**APLICACIÓN EN KOTLIN PARA VIAJES EN TRANSPORTE PÚBLICO: RAPIBUS (SEMESTRE DE 2023)**

*BERNAL 1,* *Nallely 1. Ingeniería de sistemas.*

*Vengoechea 2,* *Alfonso 2.*  *Ingeniería de sistemas.*

*Zambrano 3, José 3.*  *Ingeniería de sistemas.*

*Computación Móvil, Grupo* *50958, Universidad de la Costa.*

*De la Hoz Alexis*

*26/05/2023*

**RESUMEN**

Como estudiantes de Ingeniería de sistemas se nos propuso la implementación de una aplicación de dispositivo de dispositivo móvil. El proyecto en cuestión se enfocó en desarrollar una aplicación móvil para solucionar la problemática del transporte público en una determinada zona. Para ello, se utilizó la metodología de desarrollo de software ágil, donde se trabajó en ciclos iterativos y se contó con la participación del docente para asegurar que la aplicación cumpliera con sus requerimientos y expectativas.

La aplicación fue desarrollada utilizando el lenguaje de programación Kotlin y el entorno de desarrollo integrado Android Studio. Se implementó una arquitectura de tres capas, donde se separó la lógica de negocio, la interfaz de usuario y el acceso a datos. Además, se integraron diversas funcionalidades como la geolocalización, la integración con mapas y la posibilidad de guardar paradas favoritas. Para asegurar la calidad y funcionalidad de la aplicación, se llevó a cabo un mecanismo de prueba en un emulador de Pixel 3A con Android 9. Este proceso de prueba permitió detectar errores y realizar las correcciones necesarias para asegurar un óptimo rendimiento de la aplicación. Este proyecto permitirá que los usuarios puedan administrar su tiempo de traslado utilizando la APP y los encargados del transporte logren tener una administración en tiempo real.

***Palabras claves:*** *Kotlin, Programación Móvil, Android Studio*,Bus

**ABSTRACT**

As Systems Engineering students, we were tasked with implementing a mobile device application. The project focused on developing a mobile application to address public transportation issues in a specific area. To achieve this, we employed the agile software development methodology, working in iterative cycles and involving our instructor to ensure that the application met their requirements and expectations.

The application was developed using the Kotlin programming language and the Android Studio integrated development environment. We implemented a three-tier architecture, separating the business logic, user interface, and data access. Additionally, we integrated various features such as geolocation, map integration, and the ability to save favorite stops. To ensure the quality and functionality of the application, we conducted testing using a Pixel 3A emulator with Android 9. This testing process allowed us to identify errors and make necessary corrections to ensure optimal performance of the application. This project will enable users to manage their commuting time using the app, while transportation authorities achieve real-time management. ***Keywords:*** *Kotlin, Mobile Programming, Android Studio, Bus.*

1. **INTRODUCCIÓN**

En las ciudades modernas, el transporte público desempeña un papel crucial en la movilidad de los ciudadanos. Sin embargo, uno de los desafíos más comunes que enfrentan los pasajeros es la falta de información precisa y actualizada sobre los horarios de llegada de los autobuses. Esta incertidumbre puede generar frustración, retrasos y un uso menos eficiente del transporte público. Para abordar esta problemática, en este estudio nos propusimos desarrollar una aplicación móvil para dispositivos Android que consuma datos de llegada de autobuses en tiempo real y los presentes de manera conveniente y útil a los usuarios.[1][2]

En nuestro proyecto, nos enfocamos en utilizar Android Studio y el lenguaje de programación Kotlin, que se han establecido como herramientas líderes en el desarrollo de aplicaciones móviles para la plataforma Android. Android Studio, el entorno de desarrollo integrado oficial de Android, ofrece una amplia gama de funciones y recursos que permiten a los desarrolladores crear aplicaciones de alta calidad. Por su parte, Kotlin es un lenguaje de programación moderno y conciso que ha ganado popularidad en la comunidad de desarrollo de Android debido a su sintaxis clara y su interoperabilidad con Java.[2][3]

1. **MARCO TEÓRICO**

En Colombia, el transporte público desempeña un papel vital en la movilidad de las personas. Es el medio de transporte más utilizado para el traslado de personas en todo el país, especialmente en áreas urbanas y metropolitanas. En el marco teórico de este estudio, se explorará la importancia del transporte público en Colombia, así como los factores y las teorías que influyen en su uso generalizado.[4]

