

**REINGENIERÍA DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE LA
MOVILIDAD DE LA COMUNIDAD UNIVERSITARIA.**

**ELKIN DARÍO MERCADO SIERRA
NATHALY HERNÁNDEZ AGUILAR**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍA FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2019**

**REINGENIERÍA DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE LA
MOVILIDAD DE LA COMUNIDAD UNIVERSITARIA.**

**ELKIN DARÍO MERCADO SIERRA
NATHALY HERNÁNDEZ AGUILAR**

**Trabajo de grado presentado como requisito para optar el título de ingeniero
de sistemas**

Director

**FERNANDO ANTONIO ROJAS MORALES
Magíster en Ciencias Computacionales**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍA FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2019

CONTENIDO

	Pág.
INTRODUCCIÓN	12
1. DESCRIPCIÓN GENERAL	15
1.1 PLANTEAMIENTO Y JUSTIFICACIÓN	15
1.2. OBJETIVOS.....	16
1.2.1 Objetivo general.....	16
1.2.2 Objetivos específicos	16
1.3. METODOLOGÍA	16
1.3.1 Desarrollo iterativo y creciente.....	17
2. ESTADO DEL ARTE.....	21
2.1 UBER.....	21
2.2. InDriver	23
2.3. WHEELS.....	25
3. MARCO DE REFERENCIA.....	28
3.1. TÉRMINOS POR CONOCER	28
3.1.1 Carpooling.....	28
3.2. ENFOQUE DEL PROYECTO	29
3.2.1 Reingeniería de software.	29
3.3. LENGUAJES DE PROGRAMACIÓN	30
3.3.1 Java.	30
3.4. LENGUAJES DE ETIQUETA.....	31
3.4.1. XML	31
3.5. ENTORNOS DE EJECUCIÓN	32
3.5.1. Node.Js.....	32
3.6. ENTORNOS DE DESARROLLO	33
3.6.1. Android Studio.	33

3.6.2 Microsoft Visual Studio.....	34
3.7. PLATAFORMA PARA EL DESARROLLO	35
3.7.1. Firebase.....	35
3.7.1.1. Cloud Firestore	35
3.7.1.2 Realtime Database	36
4. ANÁLISIS Y DESCRIPCIÓN DEL PROTOTIPO DE APLICACIÓN UISWHEELS VERSIÓN 1.....	37
4.1 DIAGRAMAS DE CLASE UISWHEELS VERSIÓN 1.0.....	40
4.2 REPORTES DE ERRORES.....	40
5. DISEÑO Y DESARROLLO DE LA APLICACIÓN	42
5.1. REALTIME DATABASE VS FIRESTORE.....	42
5.2. LISTA DE REQUERIMIENTOS.	45
5.3 TARJETAS CRC.....	48
5.4 DIAGRAMA DE CLASES DE UISWHEELS VERSIÓN 2.0.....	51
5.5 MODELO DE DIAGRAMA DE BASES DE DATOS DE UISWHEELS VERSIÓN 2.0.....	52
6. RESULTADOS.....	53
7. UISWHEELS 2.0.....	58
8. CONCLUSIONES	63
9. RECOMENDACIONES.....	65
BIBLIOGRAFÍA.....	66
ANEXOS	68

LISTA DE TABLAS

	Pág.
Tabla 1. Análisis de las funcionalidades de la aplicación UisWheels versión 1.	39
Tabla 2. Análisis de las funcionalidades de la aplicación UisWheels versión 1.	42
Tabla 3. Tarjeta CRC Clase Aplicación UisWheels con sus responsabilidades y sus colaboradores.....	49
Tabla 4. Tarjeta CRC Clase Usuario con sus responsabilidades y sus colaboradores.	49
Tabla 5. Tarjeta CRC Clase Conductor con sus responsabilidades y sus colaboradores.	49
Tabla 6. Tarjeta CRC Clase Pasajero con sus responsabilidades y sus colaboradores.	50
Tabla 7. Tarjeta CRC Clase Servicio con sus responsabilidades y sus colaboradores.	50
Tabla 8. Tarjeta CRC Clase Vehículo con sus responsabilidades y sus colaboradores.	50
Tabla 9. Tarjeta CRC Clase Calificación con sus responsabilidades y sus colaboradores.	51
Tabla 10. Tarjeta CRC Clase Mensaje con sus responsabilidades y sus colaboradores.	51
Tabla 11. Iteración No 1.....	53
Tabla 12. Iteración No 2.....	54
Tabla 13. Iteración No 3.....	54
Tabla 14. Iteración No 4.....	55
Tabla 15. Tabla de iteración No 5.....	56
Tabla 16. Iteración No 6.....	56
Tabla 17. Iteración No 7.....	57

Tabla 18. Iteración No 8.....57

LISTA DE IMAGENES

	Pág.
Imagen 1. Metodología usada en el proyecto.	20
Imagen 2. Capturas de Aplicación Uber	23
Imagen 3. Interfaz gráfica aplicación InDriver	25
Imagen 4. Interfaz gráfica aplicación Wheels.....	26
Imagen 5. Interfaz gráfica para la aplicación UisWheels.	38
Imagen 6. Ilustración de consulta y estructura jerárquica con manejo de sub-colecciones de datos de Firestore.	44
Imagen 7. Interfaz gráfica para mostrar los módulos de la aplicación UisWheels versión 1.	58
Imagen 8. Interfaz gráfica para perfil de usuario de la aplicación UisWheels 2.0 ..	59
Imagen 9. Interfaz gráfica para crear una ruta (primera captura) y mostrar ruta publicada (segunda captura) en la aplicación UisWheels versión 2.	60
Imagen 10. Interfaz gráfica para mostrar la información del servicio que ha publicado e iniciar y finalizarlo.....	61
Imagen 11. Interfaz gráfica para ver solicitudes de usuarios aceptados y por aceptar.	62
Imagen 12. Interfaz gráfica para la sala de chat creada para la aplicación UisWheels versión 2.	62

RESUMEN

TÍTULO: REINGENIERÍA DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE LA MOVILIDAD DE LA COMUNIDAD UNIVERSITARIA*

AUTORES: ELKIN DARIO MERCADO SIERRA Y NATHALY HERNANDEZ AGUILAR**

PALABRAS CLAVE: REINGENIERÍA, APLICACIÓN MÓVIL, CARPOOLING, JAVA, XML, FIREBASE.

DESCRIPCIÓN:

La movilidad urbana en la ciudad de Bucaramanga está conformada por los medios de transporte públicos, privados y humanos. El incremento de transporte en el área metropolitana de Bucaramanga ha traído consigo consecuencias no solo de congestión vehicular, sino también una problemática social, afectando el nivel de satisfacción de los habitantes de la ciudad que utilizan los medios de transporte público, ya que no se sienten satisfacción con el tiempo de viaje que duran los trayectos, motivo por el cual ha incrementado el uso del “transporte informal” el cual es ilegal, de baja confiabilidad y poco seguro. Con el fin de dar solución a esta problemática, se realizó un prototipo de aplicación móvil que permita solventar dichas insatisfacciones de los usuarios que en este caso son los estudiantes de la Universidad Industrial de Santander.

Pensando en ofrecer una nueva alternativa a lo existente, se realiza el proceso de implementación de la segunda versión de la aplicación móvil, ofreciendo mejoras e integrando nuevas funcionalidades para desplegar a los usuarios y obtener un producto completamente funcional que brinda los servicios de rutas dirigidas a la comunidad universitaria en tiempo real, permitiendo trazar a los usuarios rutas comunes donde podría ofrecer o unirse a una ruta.

Se mantiene el mismo entorno de desarrollo integrado (IDE) Android Studio oficial, con el fin de hacer el uso adecuado de los recursos del prototipo de aplicación para dar más competitividad y alcance. Se ha rediseñado la estructura, las funcionalidades y la interfaz de usuario.

* Trabajo de grado

** Facultad de ingeniería Físico-mecánicas. Escuela de ingeniería de sistemas e informática. Director MSc. Fernando Antonio Rojas Morales

ABSTRACT

TITLE: RE-ENGINEERING A MOBILE APPLICATION FOR THE MANAGEMENT OF THE MOBILITY OF THE UNIVERSITY COMMUNITY*.

AUTHORS: ELKIN DARIO MERCADO SIERRA AND NATHALY HERNANDEZ AGUILAR**

KEY WORDS: REINGENIERÍA, MOBILE APPLICATION, CARPOOLING, JAVA, XML, FIREBASE.

DESCRIPTION:

Urban mobility in the city of Bucaramanga is made up of public, private and human transport. The increase of transport in the metropolitan area of Bucaramanga has brought with it consequences not only of vehicular congestion, but also a social problem, affecting the level of satisfaction of the inhabitants of the city who use the means of public transport, since they do not feel satisfaction with the travel time of the journeys, which is why the use of "informal transport" has increased, which is illegal, unreliable and unsafe. In order to provide a solution to this problem, a prototype of a mobile application was developed to solve these user dissatisfactions, which in this case are the students of the Industrial University of Santander.

Thinking about offering a new alternative to the existing one, the process of implementation of the second version of the mobile application is carried out, offering improvements and integrating new functionalities to deploy the users and obtain a fully functional product that offers the services of routes directed to the university community in real time, allowing to trace to the users common routes where it could offer or join to a route.

The same official Android Studio integrated development environment (IDE) has been maintained, in order to make appropriate use of the resources of the application prototype to give more competitiveness and reach. The structure, functionalities and user interface have been redesigned.

* Trabajo de grado

** Facultad de ingeniería Físico-mecánicas. Escuela de ingeniería de sistemas e informática. Director MSc. Fernando Antonio Rojas Morales

INTRODUCCIÓN

En el mundo del desarrollo se encuentra un extenso espectro de herramientas tecnológicas, enfoques y metodologías que permiten la amplia comunicación entre comunidades. Las aplicaciones móviles se han convertido en un auge en la actualidad, gracias al uso de teléfonos inteligentes y tabletas.

Cada día se descubren nuevos métodos para que las aplicaciones sean más fáciles y versátiles, las personas realizan tareas que las aplicaciones han sido capaz de ejecutar, como enviar un email, solicitar servicio de Uber, comprar en línea, seguir el recorrido de un envío. Un Smartphone, o una tableta, puede hacer como mínimo estas tareas y más donde se creó la necesidad de portar estos dispositivos y a su vez tengan más importancia a nivel empresarial como a nivel individual.

Ni Uber inventó nada nuevo (los taxis y los móviles ya existían), ni Facebook sería lo que es sin su app móvil (pese a que la app móvil solo es un reflejo de su solución desktop), el uso de nuevos métodos como la reingeniería de software tiene como principal función reestructurar o rediseñar aplicaciones que con los avances tecnológicos tanto de hardware o de software se vuelven obsoletas, para lo cual es necesaria su operación sobre nuevas plataformas.

El tener un esquema de aplicaciones software centralizado en la reestructuración, evolución, implementación, y rediseño, genera ciertas responsabilidades al tomar decisiones, para solucionar esta problemática se decide tomar algunas acciones entre las que se determinó aplicar un proceso de reingeniería en lugar de la creación de una nueva aplicación, basándose principalmente en el hecho de contar con suficiente información (documentación, prototipo de aplicación, ejemplos de

aplicación móvil) que permite agregar y reutilizar las mismas para analizar y agregar nuevas funcionalidades.

El presente proyecto contempla un proceso de reingeniería para una aplicación móvil, donde la necesidad de investigar, aprender, y consolidar conocimientos adquiridos sobre herramientas de software, y adjuntando un perfecto ejemplo de arquitectura limpia al proyecto como lo es el patrón arquitectónico MVP - Model View Presenter (Modelo Vista Presentador) que garantiza el desacoplamiento de las diferentes capas que componen una aplicación.

Adicionalmente, para este proyecto se decide migrar de base de datos teniendo en cuenta las diferentes ventajas y desventajas que ofrecían, se migro de Realtime Database a Cloud Firestore, además se debe mantener la ideología de una base de datos flexible y escalable para la programación de servicios móviles, al igual que Firebase Realtime Database también mantiene sus datos sincronizados entre apps cliente, así mismo Cloud Firestore también ofrece integración en sus datos, flexibilidad, Actualización en tiempo real entre otras funciones claves.

