formulario donde creará dicha ruta el deberá colocarle un nombre a la ruta y una descripción la cual le ayudará a reconocer fácilmente, también se desplegará un mapa en el cual el podrá dibujar su ruta.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Estar registrado en la aplicación.
Flujo principal	<ol style="list-style-type: none"> 1. Listar ruta: <ol style="list-style-type: none"> a. El sistema recuperará todas las rutas que fueron creadas por el usuario y las mostrará en la vista principal en una tabla la cual tendrá la siguiente información: nombre y descripción. 2. Nueva ruta: <ol style="list-style-type: none"> a. En la vista principal el usuario seleccionará "Nueva" b. El sistema desplegará un formulario donde creará dicha ruta el deberá colocarle un nombre a la ruta y una descripción la cual le ayudará a reconocer fácilmente, también se desplegará un mapa en el cual el podrá dibujar su ruta a través del evento click del mouse sobre el mapa. Cuando el desee podrá limpiar toda la ruta o simplemente retroceder de uno en uno. 3. Ver ruta: <ol style="list-style-type: none"> a. Sobre la ruta que se seleccione el sistema mostrará toda la información de dicha ruta con su respectivo dibujo sobre un mapa interactivo. 4. Eliminar ruta: <ol style="list-style-type: none"> a. Dentro de la vista principal del caso de uso el usuario selecciona "Eliminar" en la fila de la ruta que desee quitar. b. Aparecerá una ventana emergente pidiendo una confirmación. c. El usuario acepta. d. El sistema quita de los registros a la ruta
Post-condición	Ninguna
Excepciones	2.b. Debe al menos seleccionar 5 puntos sobre el mapa.

Caso de uso	CU05. Ofrecer servicio
Propósito	Que el conductor pueda publicar que está saliendo hacia la Universidad o retornando de ella y que está dispuesto a recoger pasajeros.
Descripción	Los usuarios que deseen ofrecer el servicio como conductores podrán publicar el horario en el que saldrán, el vehículo que utilizarán, la ruta que tomarán y el precio que costará utilizar dicho servicio.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Tener registrado al menos un vehículo y una ruta.
Flujo principal	<ol style="list-style-type: none"> 1. Listar servicios: <ol style="list-style-type: none"> a. En la ventana principal del caso de uso se listarán en una tabla todos los servicios que el usuario haya creado y aun estén pendientes a ser realizados. 2. Nuevo servicio: <ol style="list-style-type: none"> a. En la ventana principal del caso de uso el usuario seleccionará "Nuevo" b. Se desplegará un formulario, en el cual se pedirá los siguientes datos: el sentido del viaje (ida o vuelta), la fecha y hora de salir, seleccionará la ruta que realizará, el vehículo que utilizará y por último el precio del pasaje. c. El usuario rellenará todos los datos y seleccionará "Guardar". d. El sistema verificará y validará todos los campos y procederá a guardarlo. e. Se redirige a la vista principal
Post-condición	Ninguna
Excepciones	2.a. Debe llenar todos los campos que sean obligatorios.

Caso de uso	CU06. Solicitar servicio
Propósito	Que el usuario pueda encontrar el servicio de transporte más adecuado para sus necesidades, según la hora de salida y ubicación. Una vez hallado este que pueda realizar su reserva señalando un punto de encuentro en la ruta del servicio.
Descripción	El usuario que desee utilizar el servicio como medio de transporte podrá hacerlo de la siguiente manera: primero señala a qué hora y más o menos donde quiere que lo recojan, para lo cual el sistema buscará aquellos más cercanos y al horario más próximo. Una vez realizada la búsqueda se mostrarán los mejores resultados en una lista, de la cual el usuario podrá seleccionar cualquiera para ver sus detalles y si lo desea, señalar un punto de encuentro y solicitar dicho servicio.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Estar registrado en la aplicación.
Flujo principal	<ol style="list-style-type: none"> 1. Listar servicios: <ol style="list-style-type: none"> a. En la ventana principal del caso de uso se listará dentro de una tabla todos los servicios que el usuario haya solicitado siempre y cuando este no haya finalizado. 2. Solicitar servicio: <ol style="list-style-type: none"> a. En la ventana principal el usuario seleccionara "Solicitar" b. El sistema le redirigirá hacia un formulario donde se le pedirá la fecha, hora, el sentido, y un lugar marcado en el mapa para su recojo. c. El usuario rellena los campos y marca en el mapa el lugar donde desea buscar. d. El sistema procesa lo solicitud y busca en todos los servicios disponibles uno que cumpla con las necesidades del cliente. e. Se muestra una lista con todos los mejores resultados. f. El usuario selecciona uno. g. Se despliega una ficha con toda la información del servicio, mostrando la ruta, información del conductor y el vehículo q se utilizará. h. El usuario si está conforme con el servicio en el mapa señalará donde desea ser recogido y selecciona "Solicitar" i. Se procesa su solicitud, se realiza el cobro del servicio, y se notifica al conductor. j. Se redirige a la vista principal.
Post-condición	Ninguna
Excepciones	Ninguna

