ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN PROTOTIPO DE PLATAFORMA WEB Y MÓVIL PARA LA MONITORIZACIÓN DE PARÁMETROS DE LA ACTIVIDAD FÍSICA DE LOS PARTICIPANTES DE CARRERAS ATLÉTICAS.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

JENNIFER SAMANTA MINAYO YUJATO

DIRECTOR: Ph.D. FELIPE LEONEL GRIJALVA ARÉVALO

CODIRECTOR: Ph.D ANA MARÍA ZAMBRANO VIZUETE

Quito, abril 2022

AVAL

Certificamos que el presente trabajo fue desarrollado por Jennifer Samanta Minayo Yujato, bajo nuestra supervisión.

Ph.D. FELIPE LEONEL GRIJALVA ARÉVALO
DIRECTOR DEL TRABAJO DE TITULACIÓN

Ph.D ANA MARÍA ZAMBRANO VIZUETE
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, JENNIFER SAMANTA MINAYO YUJATO, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

JENNIFER SAMANTA MINAYO YUJATO

DEDICATORIA

Este trabajo esta dedicado a:

A mi abuelito,
quien en vida me enseño que
para alcanzar mis metas
debo dar lo mejor de mi a cada paso.

Samanta Minayo

AGRADECIMIENTO

Agradezco a Dios, por ayudarme a completar esta etapa de mi vida.

A mis padres por su apoyo.

A mis hermanas quienes han sido mi mayor inspiración y fuerza para nunca rendirme.

Al Ph.D. Felipe Grijalva por su tiempo y su ayuda para completar este proyecto.

A mis amigos y compañeros que me acompañaron durante mi formación académica. Su ayuda fue esencial para poder alcanzar este logro. Especialmente agradezco a mis mejores amigas del colegio Adriana, Aracely y Alejandra. A mis amigos de prepo Eve, Kevin y Juan quienes a pesar del corto tiempo compartido en un aula han estado conmigo y me han apoyado durante esta etapa de mi vida. También, agradezco a los chicos del club de robótica Mafer, Karlitas, Diego, Pato, Andres y Jona quienes han sido un ejemplo de responsabilidad, esfuerzo y humildad desde que llegaron a mi vida. Finalmente, a mis amigos de la facultad Alexa, Julio, Daniel, Yesenia, Vicky y Marilyn quienes siempre me han apoyado en los mejores y peores momentos de mi formación académica y de mi vida.

Finalmente, a la Escuela Politécnica Nacional por darme la oportunidad de obtener los conocimientos y habilidades en mi desarrollo profesional.

Gracias.

Samanta

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

SEGMENTOS DE CÓDIGOS

RESUMEN

Para monitorear parámetros de la actividad física de atletas amateurs existen múltiples aplicaciones móviles tanto para dispositivos iOS y Android, estas aplicaciones recolectan la información del atleta sin considerar si esta actividad física corresponde a la participación del usuario en una carrera, por lo que el usuario no tiene acceso a los resultados promedio de los participantes de las carreras. Además, los usuarios solo tienen acceso a esta información desde la aplicación móvil por lo que si la aplicación no esta asociada a una cuenta de usuario se perderá la información recolectada.

Por ello se desarrollo un prototipo de plataforma que permita centralizar la información de las carreras en las que el usuario puede participar y la información de la actividad física de los atletas en cada carrera que ha participado. Este prototipo consta de una aplicación web y móvil que se comunican entre sí, y utilizan los servicios de Firebase para implementar las funcionalidades del prototipo como: autenticación, lectura y escritura de datos, almacenamiento de imágenes y manejo de notificaciones.

En el primer capítulo se presenta el marco teórico sobre las tecnologías utilizadas, las herramientas de desarrollo usadas y finalmente se describe la metodología ágil Kanban.

En el segundo capítulo se presentan los resultados de la entrevista inicial, las historias de usuario, los requisitos funcionales y no funcionales, la arquitectura y finalmente se muestra el diseño e implementación del prototipo de plataforma.

El tercer capítulo presenta los resultados de las pruebas funcionales de la aplicación web y móvil. Además, se presenta los resultados de la entrevista final realizada a los usuarios. Finalmente se muestra los errores encontrados en el prototipo de plataforma y las soluciones empleadas para cada uno de ellos.

En el cuarto capítulo se presentan las conclusiones y recomendaciones obtenidas en el desarrollo de este Trabajo de Titulación. Finalmente, como anexos se incluyen: las entrevistas, el código del proyecto de la aplicación móvil, web; y el manual de usuario del módulo de administración de la aplicación web.

PALABRAS CLAVE: Firebase, Android, React, Aplicación web y móvil.

ABSTRACT

To monitor parameters of the physical activity of amateur athletes there are multiple mobile applications for both iOS and Android devices, these applications collect the athlete's information without considering whether this physical activity corresponds to the user's participation in a race, so the user does not You have access to the average results of the race participants. In addition, users only have access to this information from the mobile application, so if the application is not associated with a user account, the information collected will be lost.

For this reason, a prototype platform was developed that allows centralizing the information of the races in which the user can participate and the information of the physical activity of the athletes in each race that they have participated. This prototype consists of a web and mobile application that communicate with each other, and use Firebase services to implement the functionalities of the prototype such as: authentication, data reading and writing, image storage and notification handling.

The first chapter presents the theoretical framework on the technologies used, the development tools used and finally the agile Kanban methodology is described.

The second chapter presents the results of the initial interview, the user stories, the functional and non-functional requirements, the architecture, and finally the design and implementation of the platform prototype are shown.

The third chapter presents the results of the functional tests of the web and mobile application. In addition, the results of the final interview with users are presented. Finally, the errors found in the platform prototype and the solutions used for each of them are shown.

The fourth chapter presents the conclusions and recommendations obtained in the development of this Degree Project. Finally, as annexes are included: the interviews, the project code of the mobile application, web; and the user manual of the web application administration module.

KEYWORDS: Firebase, Android, React, Mobile and web app

1. INTRODUCCIÓN

En este capítulo se describirán los antecedentes, los objetivos, el alcance y el marco teórico. Además, se presentarán las tecnologías y herramientas usadas en el desarrollo de este Trabajo de Titulación. Finalmente, se describirá la metodología ágil Kanban.

1.1. ANTECEDENTES

En el año 2016, únicamente en el Distrito Metropolitano de Quito los atletas tuvieron la oportunidad de escoger entre 96 carreras, además de las que se desarrollaron en distintas partes del país [2].

Cada año se realizan varias carreras dentro de las cuales la mayoría de los participantes son atletas amateurs, quienes buscan medios para recolectar información de su desempeño en las carreras en las que han participado, por lo que muchos recurren a sus smartphones en conjunto con aplicaciones y accesorios como porta-celulares, los cuales permiten al usuario correr con su smartphone de manera cómoda mientras recogen información [3].

A pesar de contar con una gran variedad de Apps como Google Fit, MyFitnessPal, Pacer, Sports Tracker, entre otras [4]; las cuales permiten recolectar información relacionada a la actividad física del usuario no existe alguna que recolecta información específica a una carrera, ya que las aplicaciones existentes guardan la información como actividad física correspondiente al día de la carrera es decir similar a cualquier día de entrenamiento.

Además, podemos recalcar que a pesar de que existen carreras que se realizan a gran escala y proporcionan equipos para el monitoreo de sus participantes la única información que brindan al usuario es el tiempo en el que completó la carrera, tal es el caso de la 15K Quito Últimas Noticias y la Ruta de las Iglesias [5] [6] por lo que de igual manera los atletas recurren a aplicaciones móviles para recolectar la información.

Con el desarrollo de este Trabajo de Titulación se plantea facilitar el almacenamiento y visualización de la información recolectada por cada atleta en cada carrera en la que participe, tanto individualmente como globalmente, además se busca generar un historial de desempeño del usuario donde pueda visualizar su avance en cada carrera realizada.

El enfoque de este Trabajo de Titulación es desarrollar una plataforma que conste de una aplicación móvil para el sistema operativo Android y una aplicación web, mediante las cuales el usuario pueda almacenar y acceder a la información relacionada a la actividad física realizada por el usuario en cada carrera. Además, de contar con un módulo de administración en la aplicación web mediante el cual se administrará y monitorizará las carreras atléticas.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Desarrollar un prototipo de plataforma web y móvil para la monitorización de parámetros de la actividad física de los participantes de carreras atléticas.

1.2.2. OBJETIVOS ESPECÍFICOS

- Analizar las tecnologías necesarias para el desarrollo de este prototipo de plataforma.
- Diseñar la estructura y componentes de los módulos del prototipo de plataforma.
- Implementar el prototipo de plataforma en base al diseño realizado.
- Analizar los resultados en base a las pruebas realizadas.

1.3. ALCANCE

En este Trabajo de Titulación se presenta el desarrollo de un prototipo de plataforma orientada a monitorear parámetros relacionados a la actividad física de los participantes de una carrera atlética, mediante el desarrollo de una aplicación móvil para el sistema operativo Android y una aplicación web.

Para el monitoreo de los parámetros de actividad física la aplicación web permite al administrador compartir un código de acceso a la carrera a los usuarios que estén inscritos en la misma, mediante el cual los atletas podrán dar inicio al monitoreo de su desempeño en la carrera.

Los atletas que deseen ser monitorizados deberán correr con su dispositivo Android en el cual tendrán instalada la aplicación móvil y deberán tener habilitado el internet y GPS, ya que mediante estos se obtendrán los datos de ubicación, velocidad y distancia y específicamente a través de internet se realizará el envío de información en tiempo real a la base de datos.

Para el inicio del registro de la información de la actividad física, el usuario se inscribirá en una carrera e ingresará el código proporcionado por el administrador el cual le permitirá dirigirse a una interfaz donde mediante un botón inicio y un botón fin dará inicio y fin a la transmisión de información a la base de datos.

La información transmitida por los atletas consiste en los parámetros de la actividad física realizada como ubicación, velocidad, tiempo, distancia y número de pasos, los cuales son obtenidos desde la aplicación móvil y son enviados hacia la base de datos y a su vez estos se mostrarán en la aplicación web a la que tendrá acceso el administrador de la carrera como se indica en la Figura 1.1 y se describe a continuación mediante el establecimiento de perfiles para usuarios autenticados y no autenticados.

Figura 1.1: Diagrama del prototipo

Para el funcionamiento del prototipo de plataforma la aplicación web y móvil constaran de usuarios autenticados y no autenticados. Para los usuarios autenticados se establecen dos perfiles: Perfil Atleta y Perfil Administrador.

USUARIOS AUTENTICADOS

- **Perfil Atleta:** Es el único perfil que tendrá acceso a la aplicación móvil, desde la cual después de iniciar sesión y crear su perfil podrá inscribirse en una carrera, monitorear parámetros relacionados su actividad física en cada carrera en la que participe, visualizar sus resultados en cada carrera, visualizar los resultados promedios de cada carrera, visualizar sus resultados globales a lo largo de las carreras en las que ha ido participando, comparar sus resultados con los resultados promedio de cada carrera y visualizar gráficas estadísticas de su desempeño y el desempeño global de los participantes en cada carrera por ejemplo distancia-tiempo y distancia-velocidad. Además, este perfil tendrá acceso a la aplicación web, desde la cual después de iniciar sesión solamente podrá visualizar la misma información que se muestra en la aplicación móvil.

- **Perfil Administrador:** Este perfil solo tendrá acceso a la aplicación web desde la cual tras iniciar sesión será el encargado de la administración de las carreras atléticas, es decir las crea, modifica, elimina y visualiza, además podrá monitorear los participantes de la carrera, hacer consultas sobre los resultados de cada carrera por ejemplo: filtrar resultados por género, por edad, por tiempo promedio y por velocidad promedio, también podrá visualizar gráficas estadísticas del desempeño global de los participantes en cada carrera por ejemplo distancia-tiempo y distancia-velocidad.

USUARIOS NO AUTENTICADOS

Los usuarios no autenticados solo tendrán acceso a la aplicación web, desde la cual únicamente podrán visualizar y realizar las mismas consultas que el administrador. Cabe recalcar que este tipo de usuario solo tiene permisos de visualización.

Para la interacción entre los usuarios mencionados anteriormente y el prototipo de plataforma se tienen varios módulos tanto para la aplicación web y aplicación móvil como se indica a continuación.

APLICACIÓN MÓVIL

- **Módulo Registrar Usuario:** Si el usuario no dispone de una cuenta podrá crear una en base a un correo y contraseña, además tendrá la oportunidad de registrarse en la aplicación con el correo de Gmail, una vez creado el usuario se dirige al Submódulo Ingresar Datos.
    - **Submódulo Ingresar Datos:** Una vez creada la cuenta del usuario pasará por un formulario donde ingresará información básica como nombre, edad, estatura y peso, algunos de los cuales son parámetros necesarios para tener un perfil del estado físico del usuario.
- **Módulo Autenticación:** La aplicación móvil realiza un proceso de autenticación, en el caso de que el usuario disponga de una cuenta y haya ingresado sus datos correctamente ingresa en el Módulo Principal, caso contrario se dirige al Submódulo Ingresar Datos.
- **Módulo Principal:** En este módulo se muestra una lista de publicaciones hechas por el administrador, además consta de un menú lateral desde el cual el usuario podrá dirigirse a los diferentes módulos de la aplicación o salir de la misma.
- **Módulo Usuario:** En este módulo el usuario podrá ver su perfil, incluyendo su información personal y de cómo va su desempeño a lo largo de todas las carreras en las que ha participado como velocidad promedio por kilómetro, tiempo promedio por kilómetro, distancia total, calorías totales quemadas y número total de pasos, además se mostrarán gráficas estadísticas como velocidad promedio por carrera, tiempo promedio por carrera y calorías quemadas por carreras.
- **Módulo Carreras:** La aplicación contará con una lista de todas las carreras creadas por el administrador próximas a realizarse y en las cuales el usuario podrá inscribirse. Además, cada vez que se genere una nueva carrera se generará una notificación para que el usuario esté al día con los eventos en las que podrá participar y contará con el siguiente submódulo.
    - **Submódulo Carrera:** Este módulo consta de información proporcionada por el administrador al momento de crear la carrera. Desde este módulo el usuario podrá

inscribirse en la carrera, cancelar la inscripción y monitorear su desempeño en la carrera, para este último el usuario deberá ingresar la clave proporcionada por el administrador para dar inicio al monitoreo de la información relacionada a la actividad física del atleta en la carrera.

- **Módulo Carreras por Usuario:** En este módulo se mostrará una lista de las carreras en las que el usuario ha participado y podrá acceder a cada una mediante el siguiente submódulo.
    - **Submódulo Carrera por Usuario:** En este submódulo se mostrarán los resultados de la carrera y de la participación del usuario.

APLICACIÓN WEB

La aplicación web será orientada tanto a administradores como hacía usuarios. Esta aplicación web constará de los siguientes módulos.

- **Módulo Autenticación:** La aplicación web realiza un proceso de autenticación para el usuario y el administrador.
- **Módulo Carreras:** La aplicación contará con una lista de todas las carreras creadas por el administrado. A este módulo podrán acceder los usuarios autenticados y no autenticados y consta del siguiente submódulo.
    - **Submódulo Carrera:** Si la carrera aún no se ha realizado este submódulo consta de información proporcionada por el administrador al momento de crear la carrera. Si la carrera ya se ha realizado en este módulo se mostrarán los resultados e información de la carrera, además se podrá realizar consultas sobre los resultados de cada carrera. Cabe recalcar que la información anteriormente descrita se mostrará tanto a usuarios autenticados como no autenticados. Finalmente, para el perfil de usuario, una vez que haya iniciado sesión en este módulo se mostrará la información anteriormente mencionada y además los resultados de la participación del usuario.
- **Módulo Usuario:** En este módulo el usuario autenticado podrá ver su perfil, incluyendo su información personal y de cómo va su desempeño a lo largo de todas las carreras en las que ha participado. Además se mostrarán gráficas estadísticas como velocidad promedio por carrera, tiempo promedio por carrera y calorías quemadas por carreras.
- **Módulo Administración de Carreras:** A este módulo sólo tendrá acceso el administrador una vez que haya iniciado sesión. En este módulo el administrador podrá crear, modificar y eliminar carreras.

1.4. MARCO TEÓRICO

En esta sección se detalla el conjunto de tecnologías y herramientas usadas para el diseño y desarrollo del prototipo de plataforma. Finalmente se da una corta introducción a la metodología ágil kanban para el desarrollo de productos software.

1.4.1. TECNOLOGÍAS

1.4.1.1. BASE DE DATOS NO RELACIONAL

Una base de datos es una colección de información organizada que poseen un mismo contexto y se almacenan de forma sistemática. Dependiendo del tipo de información y la estructura que se requiera para almacenarla existen múltiples tipos de bases de datos. Este trabajo se centra en el uso de base de datos no relacional y se hará una comparación con las bases de datos relacionales.

En las bases de datos relacionales los datos se encuentran organizados en tablas, las cuales se relacionan entre si mediante identificadores y cumplen con el modelo entidad relación. Las bases de datos no relacionales no cumplen con el esquema entidad relación, no usan tablas, ni poseen identificadores que relacionen tablas. Las bases de datos no relacionales almacenan su información sin esquemas exactos. Existen varios tipos de datos no relacionales: clave-valor, documento y gráficos. En este trabajo de titulación se utilizará la base de datos no relacional tipo documento [7].

Este trabajo de titulación consiste en recolectar la mayor cantidad de información de la actividad física de un atleta en tiempo real durante su participación en una carrera, lo cual implica que por cada usuario y por cada carrera el volumen de información en la base de datos seguirá creciendo rápidamente por lo que en base a esta consideración del crecimiento de información la base de datos recomendad es una base no relacional.

1.4.1.2. FIREBASE

Es un servicio BaaS [1] (Backend-As-A-Service) propio de Google que proporciona soluciones para los desarrolladores. El servicio está diseñado para acelerar la integración de funciones basadas en la nube en dispositivos móviles y aplicaciones web.

Firebase ofrece servicios como Authentication, Realtime Database, Storage, Cloud Messaging, Hosting, entre otros.

A continuación, se detallarán los servicios que serán usados en el desarrollo de este Trabajo de Titulación [8].

- **Firebase Authentication:** Permite integrar varios proveedores de autenticación como: correo electrónico/contraseña, teléfono, Google, Apple, Microsoft, Facebook, Twitter, etc.

[1] BaaS (Backend-As-A-Service) es un servicio que permite vincular aplicaciones web y móviles hacia el backend en la nube

- **Firebase Realtime Database:** Consiste en una base de datos no relacional alojada en la nube, la cual almacena datos en formato JSON y además permite a los usuarios de las aplicaciones conectadas a la base de datos obtener la información en tiempo real.
- **Firebase Cloud Storage:** Este servicio permite almacenar archivos como imágenes, audio, video y otros tipos de contenido generado por los usuarios de las aplicaciones conectadas al storage. Además, este servicio posee una estructura de almacenamiento basado en carpetas facilitando la organización y acceso al contenido almacenado.
- **Firebase Hosting:** Permite el alojamiento seguro y rápido para aplicaciones web. El contenido de la aplicación puede ser dinámico o estático.
- **Cloud Messaging:** Es una solución de mensajería multiplataforma que permite enviar mensajes de forma segura y gratuita. Dentro de este trabajo de Titulación será utilizado para el manejo de notificaciones.

Los servicios de firebase son compatibles con múltiples plataformas y gracias a que proporciona un SDK [2] (Software Development Kit) para los desarrolladores facilita la integración de sus servicios en diferentes tipos de aplicaciones.

1.4.1.3. REACT JS

React JS es una biblioteca de Javascript desarrollada por facebook que permite implementar interfaces de usuario interactivas de forma sencilla. También, se encarga de la actualización y renderización de forma eficiente de los componentes en función de los cambios que se generen durante la interacción sobre la interfaz de usuario.

La estructura de React se basa en la creación de componentes que manejan sus propios estados y los transforma en interfaces de usuario complejas, la lógica de estos componentes está escrita en Javascript y permite el paso de información de manera sencilla a través de la aplicación [9].

React JS trabaja con HTML y CSS por lo que a continuación se dará una pequeña descripción de cada uno.

HTML

Es un lenguaje de marcas de hipertexto, es el componente más básico de la Web. Su nombre hace referencia a los enlaces que conectan páginas web entre sí, ya que utiliza marcas para etiquetar texto, imágenes y otro contenido para mostrarlo en el navegador web [10].

CSS

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo, el cual permite aplicar estilos de manera selectiva sobre los elementos del documento HTML, por ejemplo, configurar posicionamiento, colores, tamaño, animación entre otros [11].

[2] SDK (Software Development Kit) es un grupo de herramientas que permiten el desarrollo de aplicaciones

1.4.1.4. ANDROID SDK

Es una herramienta disponible para el desarrollo de aplicaciones para la plataforma de Android. Además de facilitar la creación de aplicaciones, permite la evaluación de las mismas y la reparación de errores. Esta herramienta es la base sobre la cual se han desarrollado la mayoría de las aplicaciones disponibles en Google Play [12].

Con este paquete se hace posible el desarrollo de aplicaciones de Android usando Java como lenguaje de programación. Además, está formado por varios componentes los cuales facilitan el desarrollo de aplicaciones. A continuación, se describen los componentes básicos para el desarrollo de aplicaciones [13].

- **Android SDK Tools:** este componente posee las herramientas necesarias para la creación y desarrollo de aplicaciones.
- **Android SDK Build-Tools:** son las herramientas de compilación de Android, se requieren para la creación de aplicaciones por lo que se deben mantener actualizadas constantemente.
- **Android SDK Platform-Tools:** este componente posee la ventana de comandos que permite ejecutar varias acciones en un dispositivo, así como instalar y depurar aplicaciones.
- **SDK Platform:** permite determinar la versión de Android en la que se desarrollará la aplicación. Es importante elegir una versión y considerar que en base a la versión seleccionada las versiones más antiguas de Android no serán compatibles con la aplicación desarrollada.
- **Android Emulator:** este componente permite al desarrollador probar el funcionamiento de las aplicaciones en un entorno similar al de un dispositivo Android real como: teléfonos, relojes, tablets, entre otros.

1.4.2. HERRAMIENTAS DE DESARROLLO

1.4.2.1. NODE JS

Es un entorno de ejecución para JavaScript multiplataforma y de código abierto construido con el motor de JavaScript V8 de Chrome [3], encargado de gestionar los servicios en la ejecución de la aplicación [14].

Node.js está orientado a eventos asincrónicos, sin bloqueos y se encarga de la ejecución de varias tareas con un bajo consumo de recursos convirtiéndolo en un entorno de ejecución eficiente y liviano.

La herramienta Node.js posee su propio sistema gestor de paquetes NPM (Node Package Manager), el cual permite administrar módulos, distribuir paquetes e instalar dependencias para el desarrollo de aplicaciones [15].

[3] JavaScript V8 de Chrome es un motor de código de JavaScript y WebAssembly

1.4.2.2. ANDROID STUDIO

Android Studio es el IDE[4] (Integrated Development Environment) oficial de la plataforma Android que ofrece funciones para el desarrollo de apps para Android, como las siguientes [16]:

- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Integración con GitHub y plantillas de código.
- Compatibilidad integrada con Google Cloud Platform.

La Figura 1.2 muestra una captura de pantalla de Android Studio.

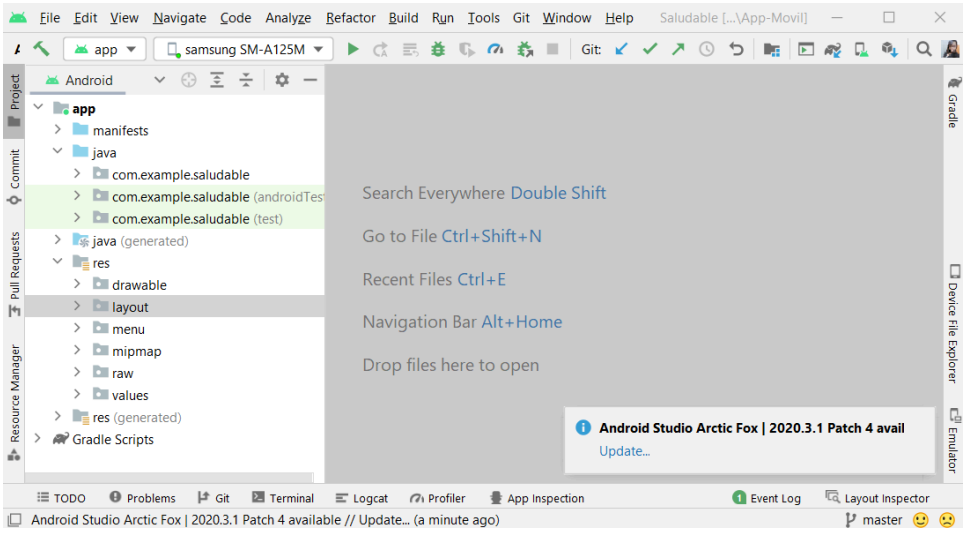


Figura 1.2: Captura de pantalla de Android Studio

1.4.2.3. VISUAL STUDIO CODE

Es un editor de código fuente desarrollado por Microsoft y está disponible para los sistemas operativos Windows, macOS y Linux.

Tiene soporte de Node.js incorporado y permite la administración de diferentes extensiones para el uso de varios lenguajes de programación. Proporciona soporte para depuración mediante el terminal, control de versiones a través de Git y brinda acceso a una tienda de extensiones descargables facilitando la integración de diferentes tecnologías y librerías [17].

La Figura 1.3 muestra una captura de pantalla del editor de código.