- El presente documento se encuentra estructurado de la siguiente manera:

En el capítulo 1 se presenta el planteamiento, la justificación junto a los objetivos generales y específicos, la metodología propuesta para el desarrollo del proyecto a fin de cumplir con los objetivos descritos.

En el capítulo 2 se explica el estado del arte, el cual está enfocado en las dificultades existentes de movilidad en la ciudad con las soluciones que se han propuesto y se han puesto a disposición de los usuarios, donde destacan, el uso de Uber además de otras aplicaciones.

Por su parte, el capítulo 3 se recopila las consideraciones teóricas y las investigaciones previas que se tomaron como introducción a una noción básica de los conceptos usados durante el desarrollo del libro.

En el capítulo 4 se documenta el análisis, la descripción, características y funcionalidades generales del prototipo de aplicación UisWheels Versión 1, con el fin de obtener resultados acerca del funcionamiento de la aplicación.

En el capítulo 5 se presenta el proceso que se llevó a cabo para poder lograr los objetivos planteados en el proyecto, usando técnicas de desarrollo software y empleando la metodología que se había establecido.

En el capítulo 6 se presentan los resultados obtenidos, este capítulo es muy importante ya que en él se reflejan los objetivos alcanzados y se muestra el producto final, este capítulo incluye las tablas de requerimientos cumplidos a lo largo del desarrollo del proyecto.

1. DESCRIPCIÓN GENERAL

1.1 PLANTEAMIENTO Y JUSTIFICACIÓN

El área metropolitana de Bucaramanga cuenta con diferentes medios de transporte público, notando que algunos de estos medios son ineficientes o en algunos casos prestan un mal servicio. Con frecuencia la comunidad universitaria hace uso de buses convencionales que actualmente brindan una deficiente oferta de rutas para diferentes sectores de la ciudad, el servicio de Metrolínea el más usado con regularidad no ha cumplido con sus propósitos.

Según Vanguardia liberal, encuestando¹ algunos ciudadanos aseguraron que prefieren movilizarse en un vehículo que presta el servicio informal, por su economía y rapidez de movilidad dentro del Área Metropolitana de Bucaramanga a donde el servicio de Metrolínea no llega.

Con el fin de encontrar mejoras y utilizar los recursos, los estudiantes han desarrollado diversos medios que buscan mejorar la movilidad de mismos, para esto se crea una aplicación llamada UisWheels. Sin embargo, esta aplicación es solo un prototipo que tiene como fin realizar diversas pruebas con diferentes usuarios.

Surge la necesidad de reevaluar el prototipo y hacer de este, una aplicación completamente funcional algo, más consolidado que cumpla las expectativas y las necesidades de los usuarios al realizar o solicitar un servicio de transporte.

¹ El Sistema Integrado de Transporte Masivo, que en el papel se adaptó para hacer más ágiles los desplazamientos de los ciudadanos, encuesta tomadas y publicadas con fecha de martes 12 de febrero de 2019. Archivo / VANGUARDIA LIBERAL.

1.2. OBJETIVOS.

1.2.1 Objetivo general. Desarrollar una versión funcional para el prototipo de aplicación UisWheels, con el fin de mejorar y ampliar su accesibilidad a toda la comunidad universitaria, utilizando Android Studio como motor de desarrollo, reutilizando parte del producto desarrollado anteriormente.

1.2.2 Objetivos específicos

1. Evaluar el software UisWheels con el fin de establecer el estado de la funcionalidad actual.
2. Revisar y actualizar los requerimientos para el mejoramiento del aplicativo.
3. Diseñar e implementar la funcionalidad faltante al software móvil usando el paradigma de orientación a objetos para el sistema operativo Android.
4. Realizar pruebas con un grupo piloto en la nueva versión del Software móvil para movilidad de la comunidad universitaria.

1.3. METODOLOGÍA

La metodología para el desarrollo del proyecto comprende de 6 fases, en el desarrollo de software actualmente se encuentra una gran cantidad de métodos que ayudan en la elaboración del proyecto, estas herramientas sirven como guía para tener un marco de referencia de cómo encaminar el proyecto desde un comienzo hasta obtener un resultado final, se puede decir que son mecanismos que permiten encontrar una solución al problema propuesto, la planificación del proyecto, la variación de los requisitos durante el proyecto, el aseguramiento de la calidad del software, la creación de la documentación, entre otras.

Este proyecto consiste en desarrollar una aplicación que será usada por una gran cantidad de usuarios, por tanto, habrá momentos en donde se maneja una cantidad considerable de carga y la calidad de la herramienta dependerá de la preparación que se tenga para afrontar estos factores.

Un software que se crea para prestar un servicio a toda una comunidad requiere de una excelente preparación y de una buena acogida por parte de los usuarios, que en últimas, de la percepción de ellos dependerá el éxito del mismo. Por esto se ha decidido realizar el proyecto basándose en la metodología de desarrollo iterativo, ya que es un modelo que se enfatiza en la implementación extrema iterativa y en la usabilidad. Al finalizar todo el proceso y luego de un seguimiento muy riguroso, se tendrá como resultado un producto muy completo que cumplirá tanto con los requerimientos técnicos de los desarrolladores como con los requerimientos de usabilidad del usuario.

1.3.1 Desarrollo iterativo y creciente. Desarrollo iterativo y creciente (o incremental) es un proceso de desarrollo de software creado en respuesta a las debilidades o falencias del modelo tradicional de cascada.

Básicamente este modelo de desarrollo, que no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones), es uno de los más utilizados últimamente ya que, como se relaciona con novedosas estrategias de software y una programación extrema, es empleado en metodologías diversas.

el modelo consta de diversas etapas de desarrollo en cada incremento, las cuales inician con el análisis y finalizan con la instauración y aprobación del sistema.

Se planifica un proyecto en distintos bloques temporales que se le denomina iteración. En una iteración se repite un determinado proceso de trabajo que brinda

un resultado más completo para un producto final, de forma que quien lo utilice reciba beneficios de este proyecto de manera creciente.

Para llegar a lograr esto, cada requerimiento debe tener un completo desarrollo en una única iteración que debe de incluir pruebas y una documentación para que el equipo pueda cumplir con todos los objetivos que sean necesarios y esté listo para ser dado al cliente. Así se evita tener arriesgadas actividades en el proyecto finalizado.

Lo que se busca es que en cada iteración los componentes logren evolucionar el producto dependiendo de los completados de las iteraciones antecesoras, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo.

Una manera muy primordial para dirigir al proceso iterativo incremental es la de priorizar los objetivos y requerimientos en función del valor que ofrece el cliente.

Para apoyar el desarrollo de proyectos por medio de este modelo se tuvieron en cuenta fases que permiten el desarrollo incremental e iterativo, la fase de análisis fue primordial y tomada en cuenta siempre para poder visualizar las mejoras en el proyecto.

Ciclo de vida

La idea principal de esta metodología que se emplea en este proyecto finalmente es mejorar un prototipo de aplicación existente de manera incremental, permitiendo sacar ventaja de lo que se ha implementado a lo largo del desarrollo del prototipo anterior, extendiendo la versión entregable de la app.

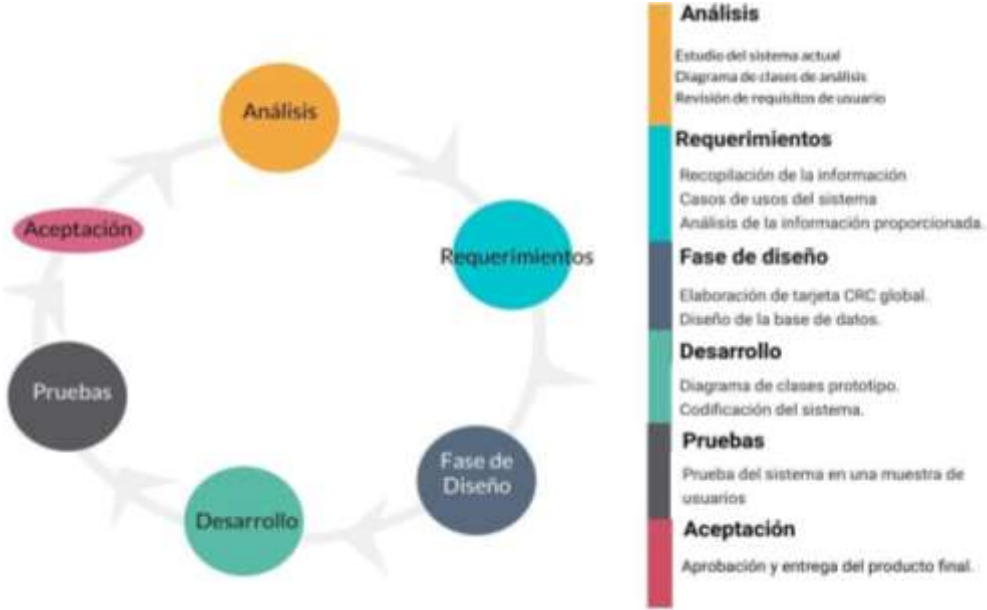
los pasos claves en el proceso son iniciar con implementación simple de los requerimientos de la app, e iterativamente mejorando la serie evolutiva de las versiones hasta que la aplicación completa esté implementada. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades. El proceso en sí se trata de crear a simple vista una versión mejorada del sistema que es la aplicación, la meta es crear un producto funcional con el que el usuario pueda interactuar y por ende retroalimentar el proceso.

En este proyecto es fundamental tener presentes los aspectos claves del problema existente en el prototipo de aplicación y proveer una solución lo suficientemente simple para ser comprendida e implementada a lo largo de cada iteración. Para guiar el proceso de iteración, una lista de controles de proyecto que se crea, y esta lista contiene un historial de todas las tareas que necesitan ser realizadas. Incluye cosas con nuevas funcionalidades para ser implementadas, y áreas de rediseño de la solución ya existente.

Esta lista de controles se revisará periódica y constantemente como resultado de la fase de análisis.

El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia. En la siguiente imagen (ver Imagen 1) se ilustra la metodología que se utilizó para realizar el proyecto.

Imagen 1. Metodología usada en el proyecto.



2. ESTADO DEL ARTE

Para analizar las aplicaciones que se mencionan en el estado del arte, es necesario identificar y enfatizar los factores más relevantes que han sido causa del desarrollo de estas, claramente estos factores problema han provocado soluciones necesarias, entendiendo como ello una forma de componer herramientas que hagan la movilidad más eficiente.

2.1 UBER

De cierta forma existen aplicaciones que han facilitado ciertas tareas del diario vivir de las personas, grandes ejemplos a nivel mundial, como Uber, una aplicación que surge de la inconformidad manifestada de los usuarios por el pésimo servicio de taxis que se prestaba en la ciudad de San Francisco, claramente estas aplicaciones forman competencia para empresas que a lo largo de la historia siguen y seguirán vigentes y por decirlo legalmente están registradas como empresas, pero al caso, Uber surge como una idea en una ciudad y ahora es un ente importante a nivel mundial.

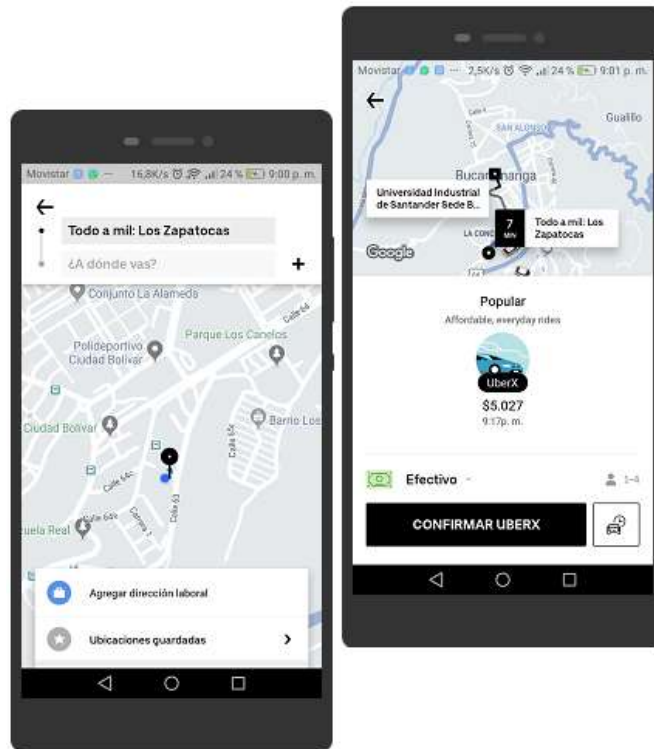
Uber actúa fundamentalmente y, en palabras llanas, como una empresa cuya actividad principal consiste en la mediación entre interesados –conductor y pasajero/s- en alcanzar un destino partiendo de un lugar determinado. Por supuesto, el término “empresa” ya indica que ostenta ánimo de lucro el desarrollo de esa actividad. Pese a haber recientemente diversificado su actividad y alcanzado otros sectores, el transporte de personas sigue siendo su seña de identidad.