Caso de uso	CU02. Ver estado de cuenta bancaria
Propósito	Ver el saldo actual junto con todos los movimientos financieros realizados, cada uno con su respectivo detalle.
Descripción	Cada usuario podrá ver su saldo actual en su cuenta bancaria junto con todos los movimientos financieros realizados, de los cuales se mostrará el código del movimiento la fecha y hora exacta, el tipo de movimiento, el detalle y el monto.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Estar registrado en la aplicación.
Flujo principal	<ol style="list-style-type: none"> 1. Listar movimientos: <ol style="list-style-type: none"> a. Se recuperará en un arreglo todos los movimientos financieros realizados dentro de la cuenta del usuario, ordenándolos desde el más reciente hasta el más lejano. b. Se listará el resultado obtenido en una tabla mostrando el código del movimiento la fecha y hora exacta, el tipo de movimiento, el detalle y el monto.
Post-condición	Ninguna.
Excepciones	Ninguna.

Caso de uso	CU01. Registrar y administrar usuario
Propósito	Registrar a los usuarios que usarán la aplicación, y permitirles modificar sus datos si fuere necesario.
Descripción	Todos los usuarios que deseen utilizar la aplicación deberán registrarse depositando algunos datos dentro de un formulario, se le pedirá su código universitario, carnet de identidad, nombre completo, teléfono celular, una fotografía, dirección, email y una contraseña. Si el usuario lo deseara podrá actualizar sus datos con excepción a su carnet de identidad y código universitario.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Pertenecer oficialmente a la universidad.
Flujo principal	<ol style="list-style-type: none"> 1. Registro: <ol style="list-style-type: none"> a. Se le despliega un formulario de registro a todos los usuarios no registrados b. El formulario pedirá los siguientes datos: código universitario, carnet de identidad, nombre completo, teléfono celular, una fotografía, dirección, email y una contraseña. c. El usuario rellena los campos y selecciona "Registrar" d. Se verificará que los campos hayan sido rellenos correctamente y se guardan los datos. 2. Editar perfil: <ol style="list-style-type: none"> a. Se desplegará un formulario donde se cargará toda la información del usuario, juntamente con su foto actual. b. El usuario modifica los campos que él desea y sea permitido c. Se verificará que los campos hayan sido rellenos correctamente y se actualizan los datos.
Post-condición	Ninguna
Excepciones	<ol style="list-style-type: none"> 1.c. Debe rellanar los campos obligatorios 1.c. El correo ingresado ya se encuentra registrado 2.c. Debe rellanar los campos obligatorios

Caso de uso	CU02. Solicitar servicio
Propósito	Que el usuario pueda encontrar el servicio de transporte más adecuado para sus necesidades, según la hora de salida y ubicación. Una vez hallado este que pueda realizar su reserva señalando un punto de encuentro en la ruta del servicio.
Descripción	El usuario que desee utilizar el servicio como medio de transporte podrá hacerlo de la siguiente manera: primero señala a qué hora y más o menos donde quiere que lo recojan, para lo cual el sistema buscará aquellos más cercanos y al horario más próximo. Una vez realizada la búsqueda se mostrarán los mejores resultados en una lista, de la cual el usuario podrá seleccionar cualquiera para ver sus detalles y si lo desea, señalar un punto de encuentro y solicitar dicho servicio.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Estar registrado en la aplicación.
Flujo principal	<ol style="list-style-type: none"> 1. Listar servicios: <ol style="list-style-type: none"> a. En la ventana principal del caso de uso se listará dentro de una tabla todos los servicios que el usuario haya solicitado siempre y cuando este no haya finalizado. 2. Solicitar servicio: <ol style="list-style-type: none"> a. En la ventana principal el usuario seleccionara "Solicitar" b. El sistema le redirigirá hacia un formulario donde se le pedirá la fecha, hora, el sentido, y un lugar marcado en el mapa para su recojo. c. El usuario rellena los campos y marca en el mapa el lugar donde desea buscar. d. El sistema procesa lo solicitud y busca en todos los servicios disponibles uno que cumpla con las necesidades del cliente. e. Se muestra una lista con todos los mejores resultados. f. El usuario selecciona uno. g. Se despliega una ficha con toda la información del servicio, mostrando la ruta, información del conductor y el vehículo q se utilizará. h. El usuario si está conforme con el servicio en el mapa señalará donde desea ser recogido y selecciona "Solicitar" i. Se procesa su solicitud, se realiza el cobro del servicio, y se notifica al conductor. j. Se redirige a la vista principal.
Post-condición	Ninguna
Excepciones	Ninguna