[4]IDE (Integrated Development Environment) es un entorno de desarrollo integrado que proporciona herramientas para el desarrollo de software

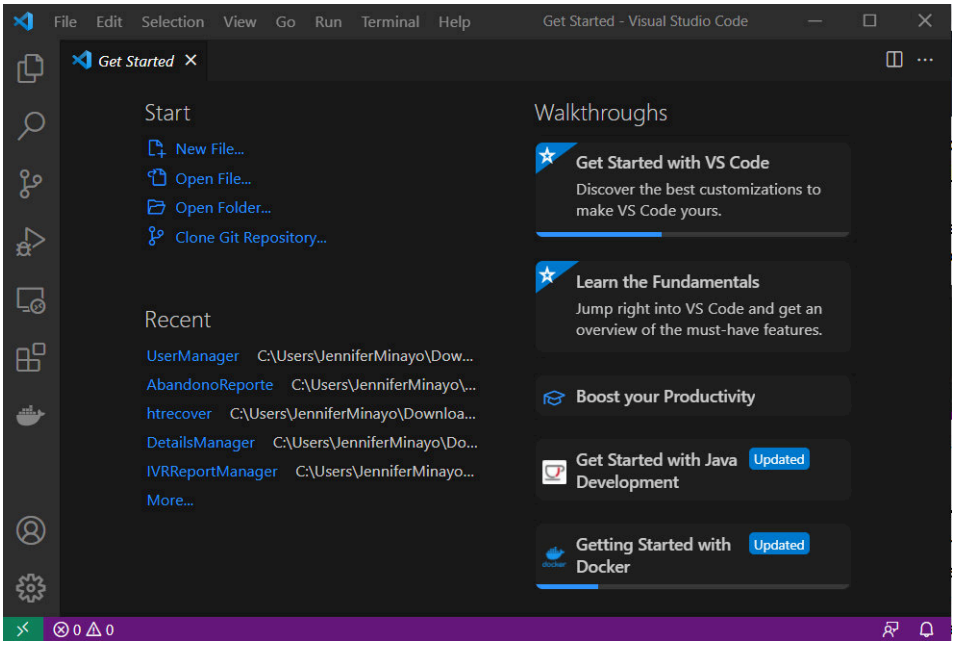


Figura 1.3: Captura de pantalla de Visual Studio Code

1.4.2.4. FIREBASE CONSOLE

Google proporciona a los desarrolladores la consola de administración de proyectos en firebase, mediante la cual el desarrollador puede crear, eliminar o configurar cada proyecto con los diferentes servicios ofrecidos por firebase como Authentication, Realtime Database, Storage, Cloud Messaging, Hosting, entre otros.

La Figura 1.4 muestra la pantalla principal de la consola de firebase donde se muestran los proyectos creados y la opción de agregar un nuevo proyecto.

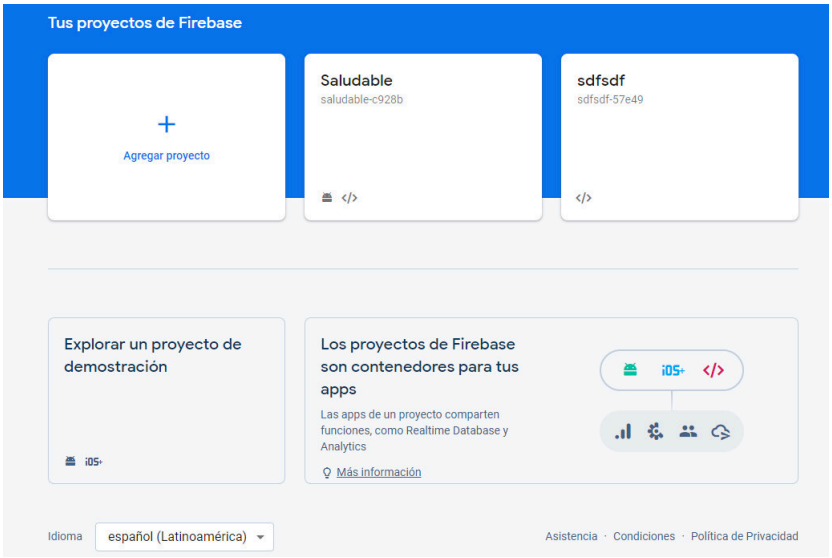


Figura 1.4: Captura de pantalla de consola de firebase

Finalmente, en la figura 1.5 se observa la consola de firebase desde la cual se configuran

los servicios del proyecto. Además. la consola de firebase permite el monitoreo del uso de los servicios de firebase consumidos por cada aplicación.

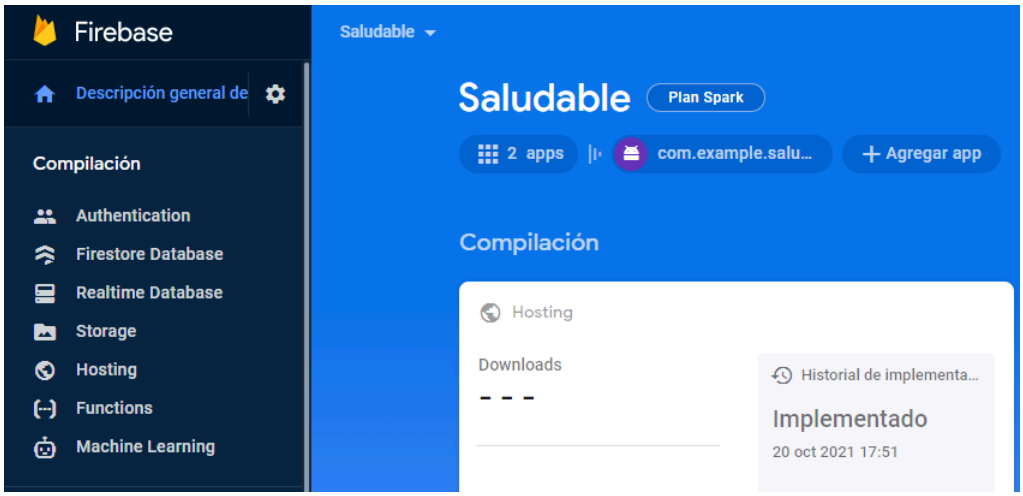


Figura 1.5: Captura de pantalla de servicios de firebase

1.4.2.5. GIT Y GITHUB

El versionamiento de código fuente dentro del desarrollo de software es fundamental ya que permite mantener un histórico de los cambios realizados sobre el proyecto, con el fin de acceder a versiones anteriores en caso de que los últimos cambios no han sido satisfactorios o han generado algún problema sobre la aplicación.

Git es una herramienta de código abierto para el control de versiones que se implementa de manera local sobre el dispositivo en el que se encuentra el ambiente de desarrollo. Git permite volver a una versión anterior y permite trabajar en equipo [18].

Github es una plataforma para el control de versiones usando Git. Esta plataforma está basada en la nube por lo que es un servicio de alojamiento y administración de repositorios de git [18].

1.4.3. METODOLOGÍA ÁGIL KANBAN

Kanban es una metodología ágil que permite visualizar y gestionar el flujo de trabajo. Kanban se basa en la filosofía de la metodología Lean [19]:

- Alta calidad: eliminar defectos, detectar y corregir problemas desde el origen.
- Reducción del desperdicio: eliminar actividades que no presentan valor y optimizar el uso de recursos.
- Mejora continua: reducir costos, aumentar calidad y productividad.
- Flexibilidad: producción de variedad de productos, priorizar y condicionar tareas de la cola.
- Productos pull: los productos deben ser solicitados por el usuario final.

“Lean es básicamente todo lo concerniente a obtener las cosas correctas en el lugar correcto, en el momento correcto, en la cantidad correcta, minimizando el desperdicio, siendo flexible y estando abierto al cambio” [20].

Kanban divide el trabajo en bloques (ítems) y los permite visualizar en el flujo de trabajo mediante el uso de tarjetas y tableros. Además, procura establecer tiempos de ejecución cortos para completar un ítem.

1.4.3.1. Tarjetas Kanban

Las tarjetas kanban se implementan por cada ítem y contienen la información de la persona responsable de la ejecución, la descripción de la tarea y el tiempo estimado de realización. En cualquier punto de tiempo como: definición, desarrollo y finalización de la tarea, la tarjeta representa el estado actual del trabajo [21].

1.4.3.2. Tablero Kanban

Permite visualizar, optimizar y estandarizar el flujo de trabajo mediante la definición de tres estados principales: por hacer, en progreso y listo, además tiene la flexibilidad de añadir nuevos estados según las necesidades y requerimientos de cada proyecto.

Cuando se ha completado un ítem, se mueve la tarjeta que lo representa al siguiente estado del flujo de trabajo. Es decir, el objetivo de un tablero Kanban es tener siempre presente el trabajo que se debe realizar y la prioridad de cada uno [22].

En la Figura 1.6 se observa el tablero kanban con los tres estados.

Figura 1.6: Tablero Kanban

2. METODOLOGÍA

El presente Trabajo de Titulación propone desarrollar un prototipo de plataforma orientada al monitoreo de parámetros relacionados a la actividad física de los participantes de una carrera atlética mediante el uso de la metodología ágil Kanban. Esta metodología propone dividir el trabajo en diferentes bloques que se conocen como ítems, los cuales serán representados en tarjetas Kanban que serán ubicadas en el tablero Kanban con el objetivo de visualizar el flujo de trabajo en todo momento.

Este capítulo se presentará en tres fases: análisis, diseño e implementación. Donde cada fase consta de un conjunto de tareas a ser realizadas llamados ítems, los cuales serán ubicados en un tablero Kanban basado en tres estados: por hacer, en progreso y listo.

Fase de Análisis. Es donde se realizará el levantamiento de información por medio de entrevistas a un mínimo de diez potenciales usuarios con el perfil de atleta, estos usuarios corresponden a atletas que no monitorean su actividad física, en base a la información recolectada de estas entrevistas se generarán historias de usuario con el objetivo de obtener los requerimientos funcionales y no funcionales de la aplicación. Por lo tanto, en esta fase es donde se crea la visión del proyecto.

Fase de Diseño. En esta fase se diagramará la arquitectura del prototipo de plataforma. Además, se elaborarán los diagramas de actividades, diagramas de casos de uso, diagrama de clases y el esquema de base de datos. Finalmente se realizarán los mockups para aplicación móvil y aplicación web.

Fase de Implementación. En primer lugar, se configurarán las interfaces de Google Cloud Platform. Después, se codificará la UI de la aplicación móvil. Luego se añadirán a la aplicación móvil las funcionalidades de autenticación, lectura y escritura en la base de datos de Firebase; Push Notifications y tracking de la actividad física realizada por el usuario. A continuación, se codificará la UI(Interfaz de Usuario) de la aplicación web y se añadieron las funcionalidades de autenticación, lectura y escritura en la base de datos y storage. Finalmente se desarrolló el módulo de administración en la aplicación web para la gestión del contenido que se mostrará a los usuarios de la aplicación móvil y web.

2.1. ANÁLISIS

La fase de Análisis inicia con el establecimiento de tres ítems de trabajo. Estos ítems consisten en el levantamiento de información, procesamiento de la información y finalmente con

la definición de los requerimientos funcionales y no funcionales del prototipo de plataforma.

En la Figura 2.1 se observan los ítems de trabajo las cuales son representados como tarjetas Kanban y se encuentran ubicadas dentro de la columna por hacer. Como se muestra en la figura las columnas haciendo y hecho se encuentran vacías ya que no se ha iniciado ninguna tarea.

Figura 2.1: Tablero Kanban para la Fase de Análisis

A continuación, se procede con el desarrollo de los ítems de trabajo planteados.

2.1.1. ENTREVISTAS

Para obtener las historias de usuario se realizaron entrevistas a diez potenciales usuarios del prototipo de plataforma con el perfil de atleta, estas entrevistas se basaron en un cuestionario formado por ocho preguntas. El modelo de la entrevista se encuentra en el ANEXO A y sus resultados se detallan a continuación:

En la Figura 2.2 se muestran los resultados obtenidos de la pregunta 1. Esta pregunta tiene como objetivo conocer la información personal que el usuario desea compartir con la aplicación móvil. Los resultados recogidos muestran que considerando el objetivo de la aplicación móvil los datos de interés al momento de registrar al usuario son la foto de perfil, nombre de usuario, género, estatura y peso.

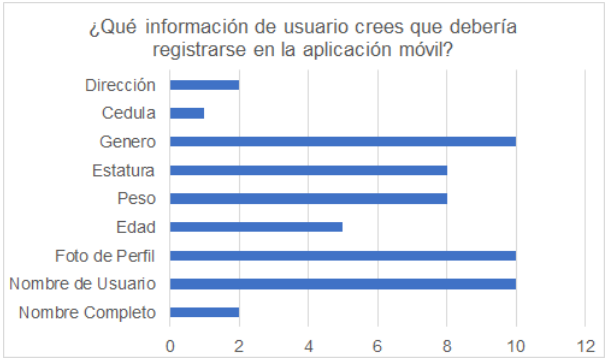


Figura 2.2: Resultados de la pregunta 1

La segunda pregunta establece ¿Cree qué es necesario y aceptaría que la aplicación móvil reciba una notificación cuando exista una nueva carrera?, esta pregunta tiene como objetivo conocer la predisposición del usuario a recibir notificaciones y como se muestra en la Figura 2.3 el 90 % de los entrevistados contestó que SI es necesario y aceptaría recibir notificaciones en la aplicación móvil.

Figura 2.3: Resultados de la pregunta 2

La tercera pregunta indica ¿Qué información de la carrera cree usted que debería mostrarse al usuario?, el objetivo de esta pregunta es conocer qué información correspondiente a una carrera es considerada importante para los usuarios. En la Figura 2.4 se muestra que los entrevistados consideran que se debe obtener información lo más completa y detallada posible para cada carrera. Esta pregunta aplica tanto a la aplicación móvil como a la aplicación web.

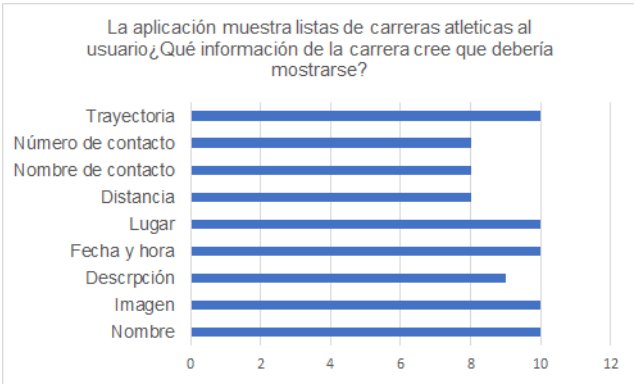


Figura 2.4: Resultados de la pregunta 3

La cuarta pregunta indica ¿Qué información monitorizada cree que debería mostrarse al finalizar la carrera?, el objetivo de esta pregunta es conocer qué información correspondiente a una carrera es considerada importante para los usuarios. En la Figura 2.5 se muestra que los entrevistados consideran que se debe obtener información lo más completa y detalla-

da posible para cada carrera. Esta pregunta aplica tanto a la aplicación móvil como a la aplicación web.

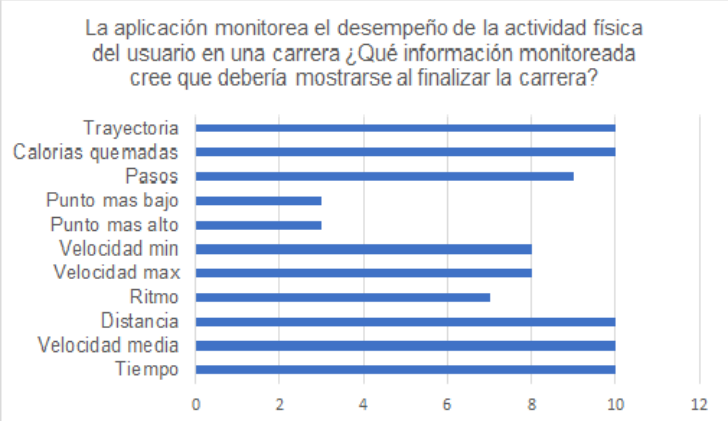


Figura 2.5: Resultados de la pregunta 4

La quinta pregunta se centra en la información recolectada por el usuario al monitorear su actividad física realizada en una carrera, para lo cual la pregunta indica ¿Qué información monitorizada del usuario cree que debería verse al finalizar la carrera?, en esta pregunta se establece una serie de parámetros físicos que pueden ser recolectados por la aplicación móvil, para posteriormente ser mostrados al usuario. Como se indica en la Figura 2.6 los entrevistados consideran la trayectoria, el tiempo, la velocidad media, la distancia y calorías quemadas como los parámetros físicos más relevantes. Esta pregunta aplica tanto a la aplicación móvil como a la aplicación web.

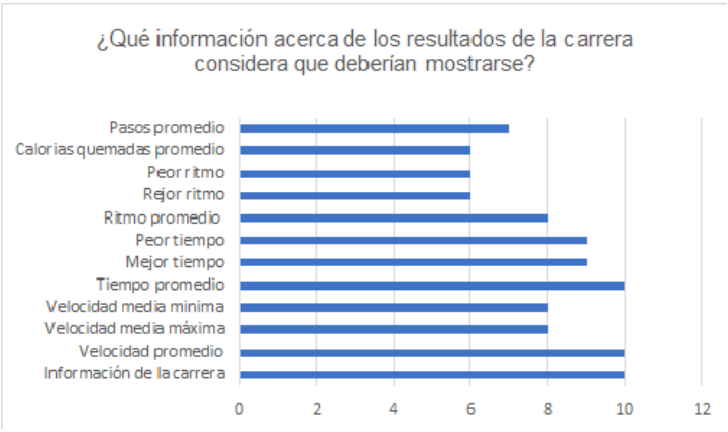


Figura 2.6: Resultados de la pregunta 5

La sexta pregunta es acerca de la visualización de las carreras en las que el usuario se ha inscrito y las carreras en las que el usuario se ha monitorizado y ha almacenado información, para ello esta pregunta indica ¿Cómo crees que se deberían visualizar las carreras en las

que se ha inscrito el usuario y las carreras en las que el usuario ya registra datos?, para esta pregunta cómo se indica en la Figura 2.7 el 80 % de los entrevistados consideró que se debe mostrar estas carreras por separado. Esta pregunta aplica tanto a la aplicación móvil como a la aplicación web.

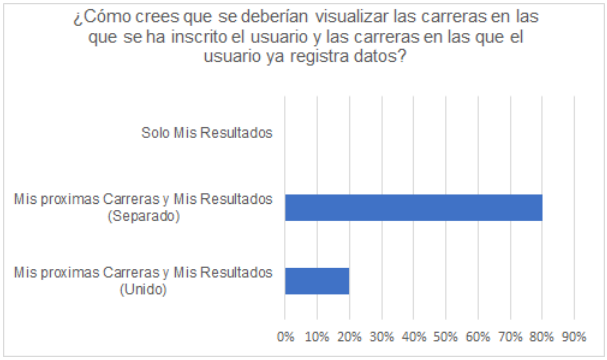


Figura 2.7: Resultados de la pregunta 6

La séptima pregunta tiene como objetivo conocer que información desea el usuario que se muestre dentro del resultado promedio de todas las carreras en las que ha participado, así como también conocer si el usuario desea ver la información de su última carrera, por lo cual esta pregunta menciona ¿Qué información cree usted que debería mostrarse en los resultados promedio de todas las carreras en las que ha participado el usuario?

Como se indica en la Figura 2.8 los entrevistados consideran importante mostrar los resultados promedios de todas las carreras y de la última carrera en la que ha participado el usuario. Esta pregunta aplica tanto a la aplicación móvil como a la aplicación web.

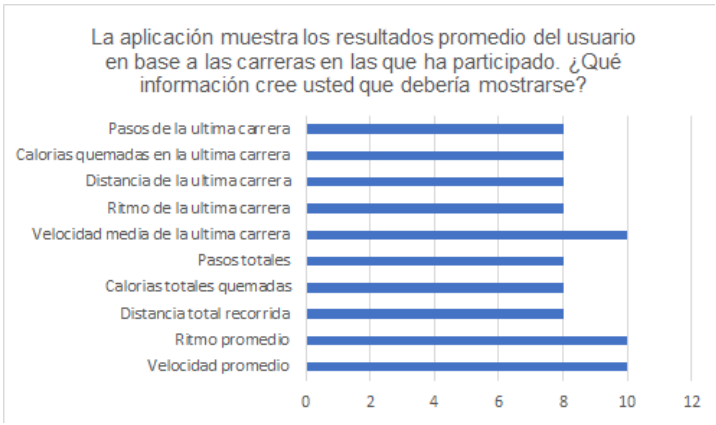


Figura 2.8: Resultados de la pregunta 7

La octava y última pregunta indica ¿Cree usted necesario un módulo de ayuda donde se explique el funcionamiento de la aplicación?, esta pregunta tiene como objetivo conocer si

es o no necesario crear un módulo de ayuda en la aplicación móvil. En la Figura 2.9 se puede observar que el 100 % de los entrevistados consideraron que Si es necesaria la existencia de este módulo. Esta pregunta aplica únicamente dentro de la aplicación móvil.

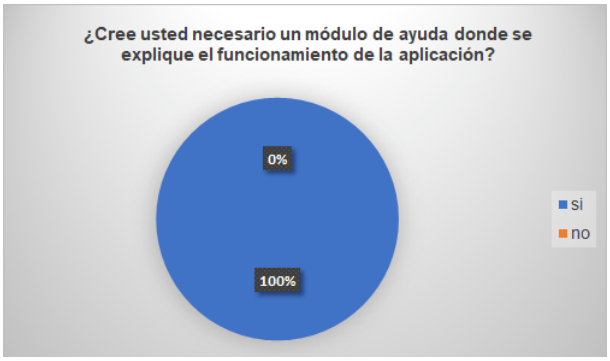


Figura 2.9: Resultados de la pregunta 8

2.1.2. HISTORIAS DE USUARIO

Las entrevistas realizadas permitieron identificar cuáles son las expectativas de los usuarios al interactuar con el prototipo de plataforma. Se agrupó la información provista por los entrevistados en categorías y cada una de estas categorías representa una historia de usuario. 2.1

Tabla 2.1: Historias de Usuario [1]

ID	Título	Descripción
1	Ingresar y Editar Información del usuario	El usuario necesita ingresar en el prototipo la información de su nombre completo, nombre de usuario, edad, estatura, peso y género.
2	Recibir Notificaciones	El usuario recibirá una notificación cada vez que exista una nueva carrera disponible.
3	Mostrar información de cada Carrera	El usuario desea ver la información de las carreras en las que podría participar.
4	Registrar inscripción en Carrera	El usuario desea poder inscribirse y retirar su inscripción de la carrera.
5	Monitorizar información de actividad física	El usuario quiere monitorear los parámetros relacionados a la actividad física realizada en una carrera.
6	Mostrar Carreras (Próximas y Realizadas)	El usuario desea poder acceder a una lista de las carreras en las que se ha inscrito y en otra lista acceder a todas las carreras en las que ha participado.
7	Mostrar Resultados por Carrera	El usuario quiere tener acceso a los resultados de la carrera esto incluye sus resultados individuales y los resultados globales de la carrera.
8	Mostrar Resultados Promedios de todas las carreras del usuario	El usuario quiere tener acceso a los resultados de la carrera esto incluye sus resultados individuales y los resultados globales de la carrera.

2.1.3. REQUERIMIENTOS DE LA APLICACIÓN

En base a las entrevistas realizadas y a las historias de usuario se han identificado los requerimientos funcionales y no funcionales de este Trabajo de Titulación.

2.1.3.1. REQUERIMIENTOS FUNCIONALES

Las historias de usuario definen la perspectiva de los usuarios acerca del funcionamiento del prototipo de plataforma por lo que es necesario definir de manera formal cuales son los requerimientos funcionales para la aplicación móvil y web definidos en base a las historias de usuario. Además, es necesario definir los requerimientos funcionales del administrador de carreras dentro de la aplicación web ya que este perfil no consta dentro de las historias de usuario.

Aplicación Móvil

- Autenticar a un usuario mediante los servicios de Authentication de Firebase.
- Registrar un usuario con su información personal.
- Editar información personal del usuario.
- Recibir notificación cada vez que exista una nueva carrera disponible.
- Registrar y eliminar inscripción del usuario en una carrera.
- Monitorizar parámetros de la actividad física del usuario en una carrera.
- Visualizar la lista de las carreras en las que se ha inscrito y la lista de las carreras en las que ha participado.
- Visualizar los resultados de la participación del usuario en la carrera.
- Visualizar los resultados promedios de todos los usuarios que participaron en la carrera
- Visualizar los resultados promedios de todas las carreras en las que ha participado el usuario.
- Cerrar Sesión.

Aplicación Web

- Autenticar a un usuario.
- Visualizar la lista de las carreras en las que se ha inscrito y la lista de las carreras en las que ha participado.

- Visualizar los resultados de la participación del usuario autenticado en la carrera.
- Visualizar los resultados promedios de todas las carreras en las que ha participado el usuario autenticado.
- Cerrar Sesión.
- Permitir a un usuario administrador crear, editar, eliminar y monitorear las carreras atléticas que se mostraran al usuario.
- Permitir a un usuario administrador enviar una notificación a los usuarios de la App móvil cada vez que se cree una carrera.
- Permitir a usuarios autenticados y no autenticados visualizar y filtrar los resultados promedios de todos los usuarios que participaron en la carrera.

2.1.3.2. REQUERIMIENTOS NO FUNCIONALES

- Crear una aplicación Android en el entorno de desarrollo Android Studio.
- Crear una aplicación web con la librería React JS.
- Alojar la base de datos en Firebase.
- Enviar y recibir notificaciones mediante Cloud Messaging de Firebase.
- Autenticar mediante Firebase Authentication.
- Publicación de la página web con Firebase Hosting.

2.2. DISEÑO

En esta fase se utilizarán diferentes tipos de diagramas UML (Unified Modeling Language), los que permitirán realizar el diseño de la estructura y definir el comportamiento del prototipo de plataforma. Iniciamos esta fase con la definición de siete ítems de trabajo, los cuales se pueden observar ubicados en las columnas por hacer y haciendo de la Figura 2.10.

2.2.1. DIAGRAMA DE LA ARQUITECTURA DEL PROTOTIPO

En la Figura 2.11 se puede observar la arquitectura del prototipo planteada. La implementación de esta arquitectura está basada en el uso del BasS de Google Firebase.