Debido al avance tecnológico acelerado en los últimos años, la utilización de rastreo satelital para servicios de navegación se ha vuelto ampliamente difundida en todo el mundo, incluyendo Colombia. Hoy en día, cualquier individuo que posea un teléfono inteligente tiene acceso al servicio de GPS.[5]

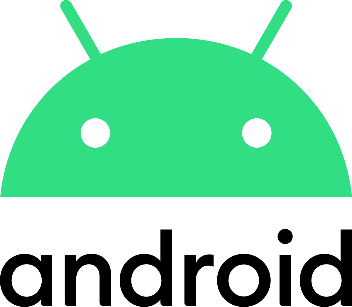
* 1. **Smartphone**

Según Techopedia, un smartphone es "un teléfono celular móvil que ofrece aplicaciones similares a las de un computador fijo o portátil, añadiéndole servicios de llamada y de mensajería instantánea"[5]. Estos dispositivos móviles cuentan con una amplia variedad de aplicaciones que facilitan la vida cotidiana y la mayoría de ellos tienen el sistema operativo Android. Para el correcto funcionamiento de una aplicación móvil, ya sea para transporte u otros fines, es crucial llevar a cabo un control y registro de los usuarios, así como almacenar datos y otra información relevante en una base de datos para su gestión efectiva.[6]

* 1. **Android**

Android es un sistema operativo móvil basado en Linux, que fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas. El sistema operativo fue inicialmente desarrollado por Android, Inc. y fue adquirido por Google en 2005[8]. Desde entonces, Android se ha convertido en uno de los sistemas operativos móviles más populares del mundo gracias a su interfaz de usuario amigable y la gran cantidad de aplicaciones disponibles en su tienda de aplicaciones, Google Play.[9][10]

**Figura 1**. Logo Android



Fuente: tomado de <https://en.wikipedia.org/wiki/Android_%28operating_system%29>

* 1. **Android Studio**

Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de App para Android, además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ.

Android Studio se ha convertido en la opción preferida de muchos desarrolladores de aplicaciones móviles debido a su amplia gama de funciones y recursos, así como a su integración con otras herramientas de desarrollo de Android. Además, Android Studio se actualiza regularmente con nuevas funciones y mejoras para mantenerse al día con las últimas tendencias y tecnologías en el desarrollo de aplicaciones móviles para Android.[10][11]

* 1. **Kotlin**

Kotlin es un lenguaje de programación de código abierto que se utiliza principalmente para el desarrollo de aplicaciones móviles para la plataforma Android, aunque también se puede utilizar para el desarrollo de aplicaciones de escritorio y web. Kotlin fue desarrollado por JetBrains y se lanzó en 2011 como una alternativa más moderna y concisa al lenguaje de programación Java. Kotlin se ha vuelto cada vez más popular en la comunidad de desarrollo de Android debido a su sintaxis clara y su interoperabilidad con Java, lo que permite a los desarrolladores utilizarlo junto con el código existente de Java en proyectos de Android.[11]

**Figura 2**. Logo Kotlin



Fuente: tomado de <https://kotlinlang.org/docs/home.html>

* 1. **Activity**

Son las funciones que ejecutará la aplicación, una aplicación Android es un conjunto finito de actividades en las que cada una tiene una única tarea u objetivo, normalmente destinadas a gestionar la visualización de una pantalla.

2**.6.Intents**

Android utiliza un mecanismo de mensajes asíncronos para asociar peticiones de tareas con su actividad correspondiente. Cada petición se empaqueta en un Intent.

**2.7.Servicios**

procesos que se lanzan en segundo plano sin la intervención del usuario.

**2.8. Notificaciones**

Alertas que recibe el usuario

**2.9 Base de datos**

Es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). Algunos ejemplos de bases de datos son Microsoft Access, MySQL, Microsoft SQL Server, FileMaker Pro y Oracle Database

**2.10 Firebase**

Firebase es una plataforma de desarrollo de aplicaciones móviles y web en la nube, propiedad de Google. Ofrece una amplia variedad de servicios y herramientas para ayudar a los desarrolladores a crear aplicaciones de alta calidad, escalables y seguras de manera más rápida y sencilla. Los servicios de Firebase incluyen alojamiento de aplicaciones, autenticación de usuarios, bases de datos en tiempo real, almacenamiento en la nube, mensajería en la nube, análisis, prueba y monitoreo de aplicaciones, entre otros.[13]

1. **JUSTIFICACION**

En este proyecto los estudiantes de ingeniería de Sistemas de la Universidad de la costa (CUC), enfrentan la problemática de los tiempos excesivos de espera de las rutas alimentadoras de transmetro, y que con el regreso a las actividades normales este servicio es insuficiente, el uso de los Smartphone es una necesidad pero también una herramienta de comunicación; para solucionar esta situación se va desarrollar una APP en Android Studio con Kotlin, que permita conocer el tiempo de espera real de las rutas alimentadoras y la ubicación de las unidades utilizando Geolocalización en servicio Web.