Caso de uso	CU03. Realizar seguimiento
Propósito	Visualizar en tiempo real el movimiento del conductor sobre la ruta, para saber de su proximidad al punto de encuentro.
Descripción	Todo usuario que haya solicitado un servicio de transporte, podrá ver el progreso del conductor sobre la ruta (desde un mapa), siempre y cuando este haya iniciado el recorrido.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Haber solicitado un servicio y que este haya sido iniciado por el conductor.
Flujo principal	<ol style="list-style-type: none"> 1. Realizar seguimiento: <ol style="list-style-type: none"> a. En la vista donde se listan todos los servicios solicitados el usuario "Seguimiento" sobre aquel servicio que desea ver su progreso. b. La aplicación verificará si este servicio se iniciado. c. Redirigirá a una vista donde se mostrar en mapa con la ruta y además un marcador en movimiento, el cual representara al conductor que ofrece el servicio.
Post-condición	Ninguna.
Excepciones	1.b. El servicio aún no se ha iniciado.

Caso de uso	CU04. Ofrecer servicio
Propósito	Que el conductor pueda publicar que está saliendo hacia la Universidad o retornando de ella y que está dispuesto a recoger pasajeros.
Descripción	Los usuarios que deseen ofrecer el servicio como conductores podrán publicar el horario en el que saldrán, el vehículo que utilizarán, la ruta que tomarán y el precio que costará utilizar dicho servicio.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Tener registrado al menos un vehículo y una ruta.
Flujo principal	<ol style="list-style-type: none"> 1. Listar servicios: <ol style="list-style-type: none"> a. En la ventana principal del caso de uso se listarán en una tabla todos los servicios que el usuario haya creado y aun estén pendientes a ser realizados. 2. Nuevo servicio: <ol style="list-style-type: none"> a. En la ventana principal del caso de uso el usuario seleccionará "Nuevo"

	<ul style="list-style-type: none"> b. Se desplegará un formulario, en el cual se pedirá los siguientes datos: el sentido del viaje (ida o vuelta), la fecha y hora de salir, seleccionará la ruta que realizará, el vehículo que utilizará y por último el precio del pasaje. c. El usuario rellenará todos los datos y seleccionará "Guardar". d. El sistema verificará y validará todos los campos y procederá a guardarlo. e. Se redirige a la vista principal
Post-condición	Ninguna
Excepciones	2.a. Debe llenar todos los campos que sean obligatorios.