El prototipo cuenta con: una aplicación móvil y una aplicación web en el lado del cliente. Además, usará Firebase que evita la necesidad de desarrollar la parte del servidor. Firebase

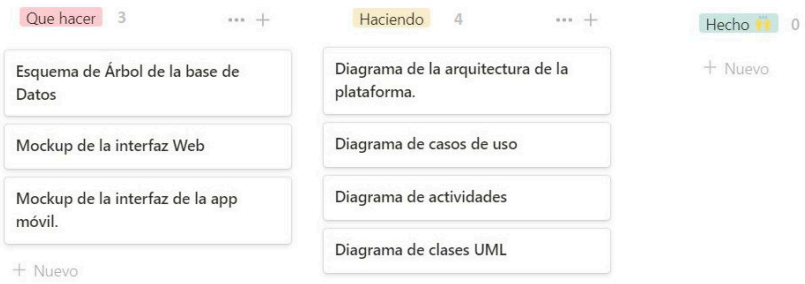


Figura 2.10: Tablero Kanban para la Fase de Diseño

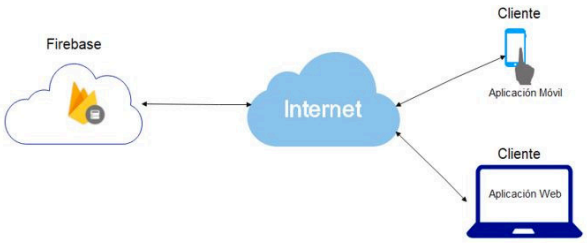


Figura 2.11: Arquitectura del prototipo

también ofrece numerosos servicios que se usarán en el desarrollo de este prototipo de plataforma, entre los cuales tenemos Realtime Database, Authentication, Storage, Hosting y Cloud Messaging.

Las aplicaciones clientes se comunicarán mediante el BasS de Firebase, para obtener información ubicada en la base de datos, imágenes almacenadas en el storage, realizar el proceso de autenticación y generar y recibir notificaciones, estas últimas están basadas en el modelo publicación y suscripción, el cual es un método que permite el envío de mensajes a varios dispositivos que se hayan suscrito en un tema específico [23].

2.2.2. CASOS DE USO

El diagrama de casos de uso representa la forma en que el sistema interactúa con las entidades externas, permite modelar el comportamiento de una aplicación con el fin de comprender el uso de la misma. Un diagrama de casos de uso está formado de tres elementos: actores, casos de uso y relaciones de uso [24].

Para la realización del diagrama de casos de uso se considerarán los siguientes actores:

- Administrador (Usuario autenticado).
- Usuario no registrado (Usuario no autenticado).
- Usuario registrado (Usuario autenticado).

En base a los requerimientos obtenidos se han generado los siguientes diagramas de caso de usos, los cuales muestran la interacción de los actores identificados con el prototipo de plataforma.

Un administrador solo tiene acceso a la aplicación web donde su principal función es gestionar las carreras atléticas, es decir la información que se brindará a los usuarios de la aplicación móvil. Este actor puede: iniciar sesión; crear, eliminar, editar y visualizar publicaciones que se mostrarán tanto en la página principal de la app móvil y la app web; crear, eliminar, editar y visualizar carreras atléticas; enviar notificaciones cada vez que el administrador cree una nueva carrera; visualizar la lista de usuarios inscritos en una carrera; monitorear carreras; visualizar la lista de usuarios que participan en una carrera; visualizar los resultados de la carreras, es decir resultados promedio, mejores resultados y peores resultados en base al desempeño de todos los participantes de la carrera atlética; filtrar los resultados de la carrera y cerrar sesión. El diagrama se muestra en la Figura 2.12

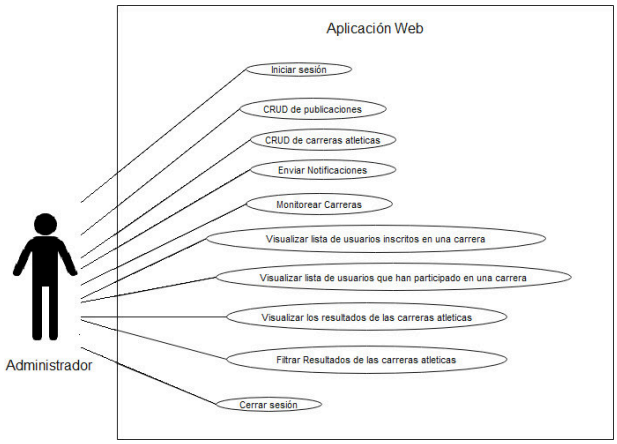


Figura 2.12: Casos de Uso: Administrador

Como se indica en la Figura 2.13 dentro de la aplicación móvil el usuario no registrado solo puede registrarse.

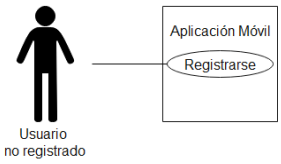


Figura 2.13: Casos de Uso: Usuario no registrado

Un usuario registrado dentro de la aplicación móvil puede emplear todas las funciones existentes en esta, como: iniciar sesión; ingresar y actualizar su información de perfil; recibir no-

tificaciones cada vez que el administrador cree una nueva carrera; visualizar la información más relevante de las carreras; registrar y eliminar la inscripción en una carrera; visualizar la lista de carreras en las que el usuario se ha inscrito y acceder a la información de cada una de ellas; monitorear parámetros de la actividad física del usuario en una carrera; visualizar lista de carreras en las que el usuario a participado; visualizar sus resultados obtenidos al monitorear su participación en una carrera; visualizar los resultados de la carreras, es decir resultados promedio, mejores resultados y peores resultados en base al desempeño de todos los participantes de la carrera atlética; visualizar su información de perfil; visualizar los resultados promedios de todas las carreras en las que ha participado el usuario y cerrar sesión. El diagrama se muestra en la Figura 2.14

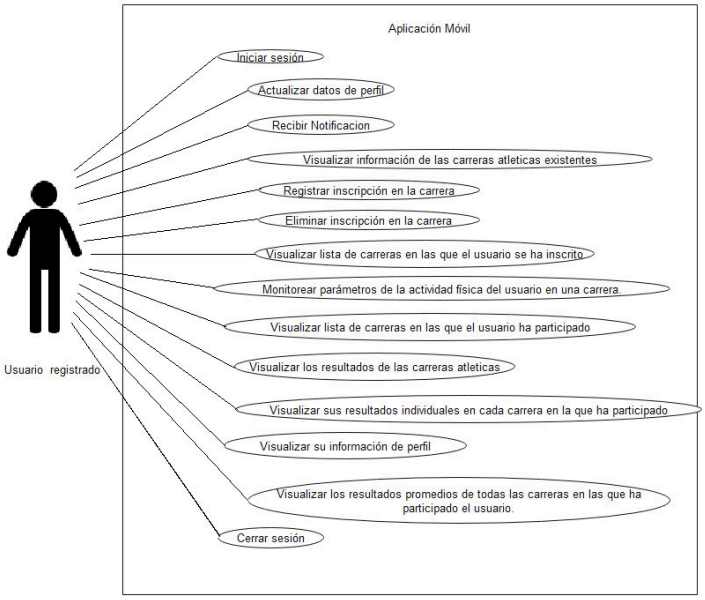


Figura 2.14: Casos de Uso: Usuario registrado

Un usuario no registrado dentro de la aplicación web se le permite: visualizar información de las carreras atlética existentes, visualizar los resultados de las carreras atléticas y filtrar resultados de las carreras atléticas realizadas. El diagrama se muestra en la Figura 2.15

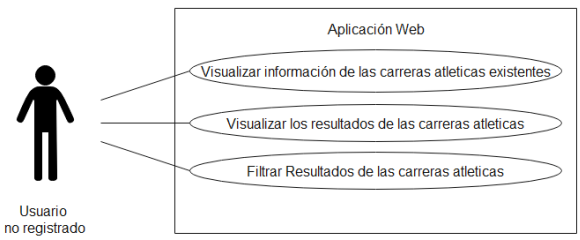


Figura 2.15: Casos de Uso: Usuario no registrado

Un usuario registrado dentro de la aplicación web puede emplear la mayoría de las funciones existentes en esta, como: iniciar sesión; visualizar la información más relevante de las carreras; visualizar la lista de carreras en las que el usuario se ha inscrito y acceder a la información de cada una de ellas; visualizar lista de carreras en las que el usuario a participado; visualizar sus resultados obtenidos al monitorear su participación en una carrera; visualizar los resultados de la carreras, es decir resultados promedio, mejores resultados y peores resultados en base al desempeño de todos los participantes de la carrera atlética; filtrar los resultados de la carrera; visualizar su información de perfil; visualizar los resultados promedios de todas las carreras en las que ha participado el usuario y cerrar sesión. El diagrama se muestra en la Figura 2.16

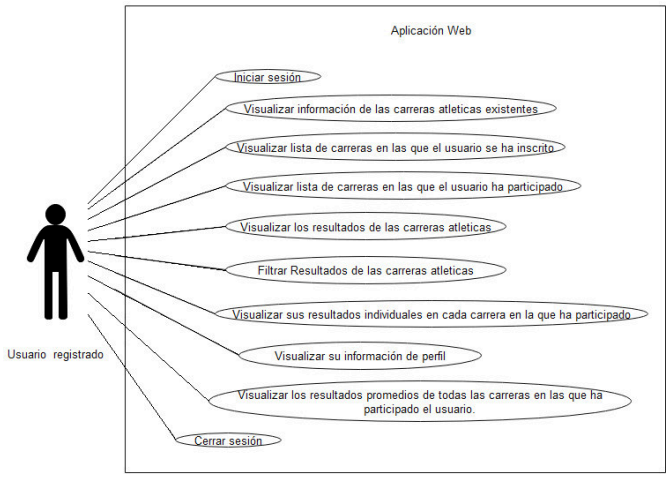


Figura 2.16: Casos de Uso: Usuario registrado

2.2.3. DIAGRAMA DE ACTIVIDADES

Un diagrama de actividades permite visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos, es decir modelar el flujo de la aplicación. Dentro del flujo se puede observar pasos secuenciales, concurrentes y condiciones [25].

Dentro del prototipo de plataforma permite modelar el flujo de integración entre las aplicaciones móvil y web con los servicios proporcionados por firebase.

2.2.3.1. ADMINISTRADOR

Partimos con el caso de uso del administrador ya que él gestionará la información que se mostrará a los usuarios registrados y no registrados tanto de la aplicación móvil como de la aplicación web.

Como se estableció anteriormente el Administrador únicamente tendrá acceso a la aplicación web, en donde tras ser autenticado podrá gestionar sus publicaciones y las carreras atléticas que se pondrán a disposición de los usuarios del prototipo.

Como se indica en la Figura 2.17 el administrador podrá gestionar los post o publicaciones, para ello dentro del menú se selecciona administración de post donde se mostrará un formulario para la creación de un nuevo post, adicional a esto se tendrá la lista de post creados anteriormente, donde el administrador podrá seleccionar cualquier post que desee editar o eliminar.

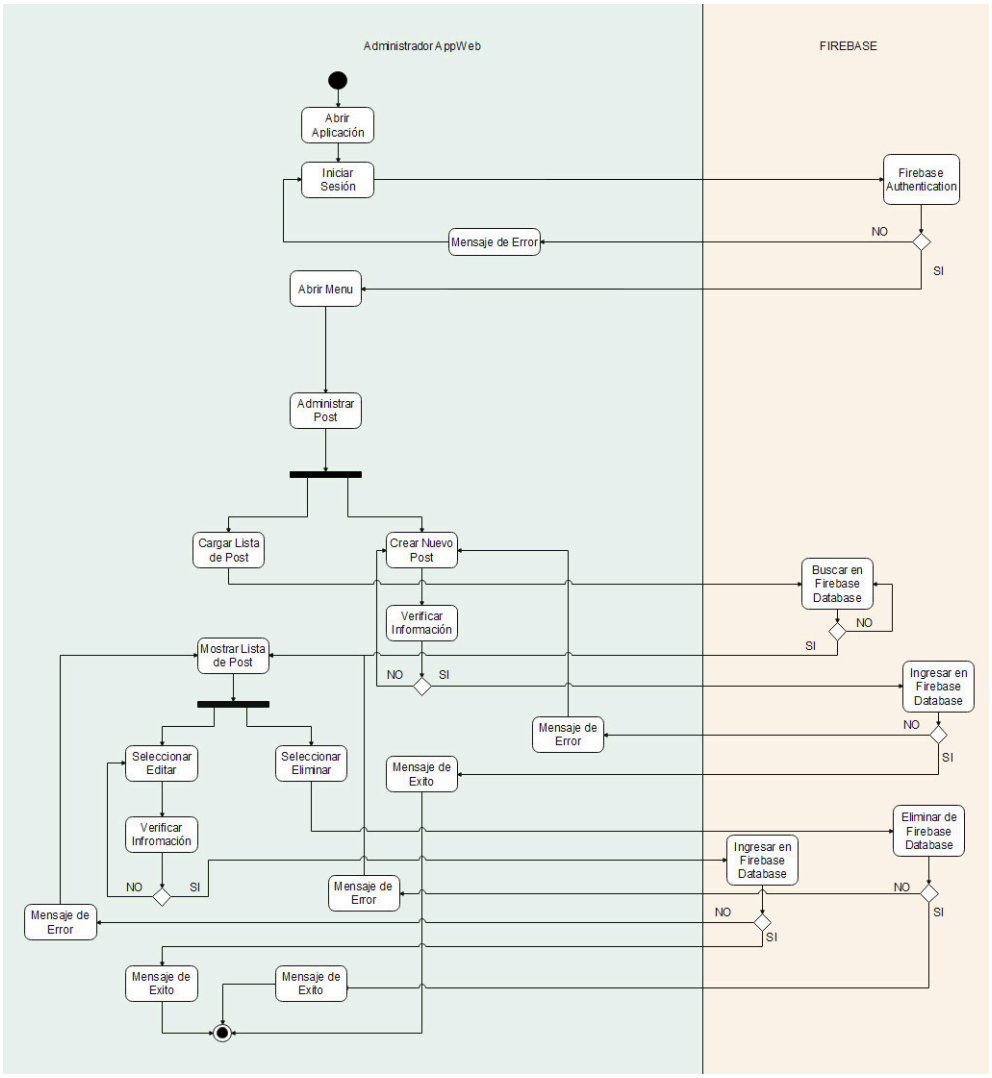


Figura 2.17: Diagrama de Actividades. CRUD Post.

Dentro de la gestión de Carreras el administrador puede crear, modificar, eliminar y monitorizar carreras. Como se indica en la Figura 2.18 para las tres primeras opciones él podrá acceder a ellas mediante dos ítems del menú.

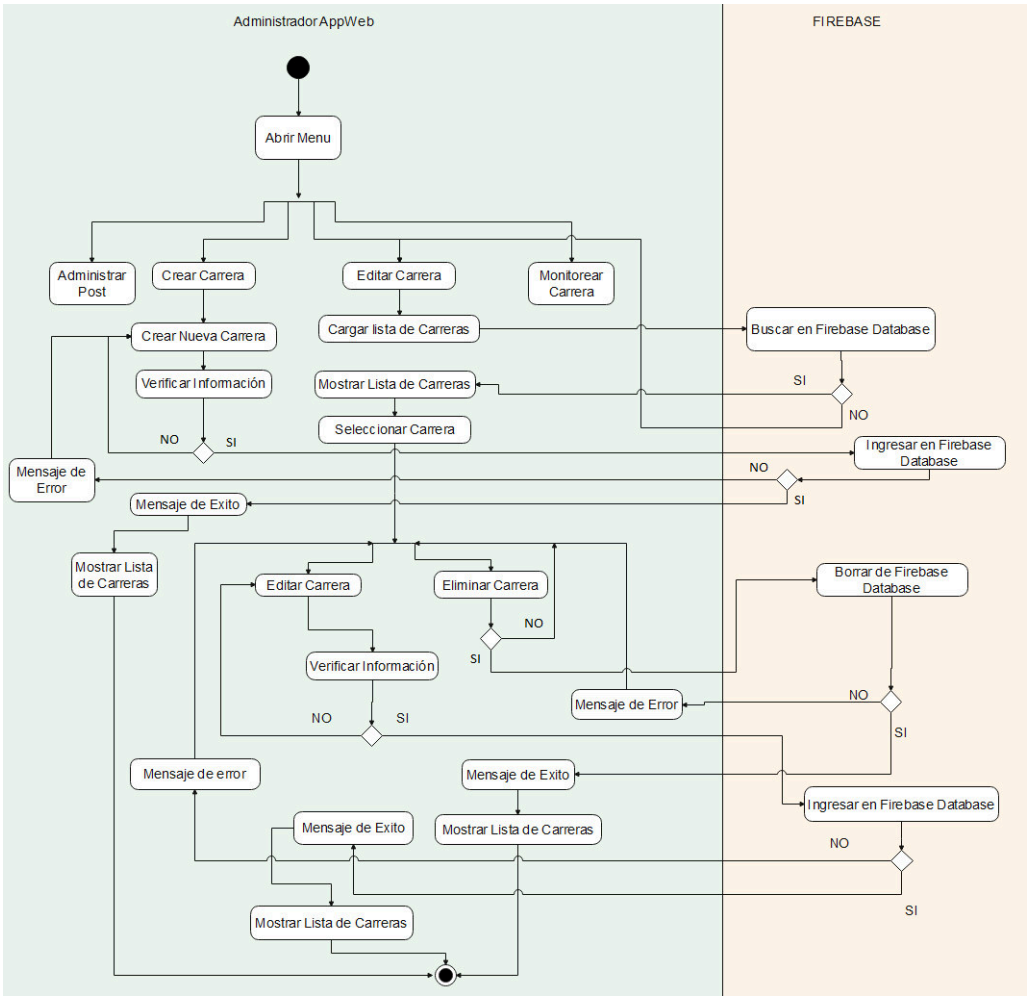


Figura 2.18: Diagrama de Actividades. CRUD Carreras.

Primero, para la creación de una nueva carrera el administrador dispondrá de un formulario donde deberá ingresar la información, la cual será verificada antes de enviarse a la base de datos de Firebase y una vez que esta haya ingresado se mostrará un mensaje de éxito al administrador y se visualizará esta nueva carrera dentro de la lista de próximas carreras.

Para editar y eliminar carreras tenemos el ítem Editar Carrera, donde se mostrará la lista de carreras existentes que aún no han sido monitorizadas. El administrador tendrá la opción de seleccionar alguna de ellas y una vez seleccionada este podrá eliminar la carrera o editar la información. Para editar la carrera el administrador contará con un formulario donde editará la información y selecciona guardar, una vez verificada esta información se la ingresará en la base de datos de Firebase, para eliminar la carrera se retira la información de la base de datos. Si la carrera se ha editado o eliminado correctamente de la base de datos de Firebase se mostrará un mensaje de éxito y se visualizará estos cambios dentro de la lista de próximas carreras, en caso contrario se mostrará un mensaje de error y se permanecerá

en la página.

Finalmente, como se indica en la Figura 2.19 para el Monitoreo de Carreras se cargará una lista de carreras no realizadas de las cuales el administrador podrá seleccionar alguna, una vez seleccionada se cargará el mapa y una lista de usuarios que han ingresado en la carrera para ser monitorizados desde la aplicación móvil. Para iniciar una carrera el administrador selecciona iniciar y se actualiza el estado de la carrera en la base de datos, permitiendo que los usuarios guarden la información de monitoreo en la base de datos y para finalizar la carrera el administrador selecciona finalizar y actualiza nuevamente el estado de la carrera, evitando que los usuarios sigan registrando datos en la carrera.

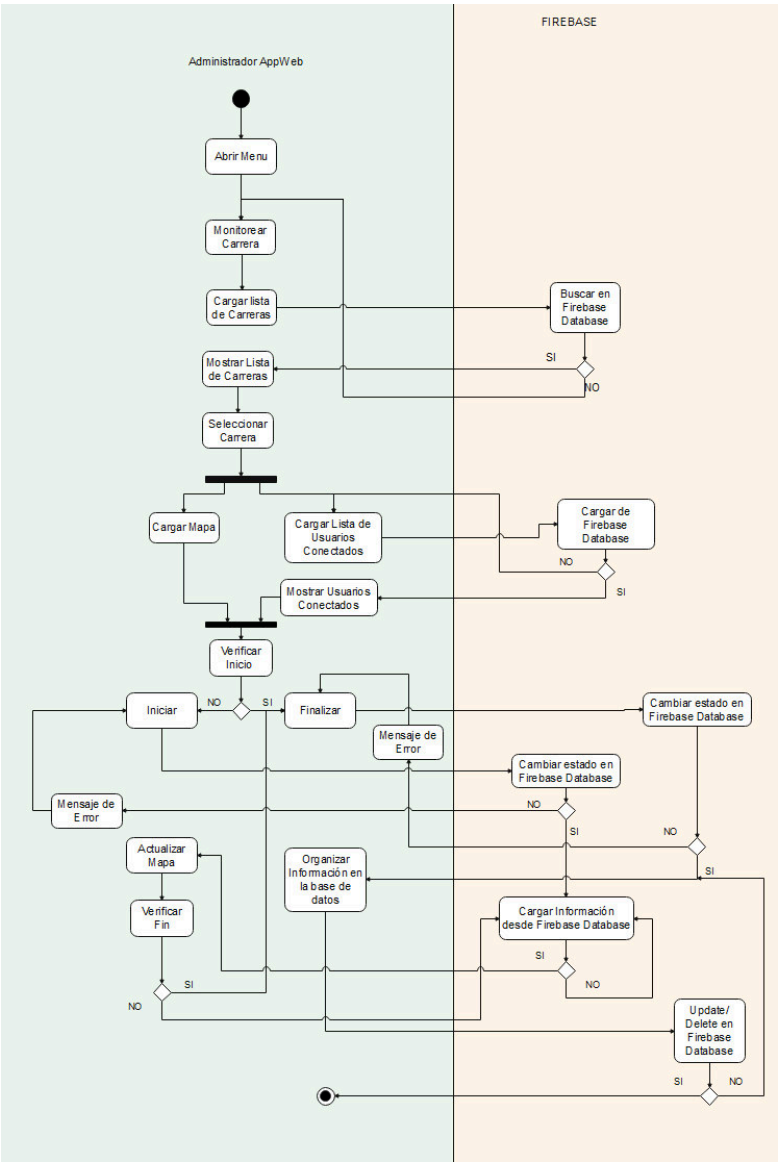


Figura 2.19: Diagrama de Actividades. Monitoreo de Carreras.

Dentro de los casos de uso se establecieron tres actores que son administrador, usuario no

registrado y usuario registrado. En la siguiente sección se realizarán los diagramas de actividades para los usuarios no registrados tanto en la aplicación móvil como en la aplicación web.

2.2.3.2. USUARIO NO REGISTRADO

Como se observa en la Figura 2.20 un usuario puede registrarse mediante la aplicación móvil ingresando su información en Firebase Authentication. Un usuario registrado inicia sesión y se verifican los datos en Firebase Database, si no existen datos se direccionará al interfaz de información de usuario donde tras ingresar la información se verificará y se ingresará en la base de datos. Si la información se ha ingresado correctamente se mostrará un mensaje de éxito y se direccionará a la interfaz de inicio de la aplicación móvil, en caso de que la información no pudiera ingresarse en la base de datos se mostrará un mensaje de error y se permanecerá en la interfaz de información de usuario.