Comencemos por describir esta app de Uber, disponible para los dos principales sistemas operativos de dispositivos móviles (Android y iOS). ¿Cómo opera? La aplicación hace gala de los sistemas integrados en buena parte de los teléfonos inteligentes o de última generación, ya asentados entre el gran público. Utilizándolos consigue conectar a un determinado usuario con el conductor que se encuentre más cerca del primero. También se prevé el hecho que el usuario pueda desconocer la dirección del emplazamiento, caso en que el conductor podría recogerle pactando el lugar.

Otro aspecto que considerar sería la elección del grado de flexibilidad horaria, variable en función de las preferencias de los conductores. La compañía recalca el hecho que ellos son “dueños de su tiempo” y hace hincapié en que en el momento que lo requieran pueden practicar tal actividad. Hay que mencionar también que los usuarios pueden valorarlos mediante una nota de 0 a 5 estrellas- de modo que podrán dejar constancia de primera mano de cómo estuvo el servicio prestado por dicho conductor, cuán hablador es, cómo es su estilo de conducción, etc. Información que puede ser consultada por otros usuarios de esta aplicación.

A continuación, se muestra (ver Imagen 2). como se ve una parte de la aplicación mencionada.

Imagen 2. Capturas de Aplicación Uber



2.2. InDriver

Otra aplicación que también sirve para mitigar el mal servicio de transporte en el área metropolitana es InDrive, surge en Rusia cuando unos estudiantes crearon el grupo de 'Controladores Independientes' (inDriver's) en la red social VKontakte, muy popular en dicho país.

Allí los miembros publicaban sus itinerarios de viaje y el precio que estaban dispuestos a pagar, y los conductores contactaban a sus clientes y tomaban sus pedidos. En cuestión de meses al grupo se habían afiliado unas 50.000 personas. Y el crecimiento no se detuvo.

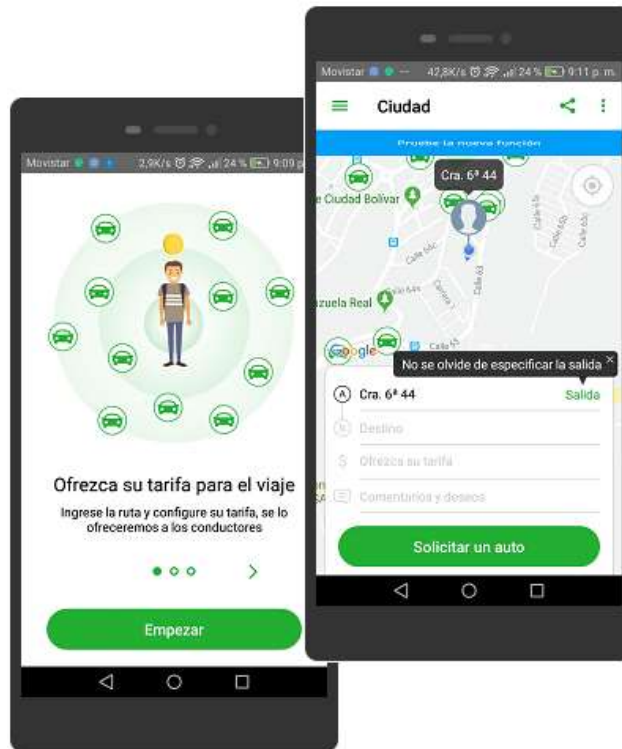
Un año después la comunidad se fusionó con la compañía Sinet que, sobre la base existente, creó un servicio de transporte de alta tecnología. Así nació InDriver, una idea que se emancipó a los servicios de taxis centralizados desde el primer día.

"A diferencia de otros servidores, InDriver no fija una tarifa mediante un algoritmo. El costo del traslado se determina de un modo democrático. Es como una subasta en tiempo real. Las partes son libres de decidir el precio. Tampoco establecemos el recorrido que deben tomar", dijo a Sputnik Eduardo Abud, gerente de relaciones públicas latinoamericano de InDriver.

Hoy su servicio alcanza a más de 20 millones de personas en más de 200 ciudades y en 14 países del mundo. En América Latina ya opera en ocho países: México, Colombia, Guatemala, Perú, Chile, El Salvador, Ecuador y Brasil, y tiene previsto mudarse a Costa Rica, República Dominicana, Panamá y Honduras.

La diferencia con Uber no es solo su forma de operar, sino su filosofía. "InDriver les devuelve a las personas la libertad, ya sea del precio, del modelo del auto, del conductor. El resultado es que se ahorran cantidades sustanciales", explicó Abud.

Imagen 3. Interfaz gráfica aplicación InDriver



2.3. WHEELS

Wheels es la plataforma digital que de alguna forma es segura ya que solo puedes interactuar con tu red de amigos de universidad o demás, gracias a los correos institucionales que brindan ciertas entidades, además permite agregar personas a tu red social y te permite acceder a ella, también puedes interactuar con tus contactos, vecinos, personas que están en tu gimnasio, etc. Interactuar en grupos de WhatsApp entre otros.

Se puede asumir para cada ruta uno de 3 roles las cuales son: WHEELER (Pasajero), TAXI SHARER (Compartidor de taxi) o DRIVER (Conductor), el pasajero puede disfrutar el confort de los carros o servicios de taxi allegas, cuando tomas el

segundo rol entonces divides la trifa con las personas que te acompañan en el viaje, y el tercero te permite compartir tu vehículo y puedes diseñar tu ruta.

Wheels es una aplicación que permite programar por días o meses los desplazamientos que creas que son recurrentes y convenientes, una vez establecida toda la información el sistema creará los match que mejor se acomodan a cada uno de los intereses.

Imagen 4. Interfaz gráfica aplicación Wheels.



En la medida podremos también se ha vivido ciertas inconformidades a partir de la falta de soluciones de herramientas, tal cual vemos que el Área Metropolitana de la ciudad de Bucaramanga ha presentado en los últimos años un crecimiento vehicular que está por sobrepasar la capacidad de movilidad en algunos sitios estratégicos, situación asociada con el incremento del flujo vehicular urbano, Al igual la contaminación vehicular por emisiones gaseosas de combustión es uno de los

problemas ambientales más importantes que deben ser estudiados en Colombia. por tal motivo para mitigar el problema de movilidad y contaminación ambiental en la ciudad de Bucaramanga, han surgido ideas de un servicio de transporte colectivo, donde diferentes seres humanos comparte un vehículo de transporte, con un destino inicial y final en común.

Para dar solución a esta realidad, se creó un prototipo de aplicación llamado UisWheels creado por dos jóvenes universitarios, UIS Wheels es una aplicación de movilidad desde y hacia el campus universitario, esta se desarrolló como una propuesta de solución al problema de movilidad y contaminación ambiental de la ciudad de Bucaramanga mencionada antes, enfocada a la comunidad de la universidad industrial de Santander donde el objetivo principal es permitir por medio de una aplicación Android compartir lugar en sus vehículos para el traslado desde casa a la universidad y viceversa.

Esta idea, quedó como un prototipo no funcional, donde carece de una estructura estable para su buen funcionamiento, y claramente la aplicación tiene algunas opciones que no se encuentran habilitadas o no cumplen la función esperada por el usuario, sin embargo, este prototipo abrió camino para hacer un nuevo proyecto como lo es el que se presenta en este documento.

3. MARCO DE REFERENCIA

El proyecto se basa en un conjunto de ideas y conceptos que usan una terminología específica, la cual es citada a lo largo del documento. No obstante, se presenta una noción básica de términos tales como: Carpooling, Reingeniería, Aplicación Móvil, Java, Xml, Firebase, Realtime Database y Cloud Firestore.

3.1. TÉRMINOS POR CONOCER

3.1.1 Carpooling. Carpooling es un paradigma de movilidad, es más una tendencia internacional que ayuda a reducir el tráfico en la ciudad y el caos vehicular teniendo en cuenta el cuidado del medio ambiente. A través de una plataforma digital, conductores y pasajeros pueden buscar viajes disponibles y autos con asientos libres, respectivamente. Es decir, una fórmula más sencilla: viajar varias personas en un mismo vehículo para llegar a un destino común. su objetivo es claro y preciso: optimizar el uso de auto al maximizar la cantidad de asientos utilizados.

El contacto se produce a través de sitios gratuitos de internet donde conductores y pasajeros publican los trayectos que necesitan recorrer, con fechas y horarios. Cuando hay coincidencia, el sistema les permite ponerse en contacto, acordar la división de costos y empezar a viajar juntos.

El carpooling cuenta además con múltiples beneficios desde las tres aristas de la sustentabilidad: económica, social y ambiental. La iniciativa surgió en los Estados Unidos en 1970 ante la crisis del petróleo, como una alternativa y necesidad para achicar los costos. Luego, al transcurrir los años, se popularizó como un hábito que,

más allá del ahorro económico, promueve la reducción del tráfico y disminuye el impacto ambiental de los gases emitidos².

Utilizar aplicaciones para viajar en un carro junto a completos desconocidos es una práctica que la economía colaborativa ha creado y que, según los expertos, ayuda a resolver problemas tangibles a un costo menor.

Se denomina 'carpooling' y en Colombia se ha definido como el hecho de conectar a pasajeros con conductores que van para el mismo destino. Además del obvio ahorro para el usuario, dicen que así logran contribuir con el cuidado del medio ambiente³.

3.2. ENFOQUE DEL PROYECTO

3.2.1 Reingeniería de software. La reingeniería de sistemas heredados tiene por finalidad reestructurar o transformar viejos sistemas en aplicaciones más fáciles de mantener, con entornos más agradables e integradas en nuevas plataformas de hardware/Software. parte de la misión de la reingeniería de software es intentar modernizar un sistema legado que ha quedado obsoleto en cuanto a operatividad, aspecto y software de base sobre el que se ejecuta, para mejorar su eficiencia y evolucionar su funcionalidad.

Existen varias las referencias que se pueden encontrar sobre concepto de reingeniería. Algunos, como Arnold⁴, la definen como una actividad que mejora la comprensión del software, o bien, lo prepara o mejora para incrementar su facilidad

² INFOBAE. Visible: Carpooling: qué es y cómo funciona. 2015 [en línea] disponible en: <https://www.infobae.com/2015/12/16/1776933-carpooling-que-es-y-como-funciona/>

³ CAMACHO Laura. Carpooling, la alternativa de transporte que ahorra tiempo y dinero: El tiempo. 2018 [en línea] disponible en: <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/que-es-y-como-funciona-el-carpooling-compartir-carro-en-colombia-237074>

⁴ ARNOLD, R.S.: Software Reengineering. IEEE Computer Society Press, 1993

de mantenimiento, reutilización o evolución. Para otro ⁵, es el examen y alteración de un sistema para reconstruirlo en una nueva forma y subsiguiente implementación de esa forma ⁶. Otros los ven como el proceso de ingeniería inversa seguida de una ingeniería directa. El concepto de reingeniería está muy relacionado con el concepto de reutilización, la reutilización está enfocada a mejorar la calidad y reducir el esfuerzo haciendo uso de partes de un sistema en un nuevo contexto. En definitiva, el concepto de reingeniería de software se refiere a la reutilización de sistemas heredados, pero transformándolos para hacerlos más mantenibles.