La cual permitirá un mejor acceso al servicio de trasporte sacándole el mejor provecho a los tiempos de espera que usualmente los usuarios se enfrentan a diario.

1. **OBJETIVOS**
   1. **OBJETIVO GENERAL**

Este proyecto beneficiará a los usuarios de la ciudad de Barranquilla y a futuro de otras ciudades de la región. La problemática detectada es que muchos usuarios pierden las rutas alimentadoras y llegan tarde a sus destinos porque el servicio es ineficiente y el largo tiempo de espera para abordar una unidad. Este proyecto permitirá que los usuarios puedan administrar su tiempo de traslado a las paradas alimentadoras utilizando la APP y los encargados del transporte logren tener una administración en tiempo real. En los últimos tiempos el servicio de transporte ha pasado por varios incumplimientos en los estándares de calidad sumado al crecimiento de servicios de transporte informales y la taza de motocicletas en el transporte por la que se evidencia una clara inconformidad de los usuarios.

* 1. **OBJETIVOS ESPECIFICOS**
* Proporcionar a los usuarios una plataforma para buscar información sobre paradas de autobús y tiempos de espera.
* Mejorar la experiencia de los usuarios al brindarles acceso rápido y fácil a la información relevante.
* Facilitar la administración de paradas favoritas para que los usuarios puedan acceder rápidamente a la información que más les interesa.
* Utilizar tecnología de mapas para visualizar la ubicación de las paradas y mejorar la navegación de los usuarios.
* Mantener la privacidad y seguridad de los usuarios al implementar medidas adecuadas de autenticación y almacenamiento de datos.

1. **METODOLOGIA**

Planificación inicial: Se realizó una planificación inicial para definir los objetivos y los requisitos de la aplicación. Se estableció una visión clara del producto y se identificaron los principales elementos y características que debían ser implementados.

Creación del backlog del producto: Se creó un backlog del producto que incluía todas las tareas y funcionalidades necesarias para el desarrollo de la aplicación. El backlog se mantuvo actualizado y se priorizaron las tareas en función de su importancia y valor para el usuario.

Sprints: El desarrollo se dividió en sprints, que son ciclos de tiempo fijos (por ejemplo, dos semanas) en los que se planificaron y desarrollaron una serie de tareas. Al inicio de cada sprint, se seleccionaron las tareas más importantes del backlog para ser implementadas.

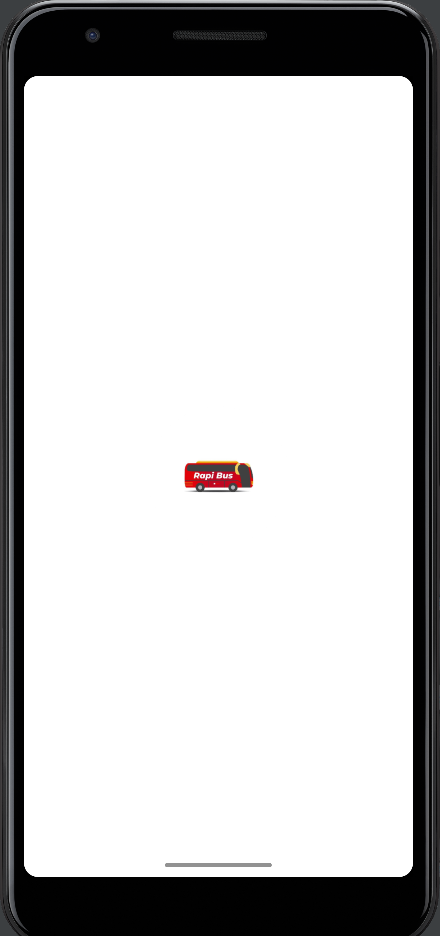
Retroalimentación y ajustes: Al finalizar cada corte, se realizó una revisión para obtener comentarios y retroalimentación de los stakeholders, como el equipo de desarrollo, los usuarios o el cliente. Esta retroalimentación se tuvo en cuenta para realizar ajustes y mejoras en las siguientes iteraciones.