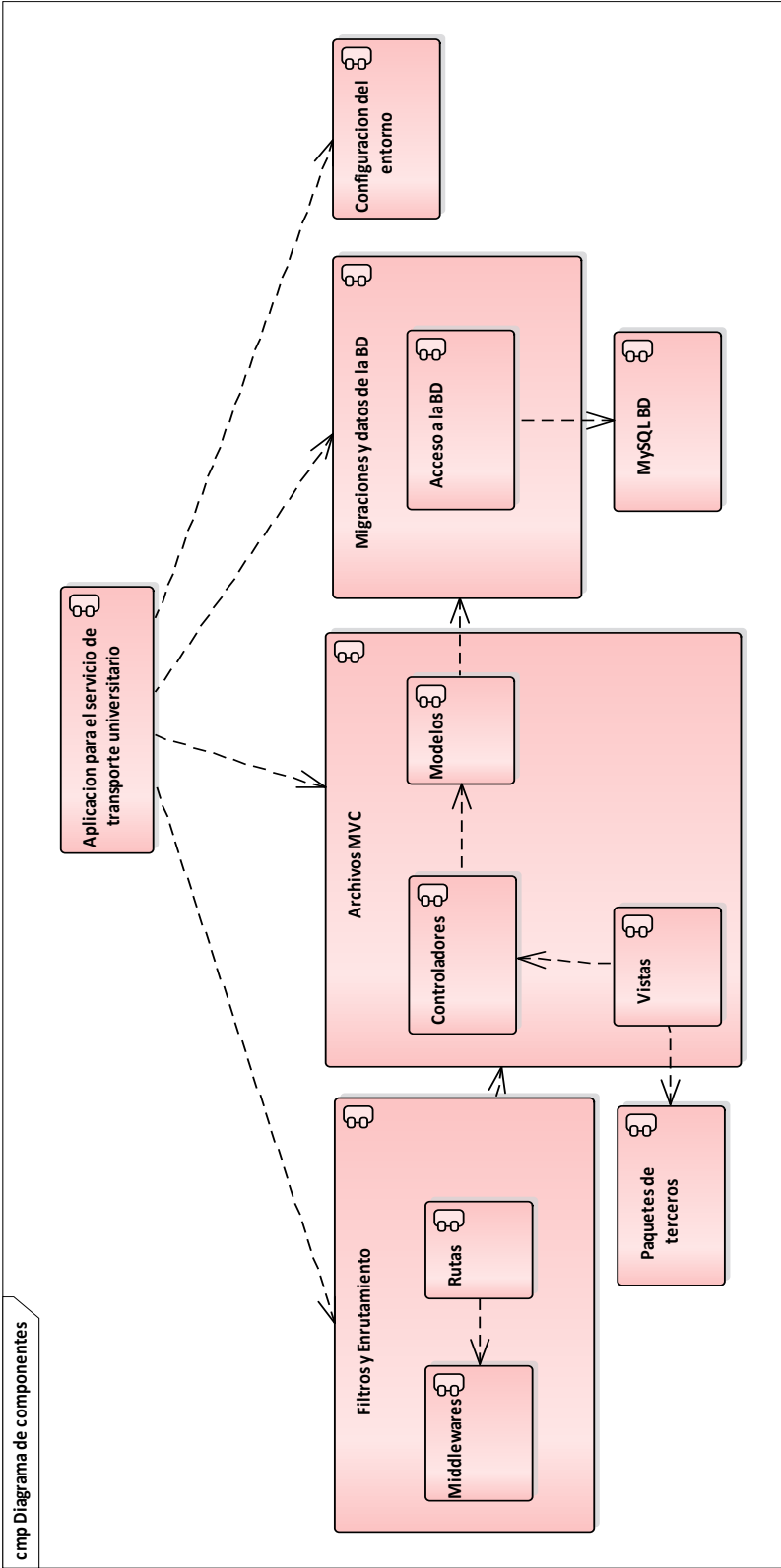
Caso de uso	CU05. Comenzar recorrido
Propósito	Mostrar al conductor su ruta establecida y los puntos de encuentro para el recojo de pasajeros.
Descripción	El usuario al comenzar un recorrido de un servicio, primero estará informando a sus pasajeros de que ya está partiendo, permitirá también que ellos puedan tener un seguimiento de su progreso y el podrá visualizar su ruta y los puntos de encuentro.
Actores	Ciudadano universitario.
Actor iniciador	Ciudadano universitario.
Pre-condición	Haber creado un servicio.
Flujo principal	1. Comenzar recorrido: <ul style="list-style-type: none"> a. En la vista principal de los servicios creados, el usuario selecciona "Iniciar". b. El sistema verificar la hora si está dentro de lo establecido y solicita una confirmación del usuario desplegando una ventana emergente. c. El usuario confirma d. Se muestra el mapa con el recorrido, además de que se señala con marcadores los puntos de encuentro donde el conductor deberá parar para recoger a los pasajeros
Post-condición	Ninguna.
Excepciones	1.b. Aun no se puede iniciar el servicio.