Según ⁷, se puede definir diez elementos del software que pueden reutilizarse: planes de proyecto, estimación de coste, arquitectura, especificaciones y modelos de requisitos, diseños, códigos fuente, documentación de usuario y técnica, interfaces humanas, datos y casos de prueba.

3.3. LENGUAJES DE PROGRAMACIÓN

3.3.1 Java. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo ⁸.

⁵ CHIKOFISKY, E.J. and CROSS, J.H.: Reverse engineering and design recovery: A taxonomy. IEEE Software. 7(1), 1990, 13-17.

⁶ BIGGERSTAFF T. and PerLis A.: Software Reusability. Addison-Wesley. November, 1990.

⁷ MONTSERRAT, M. S., MORENO, G. Metodología de reingeniería del software para la remodelación de aplicaciones científicas heredadas. Technical Report, Informe Técnico – Technical Report, DPTOIA-IT-2004-003. 2014

⁸ IBM. Conceptos básicos del lenguaje Java. [en línea] disponible en: <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/index.html>

La principal ventaja de JAVA es que es un lenguaje multiplataforma, esto quiere decir que las aplicaciones programadas en JAVA se pueden ejecutar en diferentes sistemas operativos incluyendo sistemas operativos móviles. Esta ventaja se debe a que las aplicaciones JAVA son compiladas en bytecode para finalmente ser interpretadas por la máquina virtual de JAVA alojada en el sistema operativo.

En la actualidad JAVA es uno de los lenguajes más utilizados a nivel mundial, ya que cuenta con una gran cantidad de fuentes de apoyo que ayudan en gran medida a los desarrolladores a la hora de crear software.

3.4. LENGUAJES DE ETIQUETA

3.4.1. XML . XML es uno de los estándares más utilizados en la actualidad para codificar información. Es ampliamente utilizado en internet, también se utiliza para múltiples usos en el SDK de Android. Entre otras cosas es utilizado para definir Layouts, animaciones, AndroidManifest.xml, etc.

Una de las mayores fortalezas de la plataforma Android es que se aprovecha el lenguaje de programación Java y sus librerías.⁹ El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción de esta. Lo mismo ocurre en lo referente a trabajar con XML, Java dispone de gran cantidad de API con este propósito, pero no todas están disponibles desde Android.

⁹ REINO ROMERO A. Versión 2.0 (26 de enero, 2000). [en línea] disponible en: <http://deim.urv.cat/~pedro.garcia/SOB/IntroXML.pdf>

3.5. ENTORNOS DE EJECUCIÓN

3.5.1. Node.Js. Node.js es un entorno JavaScript por parte, un servidor listo para codificar. Node proporciona una manera fácil para construir programas de red escalable.¹⁰ En lenguajes como Java™ y PHP, cada conexión genera un nuevo hilo que potencialmente viene acompañado de 2 MB de memoria. En un sistema que tiene 8 GB de RAM, esto da un número máximo teórico de conexiones concurrentes de cerca de 4.000 usuarios. La aplicación soporte más usuarios, se necesitará agregar más y más servidores.

Un usuario puede estar usando diferentes servidores para cada solicitud, así que cualquier recurso compartido debe almacenarse en todos los servidores. Por todas estas razones, el cuello de botella en toda la arquitectura de aplicación Web (incluyendo el rendimiento del tráfico, la velocidad de procesador y la velocidad de memoria) era el número máximo de conexiones concurrentes que podía manejar un servidor.

Node resuelve este problema cambiando la forma en que se realiza una conexión con el servidor. En lugar de generar un nuevo hilo de OS para cada conexión (y de asignarle la memoria acompañante), cada conexión dispara una ejecución de evento dentro del proceso del motor de Node. Node también afirma que nunca se quedará en punto muerto, porque no se permiten bloqueos y porque no se bloquea directamente para llamados E/S. Node afirma que un servidor que lo ejecute puede soportar decenas de miles de conexiones concurrentes.

Node ejecuta V8 JavaScript. El motor V8 JavaScript es el motor JavaScript subyacente que Google usa con su navegador Chrome, con el V8, Google creó un intérprete ultra-rápido escrito en C++, con otro aspecto único: puede descargar el

¹⁰ ABERNETHY Michael. IBM. (14 de junio, 2011). IBM. Node.Js. [en línea] disponible en: <https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/index.html>

motor e incorporarlo a cualquier aplicación que desee. No está restringido a ejecutarse en un navegador. Así, Node en realidad usa el motor V8 JavaScript escrito por Google y le da otro propósito para usarlo en el servidor.

3.6. ENTORNOS DE DESARROLLO

3.6.1. Android Studio. Android es un sistema operativo, inicialmente diseñado para teléfonos móviles como los sistemas operativos iOS (Apple), FireFoxOS (Mozilla) y Blackberry OS. En la actualidad, este sistema operativo se instala no sólo en móviles, sino también en múltiples dispositivos, como tabletas, televisores, discos duros multimedia, miniordenadores, etcétera. Está basado en Linux, que es un núcleo de sistema operativo libre, gratuito y multiplataforma. Este sistema operativo permite programar aplicaciones empleando una variación de Java llamada Dalvik (o ART a partir de la versión 5.0 de Android) y proporciona todas las interfaces necesarias para desarrollar fácilmente aplicaciones que acceden a las funciones del teléfono (las llamadas, la agenda, etcétera) utilizando el lenguaje de programación Java. Su sencillez principalmente, junto a la existencia de herramientas de programación gratuitas, es la causa de que existan cientos de miles de aplicaciones disponibles, que extienden la funcionalidad de los dispositivos y mejoran la experiencia del usuario.

Proyecto libre (Open Source) una de las características más importantes de este sistema operativo reside en que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Es muy popular entre los fabricantes de teléfonos y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos. Aula Mentor 8 Cualquier programador puede descargarse el código fuente, inspeccionar, compilarlo e incluso modificarlo. Por esta razón, hay muchas versiones paralelas de Android de grupos de desarrollo de lo más variopinto.

3.6.2 Microsoft Visual Studio. Visual Studio es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, permitiendo a los desarrolladores crear sitios y aplicaciones web, así como otros servicios web en cualquier entorno que soporte la plataforma.

En palabras más específicas, Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todo el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.

Desarrollo móvil: Visual Studio 2017 con Xamarin hace más rápido y fácil para los desarrolladores compilar, conectar y ajustar aplicaciones móviles para Android, iOS y Windows.

Adicional se le ha dado un enfoque renovado para mejorar la eficiencia de las tareas fundamentales que los desarrolladores desempeñan diariamente. Desde una nueva instalación más ligera y modular adaptada a las necesidades del desarrollador, un IDE más rápido desde el arranque al apagado, una nueva manera de ver, editar y depurar cualquier código sin proyectos ni soluciones.

3.7. PLATAFORMA PARA EL DESARROLLO

3.7.1. Firebase. Firebase es una plataforma integral de desarrollo para dispositivos móviles, Firebase proporciona funciones como estadísticas, bases de datos, informes de fallas y mensajería, de manera que puedas ser más eficiente y enfocarse en los usuarios. utiliza la infraestructura de Google y se escala automáticamente, incluso para las apps más grandes. Los productos de Firebase funcionan a la perfección por sí solos. ¹¹ Sin embargo, debido a que comparten datos y estadísticas, funcionan aún mejor juntos.

Las API de Firebase se incluyen en un único SDK, por lo que puede expandirse a más idiomas y plataformas, incluidos C++ y Unity, con Firebase como tu back-end unificado.

Firebase te permite compilar aplicaciones más potentes, seguras y escalables con una infraestructura de primer nivel.

3.7.1.1. Cloud Firestore

- Almacena y sincroniza los datos de tu app a escala global.

Almacena y sincroniza datos entre usuarios y dispositivos a escala global mediante una base de datos NoSQL alojada en la nube ¹²]. Cloud Firestore te ofrece sincronización en vivo, soporte sin conexión y consultas de datos eficaces. Su integración con otros productos de Firebase te permite compilar aplicaciones verdaderamente sin servidores.

¹¹ GÓMEZ TORRES Juan Guillermo Diferencias entre Realtime Database vs Firestore. [en línea] disponible en: <https://medium.com/@jggomez/diferencias-entre-realtime-database-vs-firestore-94364a2cc09e>

¹² CLOUD FIRESTORE. Documentación. Firebase: [en línea] disponible en: <https://firebase.google.com/docs/firestore/?hl=es-419>

3.7.1.2 Realtime Database

- Almacena y sincroniza datos en tiempo real.

Almacena y sincroniza datos con nuestra base de datos NoSQL alojada en la nube. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen disponibles cuando la app no tiene conexión.

Firestore Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando compilas aplicaciones multiplataforma con nuestros SDK de iOS, Android y JavaScript, todos tus clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

4. ANÁLISIS Y DESCRIPCIÓN DEL PROTOTIPO DE APLICACIÓN UISWHEELS VERSIÓN 1

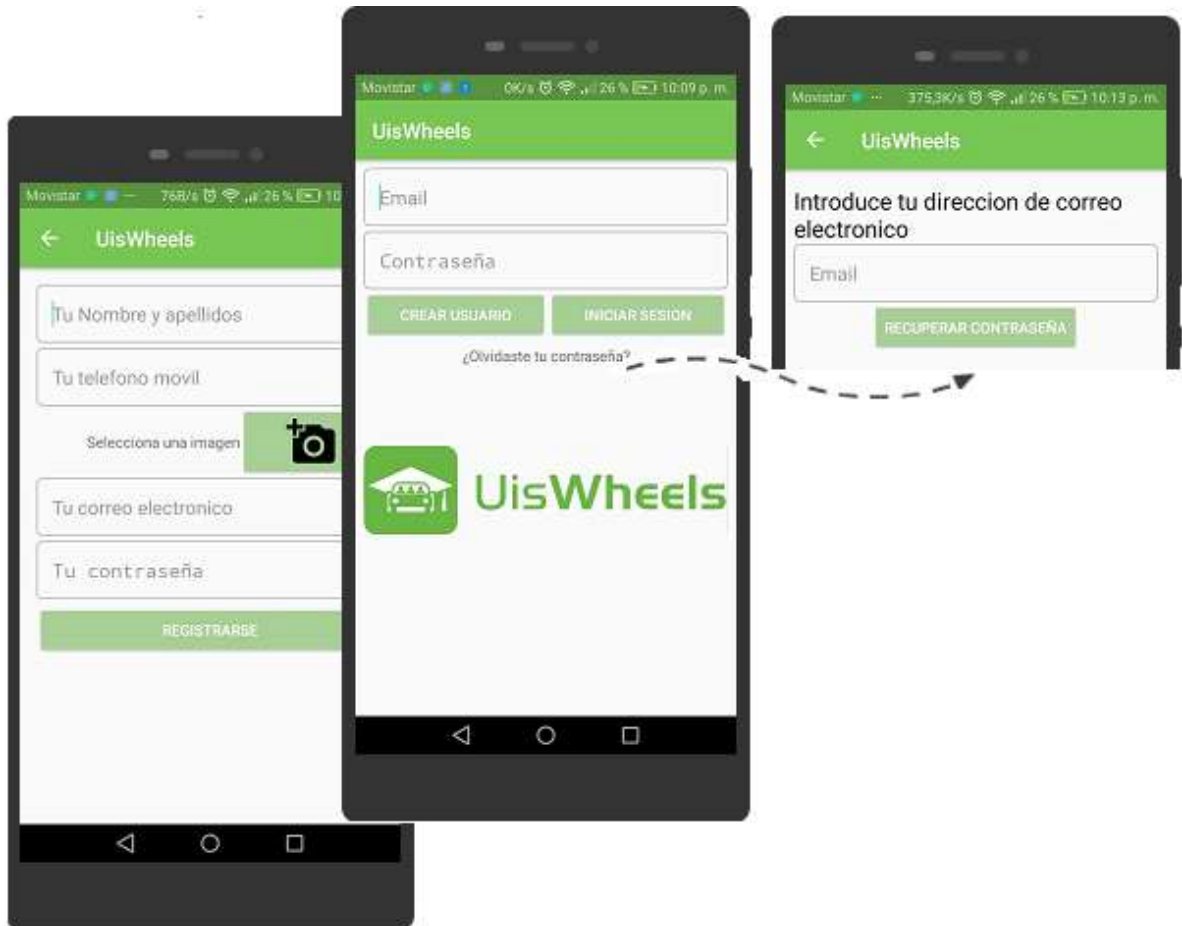
En este capítulo se presenta una descripción previa del prototipo de aplicación existente y la descripción de la aplicación a realizar y de forma iterativa las diferentes propuestas diseñadas para su construcción a medida que iteraciones evolutivas van explicando cada uno de sus componentes hasta llegar a la versión final.