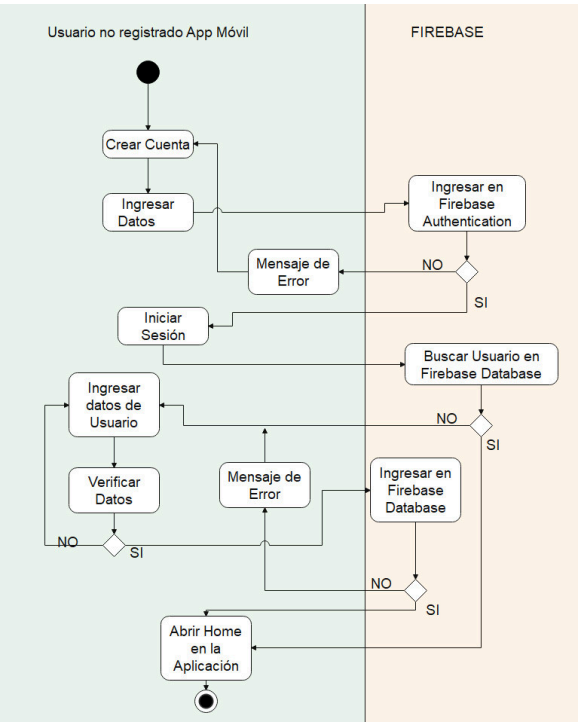


Figura 2.20: Diagrama de Actividades. Proceso de Registro

Un usuario no registrado en el prototipo de plataforma no puede registrase desde la aplicación web, pero tendrá la opción de visualizar las carreras creadas por el administrador como Próximas Carreras donde se visualizará una lista de carreras de las cuales podrá seleccionar alguna para acceder a información más completa de la carrera. Además, podrá ver los resultados de las Carreras Realizadas mediante una lista de carreras realizadas de las cua-

les podrá escoger alguna y visualizar la información de la carrera y sus resultados, adicional a esto el usuario podrá filtrar la información de los resultados de la carrera seleccionada. El diagrama se muestra en la Figura 2.21

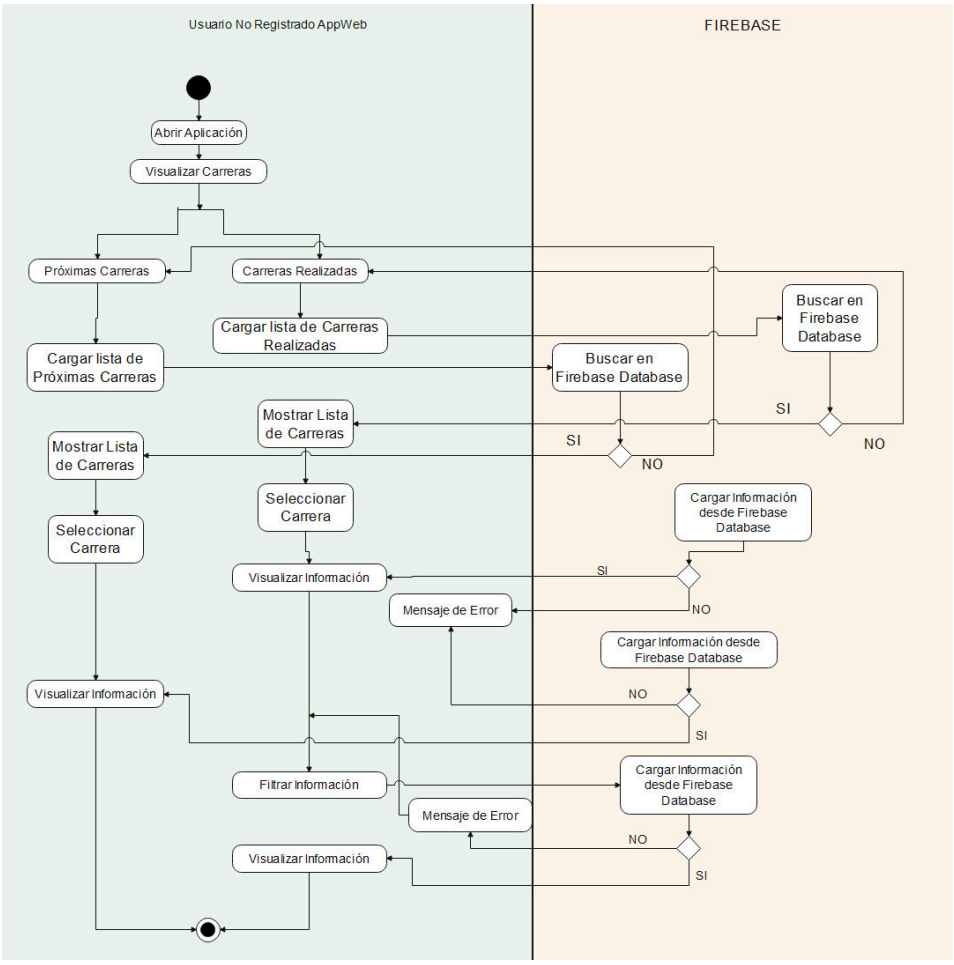


Figura 2.21: Diagrama de Actividad. Usuario No Registrado

2.2.3.3. USUARIO REGISTRADO

Aplicación Móvil

Un usuario que ya se ha registrado y autenticado en la aplicación móvil puede visualizar y editar su información de usuario dirigiéndose a su perfil y seleccionando editar. Estando en el perfil de usuario se verificará la existencia de la información en la base de datos local, si esta no existiera se la importará desde Firebase Database y se ingresará en la base de datos local. Para editar la información, una vez que se ha seleccionado esta opción se abrirá la interfaz donde se cargará la información del usuario y esta será editable, a continuación se verificará la información y se la enviara a Firebase Database, en caso de que la información se ingrese correctamente en la base de datos de Firebase se enviará un mensaje de éxito y

también se registrará esta información en la base de datos local del dispositivo, finalmente si no se completará el ingreso de información en Firebase Database se enviará un mensaje de error al usuario. Lo descrito anteriormente se lo puede observar en la Figura 2.22

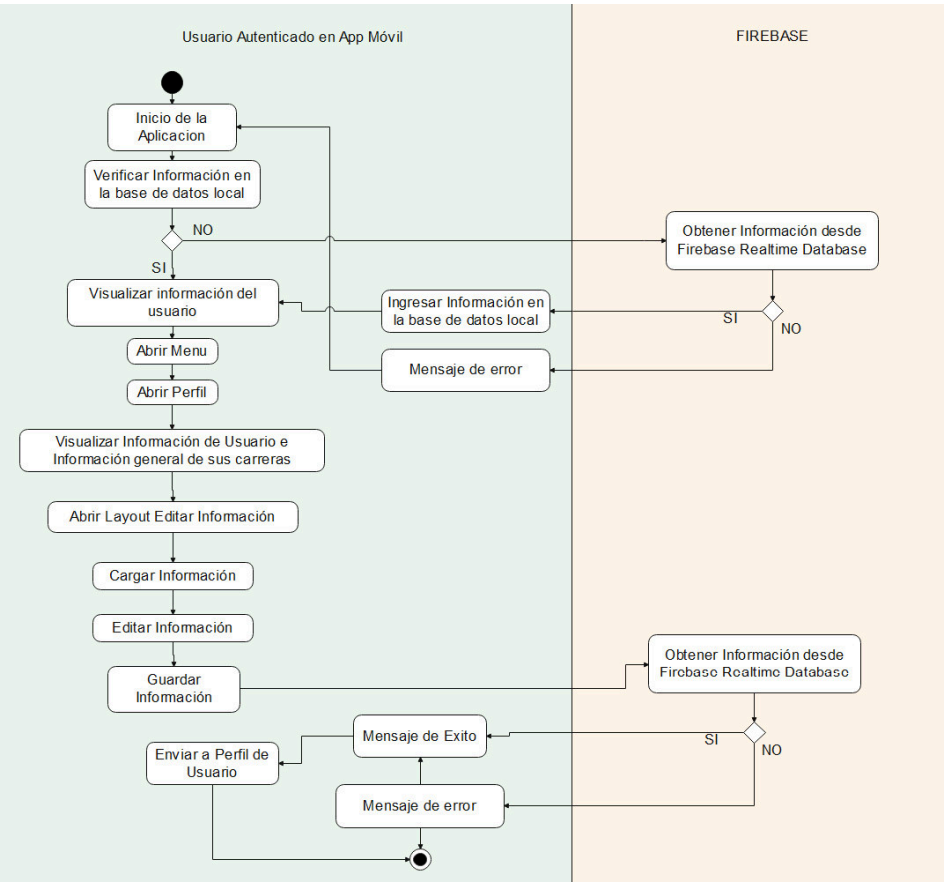


Figura 2.22: Diagrama de Actividades. Visualizar y Editar Información de Usuario

Dentro de la aplicación móvil el objetivo principal es el monitoreo de parámetros de la actividad física del usuario en cada carrera que participe para ello se ha dividido este proceso en dos diagramas de actividades, el primero consiste en el registro de inscripción del usuario en la carrera e ingreso en la interfaz de monitoreo y el segundo consiste en el proceso de monitoreo en sí.

Como se indica en la Figura 2.23 el usuario puede acceder a una lista de carreras mediante la opción buscar carreras del menú, esta lista estará compuesta por todas las carreras que el administrador ha creado y aún no han sido realizadas. El usuario selecciona alguna de ellas y podrá visualizar la información de la carrera. Además, se verificará si el usuario consta con una inscripción previa en esta carrera. Si el usuario aún no se ha inscrito la interfaz se lo permitirá enviando la información de inscripción a la base de datos de Firebase, si la información no se ingresa correctamente el usuario observará un mensaje de error y perma-

neciera en la misma interfaz, en caso de que se haya realizado la inscripción correctamente el usuario observará un mensaje de éxito y la información de la inscripción y la carrera se almacenarán en la base de datos local del dispositivo. Si el usuario ya registra una inscripción previa o si ha cumplido el proceso de inscripción la interfaz le permitirá anular esta inscripción o ingresar al interfaz de monitoreo. Para cancelar la inscripción se eliminar la información de registro en la base de datos de Firebase, si esto se ejecuta correctamente se mostrará un mensaje de éxito y se eliminará la información ingresada en la base de datos local del dispositivo al momento de la inscripción, caso contrario se mostrará un mensaje de error y se permanecerá en la interfaz. En caso de que el usuario no haya cancelado su inscripción y desee ingresar al interfaz de monitoreo deberá ingresar la clave de la carrera proporcionada por el administrador, si la clave es ingresa es correcta se ingresar en la interfaz de monitoreo, caso contrario se mostrará un mensaje de error y se permanecerá en la interfaz.

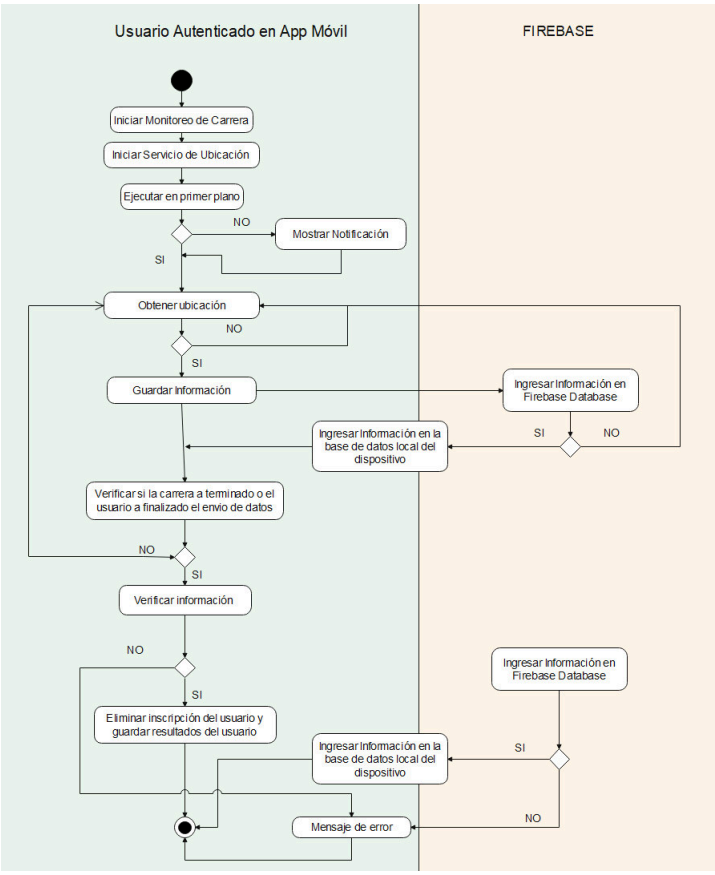


Figura 2.23: Diagrama de Actividad. Registrar Inscripción

Una vez dentro de la interfaz de monitoreo se verificará si el usuario ha otorgado permiso de ubicación a la aplicación, en caso de que no se tenga este permiso se lo solicitará y se mantendrá dentro de la interfaz de monitoreo.

En la interfaz de monitoreo se carga el mapa desde Google Maps y se tienen las opciones de iniciar y finalizar monitoreo, donde la única opción habilitada será la de iniciar. Una vez que el usuario inicia la transmisión de información se deshabilita la opción iniciar y se habilita la opción finalizar. Además, se iniciará el servicio encargado de obtener la ubicación periódicamente e interactuar con la interfaz de monitoreo, la cual almacenará la información obtenida en la base de datos de Firebase y del dispositivo móvil.

El servicio es el encargado de verificar si la aplicación se encuentra en primer plano o en background, es decir cuando se oculta la aplicación o se bloquea la pantalla del dispositivo, en este caso la aplicación mostrará una notificación persistente que indica al usuario que el monitoreo se continúa realizando. Cada vez que se obtenga la ubicación se verificará si la carrera ha finalizado, en caso de que la carrera aún no termine el usuario finalizará el monitoreo mediante la aplicación, en estos dos casos se finalizara la transmisión y se guardarán los resultados finales en la base de datos de Firebase y de la aplicación móvil tal como se indica en la Figura 2.24

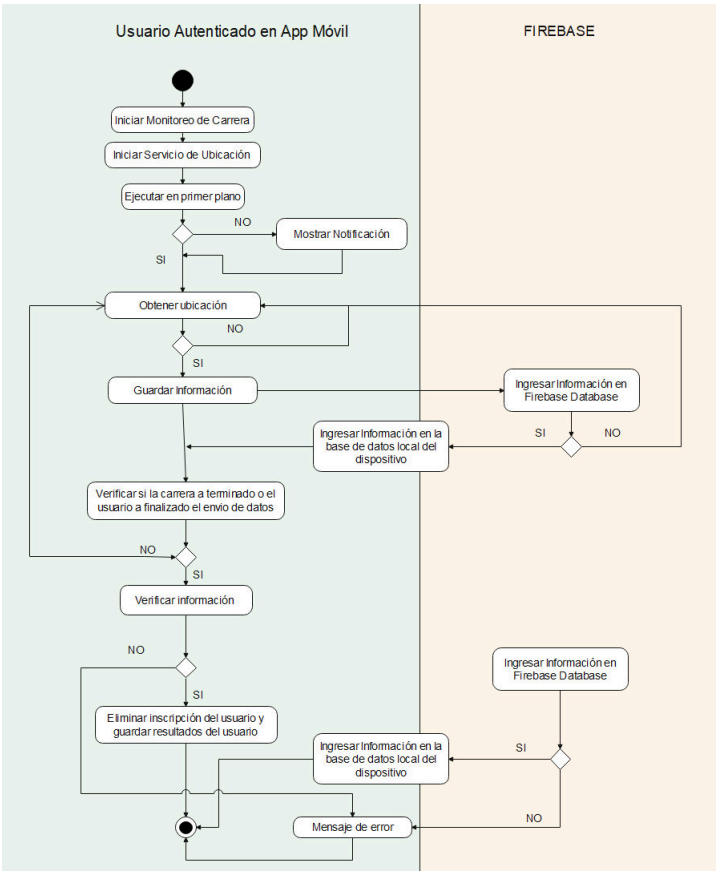


Figura 2.24: Diagrama de Actividad. Monitorear Datos

Un usuario puede visualizar las carreras en las que se ha inscrito y las carreras en las que

ha monitorizado la actividad física de su participación, en los dos casos se carga una lista de carreras de las cuales el usuario tendrá la opción de seleccionar alguna, si selecciona una de las carreras en las que se ha inscrito se observará la información de la carrera incluida su inscripción, si la selección de carrera se la hace dentro de la lista de carreras realizadas por el usuario adicional a la información de la carrera se observan los resultados de la participación del usuario y los resultados generales de la carreras. Cabe recalcar que si esta información no se encuentra en la base de datos local del dispositivo móvil se la obtendrá desde Firebase Database y se la almacenará en la base de datos local. El diagrama se muestra en la Figura 2.25

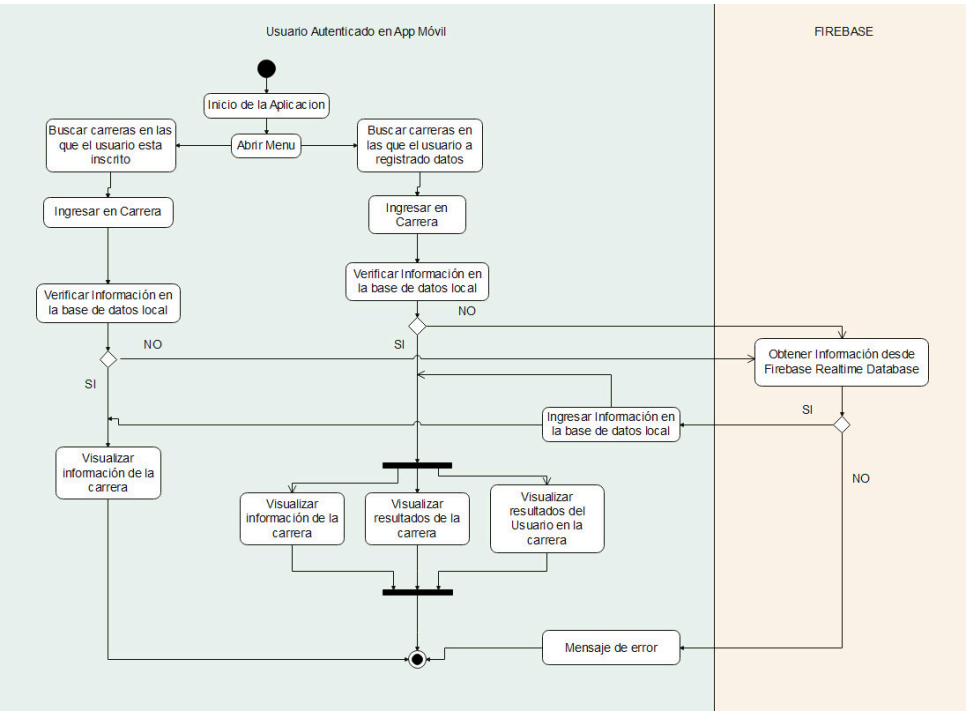


Figura 2.25: Diagrama de Actividad. Visualizar Carreras del Usuario

Aplicación Web

Dentro de la aplicación web un usuario registrado y autenticado solo podrá visualizar su información almacenada en Firebase Database. Como se indica en la Figura 2.26, el usuario podrá ingresar a su perfil donde se cargará su información de usuario y tendrá la opción de visualizar sus resultados promedios de todas las carreras en las que ha participado, las carreras en las que se ha inscrito y carreras en las que se ha monitorizado, para visualizar la información de estas dos últimas el proceso es el mismo que en la aplicación móvil, es decir se carga una lista de carreras para cada caso y el usuario selecciona alguna de ellas, en el caso de carreras inscritas solo visualizará la información de la carrera, en el caso de

sus carreras realizadas además de la información de la carrera visualizará sus resultados, los resultados globales de la carrera y podrá hacer consultas sobre los resultados de la carrera. Finalmente, el usuario podrá visualizar su desempeño promedio en base a todas las carreras en las que ha participado, similar a la información mostrada en el perfil de usuario de la aplicación móvil.

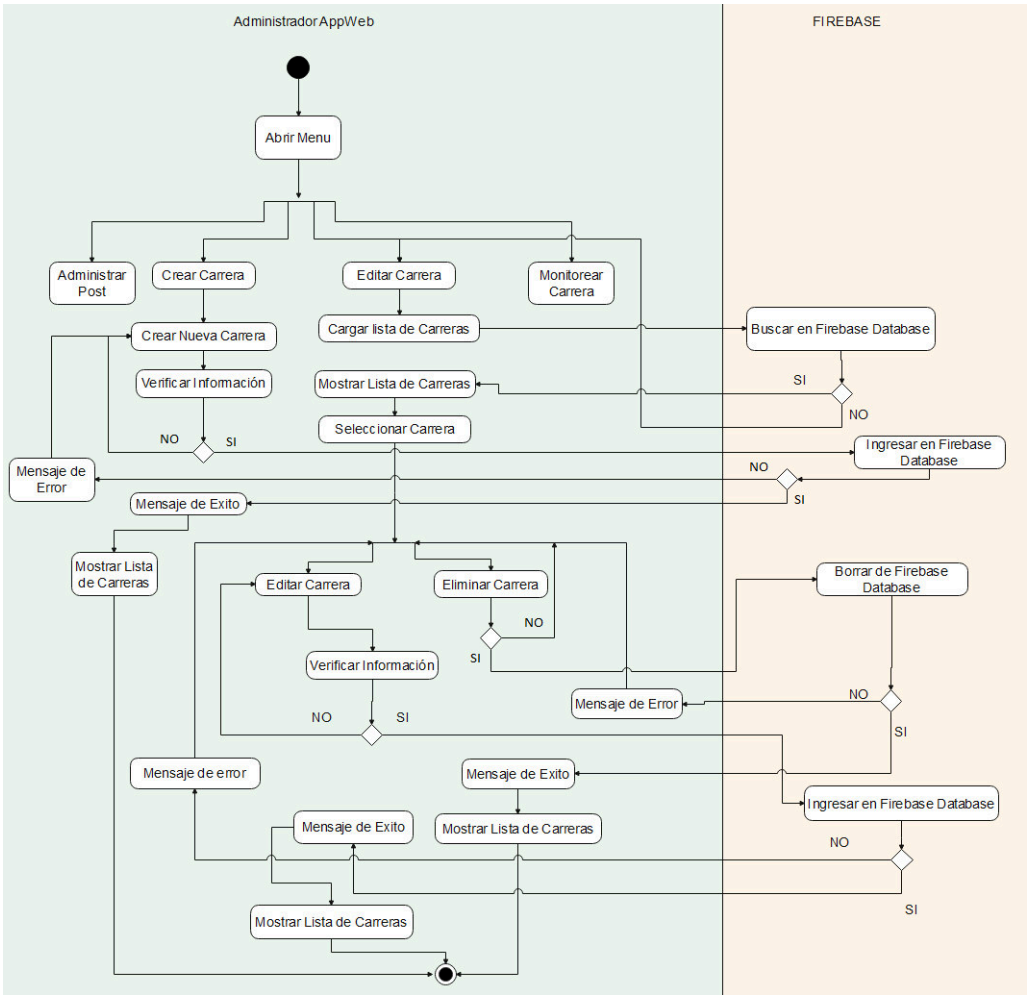


Figura 2.26: Diagrama de Actividad. Usuario Registrado

2.2.4. DIAGRAMA DE CLASES

Mediante el diagrama de clases se visualizará las relaciones entre las clases que conforman el prototipo de plataforma. En este diagrama se definieron seis clases.

La clase Carrera representa todas las carreras atléticas dentro del prototipo de plataforma.

La clase Usuario representa a todos los atletas que se registren en el prototipo de plataforma.

La clase Post o Publicación representa las publicaciones compartidas por el administrador.

La clase Resultado-Usuario representa los resultados finales del usuario al participar en una Carrera.

La clase Resultado-Carrera, esta clase representa los resultados promedio del desempeño de todos los participantes de una carrera.

La clase Punto representa la información que se almacena en la base de datos al momento del monitoreo de la actividad física de un usuario en tiempo real.

Como se observa en la Figura 2.27 se ha representado una relación doble entre la clase Carrera y la clase Usuario, esto se debe a que existen dos tipos de relaciones entre estas clases. La primera relación se representa por la cardinalidad muchos a muchos y consiste en los usuarios inscritos en las carreras. La segunda relación de igual manera se representa por la cardinalidad de muchos a muchos y consiste en los usuarios que han sido monitorizados en las carreras, esta relación es la que genera el resto del diagrama de clases, es decir cada usuario monitorizado en cualquier carrera posee una colección de puntos que son generados mientras el atleta se está monitorizando, cada usuario monitorizado posee solo un resultado final correspondiente a cada carrera en la que fue monitorizado y una carrera posee solo un resultado final al terminar de monitorear a todos sus participantes.

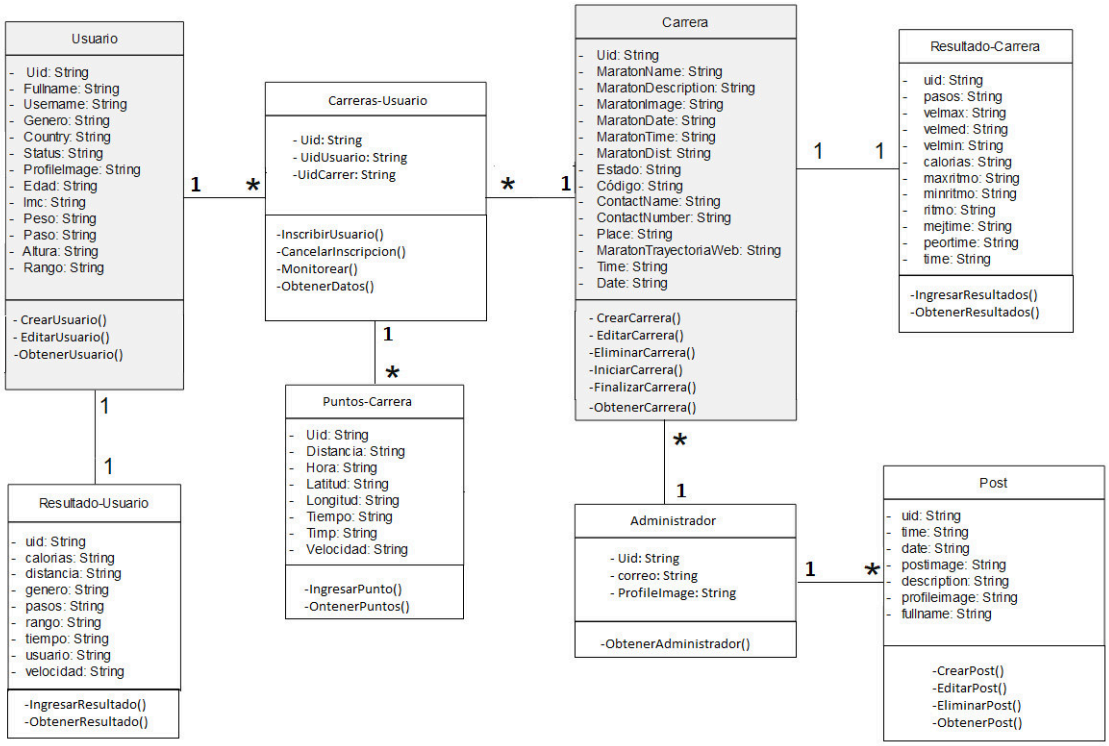


Figura 2.27: Diagrama de Clases

2.2.4.1. ESTRUCTURA DE BASE DE DATOS

Para modelar la base de datos se realizaron diagramas de árbol para los objetos necesarios en el funcionamiento del prototipo de plataforma.

Es importante conocer las relaciones bidireccionales existentes en la base de datos ya que estas generarán redundancia en la información. Dentro de la base de datos se tienen dos relaciones bidireccionales entre Carreras y Usuarios una al realizar la inscripción del usuario en la carrera y otra al registrar los resultados del usuario en la carrera ya que es necesario que esta información sea almacenada tanto en el lado del usuario como en el lado de la Carrera como se indican en las Figuras 2.28

Usuarios

El diagrama mostrado en la Figura 2.28 indica la estructura de datos que tendrá un usuario en de la base de datos. Cada usuario tendrá tres nodos: información, inscripción y resultados. Nodo Información, en este nodo se cargará la información personal del usuario en forma de atributos donde el nombre del atributo será la clave en la base de datos y el contenido del atributo será el valor en la base de datos.

Nodo Inscripción, este nodo forma parte de una relación bidireccional entre Carreras y Usuarios, es decir este nodo tendrá información de cada carrera en la que el usuario se ha inscrito y en la carrera se guardará la información del usuario inscrito.

Nodo Resultados, este nodo también tiene una relación bidireccional al igual que el nodo anterior ya que la información de resultados del usuario en una carrera se registra tanto para el usuario como para la carrera.