Entrega continua: Se siguió un enfoque de entrega continua, lo que significa que se lanzaron versiones de la aplicación en intervalos regulares, en lugar de esperar a tener una versión final completa. Esto permitió obtener comentarios tempranos del profesor.

1. **RESULTADOS**

Consideramos también que el diseño de la aplicación debe de ser una experiencia agradable para el usuario ya que actualmente los usuarios buscan una interfaz rápida, sencilla e intuitiva. Se definieron las vistas generales , algunas se presentan a continuación :

**Figura 3**.Imagen de las vistas de la aplicación movil.



**Pantalla de Carga**

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

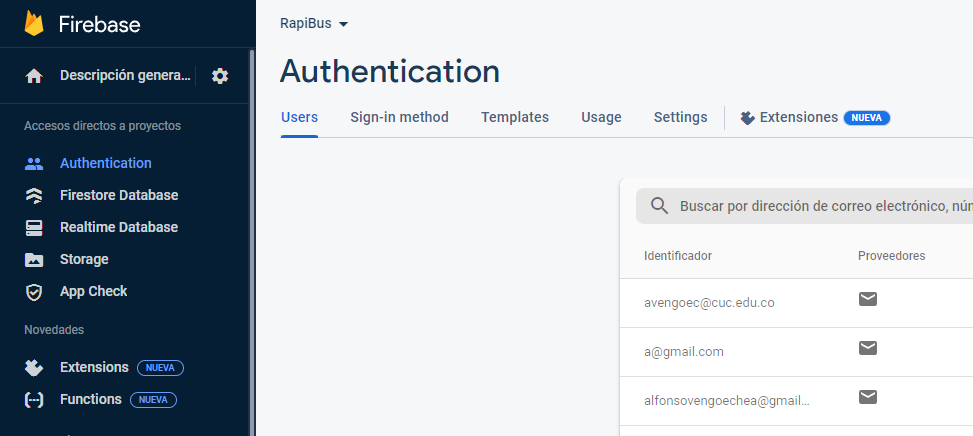
Se implementó un sistema de inicio de sesión utilizando Firebase Authentication para la autenticación de usuarios. Firebase Authentication proporciona una solución confiable y segura para la gestión de usuarios, facilitando la creación, inicio de sesión y autenticación de usuarios en la aplicación RapiBus.

El sistema de inicio de sesión consta de dos opciones: "Ingresar" y "Registrar". Al seleccionar la opción "Ingresar", se solicita al usuario que ingrese su nombre de usuario y contraseña. Estos valores se validan para asegurarse de que no estén vacíos y, luego, se utilizan para realizar una solicitud de inicio de sesión a Firebase Authentication. Si las credenciales son válidas y coinciden con una cuenta existente, el usuario puede acceder a la aplicación.

En caso de que un usuario no tenga una cuenta registrada, puede seleccionar la opción "Registrar" para crear una nueva cuenta. Esto implica proporcionar un nombre de usuario y contraseña válidos, que se validan de manera similar a la opción de inicio de sesión. Una vez que se verifica la validez de los campos, se realiza una solicitud de registro a Firebase Authentication y, si el proceso es exitoso, se crea una nueva cuenta asociada al usuario.

La integración de Firebase Authentication ofrece múltiples beneficios para el sistema de inicio de sesión. En primer lugar, proporciona un almacenamiento seguro de las credenciales de los usuarios, lo que garantiza que la información personal esté protegida. Además, Firebase Authentication maneja la lógica de autenticación y permite realizar operaciones como iniciar sesión, registrar usuarios y restablecer contraseñas de manera sencilla y segura, lo que reduce la carga de desarrollo.

Al utilizar Firebase Authentication, la aplicación RapiBus se beneficia de un sistema de inicio de sesión robusto y confiable, mejorando la experiencia de los usuarios al proteger sus datos personales y proporcionar un acceso seguro a las funcionalidades de la aplicación.