Para el desarrollo de este proyecto se utilizó la metodología iterativa y creciente como como se mencionó en los capítulos anteriores, de este modo también se analizar el prototipo de aplicación que se va a implementar, en este sentido y enfatizando esta parte del capítulo en UisWheels se puede decir que: es un prototipo de aplicación existente en la tienda PlayStore.

UisWheels tiene funciones básicas para poder crear un nuevo usuario, tales como: ingresar el nombre, teléfono, cargar una imagen, ingresar un correo electrónico, y diligenciar una contraseña (ver imagen 5.). Cuando el usuario ingresa una contraseña incorrecta o la olvida, tiene la opción de volver a ingresar una nueva contraseña, esto lo puede hacer ingresando el correo con el que se registró, de este modo se envía un enlace y continua con el proceso, estos pasos el prototipo los hace de forma correcta.

- Interfaz gráfica para registro e inicio de sesión de usuario y recuperación de contraseña de la aplicación UisWheels versión 1.

Imagen 5. Interfaz gráfica para la aplicación UisWheels.



El prototipo de aplicación también cuenta con un menú principal, que cuenta con las siguientes opciones: Home, Perfil, Tus Rutas, y configuración que conforma los reportes de problemas y cerrar sesión.

Para realizar el análisis de la aplicación se hicieron pruebas con varios dispositivos móviles, en la siguiente tabla (tabla 1.) se muestran las funcionalidades de la aplicación y las observaciones que se realizaron al interactuar con ella.

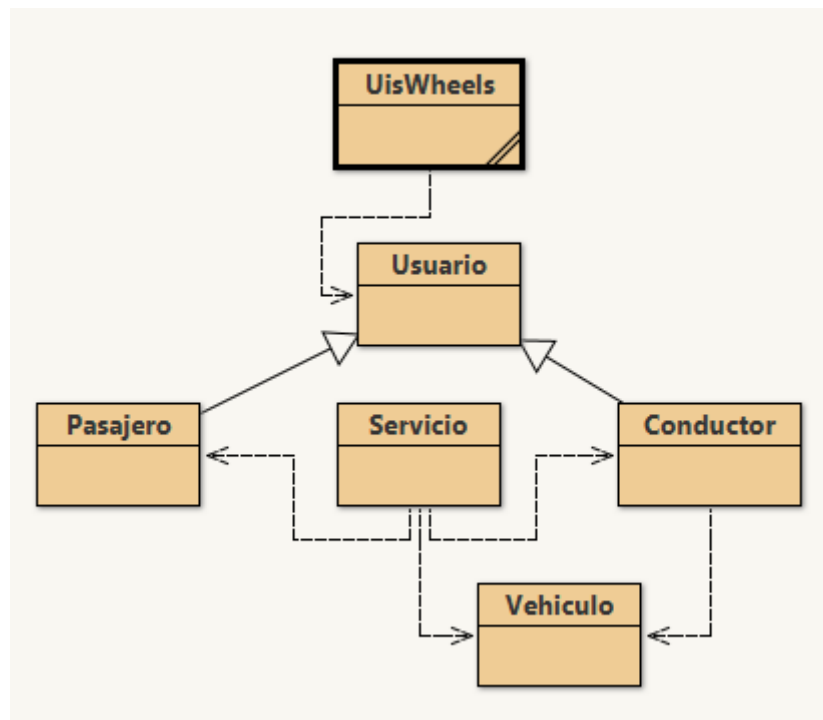
Tabla 1. Análisis de las funcionalidades de la aplicación UisWheels versión 1.

Módulos	Funcionalidad	Observación
Home	Muestra rutas creadas en fechas futuras.	No actualiza las publicaciones de rutas después de pasada la fecha
Perfil	Muestra nombre, número de teléfono. Permite ver marca, modelo y placa del vehículo. Muestra imagen de usuario.	Falta implementar el botón de calificar usuario. No es posible editar la imagen de perfil, no carga la imagen en algunos momentos. Placa deber ser un campo obligatorio. Modelo del vehículo es un campo incensario. Un usuario no debería calificarse a el mismo.
Tus Rutas	Permite editar las rutas. Muestra todas las rutas creadas por el usuario.	Falta implementar el botón de borrar las rutas. El creador de una ruta no deber tener la opción de unirse a la ruta. Borrar de la base de Datos los servicios que han finalizado. No debería permitir editar una ruta cuando ya se han unido usuarios. El swich de activar ruta no está implementado. El usuario puede solicitar unirse aun estando lleno el cupo de un servicio. Se están generando errores al crear nuevos servicios. No hay notificaciones para conductor o pasajero.
Reportar problemas	Permite reportar un problema por correo.	
Cerrar sesión	Realiza de forma correcta el cierre de sesión.	

4.1 DIAGRAMAS DE CLASE UISWHEELS VERSIÓN 1.0

Por medio de las tarjetas CRC anteriormente mencionadas fue posible realizar el diagrama de clases de la versión anterior con el fin de analizar la forma en como interactuaban las diferentes clases en el sistema anterior.

Imagen 6. Modelo del Diagrama de clases UisWheels 1.0



4.2 REPORTE DE ERRORES

Se realiza un análisis profundo de la aplicación uiswheels versión 1.0, en este análisis es posible encontrar una serie de errores a tener en cuenta para la realización de dicho proyecto.

Bloquear estrellas de puntuación: menú>perfil> estrellas bloquear las estrellas de puntuación y ajustar su valor según la calificación general del usuario.

Activar ruta: mis rutas>ruta el swich de activar ruta no funciona, no es posible activar la ruta, esta no es visualizada en otro dispositivo.

Unirse un usuario a un servicio: un usuario no puede volver a tomar otro servicio con el mismo conductor.

No hay notificar conductor: no hay notificaciones al conductor de solicitud de cupo.

No hay notificaciones al pasajero: el pasajero nunca sabe si el conductor acepto o no su solicitud.

Registrar cuenta: en un dispositivo Huawei no me permitió crear cuenta, si hay datos en la actividad de inicio no permite registrar al usuario.

Imagen de perfil: a veces no sale la imagen de perfil.

Múltiples calificaciones: un usuario puede hacer múltiples calificaciones a otro.

Búsqueda repite servicios: cuando se crea un nuevo servicio los otros dispositivos se cambia el activity a crear servicio, o les genera error.

Solicitar servicio: un usuario puede solicitar servicio al haber cupo completo.

5. DISEÑO Y DESARROLLO DE LA APLICACIÓN

5.1. REALTIME DATABASE VS FIRESTORE

Cuando se realizó el análisis de la aplicación se realizó la observación sobre la forma como se estaban almacenando los datos suministrados por los usuarios en ella, a partir de este análisis se supo que UisWheels usa una plataforma de desarrollo móvil en la nube llamada Firebase, esta plataforma suministra una base de datos llamada Realtime Database que reamente es la primera base de datos lanzada por ellos, pero tiene baja latencia y capacidad para consultas en tiempo real, lo cual es una desventaja para una aplicación, por tal razón se trata de indagar que servicios podría ofrecer Firebase, se encontró que se lanzó recientemente una nueva base de datos llamada Firestore.

Firestore es una nueva base de datos, tiene una estructura más amigable en su entorno de diseño, trae mejoras en los SDK para la realización de consultas, donde se realizó una comparación de alcance, estructura y otras características que encontramos en estas bases de datos con el fin de usar la mejor, en la siguiente tabla (Tabla 2.) se muestras algunas comparaciones entre estas dos bases de datos.

Tabla 2. Análisis de las funcionalidades de la aplicación UisWheels versión 1.

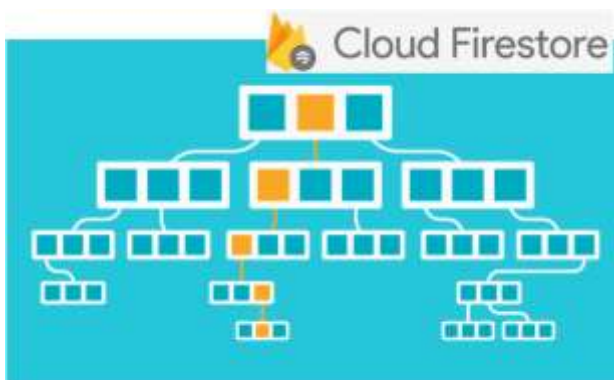
	Realtime Database	Firestore
Estructura de datos	Realtime Database almacena los datos en JSON, pero se organiza o se representa en la plataforma como un árbol. Esta forma es compleja cuando	Firestore tiene una estructura visible en la consola más sencilla y fácil de escalar cuando se tiene grandes jerarquías con el manejo de sub-colecciones.

	Realtime Database	Firestore
	crecen las anidaciones, las jerarquías o la información.	
Querys	Realtime Database soporta solo filtrar y ordenar por un campo en un solo query. Además, cuando se realiza una consulta retorna todo el subárbol del nodo	Firestore soporta consultas compuestas y puede combinar filtros con ordenamiento. Al consultar solo trae el documento o los documentos sin traer sub-colecciones o hijos, esto se pide a demanda. Por otro lado, las consultas son indexadas por defecto. Soporta también el ordenamiento descendente.
Escrituras y transacciones	Soporta escritura de forma individual.	Firestore soporta operaciones batch las cuales se pueden realizar operaciones de insertar, actualizar o eliminar al mismo tiempo.
Fiabilidad y rendimiento	Realtime Database es una base de datos usada por varias empresas con muchas operaciones y ha estado disponible por varios años, lo cual significa que ofrece una gran fiabilidad y baja latencia. Realtime Database está limitado a una sola región.	Firestore está en beta todavía lo cual deja dudas sobre su estabilidad a nivel de su uso ya que puede cambiar su SDK. Sin embargo, Firestore almacena los datos en múltiples Datacenters y en distintas regiones, brindando una gran escalabilidad y confiabilidad.
Escalabilidad		Escala de forma automática, por lo cual no se requiere compartir los datos a través múltiples instancias.

	Realtime Database	Firestore
Reglas de seguridad	En Realtime Database las reglas son en cascada. La validación de los datos en Realtime Database no son automáticas, en Realtime Database hay que definirlos.	Realizar las reglas en Firestore es más simple y más robusto. Las reglas en Firestore no son en cascada. La validación de los datos en Firestore es automática.

De acuerdo con el análisis que se realizó (ver tabla 2), se decide migrar a Firestore como la base de datos del proyecto, manteniendo la misma plataforma, sola cambiar el servicio que ofrece la misma, enfatizando la estructura de datos la de Firestore brinda una estructura de datos más ordenada y estructurada que la usada en el anterior prototipo. Se puede observar de forma ilustrada como se estructuran los datos con Firestore (ver imagen 6.).

Imagen 7. Ilustración de consulta y estructura jerárquica con manejo de sub-colecciones de datos de Firestore.



5.2. LISTA DE REQUERIMIENTOS.

Para el desarrollo funcional de la aplicación móvil se tuvo en cuenta el análisis de la primera versión de la aplicación, a partir de las observaciones que se obtuvieron anteriormente se realizó un listado de requerimientos.

Teniendo en cuenta la especificación y levantamiento de requerimientos inicial, a continuación, se adjunta el listado:

MODULO HOME

1. Ruta creada:

- Crear rutas en tiempo real.
- mantener actualizadas las rutas que se encuentren disponibles.
- Quitar las rutas que ya han terminada un servicio.

2. Solicitar unirse a una Servicio:

- Cambiar el nombre del botón unirse a solicitar cupo en una ruta publicada.
- Mostrar de forma ordenada la información que contenga la ruta publicada.
- Al solicitar un cupo que el usuario pueda describir un punto de destino.
- Que el usuario pasajero pueda cancelar el cupo que ha pedido.
- Enviar notificación al usuario pasajero cuando el usuario conductor rechace la solicitud.