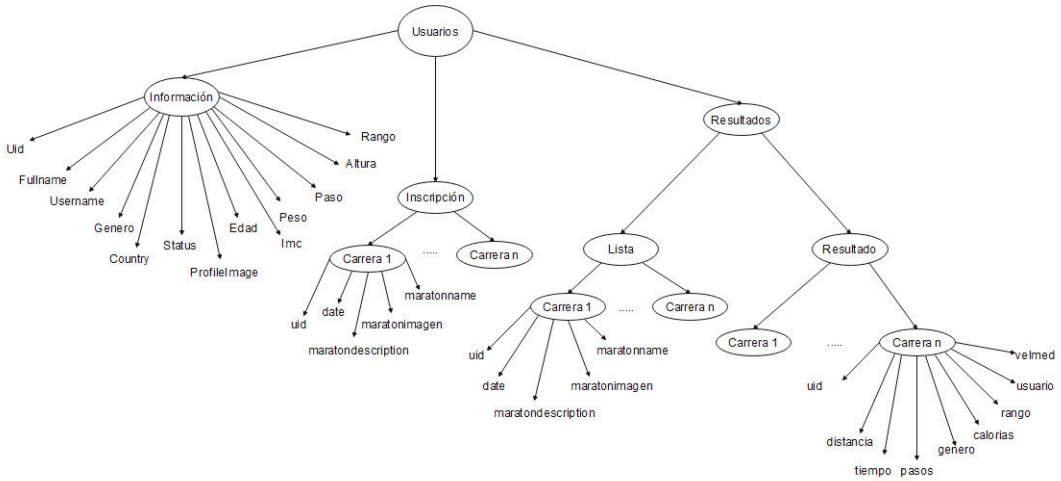


Figura 2.28: Estructura de Árbol para Usuarios

Carreras

Las carreras almacenadas en la base de datos tienen una estructura formada por seis nodos los cuales permitirán al administrador y a los usuarios realizar el monitoreo de parámetros de la actividad física de cada usuario participante en una carrera.

Nodo Nuevas, este nodo contiene todas las carreras que ha creado el administrador, cada sub-nodo correspondiente a una carrera, además de su información principal contiene el atributo estado el cual permitirá clasificar a las carreras por realizados y no realizadas.

Nodo Inscripción, este nodo es parte de la relación bidireccional entre carreras y usuario en base a la relación inscripción del diagrama de clases, ya que almacena la información de los usuarios inscritos en cada carrera, esta información permitirá al administrador visualizar a todos los usuarios inscritos por carrera.

Nodo Resultados, este nodo es parte de la relación bidireccional entre carreras y usuario en la base a la relación de usuarios monitorizados, ya que almacena los resultados finales de la participación de cada usuario en una carrera.

Estos tres nodos se los pueden observar en la Figura 2.29

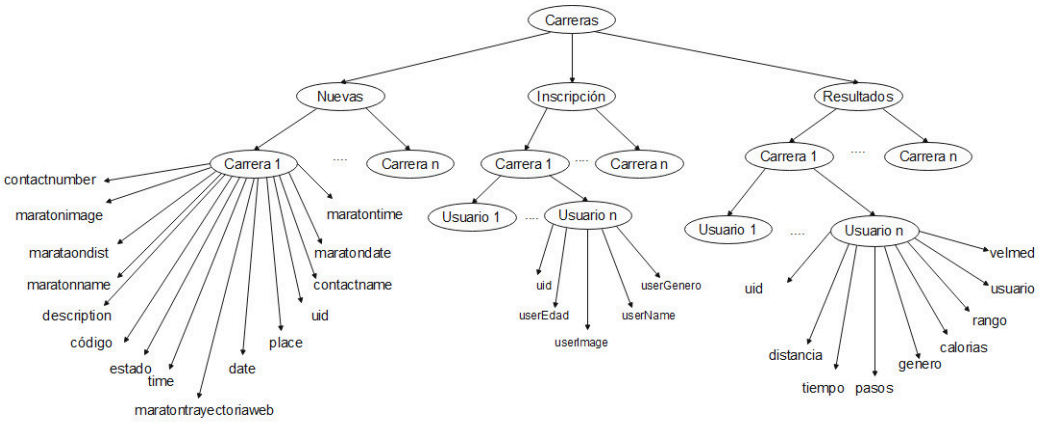


Figura 2.29: Estructura de Árbol para Carreras I

Nodo Inicio, este nodo contiene la información de todos los usuarios que ya han ingresado a la interfaz de monitoreo de la carrera desde la aplicación móvil, permitiendo al administrador visualizar una lista de usuarios listos para ser monitorizados.

Nodo Monitoreo, este nodo tendrá la información de la ubicación en tiempo real de los usuarios. Esta ubicación es la que se mostrará en el mapa del administrador.

Nodo Datos, este nodo almacenará una lista de parámetros de la actividad física de cada

usuario obtenida mediante la aplicación móvil, la cual se mostrará como resultados en la aplicación móvil y web para los usuarios.

Estos tres nodos se los pueden observar en la Figura 2.30

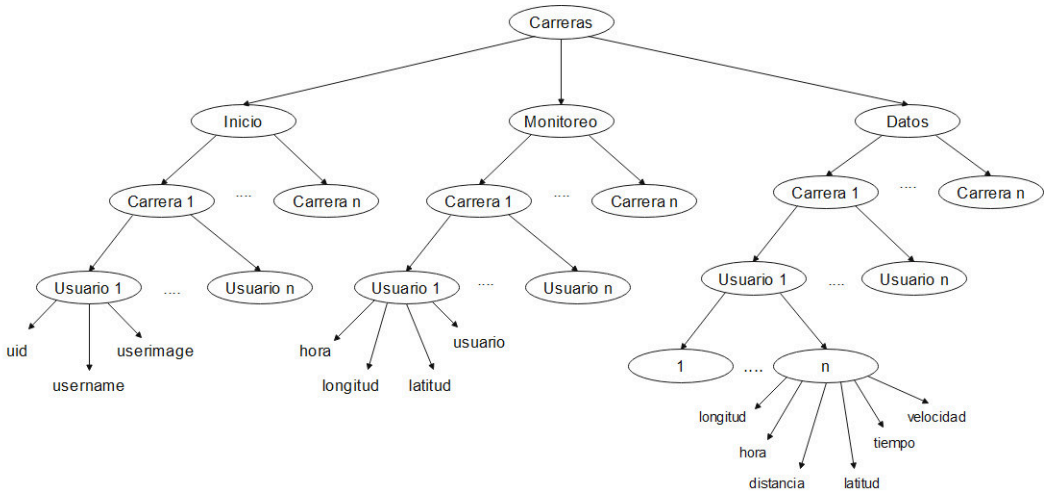


Figura 2.30: Estructura de Árbol para Carreras II

Como se indica en la figura 2.31 en la base de datos también existe el nodo Post el cual almacenara la información de los post o publicaciones que se le mostraran al usuario en la página principal de la aplicación móvil y aplicación web.

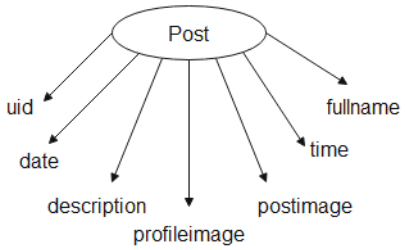


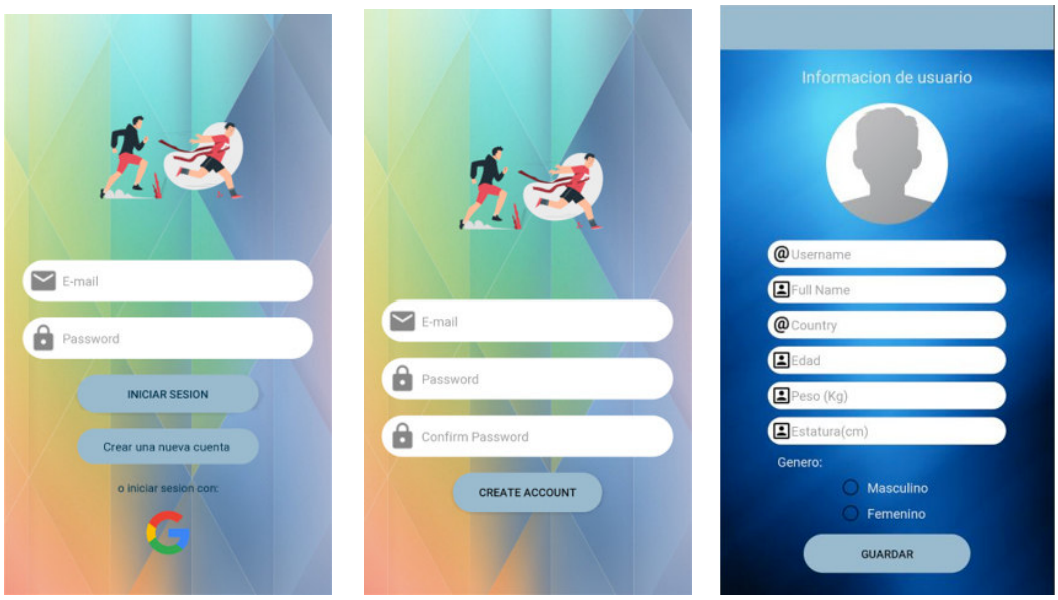
Figura 2.31: Estructura de árbol para Posts

2.2.5. MOCKUPS

A continuación, se muestran las diferentes Muckup de las interfaces que tendrá la aplicación móvil y se describirá de forma breve la interacción de los usuarios con cada una de ellas.

En la Figura 2.32 (a) se muestra la interfaz de Inicio de sesión de la aplicación donde el usuario podrá autenticarse o registrarse mediante Google, si el usuario no desea registrarse con su cuenta de Google podrá crear una cuenta mediante la interfaz mostrada en la Figura 2.32 (b).

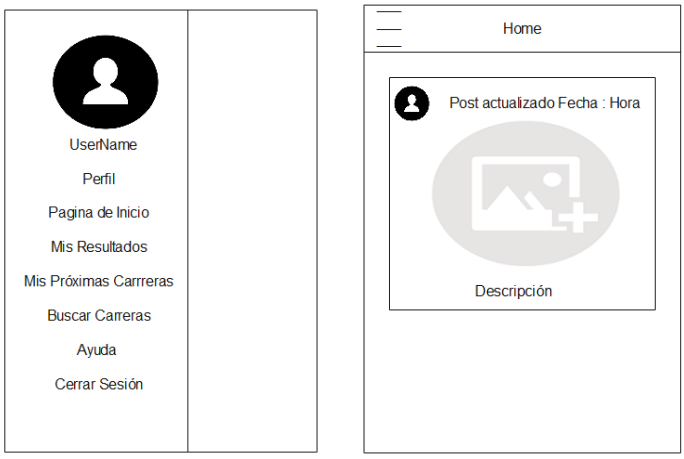
Una vez realizada la autenticación, al usuario se le presentará la pantalla de registro de información mostrada en la Figura 2.32 (c). Esta pantalla consta de un formulario donde deberá ingresar sus datos personales y se culminará con el registro del usuario.



(a) Autenticación (b) Registro de usuario (c) Registro de información de usuario

Figura 2.32: Muckups del módulo autenticación de App Móvil

La aplicación mostrará al usuario la interfaz de Inicio donde se visualizarán los post o publicaciones compartidas por el administrador. Además, en esta interfaz el usuario tendrá acceso al menú el cual le permitirá navegar a través de la aplicación. Estas dos interfaces se muestran en la Figura 2.33 (a) y Figura 2.33 (b)



(a) Menú (b) Lista de posts

Figura 2.33: Muckup de la pantalla de inicio

En la Figura 2.34 se muestra la interfaz en la que el usuario visualizará la lista de carreras no realizadas existentes en el prototipo de plataforma, de las cuales podrá seleccionar alguna.

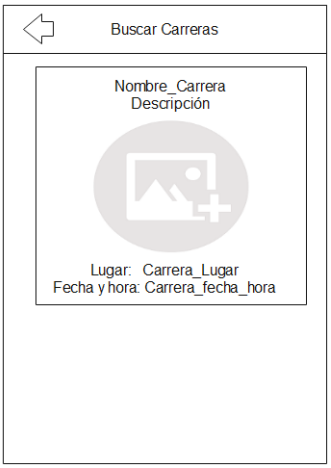


Figura 2.34: Mockup de la pantalla buscar carreras

A continuación, en la Figura 2.35 se muestra la interfaz de Carrera donde se visualizará información de la carrera. Además, el usuario podrá registrar su inscripción o cancelarla. Finalmente, esta interfaz le permite ingresar a la interfaz de Monitoreo mediante el ingreso del código proporcionado por el administrador.

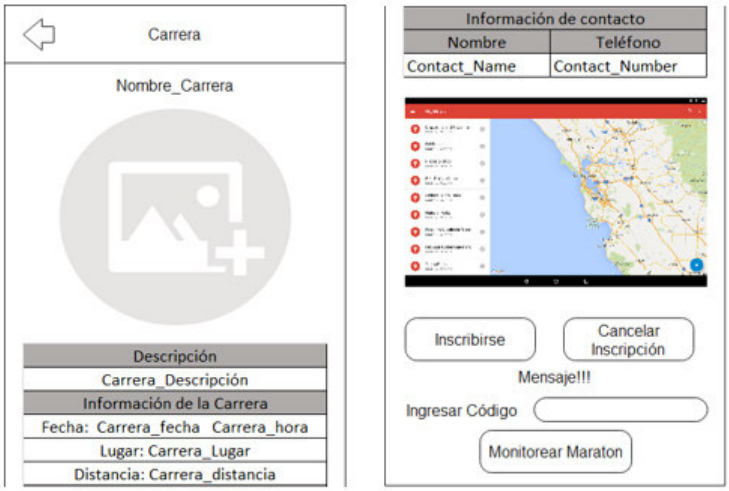
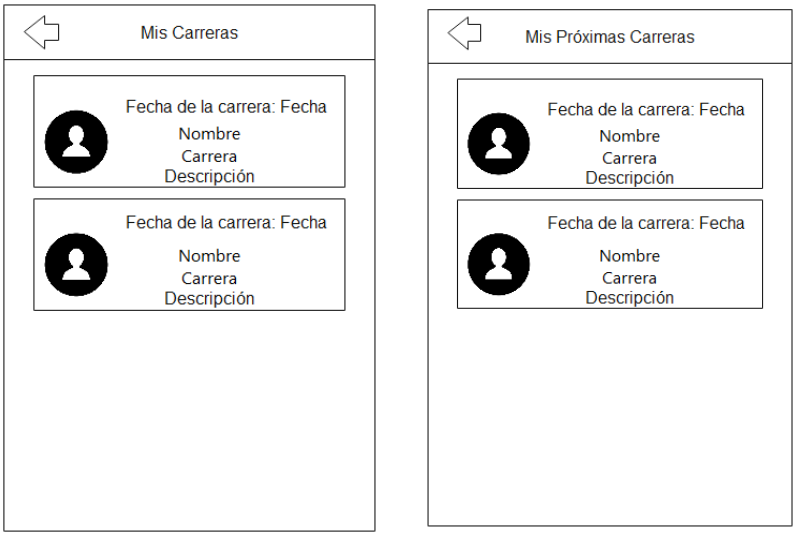


Figura 2.35: Muckup de la pantalla información de carrera

En la Figura 2.36 se muestra la interfaz mediante la cual el usuario monitorizará su actividad física en una carrera. El usuario dará inicio al monitoreo enviando en tiempo real su información a la base de datos de Firebase y la base de datos local, si el usuario desea finalizar el monitoreo lo hará mediante el botón finalizar y guardará la información final de la carrera a las bases de datos.

Figura 2.36: Mockup de la pantalla de monitoreo

El usuario tendrá acceso a la interfaz Mis Carreras y Mis Próximas Carreras a través del menú. Como se indica en la Figura 2.37 (a) y en la Figura 2.37 (b) se mostrará al usuario una lista de las carreras en las que ha participado y una lista de las carreras en las que se ha inscrito.



(a) Mis Carreras (b) Mis próximas carreras

Figura 2.37: Muckups de las pantallas de carreras

Si el usuario selecciona alguna de las carreras de la lista de la interfaz Mis Próximas Carreras se mostrará la interfaz de la Figura 2.35, en caso de que el usuario seleccione una carrera de la interfaz Mis Carreras se mostrará la interfaz Resultados de la Carrera mostrada en la Figura 2.38, esta interfaz mostrará los resultados del desempeño del usuario en la carrera y los resultados globales de la carrera.

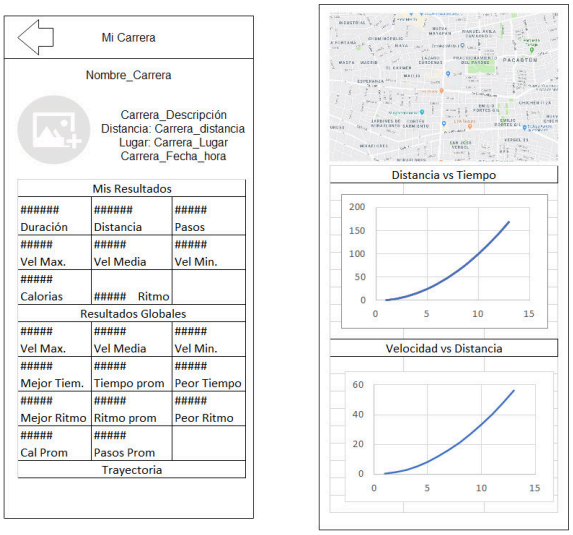


Figura 2.38: Muckup de la pantalla resultados de mi carrera

En la Figura 2.39 se observa la Interfaz Mi Perfil a la cual se ingresa mediante el menú. Esta interfaz muestra la información del usuario, así como la información promedio del desempeño del usuario a lo largo de todas las carreras en las que ha participado, además se mostrarán gráficas de los parámetros monitorizados en función de cada carrera.

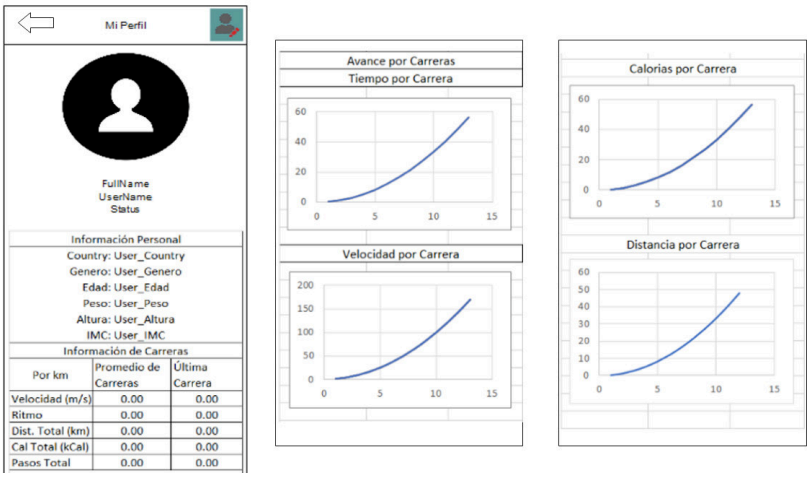
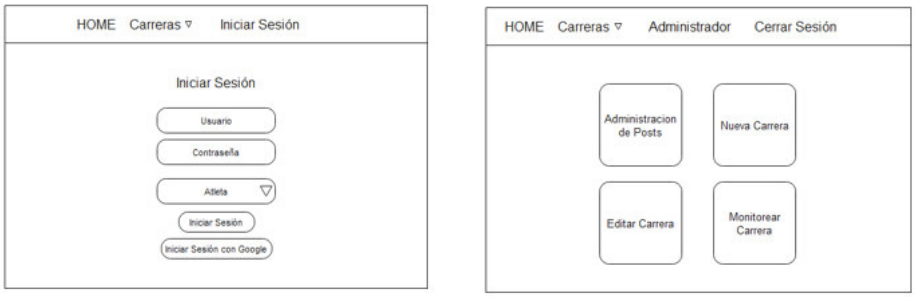


Figura 2.39: Muckup de la pantalla mi perfil

A continuación, se muestran las diferentes Muckup de las interfaces que tendrá la aplicación web y se describirá de forma breve la interacción de los usuarios con cada una de ellas. En la Figura 2.40 (a) se muestra la interfaz para inicio de sesión donde se autentican los usuarios y el administrador, además el usuario tiene la opción de autenticarse con Google mediante el servicio de Firebase Authentication. Como se indica en la Figura 2.40 (b) una vez que el administrador se ha autenticado tiene acceso al menú que le permitirá gestionar la información mostrada a los usuarios del prototipo de plataforma.



(a) Autenticación (b) Menú de Administración

Figura 2.40: Muckups del módulo de administración

A continuación, describiremos las interfaces a las que podrá acceder el administrador para gestionar las publicaciones y las carreras atléticas. Para la administración de posts o publicaciones se mostrará al administrador el formulario mostrado en la Figura 2.41 además se cargará una lista de los posts existentes de los cuales se puede seleccionar alguna publicación para editarla o para eliminarla.

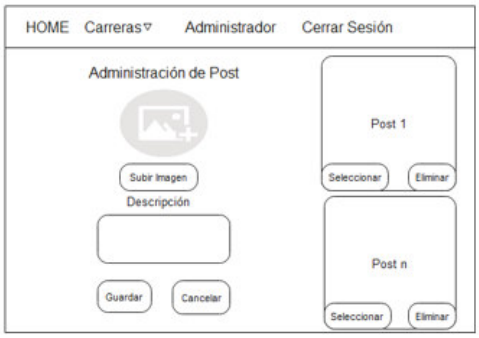


Figura 2.41: Muckup de la pantalla administración de posts

Para tener una gestión completa de las carreras atléticas el administrador puede crear, editar, eliminar y monitorear las carreras.

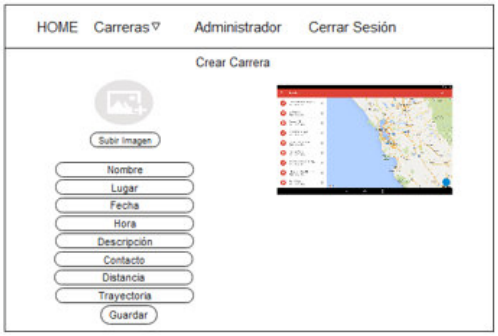


Figura 2.42: Muckup de la pantalla crear carrera

En la gestión de carreras atléticas se inicia con creación de estas, para lo cual el administra-

dor tendrá acceso a la interfaz de la Figura 2.42 donde ingresará la información solicitada. Además, se proporcionarán instrucciones que permitirán al administrador crear y cargar un proyecto desde My Maps en la trayectoria de la carrera.

Para editar y eliminar carreras el administrador tendrá acceso a una lista de carreras aun no realizadas, de las cuales al seleccionar alguna se ingresará a la interfaz mostrada en la Figura 2.43 donde podrá actualizar la información de la carrera o eliminarla. Además, se mostrará una lista de los usuarios que se han inscrito en ella.

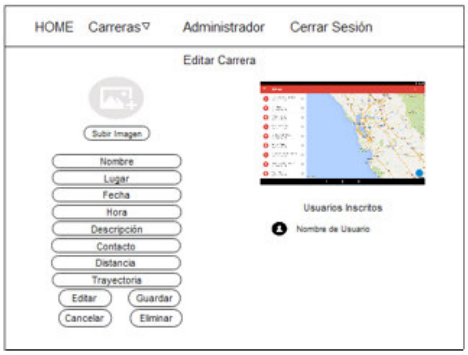


Figura 2.43: Muckup de la pantalla editar carrera

Para monitorear carreras de igual manera el administrador deberá seleccionar alguna de la lista de carreras aun no realizadas e ingresará a la interfaz mostrada en la Figura 2.44.

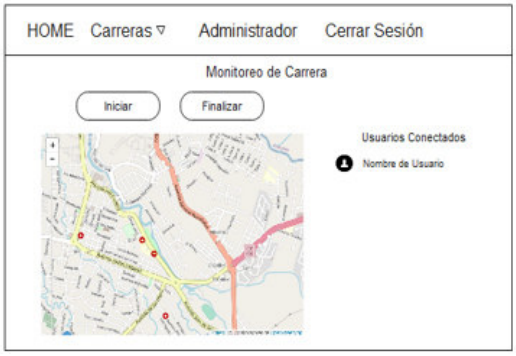


Figura 2.44: Muckup de la pantalla monitoreo de carrera

Dentro de esta interfaz el administrador podrá dar inicio y fin a la carrera. Además, se mostrará una lista de usuarios que se han conectado a la carrera para ser monitorizados y un mapa donde se mostrará la ubicación de los usuarios monitorizados en tiempo real una vez que haya iniciado la carrera.

A continuación, se muestran las diferentes interfaces de la aplicación web a la que tendrán acceso los usuarios autenticados.

Como se indica en la Figura 2.45 una vez que el usuario se ha autenticado podrá acceder a

su perfil donde se mostrará su información de usuario y tendrá la opción de seleccionar Mis Próximas Carreras, Mis Carreras Realizadas y Mis Resultados.