Caso de uso	CU06. Finalizar recorrido
Propósito	Terminar el seguimiento, y realizar un control de la cantidad de pasajeros que abordaron el vehículo y si es que existe algo que notificar.
Descripción	Una vez el conductor haya llegado a su destino junto con los pasajeros el conductor podrá finalizar el recorrido, donde se le consultará cuales de los pasajeros asignados pudo recoger.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	Haber iniciado un recorrido.
Flujo principal	<ol style="list-style-type: none"> 1. Finalizar recorrido: <ol style="list-style-type: none"> a. Dentro de la ventana del recorrido el conductor seleccionar "Finalizar" b. El sistema revisa si el conductor se encuentra alrededor del punto final establecido. c. Despliega formulario donde se le pedirá marcar a todos los usuarios que no recogió y el motivo. d. El conductor rellena el formulario y guarda. e. Se guardan todos los datos y se procede a hacer las sanciones si las hubiera y a depositar el monto total ganado para el conductor. f. Se redirige a la pantalla de los servicios creados.
Post-condición	Ninguna.
Excepciones	Ninguna.

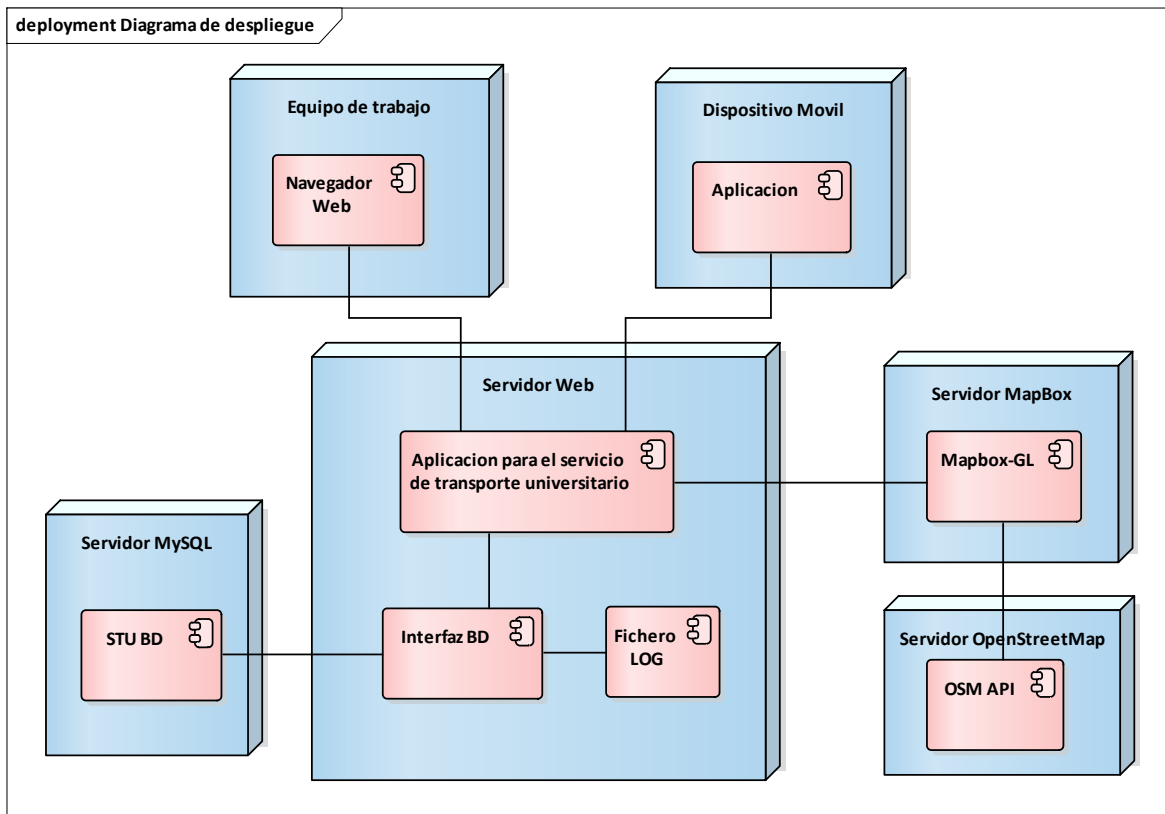
Caso de uso	CU07. Verificar identidad
Propósito	Verificar a través de un medio confiable la identidad tanto del conductor como del pasajero.
Descripción	Como parte importante de la seguridad del sistema la aplicación traerá un módulo de reconocimiento facial el cual permitirá realizar una comprobación en el momento de abordar el vehículo del conductor.
Actores	Ciudadano universitario
Actor iniciador	Ciudadano universitario
Pre-condición	El usuario que solicitó el servicio debe estar realizando el seguimiento. El usuario que ofreció el servicio debe haber iniciado el recorrido.
Flujo principal	<ol style="list-style-type: none"> 1. Verificar identidad del conductor:

	<ul style="list-style-type: none"> a. Dentro del seguimiento el usuario selecciona "Verificar identidad". b. El sistema consulta los datos del conductor y permite tomar una fotografía del conductor la cual será comparada a la guardada en los registros. c. Se toma la foto. d. Se notifica el resultado. <p>2. Verificar identidad del pasajero:</p> <ul style="list-style-type: none"> a. Dentro de la vista del inicio del recorrido el usuario selecciona "Verificar identidad". b. El sistema consulta los datos de todos los pasajeros y permite tomar una fotografía del pasajero que esta por aborda la cual será comparada a la guardada en los registros. c. Se toma la foto. d. Se notifica el resultado.
Post-condición	Ninguna.
Excepciones	Ninguna.

14.2. Modelo de arquitectura del software a nivel del código

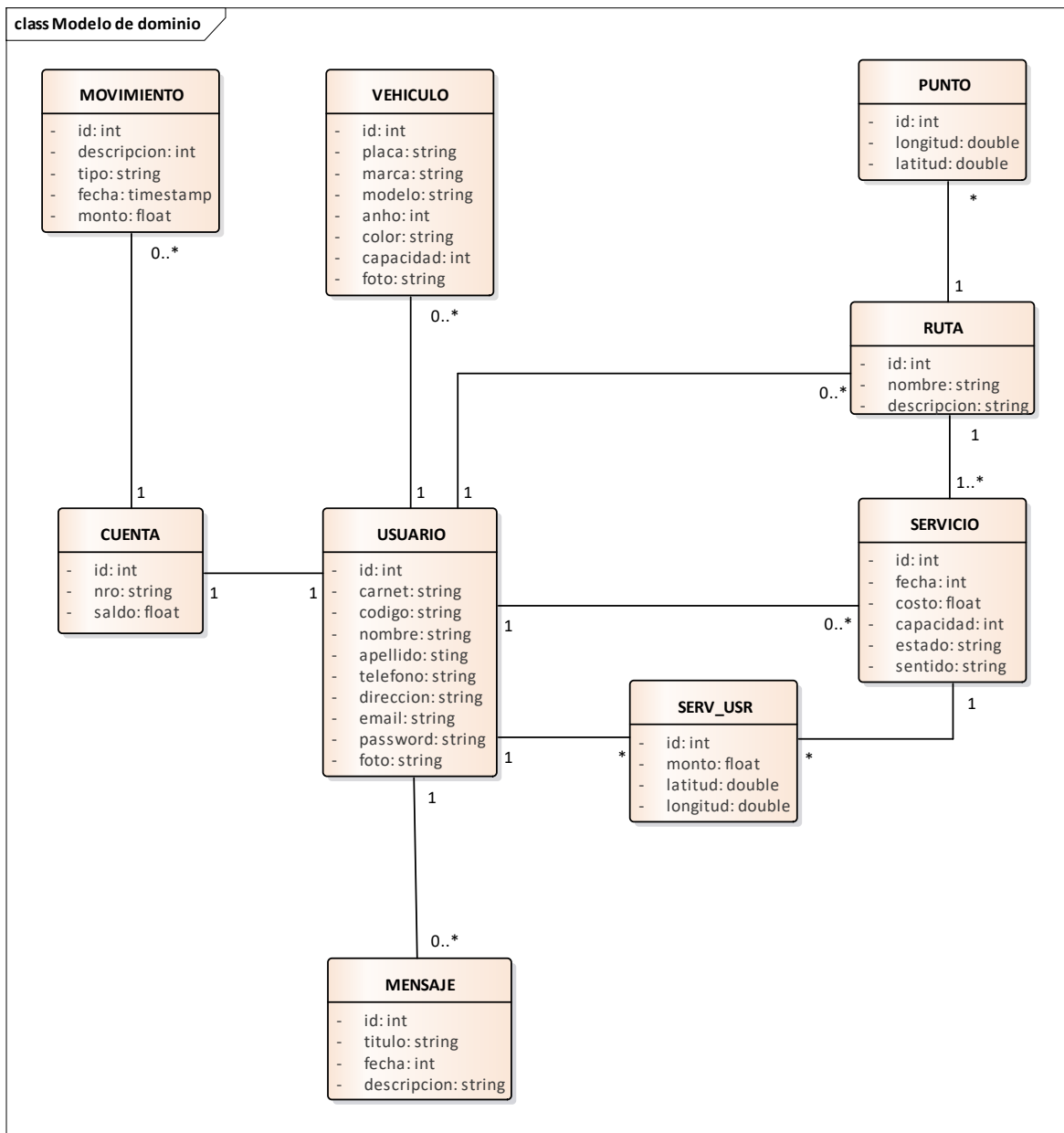


14.3. Modelo de arquitectura del software a nivel de despliegue



14.4. Modelo de persistencia

14.4.1. Diagrama de clases




14.4.2. Tabla de volumen


Cuenta:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
2	nro	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
3	saldo	float			No	<i>Ninguna</i>	


Movimiento:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
2	descripcion	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
3	tipo	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
4	monto	float			No	<i>Ninguna</i>	
5	fecha	int(10)		UNSIGNED	No	<i>Ninguna</i>	
6	cuenta_id 	int(11)			No	<i>Ninguna</i>	


User:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
2	nombre	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
3	apellido	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
4	foto	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
5	carnet	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
6	telefono	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
7	direccion	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