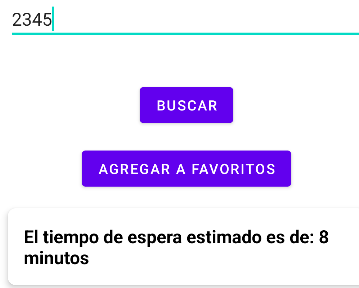


**Usuarios Registrados en Firebase**

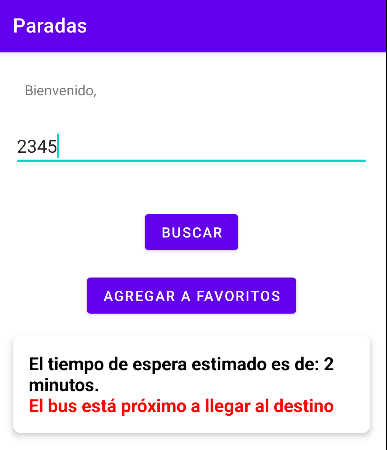
**Figura 4**. Buscador por N.Parada.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente



Se agrego un condicional, en la aplicación para notificar al usuario cuando el Autobus está próximo a llegar en menos de 3 minutos.



Se le permitió al usuario agregar las paradas mas usadas en una tabla en favoritos, para posteriormente volver a utilizarlas.

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

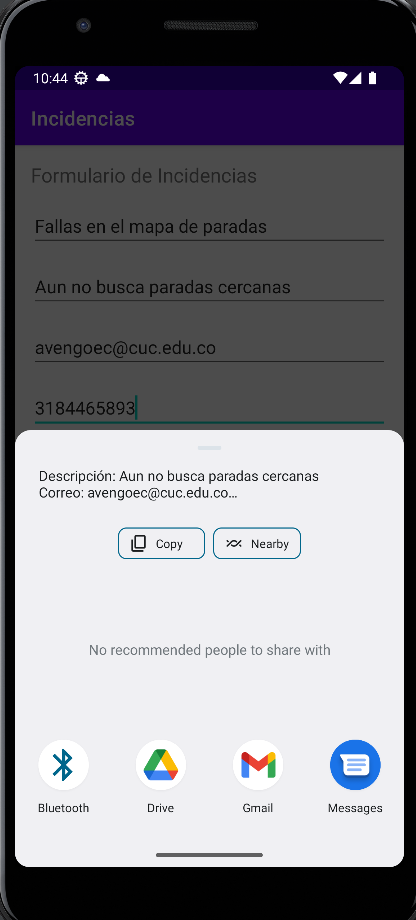
Se implemento una ventana de Mapa , que esta en periodo de testeo, mediante la API de Google Maps, el cual se podrá ver las paradas cercanas a la zona por la ubicacion