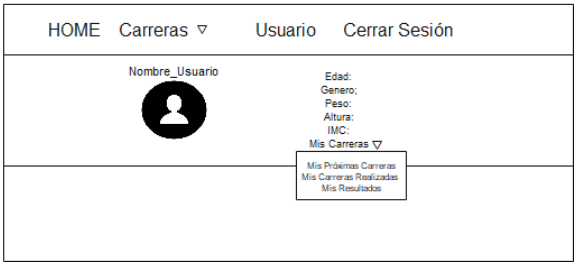


Figura 2.45: Muckup Perfil de Usuario

Dentro del Perfil de Usuario en la Sección de Información Carreras de Usuario se visualizará la información correspondiente al ítem seleccionado dentro del menú de despliegue Mis Carreras. A continuación, se describirán estos ítems. Cuando el usuario selecciona Mis Próximas Carreras se mostrará la lista de carreras en las que el usuario se ha inscrito y una vez que seleccione alguna se mostrará la información de la carrera similar a la Figura 2.48 Para el ítem Mis Carreras Realizadas se mostrará una lista de carreras en las que el usuario a participado, una vez que seleccione alguna de estas carreras se mostrará la información de esta incluyendo los resultados del usuario y los resultados promedio de todos los participantes de la carrera, además podrá filtrar la información de los resultados de la carrera. Esta interfaz se muestra en la Figura 2.46

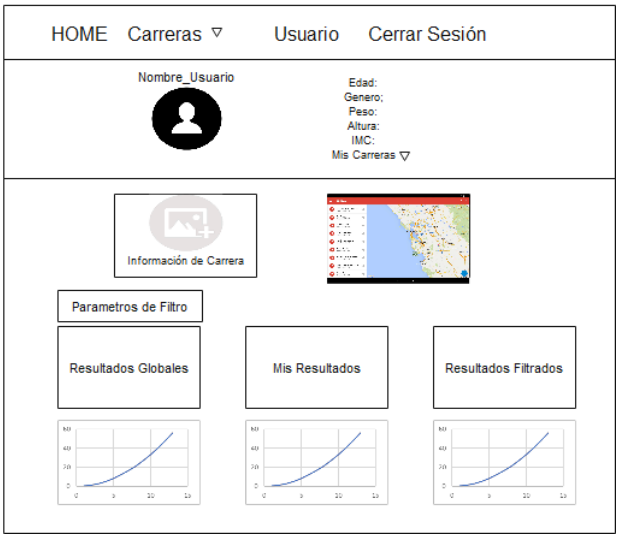


Figura 2.46: Muckup de la pantalla resultados por carrera de usuario

El último ítem es Mis Resultados donde el usuario visualizará los resultados promedio de su

desempeño a lo largo de todas las carreras en las que ha participado. Además, visualizará la información de la última carrera en la que participó tal como se indica en la Figura 2.47

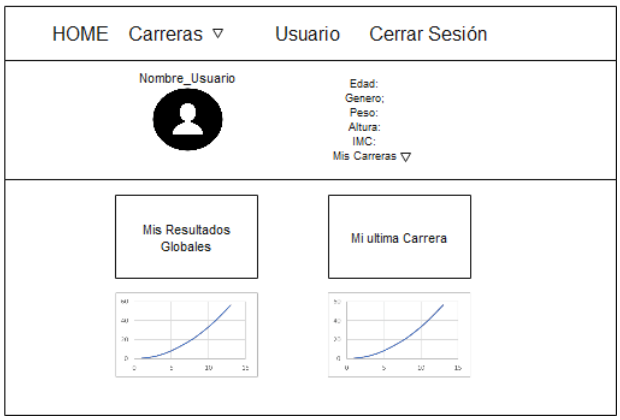


Figura 2.47: Muckup Resultados Generales de Usuario

2.3. IMPLEMENTACIÓN

En esta fase se utilizarán IDEs de programación Android Studio y Visual Studio Code en conjunto de los servicios Authentication, Realtime Database, Storage y Cloud messaging de firebase para el desarrollo de la aplicación móvil y web. Además, se usará la tecnología React JS para la implementación de la aplicación web y las tecnologías de git y github para el versionamiento y respaldo del código fuente del prototipo de plataforma.

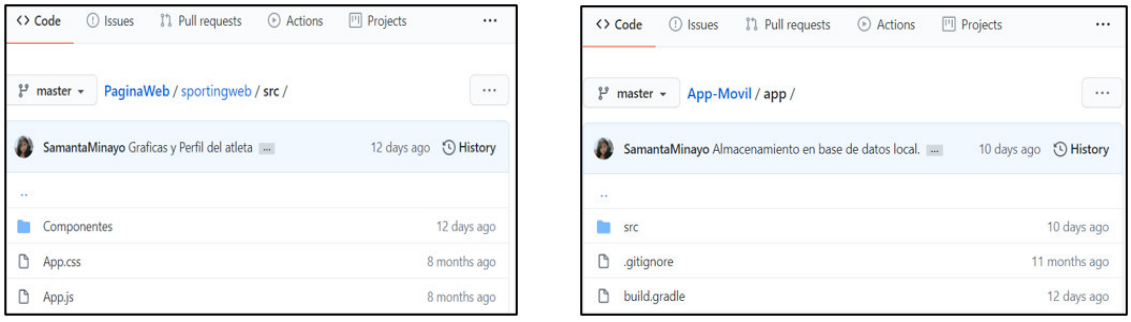
En la Figura 2.48 se observan los ítems de trabajo por hacer, en desarrollo y realizados para esta fase de implementación organizados en forma de tarjetas en el tablero kanban.

Figura 2.48: Tablero Kanban para la Fase de Implementación

A continuación, se procede con el desarrollo de los ítems de trabajo planteados.

2.3.1. VERSIONAMIENTO DEL CÓDIGO FUENTE

Para mantener un control de versionamiento y respaldo del código fuente del prototipo de plataforma se levantaron dos repositorios privados dentro de la plataforma Github los cuales se enlazaron a las aplicaciones que forman parte del prototipo. Como se indica en la Figura 2.49 (a) se ha creado el repositorio de la aplicación web y en la Figura 2.49 (b) se muestra el repositorio de la aplicación móvil.



(a) Aplicación Web (b) Aplicación Móvil

Figura 2.49: Repositorios levantados en Github

2.3.2. IMPLEMENTACIÓN DE FIREBASE

Para iniciar la implementación de firebase en este prototipo de plataforma se procede a crear un proyecto en la consola de firebase la cual proporciona una clave privada tipo json que será integrada en los ambientes de desarrollo de la aplicación web y aplicación móvil. A continuación, se explica las correspondientes configuraciones en la consola de firebase y en los diferentes ambientes de desarrollo.

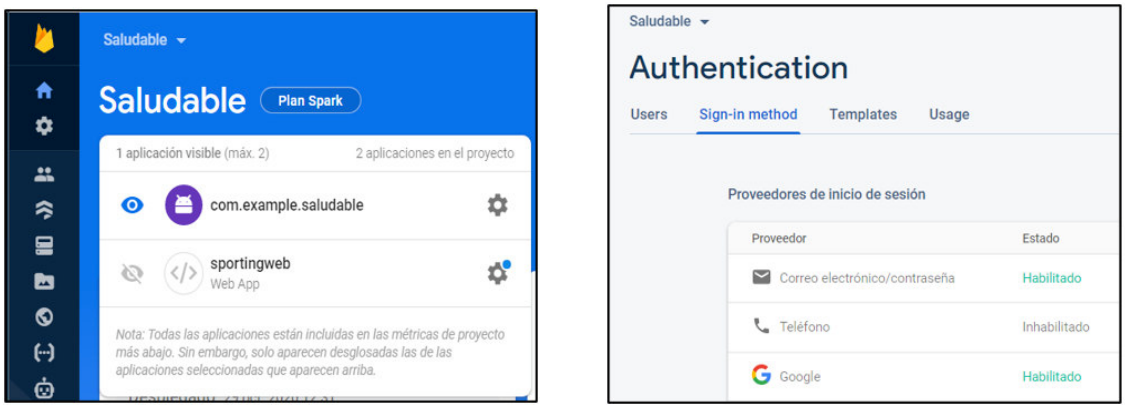
2.3.2.1. CONFIGURACIÓN EN CONSOLA DE FIREBASE

Una vez creado el proyecto en la consola de firebase se procede a la configuración de las aplicaciones del proyecto que se asociarán a los IDEs Android Studio y Visual Studio Code para la creación de la aplicación web y la aplicación móvil. La Figura 2.50 (a) muestra las aplicaciones creadas en el proyecto.

A continuación, se describen los servicios a los que se acceden en el proyecto de firebase.

Servicio Authentication

En este prototipo de plataforma se manejarán dos proveedores de inicio de sesión Google y Correo electrónico/Contraseña por lo que estos han sido activados mediante la consola de firebase en el servicio de Authentication como se muestra en la Figura 2.50 (b).



(a) Proyecto Creado (b) Servicio Authentication

Figura 2.50: Consola de Firebase

Servicios Storage y Realtime Database

Los servicios de almacenamiento (Storage) y base de datos (Realtime Database) manejan permisos de lectura y escritura mediante reglas, las cuales deben ser configuradas mediante la consola de Firebase. De acuerdo a los requerimientos del prototipo de plataforma la lectura de datos tanto para Storage y Realtime Database serán públicos como se indican en la Figura 2.51 (a) y (b) y los permisos de escritura los tendrán únicamente los usuarios autenticados dentro de la plataforma, esto se lo puede visualizar en la Figura 2.51 (a) y (b).

(a) Realtime Database (b) Storage

Figura 2.51: Consola de Firebase

Servicios Hosting y Cloud Messaging

La configuración de Hosting se la realizará mediante la aplicación Web y en consola se visualizará un historial de los publicaciones realizadas de la aplicación web en firebase.

Cloud Messaging al ser un servicio que se usará tanto en la aplicación web como en la aplicación móvil será configurado desde la aplicación web y aplicación móvil, por lo que

en la consola de firebase se mostrarán el par de claves que se generan al acceder a este servicio.

2.3.2.2. FIREBASE EN EL AMBIENTE DE DESARROLLO

Para la implementación de firebase en la aplicación móvil el ambiente de desarrollo Android Studio posee un asistente mediante el cual se conecta a la cuenta de firebase y, además permite la instalación y configuración de los servicios de Authentication, Storage, Realtime Database y Cloud Messaging. Finalmente se importará la clave privada generada por firebase para conectarse al proyecto creado anteriormente.

Para la implementación en la Aplicación Web se utilizarán las librerías firebase y firebase-tools propias de firebase mediante las cuales se puede acceder a sus servicios a través del IDE Visual Studio Code. Mediante comandos de consola se configurará la cuenta de firebase y el acceso a los correspondientes servicios. Finalmente se importará la clave privada generada por firebase para conectarse al proyecto creado anteriormente.

2.3.3. APLICACIÓN MÓVIL

El prototipo de plataforma consta de la aplicación móvil y la aplicación web. En esta sección se describe la implementación de la aplicación móvil con el IDE Android Studio.

En el ANEXO B se puede observar toda la solución de la aplicación móvil.

Una vez conectado el proyecto de Android Studio con Firebase se crearán los diferentes Activities, los cuales corresponden a las pantallas o vistas que existirán en la aplicación móvil. Al crear un activity se generan dos archivos: el primero es un archivo en formato xml conocido como layout el cual corresponde a la interfaz gráfica del activity y el segundo es un archivo en formato java que contiene la lógica de programación que dará funcionalidad al activity.

Dentro de la aplicación móvil se crearán doce activities con sus correspondientes layouts: LoginActivity, RegisterActivity, SetupActivity, MainActivity, MaratonActivity, ClickMaratonActivity, ProfileActivity. SettingsActivity, MiMaratonActivity, MiInscripcionActivity, ClickMiMaratonActivity y MapsActivity.

2.3.3.1. INTERFAZ GRÁFICA

Android Studio posee los componentes básicos para la creación de las interfaces de usuario como Button, EditText, TextView, ImageView, Chronometer, fragment, entre otros. Además,

si se requiere de algún componente externo Android Studio permite la importación de librerías.

Para el diseño de los componentes mencionados anteriormente cada uno de ellos posee propiedades mediante las cuales se puede modificar la vista de cada componente. Finalmente cabe mencionar que el componente fragment es el que permitirá importar un mapa desde Google maps mediante el cual se mostrará la trayectoria recorrida por el usuario.

En este prototipo de plataforma la aplicación móvil maneja varias vistas que consisten en listas de publicaciones o carreras como: MainActivity, MaratonActivty, MiMaratonActivity y MiInscripcionActivity, para ello se manejarán layouts extras, los cuales tienen el formato de visualización de cada publicación o carrera dentro del layout principal.

2.3.3.2. IMPLEMENTACIÓN DE CLASES

La etapa de desarrollo se inició implementando las clases Maraton, MaratonResult, Post, Punto y User, estas clases corresponden a los objetos dentro de la aplicación móvil. La Figura 2.52 muestra los objetos creados en Android Studio para la aplicación móvil. La información de carreras, los resultados por carrera, la información del usuario y la información de las publicaciones del administrador del prototipo de plataforma.

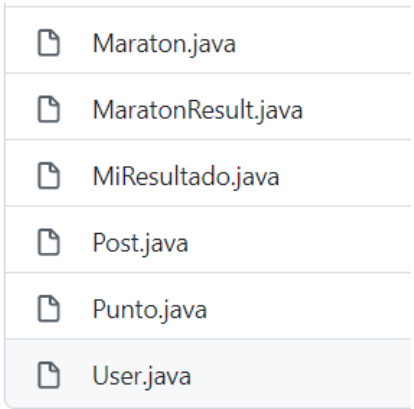


Figura 2.52: Clases de Objetos creados en Android Studio

2.3.3.3. NAVEGACIÓN ENTRE PANTALLAS

La interacción del usuario con la aplicación móvil consiste en la navegación entre varias pantallas, para esta navegación se considera hacía que pantalla o vista se dirige el usuario y a cual regresa.

El segmento de código 2.1 muestra el método usado para la interacción entre pantallas. La

línea 2 instancia el objeto mediante el cual se define el activity al que se dirigirá el usuario y la línea 3 inicia la instancia creada en la línea 2 abriendo en nuevo activity.

Segmento de código 2.1: Navegación entre Activities

```
1 private void SendUserToLoginActivity() {
2 Intent mainIntent = new Intent ( LoginActivity.this, LoginActivity.class );
3 startActivity ( mainIntent );
4 }
```

2.3.3.4. MANEJO DE PERMISOS

Debido a los requerimientos funcionales de la aplicación móvil, esta debe acceder a las funcionalidades de servicios en segundo plano, ubicación, internet y galería del dispositivo. Estas funcionalidades requieren de permisos los cuales deben ser configurados en el archivo AndroidManifest.xml como se indica en el segmento de código 2.2

Segmento de código 2.2: Permisos Manifest

```
1 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
2 <uses-permission ...
      android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
3 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
4 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
5 <uses-permission android:name="android.permission.INTERNET" />
6 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7 <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

Además de configurar los permisos en el archivo AndroidManifest.xml estos deben ser solicitados al usuario mediante la aplicación móvil. El segmento de código 2.3 muestra en la línea 1 el método requestPermissions dentro del código de java. Este método puede solicitar uno o varios permisos a la vez, por lo que en la primera vista tras la instalación de la aplicación serán solicitados todos los permisos y en caso de no ser aceptados se volverán a solicitar individualmente dependiendo de la funcionalidad que tenga el Activity.

Segmento de código 2.3: Código para solicitar permisos

```
1 private void requestPermissions() {
```

```
2 ActivityCompat.requestPermissions ( LoginActivity.this , new String[]{
3     Manifest.permission.ACCESS_FINE_LOCATION,
4     Manifest.permission.WRITE_EXTERNAL_STORAGE
5 }, CODIGO_PERMISO );}
```

2.3.3.5. LISTAS DINÁMICAS

La aplicación móvil maneja la vista de cuatro listas dinámicas: publicaciones del administrador, nuevas carreras, carreras en las que el usuario está inscrito y carreras en las que el usuario a participado, por lo que es necesario usar RecyclerView.ViewHolder, ya que permite el manejo eficiente de conjuntos de datos además define el aspecto de cada elemento y lo crea de forma dinámica en base al conjunto de datos.

El segmento de código 2.4 muestra la estructura del ViewHolder para la lista de carreras en las que el usuario a participado. Las líneas 4,5,8.9,10 y 11 instancian los componentes de la vista del elemento y le asignan el valor correspondiente de cada elemento que se mostrara en la interfaz de usuario de la aplicación móvil.

Segmento de código 2.4: Código de View Holder

```
1  public class MiMaratonViewHolder extends RecyclerView.ViewHolder implements
2  View.OnClickListener {
3      public TextView username, maratondate, maratonname, maratondescription;
4      public CircleImageView maratonimage;
5      IRecyclerItemClickListener iRecyclerItemClickListener;
6      public MiMaratonViewHolder(@NonNull View itemView) {
7          super ( itemView );
8          maratondate = itemView.findViewById ( R.id.my_maraton_date );
9          maratonname = itemView.findViewById ( R.id.my_maraton_name );
10         maratonimage = itemView.findViewById ( R.id.my_maraton_image );
11         maratondescription = itemView.findViewById ( ...
               R.id.my_maraton_description );
12         itemView.setOnClickListener ( this );
13     }
14 }
```

2.3.3.6. BASE DE DATOS LOCAL

La aplicación móvil accede a la información del usuario desde la base de datos de firebase por lo que requiere que el dispositivo se encuentre conectado a internet, por ello se agregará

la funcionalidad de mantener la información del usuario en una base de datos local del dispositivo para que el usuario tenga acceso a su información sin requerir de internet.

Android Studio posee la librería SQLite la cual permite tener una base de datos conectada a la aplicación móvil dentro del dispositivo, esta base de datos tendrá las tablas user, maraton, maresult, puntos y result. Estas tablas manejarán la información del usuario de la siguiente manera:

- Usuarios (user). Esta tabla tiene la información del perfil de usuario, es decir id, nombre, edad, estatura, peso, género, imc, url de la imagen, etc.
- Carrera (maraton). Esta tabla contiene la información de las carreras en las que el usuario se ha inscrito o ha participado como: id, fecha, hora, descripción, nombre de contacto, número de contacto, lugar, estado, distancia, trayectoria, etc.
- Resultados por carrera (maresult). Esta tabla contiene los resultados de las carreras en las que el usuario ha participado como: velocidad promedio, velocidad máxima, velocidad mínima, mejor tiempo, peor tiempo, etc.
- Puntos (puntos). Esta tabla contiene la información punto a punto de la trayectoria recorrida por el usuario en cada carrera en la que ha participado.
- Resultados (result). Esta tabla contiene la información de los resultados de la participación del usuario por carrera.

Para la administración de información en la base de datos se creó un objeto de acceso de datos (DAO) para cada tabla, en el cual se codificaron las consultas básicas como insertar, editar, obtener y eliminar para el manejo de información.

El segmento de código 2.5 muestra la creación de la clase Dao para el manejo de la tabla de usuarios. La línea de código 5 instancia el nombre de la base de datos y la línea 6 define la creación y estructura que tendrá la tabla user. Además, en las líneas 7 a 10 se observa el constructor de la clase donde se instancia la base de datos y la tabla correspondiente a la clase.

Segmento de código 2.5: Código de la clase DAO para usuarios

```
1     SQLiteDatabase db;
2     User usr;
3     Context ctx;
4     String nombredb = "SaludableDB";
```

```
5       String tabla = " create table if not exists user(" +"uid text primary ...
            key,email text,altura text, edad text, peso text, genero text," ...
            +"fullname text,username text, country text, imc text, profileimage ...
            text, status text" +  ",rango text, paso text, image text)";
6       public DaoUsers(Context c) {
7           this.ctx = c;
8           db = c.openOrCreateDatabase ( nombredb, Context.MODE_PRIVATE, null );
9           db.execSQL ( tabla );
10      }
```

El segmento de código 2.6 muestra un ejemplo de consultas SQL como: insertar, actualizar, obtener y eliminar sobre la base de datos. A continuación, se detallara cada una de ellas.

- Insertar. Este método inserta un registro en la tabla. Recibe como parámetros el nombre de la tabla, el valor que se escribirá en cada columna donde no exista un valor asignado para el elemento a ingresar, finalmente recibe un elemento del tipo ContentValues donde se asignarán valores para cada columna del elemento a ser ingresado.
- Actualizar. Este método actualiza la información de una fila en base a un parámetro. El método update requiere el nombre de la tabla, el objeto de tipo ContentValues y el valor del parámetro usado para la actualización del registro.
- Obtener. El método rawQuery permite obtener elementos de la base de datos dependiendo de la consulta que se envíe mediante este método, el segmento de código 2.6 muestra en la línea 3 la consulta enviada para obtener el usuario guardado en la tabla user y además está configurado para retornar el valor null en caso de no encontrar registros.
- Eliminar. El método delete elimina un registro de la tabla en base a un parámetro. Este método recibe como parámetros el nombre de la tabla y el valor del parámetro usado para eliminar el registro.

Segmento de código 2.6: Ejemplo de código de consultas sql

```
1 db.insert ( "user", null, contenedor )
2 db.update ( "user", contenedor, "uid=" + "'" + uid + "'")
3 db.rawQuery ( "select *from user", null )
4 db.delete ( "user", "uid=" + "'" + uid + "'")
```

2.3.3.7. MÉTODOS DE FIREBASE

En esta sección se mostrarán los métodos utilizados para acceder a los servicios de Authentication, Storage, Realtime Database y Cloud Messaging de firebase. Para acceder a estos servicios es necesario que cada uno de ellos sea instanciado dentro de cada activity en el que se lo utilice.

Servicio Authentication

El manejo de autenticación tiene dos Activities: RegisterActivity y LoginActivity. El primero permite registrar un nuevo usuario con el proveedor de inicio de sesión Correo Electrónico/-Contraseña, el segundo permite registrar un nuevo usuario e iniciar sesión con el proveedor de Google, es decir gmail y además permite el inicio de sesión basado en correo electrónico/contraseña.

El segmento de código 2.7 muestra la creación de un nuevo usuario en el servicio Authentication de firebase con el proveedor correo electrónico/contraseña mediante el método createUserWithEmailAndPassword desde RegisterActivity. En la línea 4 se observa el método onComplete el cual se ejecuta una vez que la petición de ingreso de usuario hacia firebase se ha completo, además en la línea de código 5 se verifica que se haya completado exitosamente el proceso y se procede al ingreso de la información de usuario en la base de datos de firebase desde la aplicación.

Segmento de código 2.7: Creación de nuevo usuario en Firebase desde Android Studio

```
1  mAuth.createUserWithEmailAndPassword ( email, password ).
2  addOnCompleteListener ( new OnCompleteListener<AuthResult> () {
3      @Override
4      public void onComplete(@NonNull Task<AuthResult> task) {
5          if (task.isSuccessful ()) { SendUserToSetupActivity ();}
6          else {
7              Toast.makeText ( RegisterActivity.this, "Error ...
                   "+task.getException ().toString (), Toast.LENGTH_SHORT).show ...
                   ();
8          }
9      }
10 });
```

LoginActivity maneja dos proveedores de autenticación mediante la instancia mAuth de FirebaseAuth. El primer proveedor basado en correo electrónico/contraseña requiere la exis-

tencia previa de un usuario en el servicio Authentication de firebase y el segundo basado en google autentica al usuario a través de gmail y en caso de no existir el usuario en firebase lo crea. Al finalizar el proceso de autenticación si este se realizó correctamente la aplicación dirige al usuario a la pantalla de ingreso de datos.

El segmento de código 2.8 muestra el método signInWithEmailAndPassword mediante el cual se autentica el usuario en Firebase en base a correo y contraseña. La línea de código 4 muestra el método onComplete mediante el cual se obtiene el resultado a la petición de autenticación, si el usuario no fue autenticado se mostrará el mensaje de error devuelto desde firebase.

Segmento de código 2.8: Autenticación correo/contraseña

```
1  mAuth.signInWithEmailAndPassword ( email, password )
2  .addOnCompleteListener ( new OnCompleteListener<AuthResult> () {
3  @Override
4  public void onComplete(@NonNull Task<AuthResult> task) {
5      if (task.isSuccessful()) {
6          SendUserToMainActivity ();
7      } else {
8          Toast.makeText ( LoginActivity.this, "Error "+task.getException ...
               ().toString (), Toast.LENGTH_SHORT ).show ();
9      }
10 }});
```

El segmento de código 2.9 muestra el funcionamiento del método firebaseAuthWithGoogle. En la línea 1 se observa que el método recibe el parámetro de entrada del tipo GoogleSignInAccount el cual genera una nueva vista para la autenticación del usuario en su cuenta de google y devuelve un token que será usado para la autenticación dentro de firebase a través del método signInWithCredential. Finalmente, la línea de código 5 muestra el método onComplete mediante el cual se obtiene el resultado a la petición de autenticación, si el usuario no fue autenticado se mostrará el mensaje de error devuelto desde firebase.

Segmento de código 2.9: Autenticación Google

```
1  private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
2      AuthCredential credential = GoogleAuthProvider.getCredential ( ...
           acct.getIdToken (), null );
```

```
3           mAuth.signInWithCredential ( credential ).addOnCompleteListener ( ...
                this, new OnCompleteListener<AuthResult> () {
4           @Override
5           public void onComplete(@NonNull Task<AuthResult> task) {
6               if (task.isSuccessful ())SendUserToMainActivity ();
7               else SendUserToLoginActivity ();
8       }}
9   );}
```

Servicio Storage

El Storage de Firebase se utilizará por la aplicación móvil para el almacenamiento de imágenes de perfil de cada usuario en la carpeta ProfileImages. Cada imagen almacenada estará en formato JPG y tendrá como nombre el uid del usuario generado por el servicio de Authentication de firebase. El método de almacenamiento de imagen será usado en las pantallas de ingreso de datos del usuario y actualización de información de usuario.

El segmento de código 2.10 contiene los métodos usados para insertar la imagen en el storage de firebase en la línea 2 y poder acceder a la url de descarga en la línea 7. La url de descarga obtenida será almacenada en la base de datos de firebase como un elemento de la información de usuario para posteriormente acceder a la url y visualizar la imagen de perfil del storage en la aplicación móvil.

Segmento de código 2.10: Insertar Imagen en Storage y obtener url

```
1  StorageReference filePath = UserProfileImageRef.child(currentUserID+".jpg");
2  filePath.putFile ( resultUri ).addOnCompleteListener (
3  new OnCompleteListener<UploadTask.TaskSnapshot> () {
4  @Override
5  public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {
6      if (task.isSuccessful ()) {
7          filePath.getDownloadUrl ().addOnCompleteListener (
8           new OnCompleteListener<Uri> () {
9          @Override
10         public void onComplete(@NonNull Task<Uri> task) {
11             final String downloadUri = task.getResult ().toString ();
12         }});
13     }});
14 });
```

Servicio Realtime Database

La aplicación móvil accede a la base de datos Realtime Database para ingresar la información de usuario, inscripciones de usuarios en carreras, resultados de participación del usuario en una carrera y además permite la lectura de toda esta información incluyendo la información publicada por el administrador como nuevas carreras y publicaciones.

A continuación, se detalla el código usado para la lectura y escritura de información en Firebase Realtime Database.

El segmento de código 2.11 muestra un ejemplo de la lógica de programación usada para ingresar datos en Realtime Database de Firebase. La función updateChildren permite ingresar o actualizar información en la base de datos dependiendo del nodo de realtime database instanciado en el objeto SettingsUserRef. Además, se tiene el método addOnCompleteListener el que se ejecuta una vez que se ha completado el proceso de ingreso de información en el nodo.

Segmento de código 2.11: Insertar información en Realtime Database