8	codigo	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
9	email	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
10	tipo	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
11	password	varchar(255)	latin1_swedish_ci		No	<i>Ninguna</i>	
12	remember_token	varchar(255)	latin1_swedish_ci		Si	<i>NULL</i>	
13	cuenta_id 	int(11)			No	<i>Ninguna</i>	


Vehículo:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	Ninguna	AUTO_INCREMENT
2	placa	varchar(255)	latin1_swedish_ci		No	Ninguna	
3	marca	varchar(255)	latin1_swedish_ci		No	Ninguna	
4	modelo	varchar(255)	latin1_swedish_ci		No	Ninguna	
5	año	varchar(255)	latin1_swedish_ci		No	Ninguna	
6	color	varchar(255)	latin1_swedish_ci		No	Ninguna	
7	foto	varchar(255)	latin1_swedish_ci		No	Ninguna	
8	capacidad	int(11)			No	Ninguna	
9	visible	char(1)	latin1_swedish_ci		No	1	
10	user_id 	int(11)			No	Ninguna	




Ruta:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	Ninguna	AUTO_INCREMENT
2	nombre	varchar(255)	latin1_swedish_ci		No	Ninguna	
3	visible	char(1)	latin1_swedish_ci		No	1	
4	descripcion	varchar(255)	latin1_swedish_ci		No	Ninguna	
5	user_id 	int(11)			No	Ninguna	

Punto:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	Ninguna	AUTO_INCREMENT
2	longitud	double			No	Ninguna	
3	latitud	double			No	Ninguna	
4	ruta_id 	int(11)			No	Ninguna	

Servicio:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	Ninguna	AUTO_INCREMENT
2	sentido	varchar(255)	latin1_swedish_ci		No	Ninguna	
3	fecha	int(10)		UNSIGNED	No	Ninguna	
4	estado	varchar(255)	latin1_swedish_ci		No	Ninguna	
5	cant_p	int(11)			No	Ninguna	
6	costo	float			No	Ninguna	
7	user_id 	int(11)			No	Ninguna	
8	vehiculo_id 	int(11)			No	Ninguna	
9	ruta_id 	int(11)			No	Ninguna	

Serv_Usr:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	Ninguna	
2	estado	varchar(255)	latin1_swedish_ci		Sí	En espera	
3	monto	float			No	Ninguna	
4	servicio_id 	int(11)			No	Ninguna	
5	user_id 	int(11)			No	Ninguna	