Mapa

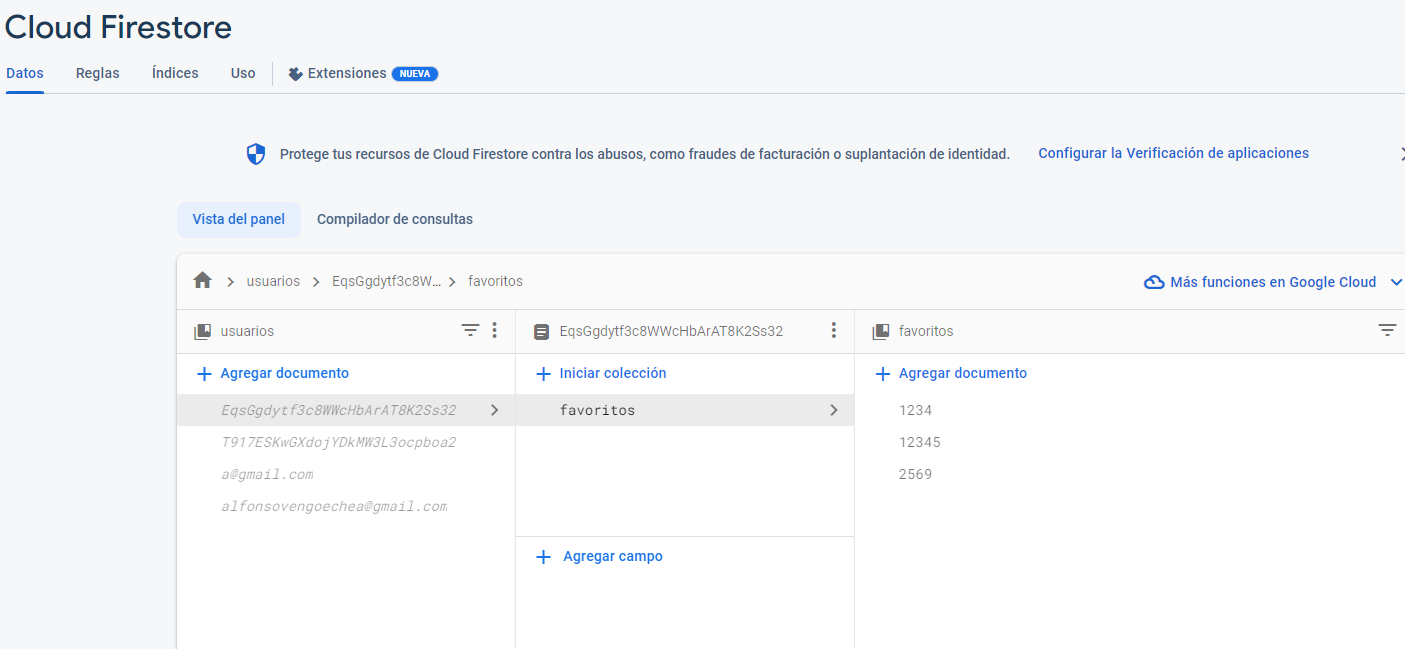
Descripción generada automáticamente

Finalmente se agrego una ventana de incidencias, en la que el usuario notificara por correo electrónico, las posibles fallas que se encuentren en la aplicación.





Se uso los servicios de Firebase con Cloud Firestore para almacenar las paradas favoritas por usuario.



1. **ANÁLISIS DE RESULTADOS**

Usabilidad: La aplicación RapiBus ha sido diseñada con una interfaz de usuario intuitiva y fácil de usar. Los elementos de navegación, como menús y botones, son claros y proporcionan una experiencia de usuario fluida. Los usuarios pueden acceder fácilmente a las funciones principales de la aplicación, como buscar paradas, ver mapas y administrar paradas favoritas.

Rendimiento: La aplicación RapiBus ha demostrado un rendimiento sólido en términos de velocidad de carga de datos y respuesta a las interacciones del usuario. Las consultas a la base de datos y los servicios externos se ejecutan de manera eficiente, lo que permite una experiencia fluida y sin retrasos para los usuarios.

Respuesta del usuario: Las acciones, como buscar paradas, agregar paradas favoritas y ver los tiempos de espera, se ejecutan rápidamente sin una espera significativa. Esto ha contribuido a una experiencia de usuario satisfactoria y sin frustraciones.

Estabilidad: La aplicación RapiBus ha demostrado ser estable y confiable durante las pruebas y el uso regular. No se han reportado bloqueos frecuentes ni errores graves que afecten negativamente la funcionalidad de la aplicación. Los usuarios pueden confiar en que la aplicación funcionará correctamente y no experimentarán interrupciones significativas.

La capacidad de administrar paradas favoritas y acceder rápidamente a la información relevante ha mejorado la experiencia del usuario y ha generado satisfacción.

la aplicación RapiBus ha demostrado un rendimiento sólido, una usabilidad efectiva, estabilidad y ha obtenido una alta satisfacción del testeo.Estos resultados indican que la aplicación cumple con su propósito de proporcionar a los usuarios una herramienta confiable y conveniente para el seguimiento de las paradas de autobús y los tiempos de espera. Sin embargo, es importante continuar recopilando comentarios de los usuarios y realizar mejoras continuas para mantener y mejorar la calidad de la aplicación.

1. **CONCLUSIONES**

La aplicación RapiBus ha sido desarrollada con éxito para brindar a los usuarios una herramienta eficiente y conveniente para el seguimiento de paradas de autobús y los tiempos de espera. A lo largo del desarrollo y la implementación de la aplicación, se han logrado los siguientes aspectos destacados:

Usabilidad mejorada: La aplicación se ha diseñado teniendo en cuenta la facilidad de uso y la experiencia del usuario. Se ha logrado una interfaz intuitiva que permite a los usuarios navegar sin dificultad por las diferentes funciones y acceder rápidamente a la información relevante. Esto ha contribuido a una mayor comodidad y satisfacción del usuario.

Integración con servicios de terceros: La aplicación ha aprovechado la integración con servicios externos, como Firebase, para la autenticación de usuarios y el almacenamiento de datos. Esto ha permitido una gestión segura de la información del usuario y una autenticación confiable.