3. Unirse a una Servicio:

- Permitir visualizar la información del conductor y del vehículo.
- Habilitar la una opción de llamar para que el pasajero pueda comunicarse con el conductor.
- Habilitar la opción de cancelar un servicio.
- Deshabilitar la opción de cancelar, después de iniciar un servicio.

- Crear un chat dentro del servicio.

4. Servicio creado:

- Crear una ruta para servio, solicitudes y sala de chat.

4.1. Servicio

- Permitir visualizar el servio que se creó.
- Mostrar opción de calificar usuario pasajero después de finalizar el servicio.

4.2. Solicitudes

- Ver la solicitud de los usuarios que quieran hacer parte del servio.
- visualizar la información de los usuarios solicitantes.
- Que pueda aceptar o rechazar un servicio.
- Si acepta un servio, pueda llamar al usuario pasajero.
- Que lo pueda expulsar antes de iniciar el servicio.
- Crear opción de iniciar servicio.
- deshabilitar opción de expulsar usuario pasajero si ya se ha iniciado el servicio.

4.3. Sala de chat

- Permitir al usuario conductor interactuar con los usuarios pasajeros por medio de un chat.

MODULO PERFIL

- Permitir Visualización de correo electrónico, número de teléfono y campos de vehículo en perfil de usuario.
- Permitir Visualizar calificación obtenido por los usuarios
- Deshabilitar opción de calificación para el mismo usuario que presta o toma el servicio.

1. Editar perfil:

- Cargar una nueva foto para perfil de usuario.

- Visualizar correo registrado.
- Denegar edición de correo.
- Eliminar botón de volver.
- Eliminar campo para agregar marca y modelo del vehículo.

2. Agregar vehículo:

- Cambiar el campo Modelo del vehículo por Color de vehículo.
- Cambiar nombre del botón volver por cancelar.
- Cambiar botón nombre del botón enviar por agregar.

MODULO TUS RUTAS

- Permitir agregar a una ruta la hora de salida o de llegada del servicio.
- Permitir al conductor seleccionar el tipo de vehículo (InDrive, Uber o Propio).
- Habilitar swicht para agregar la ruta a creación rápida.
- Crear opción para visualizar solicitudes para unirse a una ruta.
- Crear opción para visualizar los servicios creados por el usuario.

1. Mis solicitudes:

- Crear una ruta que permita ver los servicios a los que ha solicitado unirse.

2. Mis Servicios

- Mis servicios
- Crear una ruta que permita ver los servicios que ha creado y publicado.

CREAR MODULO TUS RUTAS RÁPIDAS

- Mostrar las rutas que se han creado.
- Tener la opción de activar la ruta o eliminarla.
- Si la ruta ha sido activada, que se pueda editar

CREAR MODULO AYUDA

- permitir a los usuarios ver los nombres de los integrantes de cada versión de la aplicación.
- Habilitar una opción para calificar la aplicación.

5.3 TARJETAS CRC

La utilización de las tarjetas CRC (Class-Responsibility-Collaboration) es una técnica de diseño orientada a objetos propuesta por Kent Beck (introdutor de la metodología de programación extrema).

El objetivo de esta técnica es realizar, mediante las tarjetas, un inventario de las clases necesarias para implementar la aplicación y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de estas por parte de varios actores del equipo del proyecto con el fin de que el diseño sea lo más sencillo posible verificando las especificaciones de la aplicación.

A continuación, se presenta un esquema típico de tarjetas CRC en las siguientes tablas en el que se indicaran los siguientes datos:

- Nombre de la clase
- Responsabilidades de la clase
- Las clases con las que va a colaborar para poder realizar las responsabilidades indicadas.

Tabla 3. Tarjeta CRC Clase Aplicación UisWheels con sus responsabilidades y sus colaboradores.

Aplicación UisWheels	
Responsabilidades	Colaboradores
Ofrece servicio de transporte.	Usuario. Pasajero. Conductor. Ruta. Tabla 18. Tabla de iteración No 7.

Tabla 4. Tarjeta CRC Clase Usuario con sus responsabilidades y sus colaboradores.

Usuario	
Responsabilidades	Colaboradores
Creación de nuevos usuarios. Validación de usuarios. Solicitar servicio de transporte. Ofrece servicio de transporte.	

Tabla 5. Tarjeta CRC Clase Conductor con sus responsabilidades y sus colaboradores.

Conductor	
Responsabilidades	Colaboradores
Registro, modificación vehículo y cupo. Modificar servicio de ruta. Cancelar (servicio de rutas). Notificar (cambios a pasajeros). Valoración de pasajero. Hereda de usuario.	Vehículo. Servicio. Pasajeros. .

Tabla 6. Tarjeta CRC Clase Pasajero con sus responsabilidades y sus colaboradores.

Pasajero	
Responsabilidades	Colaboradores
Cancelar (servicio obtenido) informar a servicio. Consultar (cupos disponibles) consultar a servicio. Calificación de conductor Hereda de usuario.	Usuario. Servicio.

Tabla 7. Tarjeta CRC Clase Servicio con sus responsabilidades y sus colaboradores.

Servicio	
Responsabilidades	Colaboradores
Permite la creación de un recorrido por un usuario con destino, origen, hora y cupo. Consultar servicios disponibles. Consultar cupos disponibles e informa a solicitante. Descripción servicio disponibles	Usuario. Pasajero. Conductor.

Tabla 8. Tarjeta CRC Clase Vehículo con sus responsabilidades y sus colaboradores.

Vehículo	
Responsabilidades	Colaboradores
Agregar vehículo. Modificar vehículo. Eliminar vehículo.	

Tabla 9. Tarjeta CRC Clase Calificación con sus responsabilidades y sus colaboradores.

Calificación	
Responsabilidades	Colaboradores
Realizar calificación de usuario.	Conductor. Pasajero.

Tabla 10. Tarjeta CRC Clase Mensaje con sus responsabilidades y sus colaboradores.

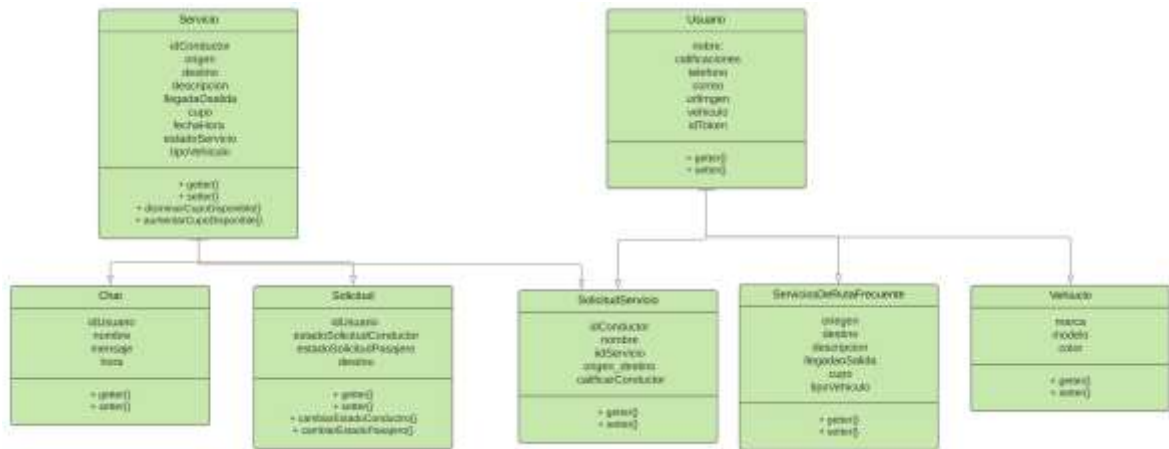
Mensaje	
Responsabilidades	Colaboradores
Permitir enviar mensajes Permitir recibir mensajes	Conductor Pasajero

A partir de las Tarjetas CRC se inició la implementación de la aplicación, con forme avanza el proyecto también varían ciertas propuestas que se tenían pensado inicialmente.

5.4 DIAGRAMA DE CLASES DE UISWHEELS VERSIÓN 2.0

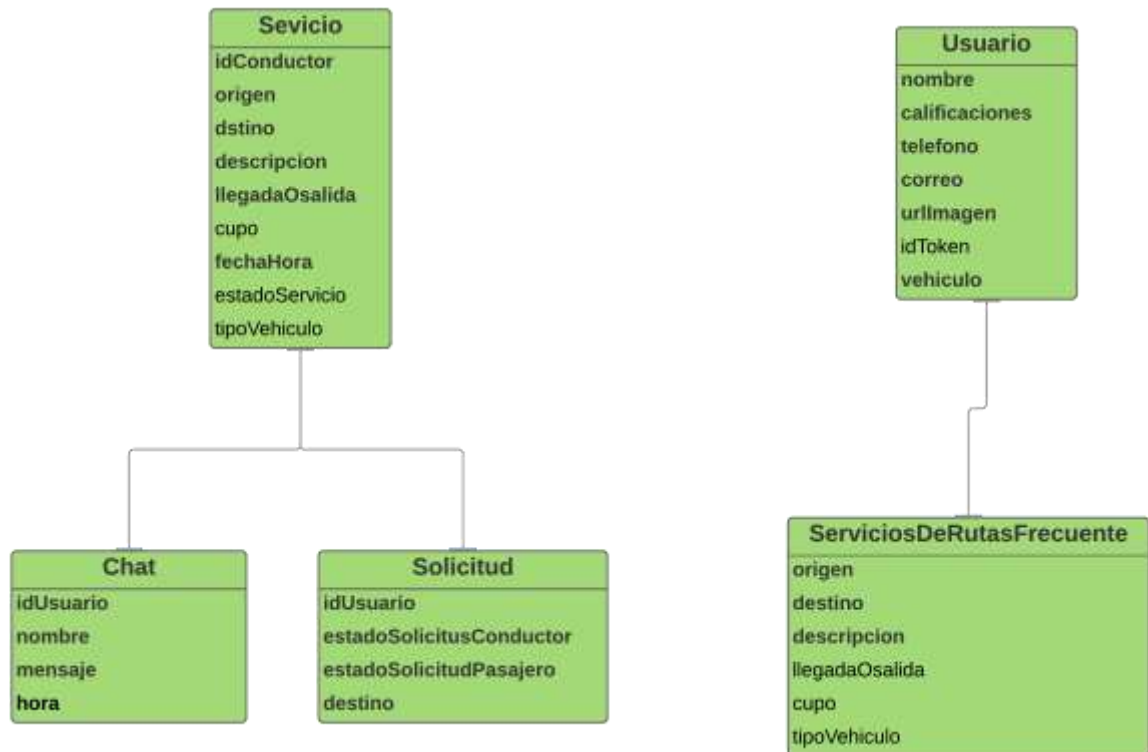
A partir de las tarjetas CRC es posible realizar el diagrama de clases para el desarrollo de la aplicación, con los atributos característicos de cada clase y los métodos consecuentes a ellos.

Imagen 8. Modelo del Diagrama de clases UisWheels 2.0



5.5 MODELO DE DIAGRAMA DE BASES DE DATOS DE UISWHEELS VERSIÓN 2.0

Imagen 9. Modelo del diagrama de base de datos



6. RESULTADOS

Este capítulo parte de los resultados obtenidos a partir de la metodología de desarrollo iterativa creciente que dirigió el curso del proyecto, a partir de dicho desarrollo se obtuvo un total de 8 iteraciones, donde cada iteración tuvo un periodo de tiempo de 8 días, partiendo del análisis de especificación y requerimientos se descubrió una totalidad de 59 requerimientos contenidos en 7 módulos.

A continuación, se muestra la relación de los requerimientos desarrollados por iteración.

- **Iteración 1**

Tabla 11. Iteración No 1.

Iteración No 1.		
Requerimiento	Cumple	No cumple
• Cargar una nueva foto para perfil de usuario.	X	
• Eliminar botón de volver.	X	
• Eliminar campo para agregar marca y modelo del vehículo.	X	
• Cambiar el campo Modelo del vehículo por Color de vehículo.	X	
• Cambiar nombre del botón volver por cancelar en agregar vehículo.	X	X
• Al agregar vehículo, cambiar botón nombre del botón enviar por agregar.	X	
• Crear rutas en tiempo real.	X	
• mantener actualizadas las rutas que se encuentren disponibles.	X	
• Deshabilitar opción de calificación para el mismo usuario que presta o toma el servicio.		