```
1 SettingsUserRef.updateChildren ( userMap ).addOnCompleteListener ( new ...
      OnCompleteListener () {
2 @Override
3 public void onComplete(@NonNull Task task) {
4     if (task.isSuccessful ()) Toast.makeText ( SettingsActivity.this, ...
          "Usuario actualizado correctamente", Toast.LENGTH_SHORT ).show ();
5     else Toast.makeText ( SettingsActivity.this, "Error Ocurred: " + ...
          task.getException ().getMessage (), Toast.LENGTH_SHORT ).show ();
6 }});
```

En el segmento de código 2.12 se observa un ejemplo de la lógica de programación usada para la lectura de información desde Realtime Database. La línea 1 muestra el método addListenerForSingleValueEvent el cual crea un listener que permite obtener en tiempo real la información existente en ese nodo por lo que este método se ejecutará cada vez que la información dentro de ese nodo sea actualizada.

Segmento de código 2.12: Leer Información de Realtime Database

```
1 UsersRef.addListenerForSingleValueEvent ( new ValueEventListener () {
2     @Override
```

```
3       public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
4           if (!dataSnapshot.exists ()) {
5               Common.loggedUser = dataSnapshot.child ( "Informacion" ...
                    ).getValue ( User.class );
6           }
7       }
8   });
```

Servicio Cloud Messaging

La aplicación móvil recibe una notificación cada vez que el administrador crea una nueva carrera, para ello una vez configurado cloud messaging en el proyecto de Android Studio se codifico la línea mostrada en el segmento de código 2.21 la cual consiste en que el usuario se suscribe a un tema en el que el administrador publicará las nuevas carreras.

Segmento de código 2.13: Suscribirse a un tema en Cloud Messaging

```
1 FirebaseMessaging.getInstance ().subscribeToTopic ( "NewMaraton" );
```

2.3.3.8. SERVICIO DE MONITOREO

Para el monitoreo de la ubicación del atleta se configuro un servicio que se ejecuta constantemente una vez que la carrera ha dado inicio y el atleta ha activado el monitoreo.

El servicio de monitoreo se implementa mediante la librería de android Localbroadcastmanager la cual permite configurar un canal de comunicación dentro de la aplicación móvil mediante la creación de un sender y un reciver.

El sender es el encargado de obtener la ubicación del usuario y enviarla al receiver de la aplicación móvil, para ello se consideran dos casos, cuando la aplicación se encuentra abierta en primer plano y en segundo plano, ya que en el segundo caso por políticas del sistema operativo Android la aplicación móvil debe enviar una notificación persistente al usuario.

En el segmento de código 2.14 muestra el envío de la ubicación desde el sender hacia el receiver mediante la función LocalBroadcastManager.getInstance ().sendBroadcast (). Además, en la línea 9 y 10 se observa la verificación del estado de ejecución del servicio para la configuración y envió de notificaciones persistentes hacia el usuario de la aplicación.

Segmento de código 2.14: Obtener nueva ubicación y envío al receiver

```
1  public void onNewLocation(Location location) {
2      mLocation = location;
3      if (mLocationant == null) {mLocationant = mLocation; }
4      Intent intent = new Intent ( ACTION_BROADCAST );
5      intent.putExtra ( EXTRA_LOCATION, location );
6      LocalBroadcastManager.getInstance ( getApplicationContext () ...
           ).sendBroadcast ( intent );
7      distancia = distancia + mLocationant.distanceTo ( mLocation );
8      mLocationant = mLocation;
9      if (serviceIsRunningInForeground ( this )) {
10         mNotificationManager.notify ( NOTIFICATION_ID, getNotification () );
11     }
12 }
```

El receiver será el encargado de recibir la información de la ubicación del usuario y actualizarla dentro de la base de datos de firebase y en la interfaz gráfica.

El segmento de código 2.15 muestra la configuración del receiver donde se ejecuta la función agregarMarcador(), la cual se encarga del procesamiento de la información obtenida desde el gps del dispositivo móvil e ingresarla en firebase.

Segmento de código 2.15: Configuración del Receiver

```
1  public class MyReceiver extends BroadcastReceiver {
2  @Override
3  public void onReceive(Context context, Intent intent) {
4  Location location = intent.getParcelableExtra ( MyService.EXTRA_LOCATION );
5      if (location != null) {
6          agrerarMarcador ( location );
7          if (fin) GuardarFinal ();
8      }
9  }}
```

Finalmente, una vez que el administrador a finalizado la carrera desde la aplicación web o el usuario a finalizado la monitorización de información se genera la información promedio de la participación del usuario en la carrera y se actualiza la información de usuario por carrera de carreras inscritas a carreras realizadas.

2.3.4. APLICACIÓN WEB

El prototipo de plataforma consta de la aplicación móvil y la aplicación web. En esta sección se describe la implementación de la aplicación web con la tecnología React JS y el IDE Visual Studio Code.

En el ANEXO C se puede observar toda la solución de la aplicación web.

Dentro de la aplicación web se definen rutas mediante la librería react-router-dom. El segmento de código 2.16 muestra un ejemplo de las rutas configuradas para la aplicación web, se configuraron rutas estáticas y rutas dinámicas, estas últimas son necesarias para el manejo de información de los diferentes usuarios autenticados que acceden a la aplicación web.

Segmento de código 2.16: Configuración de rutas para la aplicación web

```
<Router history={hist}>
    <Switch>
      <Route path="/crudadmin" component={CrudCarreras} />
      <Route path="/login" component={Login} />
      <Route path="/admin" component={Login} />
      <Route path="/" component={Home} />
    </Switch>
  </Router>,
```

Adicional a la implementación de firebase mediante consola en Visual Studio Code es necesario configurar las claves de acceso al servicio proporcionadas por firebase. El segmento de código 2.17 muestra la configuración de las claves de acceso e inicialización de firebase en el proyecto React JS. Las líneas de código 1 a 9 indican la configuración de la clave de acceso de firebase y la línea 10 muestra la inicialización de la conexión con firebase.

Segmento de código 2.17: Configuración de parámetros de firebase

```
const config={
  apiKey: "AIzaSyCAia3ylYt4CYfsoPHGzCj2OoQyr86Kv5A",
  authDomain: "saludable-c928b.firebaseapp.com",
  databaseURL: "https://saludable-c928b.firebaseio.com",
  projectId: "saludable-c928b",
  storageBucket: "saludable-c928b.appspot.com",
  messagingSenderId: "140819437335",
```

```
8     appId: "1:140819437335:web:b9c1192bdd5724b943c952"
9   }
10  firebase.initializeApp(config)
```

A continuación, se describen los módulos implementados en la aplicación web:

2.3.4.1. MÓDULO DE AUTENTICACIÓN

El prototipo de plataforma maneja dos tipos de autenticación, las cuales se implementaron tanto para la aplicación móvil como para la aplicación web dentro del módulo de autenticación.

Autenticación correo electrónico/contraseña

Para la autenticación basada en correo electrónico/contraseña se implementó el método Login mostrado en el segmento de código 2.18, el cual recibe como parámetros de entrada el correo electrónico y la contraseña, además dentro de la librería de firebase se accede al servicio de autenticación mediante la función signInWithEmailAndPassword del cual en caso de generarse un error será capturado para generar los mensajes que se le mostraran al usuario.

Segmento de código 2.18: Autenticación basada en correo electrónico/contraseña

```
1  Login(email,password){
2      firebase.auth().signInWithEmailAndPassword(email,password)
3      .then(result => { window.location.href="/"  });
4  }
```

Autenticación del Administrador

El prototipo de plataforma maneja un solo administrador, por lo que para la autenticación de este se creó el usuario en el nodo Admin de la base de datos realtime database donde se verificara la existencia del usuario para posteriormente autenticarlo como se indica el en el segmento de código 2.19, una vez autenticado se configura el elemento rol del objeto state para acceder al módulo de administración.

Segmento de código 2.19: Autenticación de administrador

```
1  firebase.database().ref().child("Admin").on('child_added', snapshot =>{
2      if(snapshot.child("email").val()===this.state.email){
```

```
3           this.setState({ rol:"Admin" })
4           this.Login(email,password)
5       }else{ window.alert("No existe usuario") }
6   })
```

Autenticación basada en Google

Para la autenticación basada en el proveedor de Google, dentro de la interfaz de autenticación se implementó un botón que ejecuta el método handleAuthwithGmail mostrado en el segmento de código 2.20, en la línea 2 se invoca al proveedor de autenticación de Google y en la línea 3 se accede a la función signInWithPopup(provider), el cual despliega una ventana emergente para la autenticación de Gmail y retorna el resultado de la autenticación con el proveedor de Google.

Segmento de código 2.20: Autenticación basada en proveedor de Google

```
1  handleAuthwithGmail() {
2      const provider = new firebase.auth.GoogleAuthProvider();
3      firebase.auth().signInWithPopup(provider)
4      .then(result => { window.location.href="/" });
5  }
```

Cierre de sesión

La finalización de sesión se maneja mediante la función signOut invocada desde el servicio de autenticación de la librería de firebase.

Segmento de código 2.21: Leeer Información de Realtime Database

```
1  handleLogout(){
2      firebase.auth().signOut();
3  }
```

Estado de sesión

La verificación de los usuarios autenticados se la realiza mediante la función onAuthStateChanged mostrada en el segmento de código 2.22, esta función crea un listener que se ejecuta cada vez que existe un cambio de estado en la sesión, es decir si existe un usuario autenticado en el navegador se lo verifica desde el módulo de autenticación de firebase y si

este cierra sesión inmediatamente se invoca nuevamente esta función y dentro de la lógica de programación para el manejo de sesión de la aplicación web se actualiza elemento user del objeto state.

Segmento de código 2.22: Estado de sesión

```
1 firebase.auth().onAuthStateChanged(user => {
2 this.setState({user});
3 if(user){
4     this.setState({
5         email: user.email,
6         sesion: 'Cerrar Sesion'
7     })
8 }});
```

2.3.4.2. MÓDULO DE ADMINISTRACIÓN

Es un módulo dentro de la aplicación web que permite al administrador alimentar a la plataforma con información de carreras para los atletas, además le permite al administrador crear publicaciones para la pantalla de inicio de la aplicación web y móvil. El administrador podrá añadir carreras, editarlas, eliminarlas y monitorizarlas. La interfaz del módulo consta de cuatro opciones dentro del menú de administración: Administración de publicaciones, Nueva Carrera, Editar Carrera y Monitorizar Carrera.

La opción administración de publicaciones permite crear y editar publicaciones, para esto se creó el método GuardarDatos mostrado en el segmento de código 2.23, este método recibe la información necesaria para crear o editar la publicación, las líneas de código 2 y 3 muestran las instancias de firebase storage para ingresar la imagen de la publicación en el Storage de firebase dentro de la carpeta PostImage, una vez que se ha completado la carga de la imagen se ejecuta el segmento de código de la línea 5 en adelante. La línea 5 obtiene la url de la imagen que será guardada como parte de la información de la publicación en la base de datos realtime database, a partir de la línea de código 7 se muestra el ingreso de la publicación en el nodo Post dentro de la base de datos realtime database.

Segmento de código 2.23: Guardar información de publicación

```
1 GuardarDatos(fullname,hora,fecha,descripcion,file,profileimage,filename){
2     const storageRef = firebase.storage().ref(`/PostImage/${filename}`);
```

```
3       const uploadTask = storageRef.put(file);
4       uploadTask.on('state_changed', snapshot =>{},()=>{
5       uploadTask.snapshot.ref.getDownloadURL().then(function(downloadURL) {
6           firebase.database().ref().child("Posts").child(filename).update({
7               fullname: fullname,
8               description: descripcion,
9               uid: filename,
10              postimage: downloadURL,
11              date: fecha,
12              time: hora,
13              profileimage:profileimage
14          });
15      });
16  });
17  }
```

La opción nueva carrera permite al administrador crear carreras y enviar una notificación a los usuarios de la aplicación móvil cada vez que se crea la carrera. El segmento de código 2.23 muestra el método GuardarDatos, el cual sigue la lógica de programación para el ingreso de imagenes en el storage de firebase y almacenamiento de información en realtime database, la carpeta asignada a imágenes de carreras dentro del storage es MarathonImages y el nodo dentro de realtime database es Carreras. Finalmente. una vez ingresada la información de la carrera en firebase se envía la notificación.

Dentro de la información de la carrera que se almacenará en realtime database está la trayectoria de la carrera, para ello se usara la aplicación My Maps de Google la cual tiene herramientas para dibujar líneas, medir distancias, ingresar marcadores y además permite compartir el mapa con la trayectoria creada mediante código html, el cual será insertado tanto en la aplicación web como en la aplicación móvil.

En el segmento de código 2.24 se muestra el envío de la notificación mediante el consumo del api rest de firebase proporcionada para el envío de cloud messaging. En la línea de código 2 se configura el host al que se realizara la petición, la línea de código 3 muestra que se usara el método POST, las líneas de código 4 a 7 indican la configuración de la cabecera de la petición, está contiene la autorización, la cual corresponde a la clave proporcionada por el servicio de firebase cloud messaging, las líneas 9 a 13 contienen el título, texto y sonido correspondiente a la estructura de la notificación, la línea 14 contiene el tema al que están suscritos los usuarios de la aplicación móvil y será el que reciba la notificación y la

enviará a los usuarios, finalmente se enviará la prioridad de la notificación.

Segmento de código 2.24: Envío de notificación

```
1  axios({
2      url: 'https://fcm.googleapis.com/fcm/send',
3      method: 'post',
4      headers: {
5          "Content-Type":"application/json",
6          "Authorization":"key=AAAAIMl-Fxc:APA91bF6NbcUkof6Ztt4DT3BSlhcDM ...
               LQMV0Qo1thDGucKF2t5t49UIqVsHJm35EKSOmRt4ZK1LDIMwFCYKOnN5fmm ...
               3_QTGSmtcyj95yLu4okE59tdX94OSGrRqBKZ7kOWo4hXIFHjE6"
7      },
8      data: {
9          "notification": {
10             "title":"Saludable",
11             "text":`Hay una nueva carrera disponible para ti. Inscribete en: ...
                   ${nombreCarrera}`,
12             "sound":"default"
13         },
14         "to":"/topics/NewMaraton",
15         "priority":"high"
16     }
17 }).then((r) => {
18     window.alert("INFORMACION ENVIADA CORRECTAMENTE")
19     window.location.href='/lista'
20 })
```

La opción editar carrera muestra una lista de carreras no realizadas y permite al administrador ingresar a cualquiera de ellas donde podrá editar carrera, generar código y eliminar carrera.

Para editar carreras se utiliza el método GuardarDatos de la opción crear carrera mostrada el segmento de código 2.23, ya que este método utiliza la función update de la instancia de realtime database de firebase, el cual crea o actualiza el nodo dependiendo de si este existe.

Para eliminar carreras se utiliza la función remove de la instancia de la base de datos de firebase especificando el nodo que se desea eliminar, la línea 2 en el segmento de código 2.25 muestra el nodo correspondiente a la carrera que se eliminará.

Segmento de código 2.25: Eliminar carrera

```
1  Delete(){
2    firebase.database().ref().child("Carreras").child("Nuevas").  ...
       child(this.state.uid).remove().then(function(){
3        window.alert("INFORMACION ELIMINADA CORRECTAMENTE");
4        window.location.href='/listacarreras'
5      },function(error){
6          if(error){
7            window.alert("ERROR: "+ error.message)
8          }
9      });
10 }
```

La opción monitorizar carreras muestra la lista de carreras que aún no se realizan y permite al administrador acceder a ellas, donde podrá iniciar la carrera, finalizar la carrera y visualizar en tiempo real la ubicación de los atletas participantes de la carrera. Esta página de monitoreo consta de un mapa el cual se actualiza en tiempo real cada vez que un atleta actualiza su ubicación en la base de datos realtime database. La ubicación de los usuarios se la muestra en el mapa que se importa desde la librería react-leaflet y la actualización de la ubicación se la realiza creando dos listeners sobre el nodo de la base de datos que contiene la información de las ubicaciones en tiempo real, estos listeners corresponden a los eventos child_added y child_changed, child_changed corresponde a la actualización de la ubicación de cualquier atleta sobre la base de datos y el evento child_added obtiene la información de todos los elementos dentro del nodo.

El segmento de código 2.26 muestra la lógica de programación utilizada para actualizar la lista de marcadores que se mostrarán en el mapa, la línea 1 muestra el listener configurado para actualizarse cada vez que exista un cambio en los elementos del nodo de monitoreo de ubicación, las líneas de código 2 a 4 limpian el array de marcadores y mediante la línea 5 obtenemos los nuevos valores que llenarán el array de marcadores y serán los que se muestren en el mapa.

Segmento de código 2.26: Obtener ubicación de atletas en tiempo real

```
1  firebase.database().ref().child("Carreras").child("Monitoreo").child( ...
     this.props.uid).on('child_changed', snapshot =>{
2      this.setState({
3          markers : []
```

```
4       })
5       firebase.database().ref().child("Carreras").child("Monitoreo"). ...
            child(this.props.uid).on('child_added', snapshot =>{
6           this.setState({
7               markers : this.state.markers.concat(snapshot.val(),
8               currentLocation : position
9           })
10      })
11  })
```

2.3.4.3. USUARIOS AUTENTICADOS

La aplicación web permite la autenticación de los atletas para que accedan a su información en la aplicación web de forma similar a la que accede a su información desde la aplicación móvil como su información de perfil, carreras inscritas y carreras en las que ha participado.

Perfil

La aplicación web proporciona una interfaz de perfil al usuario donde visualizará su información, la información de todas las carreras en las que ha participado y la última carrera en la que ha participado. Además, mediante esta interfaz de perfil podrá acceder a la lista de carreras en las que el usuario se ha inscrito o ha participado

El segmento de código 2.27 muestra la función CargarResultados, el cual se ejecuta una vez obtenida la lista de carreras en las que ha participado el usuario. Los resultados de las carreras se obtienen recorriendo la lista de carreras y procesando los resultados individuales de cada carrera como se muestran en las líneas comprendidas entre la línea 2 y 13. Las líneas de código 14 a 19 generan el arreglo necesario para realizar la gráfica de distancia en función de las carreras que el atleta ha participado.

Segmento de código 2.27: Resultados de la lista de carreras por usuario

```
1  CargarReultados(){
2      this.state.carreras.forEach(element => {
3          cont=cont+1;
4          let timu=parseFloat( ...
               (element.tiempo/(element.distancia/1000)).toFixed(2));
5          let horasu=Math.trunc(timu/60);
6          let minu=timu-horasu*60;
7          let minutosu= Math.trunc(minu);
```

```
8           let segundosu=(minu-minutosu)*60;
9           velocidad=parseFloat(velocidad)+parseFloat(element.velocidad);
10          tiempo=parseFloat(tiempo)+parseFloat(element.tiempo/ ...
                (element.distancia/1000));
11          pasos=parseFloat(pasos)+parseFloat(element.pasos);
12          calorias=parseFloat(calorias)+parseFloat(element.calorias);
13          distancia=parseFloat(distancia)+parseFloat(element.distancia);
14          this.setState({
15              distanciag:this.state.distanciag.concat({
16                  x:cont,
17                  y:distancia/1000
18              })
19          });
20      })
21      let tim=parseFloat( tiempo.toFixed(2))/this.state.nump;
22      let min=tim-horas*60;
23      let minutos= Math.trunc(min);
24  }
```

La aplicación web muestra gráficas de los resultados de la participación de los atletas, para ello se importó la librería canvasjs.react, la cual permite insertar en la aplicación web gráficas de barras, líneas, dispersión, pasteles, etc.

El segmento de código 2.28 muestra un ejemplo de la configuración del objeto que se enviará como parámetro al componente CanvasJSChart para generar gráficas, este objeto permite configurar el título en la línea 6, los ejes en las líneas 8 a 16 y los datos de la gráfica en las líneas 18 a 22 como el tipo de gráfica, el formato de par de coordenadas y el array de los datos a graficar.

Segmento de código 2.28: Gráfica Distancia vs Carrera

```
1  const optionsd = {
2      animationEnabled: true,
3      exportEnabled: true,
4      theme: "light2", // "light1", "dark1", "dark2"
5      title:{ text: "Distancia vs Carrera" },
6      axisY: {
7          title: "Distancia",
8          includeZero: false,
9          suffix: "km"
```

```
10      },
11      axisX: {
12          title: "Carrera",
13          prefix: "",
14          interval: 1
15      },
16      data: [{
17          type: "line",
18          toolTipContent: "Carrera {x}: {y}km",
19          dataPoints: this.state.distanciag
20      }]}
```

Carreras Inscritas

Desde el perfil de usuario el atleta puede acceder a la lista de las carreras en las que se ha inscrito, esto se realiza consultando el nodo de la base de datos correspondiente a la lista de carreras en las que el usuario se ha inscrito. Además, al seleccionar cualquiera de ellas podrá visualizar la información de la carrera.

Carreras Realizadas

Desde el perfil de usuario el atleta puede acceder a la lista de las carreras en las que ha participado consultándolas desde la base de datos apuntando al nodo que contiene las carreras en las que el usuario ha participado.

La carrera realizada muestra los resultados del usuario autenticado, los resultados promedio de los participantes de la carrera y una sección para filtrar los resultados en base a género, edad, tiempo y velocidad.

La línea 1 del segmento de código 2.29 muestra la instancia del nodo de la base de datos realtime database en la que se encuentra la información correspondiente a la carrera realizada por el atleta. Las líneas de código 3 a 7 muestran cómo se asigna cada elemento del nodo a un elemento del objeto state que permitirá mostrar estos valores en la aplicación web.

Segmento de código 2.29: Obtener resultados por carrera realizada del atleta

```
1 firebase.database().ref().child("Users").child(this.props.uidu).child(" ...
      Resultados").child("Resultado").child(this.props.uidg)
2 .on('child_added', snapshot =>{
3     var name=snapshot.key;
```

```
4       var value= snapshot.val();
5       this.setState({
6           [name]:value,
7       })}
```

Los resultados promedios de la participación de los atletas de una carrera se obtienen consultando y procesando la lista de resultados finales de la participación de cada usuario en la base de datos de datos realtime database en el nodo Carreras/Resultados/id_carrera.

Para obtener los resultados de los mejores y peores parámetros de desempeño de los participantes se utiliza una combinación de los métodos orderByChild, limitTolast y limitToFirst. El segmento de código 2.30 muestra un ejemplo de la combinación de los métodos orderByChild y limitToLast donde se organizan los datos en base a la velocidad de menor a mayor y obtenemos únicamente el último valor al pasar el parámetro cantidad de registros dentro del método limitToLast y así se obtiene la velocidad más alta.

Segmento de código 2.30: Código para obtener mejor velocidad