Funcionalidad sólida: La aplicación ha demostrado una funcionalidad sólida en el testeo ya que se ha podido buscar paradas, agregar paradas favoritas y obtener información relevante de manera efectiva.

Rendimiento y estabilidad: La aplicación ha mostrado un buen rendimiento en términos de velocidad de carga, respuesta del usuario y estabilidad general. No se han informado problemas graves de rendimiento o errores que afecten negativamente la experiencia del usuario.

1. **RECOMENDACIONES**

**Mejora de la precisión de los tiempos de espera:** Investigar la posibilidad de integrar fuentes de datos en tiempo real, como datos GPS de los autobuses, para mejorar la precisión de los tiempos de espera. Esto proporcionaría a los usuarios una información más actualizada y confiable.

**Implementación de notificaciones:** Agregar la capacidad de enviar notificaciones a los usuarios sobre cambios en los tiempos de espera, retrasos en las rutas o cualquier otra información relevante. Esto mantendría a los usuarios informados y les permitiría planificar sus viajes de manera más efectiva.

**Integración de mapas en tiempo real:** Incorporar un mapa interactivo en la aplicación que muestre la ubicación en tiempo real de los autobuses en las rutas seleccionadas. Esto ayudaría a los usuarios a tener una mejor comprensión de la ubicación actual de los autobuses y a planificar sus desplazamientos de manera más eficiente.

**Ampliación de la cobertura geográfica:** Investigar la posibilidad de expandir la aplicación a otras ciudades o regiones. Esto permitiría a más usuarios beneficiarse de la funcionalidad de seguimiento de paradas de autobús y tiempos de espera.

**Mejoras en la interfaz de usuario:** Continuar mejorando la interfaz de usuario para hacerla más intuitiva y atractiva visualmente. Esto incluye la organización de la información de manera clara, el uso de iconos descriptivos y la mejora de la navegación en la aplicación.

**Recopilación y análisis de comentarios de los usuarios:** Establecer canales de retroalimentación de usuarios y realizar encuestas periódicas para recopilar comentarios y opiniones. Utilizar estos comentarios para identificar áreas de mejora y priorizar nuevas funcionalidades.

**Investigación de oportunidades de monetización:** Explorar opciones de monetización, como la incorporación de publicidad relevante o la implementación de una versión premium con características adicionales. Esto podría ayudar a generar ingresos para mantener y mejorar la aplicación a largo plazo.

Estas sugerencias y recomendaciones buscan impulsar la evolución y el crecimiento continuo de la aplicación RapiBus, brindando una experiencia aún mejor a los usuarios y expandiendo su alcance para beneficiar a más personas.

1. **REFERENCIAS**

[1] González, R., & Sánchez, J. (2019). Desarrollo de una aplicación móvil para el monitoreo de rutas de transporte público en la ciudad de La Paz, México. Revista Científica y Tecnológica UPSE, 6(1), 45-50.

[2] Liu, Y., & Wu, X. (2019). Design and implementation of real-time bus arrival prediction system based on GPS and big data technology. Journal of Ambient Intelligence and Humanized Computing, 10(5), 1885-1897.

[3] Google LLC. (n.d.). Android Studio. Retrieved from https://developer.android.com/studio

[4] Gómez, L. A., & Santamaría, J. M. (2019). Análisis de la importancia del transporte público en Colombia: Factores y teorías que influyen en su uso generalizado. Revista de Investigación Académica, 55, 1-15.

[5] Techopedia. (n.d.). Definition - What does Smartphone mean? Techopedia. Retrieved from <https://www.techopedia.com/definition/23586/smartphone>

[6] Sharma, R. (2021). Building a Mobile App Database: Tips and Tricks. Appinventiv. <https://appinventiv.com/blog/building-mobile-app-database-tips-tricks/>

[8] Google LLC. (n.d.). Android. Retrieved from https://www.google.com/android/

[9] Android Developers. (n.d.). Android. Retrieved from <https://developer.android.com/about>

[10] Google LLC. (n.d.). Android Studio. [Online]. Available: https://developer.android.com/studio

[11] IntelliJ IDEA. (n.d.). IntelliJ IDEA. [Online]. Available: <https://www.jetbrains.com/idea/>

[12] JetBrains. (n.d.). Kotlin. [Online]. Available: <https://kotlinlang.org/>

[13] Google LLC. (n.d.). Firebase. [Online]. Available: https://firebase.google.com/