- **Iteración 2.**

Tabla 12. Iteración No 2.

Iteración No 2.		
Requerimiento	Cumple	No cumple
• Visualizar correo registrado.	X	
• Denegar edición de correo.	X	
• Permitir Visualización de correo electrónico, número de teléfono y campos de vehículo en perfil de usuario.		x
• Permitir Visualizar calificación obtenido por los usuarios.		x
• Permitir visualizar la información del conductor y del vehículo.	x	
• Permitir agregar a una ruta la hora de salida o de llegada del servicio.	x	
• Permitir al conductor seleccionar el tipo de vehículo (InDrive, Uber o Propio).		
• Habilitar swicht para agregar la ruta a creación rápida.		
• Crear opción para visualizar solicitudes para unirse a una ruta.	X	x
	X	
		x
	x	

- **Iteración 3**

Tabla 13. Iteración No 3.

Iteración No 3.		
Requerimiento	Cumple	No cumple
• Crear opción para visualizar los servicios creados por el usuario	X	
• Deshabilitar opción de calificación para el mismo usuario que presta o toma el servicio	X	

Iteración No 3.		
Requerimiento	Cumple	No cumple
• Permitir Visualización de correo electrónico, número de teléfono y campos de vehículo en perfil de usuario.	X	
• Permitir Visualizar calificación obtenido por los usuarios.	X	
• Crear opción para visualizar solicitudes para unirse a una ruta.	X	
• Crear opción para visualizar los servicios creados por el usuario	X	
• Cambiar el nombre del botón unirse a solicitar cupo en una ruta publicada.	X	
• Mostrar de forma ordenada la información que contenga la ruta publicada.	x	
• Al solicitar un cupo que el usuario pueda describir un punto de destino.		

Iteración 4.

Tabla 14. Iteración No 4.

Iteración No 4.		
Requerimiento	Cumple	No cumple
• Permitir visualizar el servio que se creó.	X	
• Mostrar opción de calificar usuario pasajero después de finalizar el servicio.		x
• Que el usuario pasajero pueda cancelar el cupo que ha pedido.	X	
• Crear una ruta que permita ver los servicios a los que ha solicitado unirse.	X	
• Crear una ruta que permita ver los servicios que ha creado y publicado.	X	
• Ver la solicitud de los usuarios que quieran hacer parte del servio.	X	
• visualizar la información de los usuarios solicitantes.	x	
• Que pueda aceptar o rechazar un servicio.		

- **Iteración 5.**

Tabla 15. Tabla de iteración No 5.

Iteración No 5.		
Requerimiento	Cumple	No cumple
<ul style="list-style-type: none"> • Si acepta un servio, pueda llamar al usuario pasajero. • Enviar notificación al usuario pasajero cuando el usuario conductor rechace la solicitud. • Que lo pueda expulsar antes de iniciar el servicio. • Crear opción de iniciar servicio. • deshabilitar opción de expulsar usuario pasajero si ya se ha iniciado el servicio. • Crear una ruta que permita ver los servicios a los que ha solicitado unirse. 	<p>X</p> <p>X</p> <p>X</p> <p>x</p>	<p>x</p>

- **Iteración 6.**

Tabla 16. Iteración No 6.

Iteración No 6.		
Requerimiento	Cumple	No cumple
<ul style="list-style-type: none"> • Enviar notificación al usuario pasajero cuando el usuario conductor rechace la solicitud. • Crear un nuevo módulo para pare tener rutas y activarlas fácilmente. • Mostrar las rutas que se han creado. • Tener la opción de activar la ruta o eliminarla. • Si la ruta ha sido activada, que se pueda editar. • Quitar las rutas que ya han terminada un servicio. • Habilitar la opción de cancelar un servicio. 	<p>x</p>	

- **Iteración 7**

Tabla 17. Iteración No 7.

Iteración No 7.		
Requerimiento	Cumple	No cumple
<ul style="list-style-type: none"> • Crear una ruta para servio, solicitudes y sala de chat. • Habilitar la una opción de llamar para que el pasajero pueda comunicarse con el conductor. • Habilitar la opción de cancelar un servicio. • Deshabilitar la opción de cancelar, después de iniciar un servicio. • Crear un chat dentro del servicio. 		x

- **Iteración 8.**

Tabla 18. Iteración No 8.

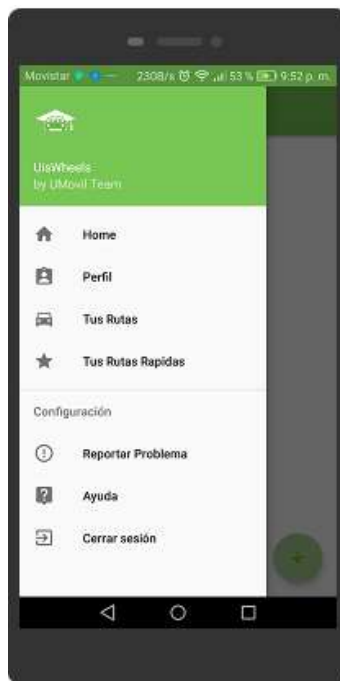
Iteración No 8.		
Requerimiento	Cumple	No cumple
<ul style="list-style-type: none"> • Crear un chat dentro del servicio. 	X	
<ul style="list-style-type: none"> • Permitir al usuario conductor interactuar con los usuarios pasajeros por medio de un chat. 	X	
<ul style="list-style-type: none"> • permitir a los usuarios ver los nombres de los integrantes de cada versión de la aplicación. 	X	
<ul style="list-style-type: none"> • Habilitar una opción para calificar la aplicación. 	x	

7. UISWHEELS 2.0

A continuación, se adjuntarán las imágenes ilustrativas de la segunda versión de la aplicación móvil UisWheels versión 2.0 sustentada en el presente documento.

- En la siguiente imagen (ver imagen 7.) se puede observar los diferentes módulos creados como: Tus Rutas y Ayuda, estos han sido implementados junto con los demás módulos para la segunda versión de la aplicación UisWheels.

Imagen 10. Interfaz gráfica para mostrar los módulos de la aplicación UisWheels versión 1.



- En la siguiente imagen (ver imagen 9.) se puede observar la interfaz gráfica para perfil de usuario modificada, las opciones que tiene el mismo para agregar un

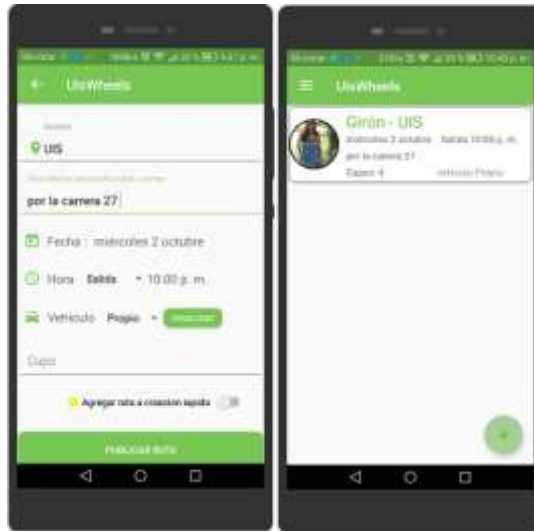
vehículo o modificar parte de la información suministrada en el registro con la opción de editar perfil.

Imagen 11. Interfaz gráfica para perfil de usuario de la aplicación UisWheels 2.0



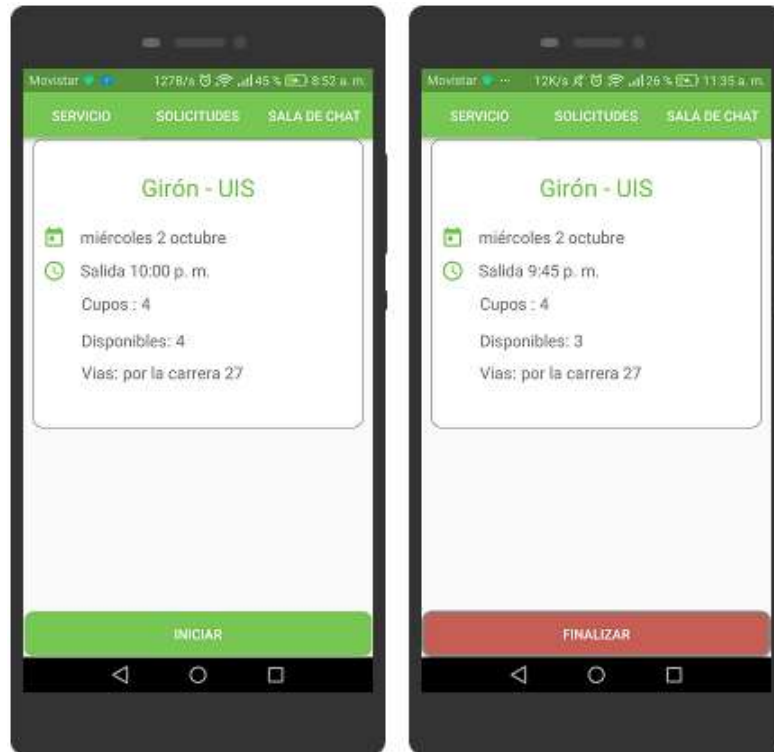
- En la siguiente imagen (ver imagen 8.) se observa dos capturas de planta de la aplicación, en la primera puede visualizarse los campos que debe agregar un conductor y las diferentes opciones que puede tener el mismo como: crear un servicio, o agregarlo a una ruta rápida y escoger el tipo de vehículo que dispondrá. En la segunda captura, visualización de la publicación.

Imagen 12. Interfaz gráfica para crear una ruta (primera captura) y mostrar ruta publicada (segunda captura) en la aplicación UisWheels versión 2.



- Cuando un usuario publica un servicio, en el mismo se muestra tres opciones que solo puede ver el usuario conductor: Servicio, Solicitudes y sala de chat. En la ruta de servicio se muestra la información proporcionada para el mismo, el conductor puede decidir iniciar el servicio y dar fin a el mismo como se muestra en la siguiente imagen (ver imagen 10.)

Imagen 13. Interfaz gráfica para mostrar la información del servicio que ha publicado e iniciar y finalizarlo.



- En la siguiente imagen (ver imagen 11.) se muestra la ruta de solicitudes que tiene un usuario conductor, así como las solicitudes enviadas para obtener un cupo y los usuarios que han sido aceptados en el servicio.

Imagen 14. Interfaz gráfica para ver solicitudes de usuarios aceptados y por aceptar.



- En la siguiente imagen (ver imagen 12.) se muestra la sala de chat creada para la aplicación UisWheels 2, en donde se puede observar el medio creado para la interacción entre los usuarios que estén interesados en el servicio.

Imagen 15. Interfaz gráfica para la sala de chat creada para la aplicación UisWheels versión 2.



8. CONCLUSIONES

- En UisWheels versión 2.0 debido a la implementación de nuevos requerimientos se nota una mejora en la usabilidad de la aplicación, comparada con la versión 1.0. Funciones integradas como la calificación de servicios (Pasajero – Conductor y viceversa), envío de notificaciones e integración del chat permiten una mayor interacción entre el conductor y el pasajero.
- A partir de la metodología de desarrollo utilizada creciente e iterativa, se permite planificar y ejecutar en diferentes bloques temporales los requerimientos establecidos, permitiendo visualizar la evolución de la aplicación de forma ágil y rápida para proporcionar un resultado que cumpla con los objetivos establecidos.
- Partiendo de la necesidad de los usuarios que utilizan transporte informal sumado a los altos tiempos que transitan diariamente en transporte público y en el auge de los dispositivos móviles, se han creado sustancialmente aplicaciones que conjugan y mitigan el acceso al transporte informal del usuario tales como Uber, InDriver y UisWheels en su versión 2.0.
- Debido a que los datos en Firebase Realtime Database se almacena como un gran árbol Json con fácil almacenamiento, pero en alta complejidad y niveles jerárquicos son más difíciles de organizar a gran escala; comparado con Cloud Firestore donde son fáciles de almacenar en documentos similares a Json y en alta complejidad su organización a gran escala da mayores facilidades con subcolecciones dentro de los documentos.
- Teniendo en cuenta el 4to objetivo específico del presente proyecto, se escogió una muestra de usuarios a quienes se les realizó una serie de pruebas con el fin

de medir la usabilidad de la aplicación UisWheels 2.0, se analizó la información consignada por los usuarios en la encuesta y se efectuaron una serie de recomendaciones (ver capítulo 9).