14.4.3. Script SQL

```
drop database if exists software;  
create database software;  
use software;
```

```
CREATE TABLE cuenta (  
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,  
    nro VARCHAR(255) NOT NULL,  
    saldo FLOAT NOT NULL  
);
```

```
CREATE TABLE movimiento (  
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,  
    descripcion VARCHAR(255) NOT NULL,  
    tipo VARCHAR(255) NOT NULL,  
    monto FLOAT NOT NULL,  
    fecha INT UNSIGNED NOT NULL,  
    cuenta_id INT NOT NULL,  
    FOREIGN KEY (cuenta_id)  
        REFERENCES cuenta (id)  
        ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    apellido VARCHAR(255) NOT NULL,  
    foto VARCHAR(255) NOT NULL,  
    carnet VARCHAR(255) NOT NULL,  
    telefono VARCHAR(255) NOT NULL,
```



```

    direccion VARCHAR(255) NOT NULL,
    codigo VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    tipo VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    remember_token VARCHAR(255),
    cuenta_id INT NOT NULL,
    FOREIGN KEY (cuenta_id)
        REFERENCES cuenta (id)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE vehiculo (
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
    placa VARCHAR(255) NOT NULL,
    marca VARCHAR(255) NOT NULL,
    modelo VARCHAR(255) NOT NULL,
    anho VARCHAR(255) NOT NULL,
    color VARCHAR(255) NOT NULL,
    foto VARCHAR(255) NOT NULL,
    capacidad INT NOT NULL,
    visible CHAR NOT NULL DEFAULT '1',
    user_id INT NOT NULL,
    FOREIGN KEY (user_id)
        REFERENCES users (id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE ruta (
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    visible CHAR NOT NULL DEFAULT '1',
    descripcion VARCHAR(255) NOT NULL,
    user_id INT NOT NULL,
    FOREIGN KEY (user_id)
        REFERENCES users (id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE punto (
    id INT AUTO_INCREMENT PRIMARY KEY,
    longitud DOUBLE NOT NULL,
    latitud DOUBLE NOT NULL,
    ruta_id INT NOT NULL,
    FOREIGN KEY (ruta_id)

```

```

        REFERENCES ruta (id)
        ON DELETE CASCADE ON UPDATE CASCADE
    );

CREATE TABLE servicio (
    id INT AUTO_INCREMENT PRIMARY KEY,
    sentido VARCHAR(255) NOT NULL,
    fecha INT UNSIGNED NOT NULL,
    estado VARCHAR(255) NOT NULL,
    cant_p INT NOT NULL,
    costo FLOAT NOT NULL,
    user_id INT NOT NULL,
    vehiculo_id INT NOT NULL,
    ruta_id INT NOT NULL,
    FOREIGN KEY (user_id)
        REFERENCES users (id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (vehiculo_id)
        REFERENCES vehiculo (id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (ruta_id)
        REFERENCES ruta (id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE serv_usr (
    id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
    estado VARCHAR(255) DEFAULT 'En espera',
    monto FLOAT NOT NULL,
    latitud DOUBLE NOT NULL,
    longitud DOUBLE NOT NULL,
    servicio_id INT NOT NULL,
    user_id INT NOT NULL,
    FOREIGN KEY (servicio_id)
        REFERENCES servicio (id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (user_id)
        REFERENCES users (id)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```
CREATE TABLE mensaje (  
  id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,  
  fecha INT UNSIGNED NOT NULL,  
  titulo VARCHAR(255) NOT NULL,  
  descripcion VARCHAR(255) NOT NULL,  
  user_id INT NOT NULL,  
  FOREIGN KEY (user_id)  
    REFERENCES users (id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

15. Referencias

DESARROLLO DE UN PROTOTIPO DE SOFTWARE MÓVIL PARA IMPLEMENTAR UN TAXIMETRO ENFOCADO AL CONTROL DEL SERVICIO, ANGÉLICA MARÍA BABATIVA GOYENECHÉ, UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

<https://www.paradigmadigital.com/dev/desarrollando-aplicaciones-moviles-nativas-con-react-native/>

<https://www.lanacion.com.ar/1670281-easy-taxi-una-aplicacion-que-llama-un-auto-con-solo-apretar-un-boton>

https://www.elconfidencial.com/tecnologia/2017-05-31/cabify-aplicacion-como-usar-servicios-tipos-coche_1391250/

<https://help.uber.com/es/h/738d1ff7-5fe0-4383-b34c-4a2480efd71e>

<https://medium.com/@MasterOfCodeGlobal/best-10-android-frameworks-for-building-android-apps-d2d0ee48e464>

https://www.ecured.cu/Arquitectura_de_software

<http://www.abiztar.com.mx/articulos/casos-a-incluir-casos-a-extender.html>