9. RECOMENDACIONES

REQUERIMIENTO	DESCRIPCIÓN
Localización de usuarios.	Para en futuras versiones de UisWheels, por medio de un mapa se pueda visualizar la ubicación en tiempo real de los diferentes actores que interesados en la ruta.
Integrar nuevo sistema operativo.	Para permitir que toda la comunidad pueda hacer uso de la aplicación, se sugiere crear versiones para sistemas operativos iOS.
Búsqueda y localización de rutas.	Para facilitar la búsqueda de rutas se sugiera que, en la próxima versión, la aplicación permita visualizar las rutas publicados por usuarios que estén en la misma localización, o en su defecto permita hacer una búsqueda de rutas por ubicación.

BIBLIOGRAFÍA

ABERNETHY Michael. IBM. (14 de junio, 2011). IBM. Node.Js. [en línea] disponible en: <https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/index.html>

ARNOLD, R.S.: Software Reengineering. IEEE Computer Society Press, 1993

BIGGERSTAFF T. and PerLis A.: Software Reusability. Addison-Wesley. November, 1990.

CAMACHO Laura. Carpooling, la alternativa de transporte que ahorra tiempo y dinero: El tiempo. 2018 [en línea] disponible en: <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/que-es-y-como-funciona-el-carpooling-compartir-carro-en-colombia-237074>

CHIKOFFSKY, E.J. and CROSS, J.H.: Reverse engineering and design recovery: A taxonomy. IEEE Software. 7(1), 1990, 13-17.

CLOUD FIRESTORE. Documentación. Firebase: [en línea] disponible en: <https://firebase.google.com/docs/firestore/?hl=es-419>

El Sistema Integrado de Transporte Masivo, que en el papel se adaptó para hacer más ágiles los desplazamientos de los ciudadanos, encuesta tomadas y publicadas con fecha de martes 12 de febrero de 2019. Archivo / VANGUARDIA LIBERAL.

GÓMEZ TORRES Juan Guillermo Diferencias entre Realtime Database vs Firestore. [en línea] disponible en: <https://medium.com/@jggomez/diferencias-entre-realtime-database-vs-firestore-94364a2cc09e>

IBM. Conceptos básicos del lenguaje Java. [en línea] disponible en: <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/index.html>

INFOBAE. Visible: Carpooling: qué es y cómo funciona. 2015 [en línea] disponible en: <https://www.infobae.com/2015/12/16/1776933-carpooling-que-es-y-como-funciona/>

MONTSERRAT, M. S., MORENO, G. Metodología de reingeniería del software para la remodelación de aplicaciones científicas heredadas. Technical Report, Informe Técnico – Technical Report, DPTOIA-IT-2004-003. 2014

REINO ROMERO A. Versión 2.0 (26 de enero, 2000). [en línea] disponible en: <http://deim.urv.cat/~pedro.garcia/SOB/IntroXML.pdf>

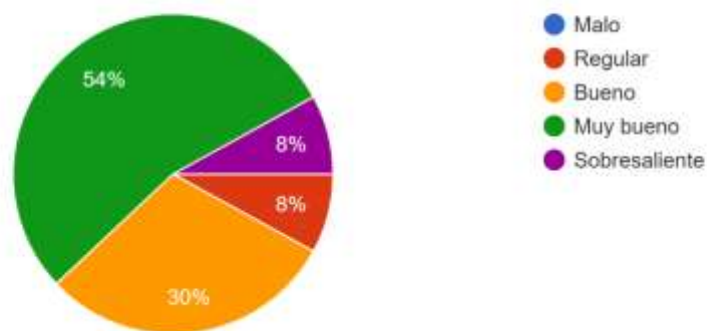
ANEXOS

Para realizar la encuesta de usabilidad fue necesario presentar la aplicación a un grupo de personas para que ellas interactuaran y luego de adquirir la experiencia con la aplicación, respondieran las preguntas anexadas a el formulario, esto con el fin de obtener datos que nos proporcionar información de satisfacción o mejoras ante el producto final.

- Presentamos los datos obtenidos ante las preguntas formuladas.

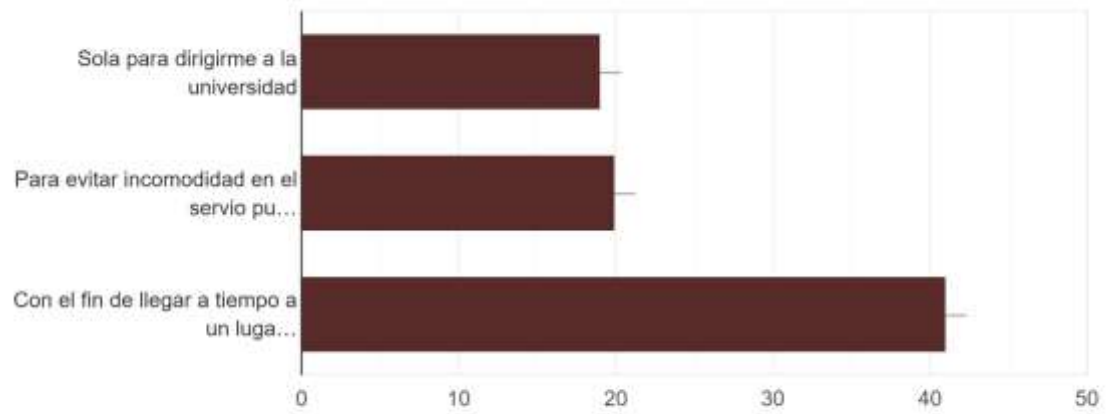
Con base al uso que dio de la aplicación. ¿Cómo calificaría su experiencia total con la app?

50 respuestas



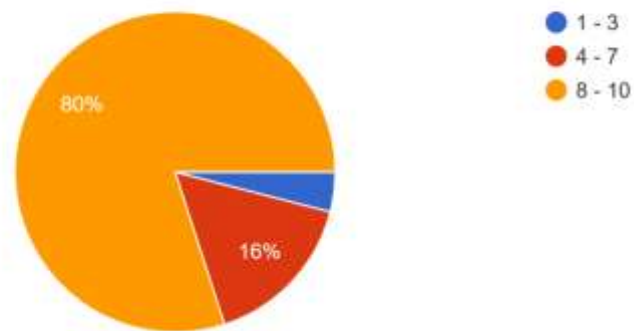
¿En qué situación usted preferiría usar el servicio que brinda UisWheels 2.0 ?

50 respuestas



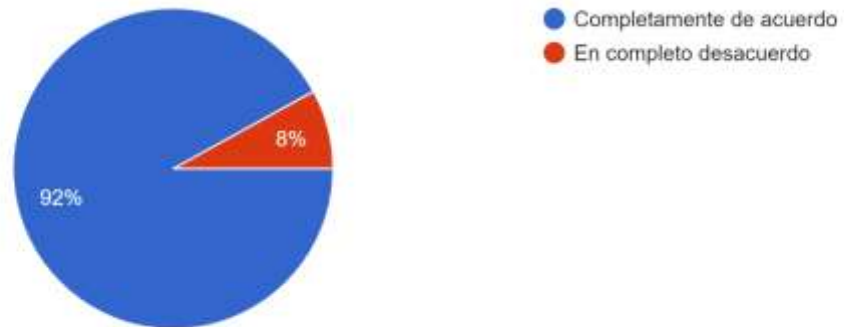
¿Cómo valora visualmente la aplicación?

50 respuestas



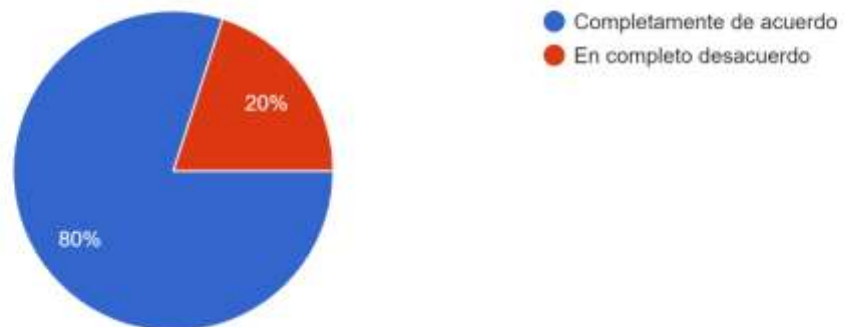
¿Cree que el servicio que ofrece esta aplicación es de utilidad para su caso personal?

50 respuestas



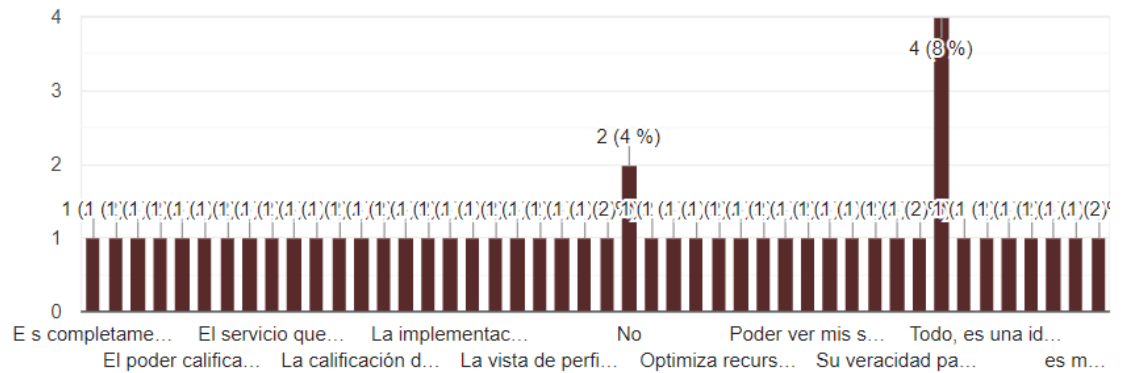
¿Le gustaría que la aplicación implementara un método de localización para facilitar la búsqueda de rutas?

50 respuestas



¿Qué fue lo que más le llamo la atención de la app?

50 respuestas



¿Tiene alguna sugerencia para mejorar el uso y servicios que ofrece la aplicación?

50 respuestas

No

Ninguna

Mapa

No, definitivamente la aplicación es muy completa y buena.

Cobertura de rutas

Hasta el momento ninguna

Implementar la ubicación en el mapa.

Estoy completamente a gusto

No me di cuenta que tenía que aceptar solicitudes de pasajeros, quizá mejor esa parte, que sean más visibles.

Ninguna, la verdad me pareció muy buena.

Quizá me gustaría que uno como pasajero pueda publicar que necesita un servicio y no esperar a que lo publiquen.

Sería bueno que colocaran localización y un buscador de rutas.

Publicidad

Noo

La UI es algo pobre

tener la opción de buscar a las rutas

Mostrar el numero del conductor antes de ser aceptado en el servicio publicado.

que se permitiera visualizar los puntos a pasar en un mapa

que la aplicación permita clasificar a los conductores dependiendo de los requisito o preferencias que se tengan.

Que contenga mas información acerca del conductor para tener mayor seguridad

NO, por el momento.

que sean más visibles las opciones destacadas.

El inicio de la aplicación es algo sencilla

Nada

Buscar rutas

Permitir buscar los servicios

Que la opción de llame también este antes de solicitar el cupo

No.

Que se incluya servicio de moto

un buscador de rutas

Creo que habilitarla a iOS

Por ahora todo esta bien.

Excelente, todo bien.

Pesa demasiado

No sabia que había algo para ver las solicitudes.

Publicidad, es una aplicación muy buena.

- Numero de descargas de la aplicación UisWheels 2.0