```
1 firebase.database().ref().child("Carreras").child("Resultados"). ...
      child(this.props.uidg)
2 .orderByChild("velocidad").limitToLast(1).on('child_added', snapshot =>{
3       this.setState({
4           velmej : snapshot.child('velocidad').val()
5       })
6 })
```

2.3.4.4. USUARIOS NO AUTENTICADOS

Para los usuarios no autenticados la aplicación web consta de información pública en las pantallas de inicio, próximas carreras y resultados.

La pantalla próximas carreras mostrara la lista de las carreras creadas por el administrador que aún no se han realizado. Además, permite ingresar a cada una de ellas y visualizar la información de la carrera incluida la trayectoria.

La pantalla resultados mostrara la lista de todas las carreras que ya se han realizado. Además, permite ingresar a cada una de ellas y adicional a la información propia de la carrera se visualizarán los resultados generales de la participación de los atletas en cada carrera. Finalmente, la pantalla de inicio mostrarán las publicaciones realizadas por el administrador.

3. RESULTADOS Y DISCUSIÓN

En el presente capítulo se describirán las pruebas realizadas al prototipo, se presentarán los resultados obtenidos y con base en los errores encontrados se realizarán las correcciones.

Inicialmente, se presentarán pruebas de integración de la aplicación móvil para verificar el correcto funcionamiento de los módulos. Aquí se presenta la conexión de la aplicación móvil con firebase. A continuación, se realizán pruebas de funcionalidad de la aplicación web incluyendo el módulo de administración desde el cual se gestionará la información de las carreras. Después, se ejecutarán pruebas de funcionalidad entre la aplicación web y la aplicación móvil. A continuación, se realizarán pruebas de funcionalidad del prototipo de plataforma con la participación de 10 atletas en dos carreras y se establecerá un modelo de entrevista para los atletas. Finalmente, se presentan los errores corregidos con base en las entrevistas realizadas.

3.1. PRUEBAS DE FUNCIONALIDAD

En este trabajo de titulación se desarrolló un prototipo de plataforma de monitoreo de carreras atléticas mediante una aplicación web y una aplicación móvil. La aplicación web contiene el módulo de administración para gestionar la información de las carreras atléticas y la aplicación móvil es la que monitoriza la participación del atleta en la carrera. Además, la aplicación web permite la visualización de todas las carreras de la plataforma a usuarios autenticados y no autenticados, también permite a los usuarios de la aplicación móvil autenticarse en la aplicación web y visualizar su información de usuario y resultados.

A continuación, se describen las pruebas de funcionalidad del módulo de administración, aplicación móvil y aplicación web.

3.1.1. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE ADMINISTRACIÓN

El módulo de administración es una interfaz gráfica que permite gestionar la información de las carreras dentro de la plataforma. Este módulo consta de varios submódulos para la administración de publicaciones, administración de carreras y monitoreo de carreras.

Inicialmente la base de datos solo constará del nodo Admin donde se registrará el usuario del administrador de la plataforma, por lo que al acceder al módulo de administración no se visualizará información de carreras ni publicaciones.

3.1.1.1. AUTENTICACIÓN

En esta prueba se verificó que el usuario administrador pueda ingresar al módulo de administración de la aplicación web.

El administrador debe ingresar sus credenciales en el login de la aplicación web y seleccionar la opción administrador, finalmente debe autenticarse mediante el botón autenticar. La figura 3.1 muestra el login de la aplicación web.

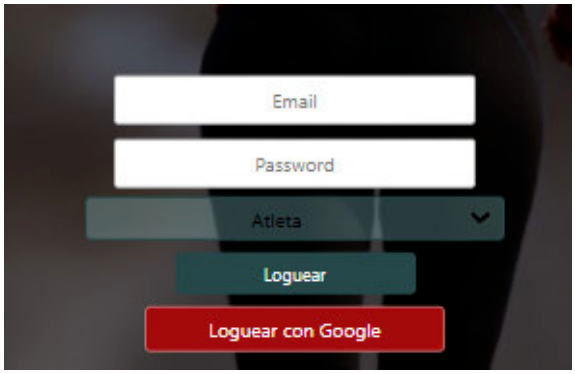


Figura 3.1: Pantalla de login

Si la información de autenticación es incorrecta la aplicación web muestra un mensaje de error. Si el correo ingresado no corresponda al administrador se mostrará el mensaje de error usuario no encontrado y en caso de corresponder al administrador los posibles mensajes de error que se muestran provienen del servicio de autenticación de firebase como contraseña incorrecta.

Si la autenticación se realiza correctamente se direccionará al administrador a la pantalla del menú de administración mostrada en la figura 3.2.

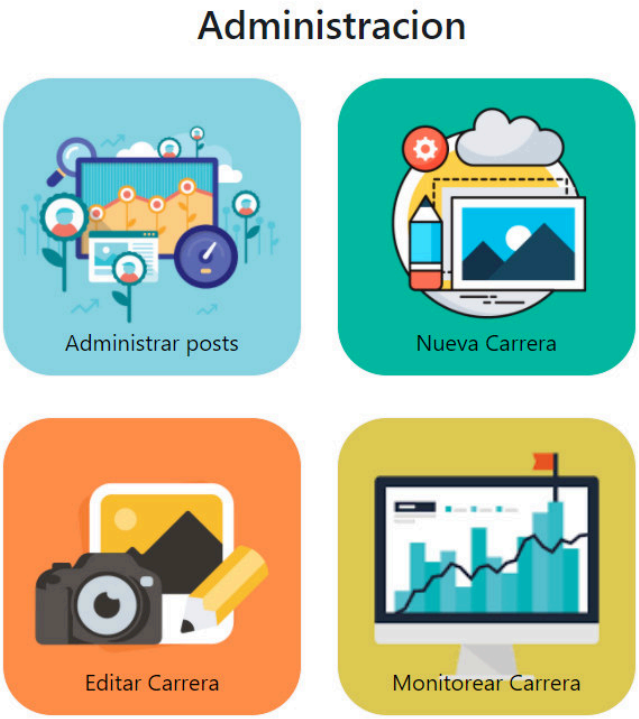


Figura 3.2: Pantalla del menú de administración

3.1.1.2. ADMINISTRACIÓN DE PUBLICACIONES

Durante esta prueba se verificó que el administrador pueda crear, editar y eliminar publicaciones.

La figura 3.3 presenta el formulario de administración para las publicaciones. El formulario muestra a la derecha la lista de todas las publicaciones existentes para poder editarlas o eliminarlas y a la izquierda el formulario de creación y edición de publicaciones.

Figura 3.3: Pantalla de administración de publicaciones

Una vez ingresada o actualizada la información de una publicación se muestra la alerta con el mensaje observado en la figura 3.4.

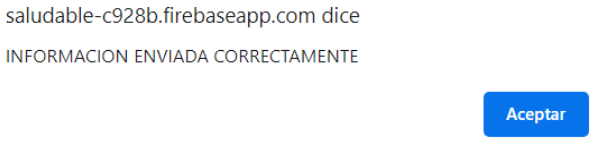


Figura 3.4: Mensaje de éxito de ingreso de información en la base de datos

Finalmente se probará la eliminación de publicación mediante el botón eliminar. Si la publicación se elimina correctamente se mostrará el mensaje observado en la figura 3.5

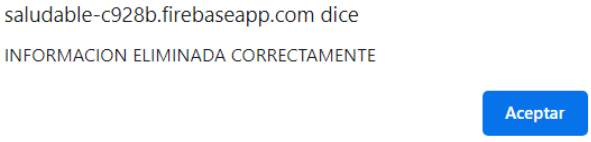


Figura 3.5: Mensaje de éxito de eliminación de información en la base de datos

3.1.1.3. ADMINISTRACIÓN DE CARRERAS

Durante esta prueba se verificó que el administrador pueda crear, editar y eliminar carreras correctamente.

NUEVA CARRERA

Desde el menú de administración se accede a la opción nueva carrera donde se observa el formulario mostrado en la imagen 3.6 y se ingresa la información de la carrera.

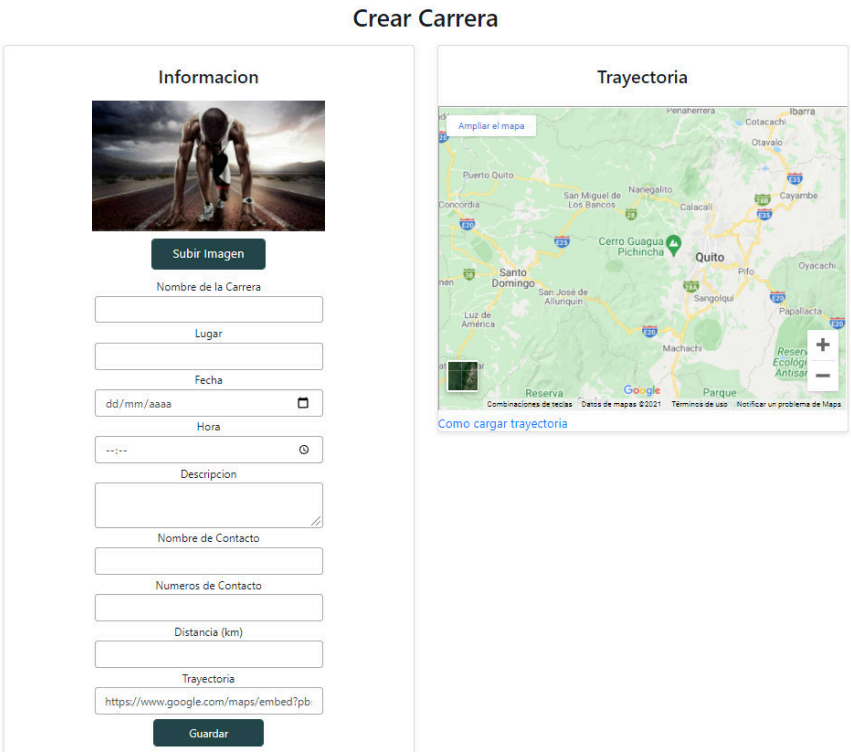


Figura 3.6: Pantalla de creación de carreras

La trayectoria de la carrera debe ser creada e importada desde la aplicación MyMaps de Google, ya que esta permite insertar marcadores, medir trayectorias y graficarlas, para ello el administrador tiene a su disposición un pequeño manual de instrucciones para la creación e importación de mapas de MyMaps de Google.

La figura 3.7 muestra la interfaz de Google MyMaps y el resultado de la creación de la trayectoria basada en el instructivo compartido con el administrador en la aplicación web. Una vez cargada la información de la trayectoria en el formulario automáticamente se actualizará el mapa mostrado en el formulario de la figura 3.6.

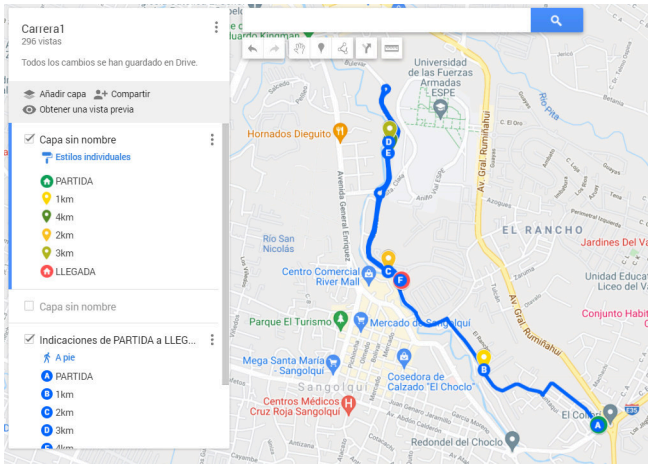


Figura 3.7: Creación de trayectoria en MyMaps

EDITAR CARRERA

La opción editar carrera muestra la lista de carreras que aún no se han realizado permitiendo ingresar a cada una de ellas para editarla, eliminarla o generar el código usado para la monitorización de carreras.

La figura 3.8 muestra la pantalla de edición de carreras con las opciones editar que habilita el formulario para actualizar la información de la carrera, generar código que crea un código aleatorio de 4 dígitos el cual se comparte con el atleta para ingresar a la monitorización de la carrera y por último la opción eliminar. Además, el formulario muestra a los atletas que se han inscrito en la carrera.

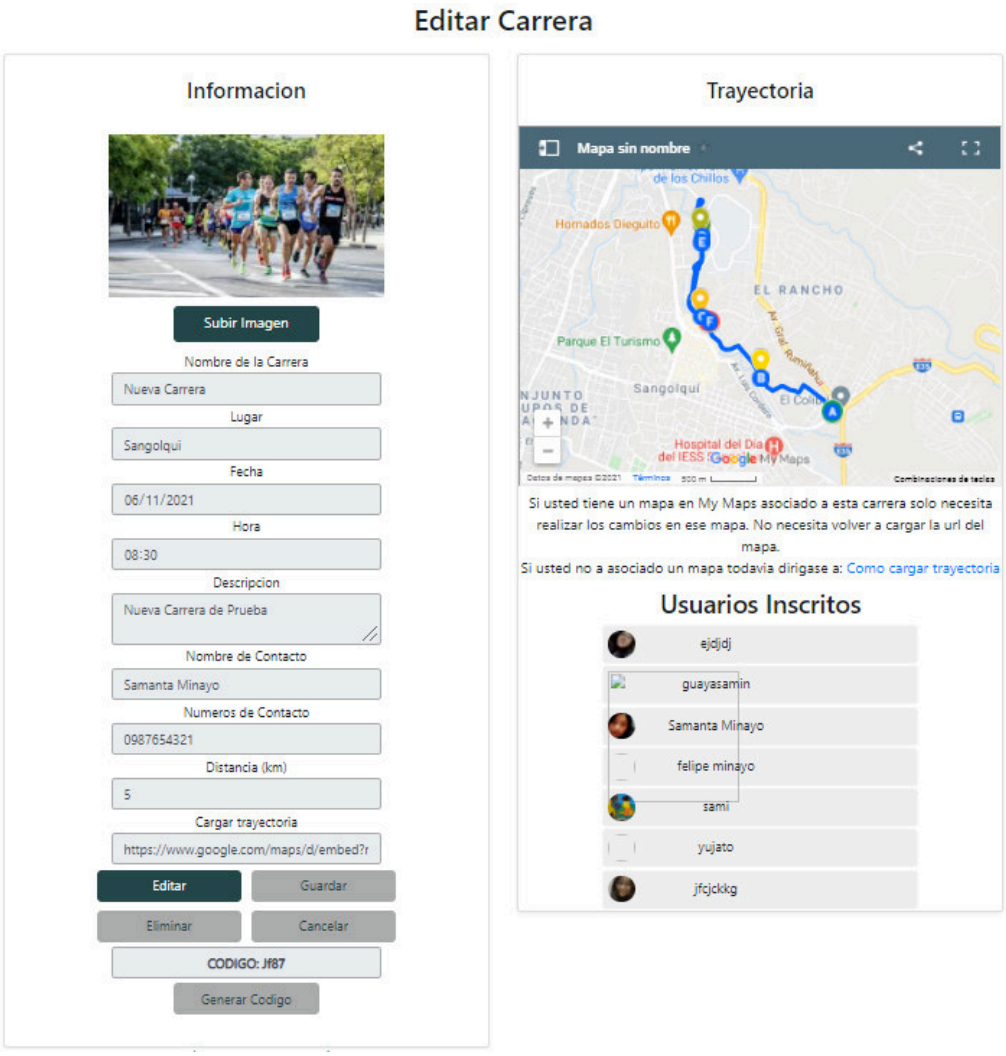


Figura 3.8: Pantalla de edición de carreras

Las acciones de crear, editar, generar código y eliminar carreras mostrarán como resultado una alerta con los mensajes: información enviada correctamente o información eliminada correctamente similar a los mensajes mostrados en las figuras 3.4 y 3.5

3.1.1.4. MONITORIZACIÓN DE CARRERAS

Durante esta prueba se verificó que el administrador pueda acceder a la pantalla de monitorización de ubicación de los atletas participantes de una carrera. Las pruebas de monitoreo de la carrera en tiempo real se realizarán en la sección pruebas de funcionalidad de monitoreo del prototipo.

La figura 3.9 muestra la pantalla de monitoreo donde se mostrarán los usuarios conectados a la carrera y sus correspondientes ubicaciones en el mapa.

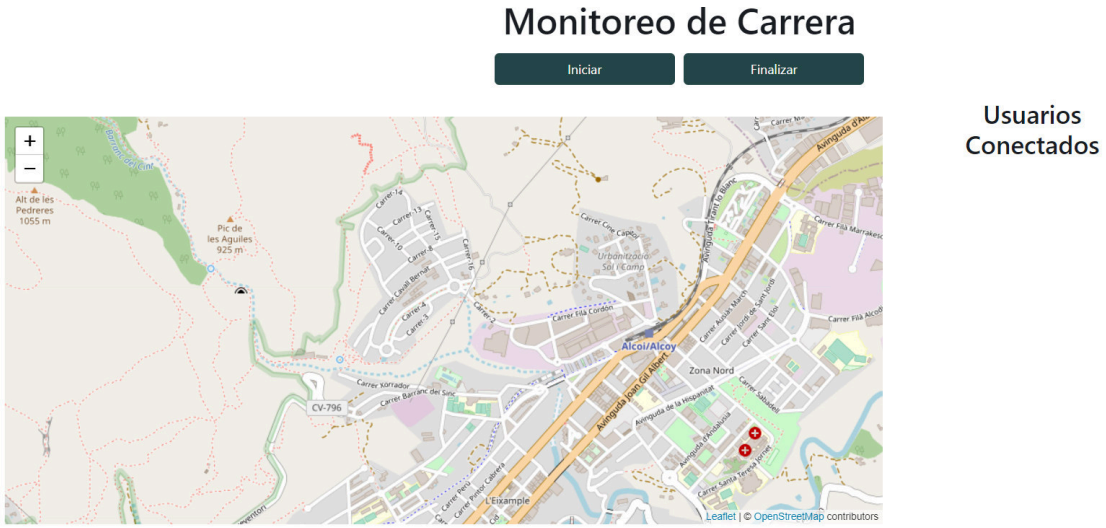


Figura 3.9: Pantalla de monitoreo de trayectoria

En el ANEXO D se localiza el manual de usuario del módulo de administración.

3.1.2. PRUEBAS DE FUNCIONALIDAD DE LA APLICACIÓN MÓVIL

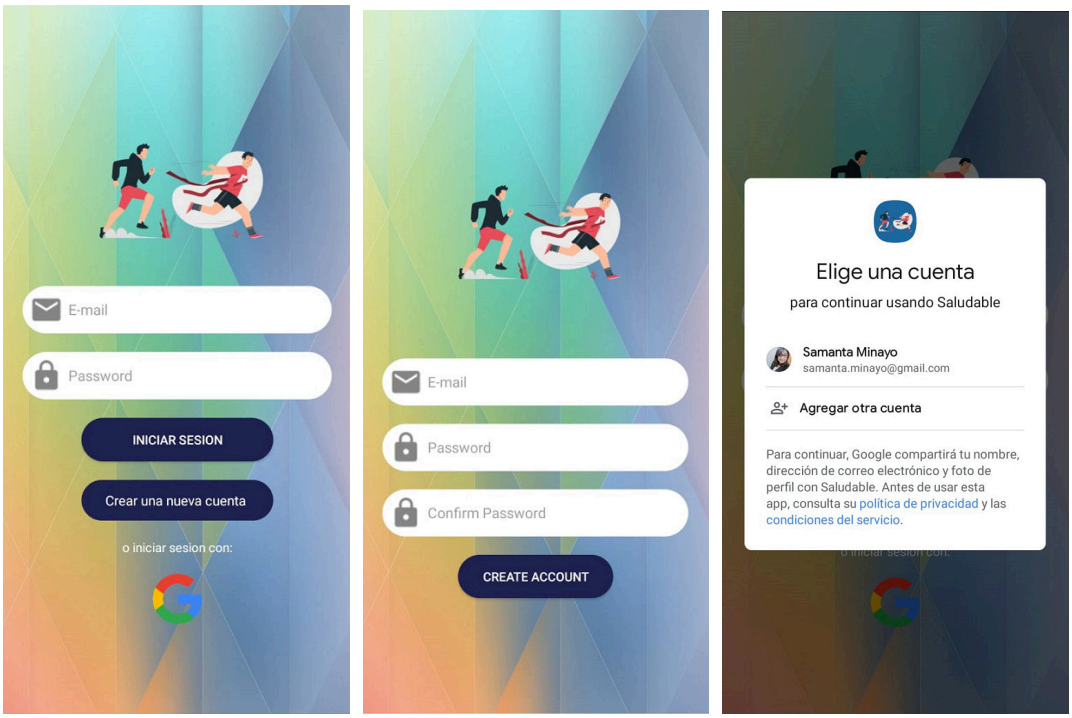
La aplicación móvil administra la información que se ingresa del usuario en el prototipo de plataforma. Las pruebas de monitoreo y visualización de resultados se las realizarán en la sección pruebas de funcionalidad de monitoreo del prototipo.

3.1.2.1. CREACIÓN Y AUTENTICACIÓN DE USUARIOS

En esta prueba se verificó que un usuario de la aplicación móvil pueda crear una cuenta con los proveedores de autenticación correo electrónico/contraseña y Google. Además, se verificó que puedan autenticarse una vez creada la cuenta en cualquiera de estos dos proveedores.

Primero al acceder a la aplicación móvil se mostrará la pantalla de autenticación mostrada en la figura 3.10(a), desde esta pantalla se crearon cuentas de dos maneras, la primera es con el botón crear una nueva cuenta el cual dirige al usuario a la pantalla de registro de cuenta con correo electrónico y contraseña mostrada en la figura 3.10(b), la segunda es mediante el proveedor de Google, el cual despliega una pantalla emergente mostrada en la figura 3.10(c) desde la cual se accede a la cuenta de Gmail del dispositivo o se ingresa una nueva cuenta.

Una vez que el usuario a creado una cuenta dentro de la plataforma puede autenticarse desde la pantalla login de la aplicación móvil ingresando correo electrónico y contraseña si creo la cuenta con este proveedor de autenticación o usando el botón de Google que despliega la ventana emergente de autenticación de Gmail.



(a) Pantalla Login (b) Pantalla nuevo usuario (c) Autenticación Google

Figura 3.10: Modulo de autenticación

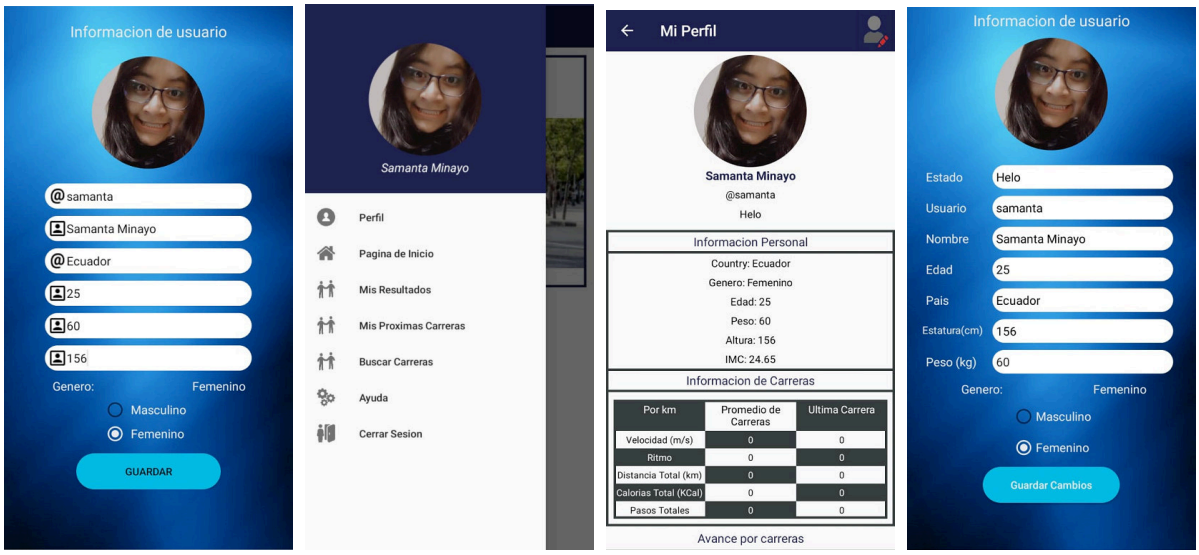
3.1.2.2. REGISTRO Y ACTUALIZACIÓN DE INFORMACIÓN DE USUARIO

Esta prueba tiene como objetivo verificar el registro y actualización de la información del usuario.

Si el usuario se ha autenticado exitosamente por primera vez la aplicación móvil muestra la pantalla de ingreso de información de usuario mostrada en la figura 3.11(a), desde la cual se ingresa la información de usuario incluyendo una imagen de usuario.

Una vez que el usuario ha registrado sus datos correctamente se redirige a la pantalla de inicio donde tendrá acceso al menú principal mostrado en la figura 3.11(b). Además, cada vez que se autentique si ya tiene registrada información de usuario se redirigirá a la pantalla de inicio.

Para la visualización de información de usuario desde la opción perfil del menú principal se dirige a la pantalla mostrada en la figura 3.11 (c), desde está se accede a la pantalla de actualización de información de usuario mostrada en la figura 3.11(d).



(a) Ingresar información de usuario (b) Menú de usuario (c) Perfil de usuario (d) Actualizar información de usuario

Figura 3.11: Registro y actualización de información de usuario

3.1.2.3. MANEJO DE INSCRIPCIONES EN CARRERAS

Durante esta prueba se verificó que los usuarios accedan a las carreras creadas por el administrador y puedan inscribirse o cancelar la inscripción en las mismas.

Para acceder a las carreras el menú principal posee la opción buscar carreras la cual redirigirá a la pantalla mostrada en la figura 3.12, esta pantalla muestra la lista de carreras próximas a realizarse en las que el usuario puede ingresar.

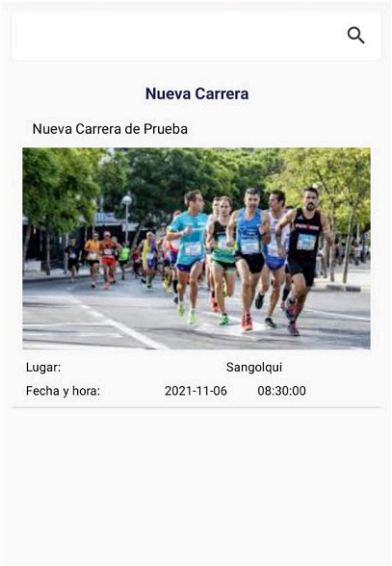
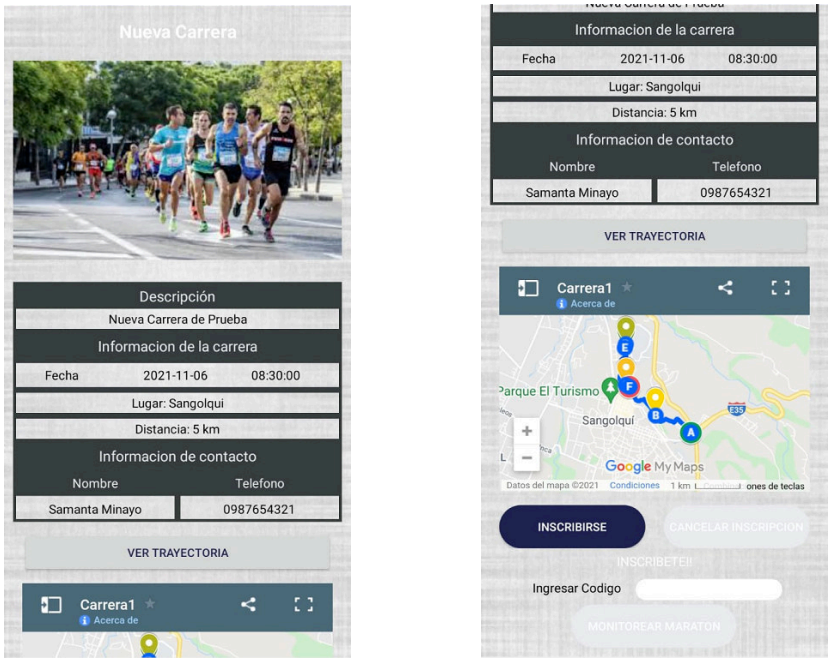


Figura 3.12: Pantalla de búsqueda de carreras

La figura 3.13(a) muestra la información de la carrera y en la figura 3.13(b) se observa la trayectoria de la carrera.



(a) Información de carrera (b) Trayectoria de carrera

Figura 3.13: Pantalla Información de Carrera

Desde la pantalla de la información de carrera el usuario tiene las opciones de inscribirse y cancelar inscripción. La figura 3.14(a) muestra el mensaje de éxito tras la inscripción del usuario en la carrera y la figura 3.14(b) muestra el mensaje de éxito de haber cancelado la inscripción de la carrera. Además, en la figura 3.14 se observa la lógica de funcionamiento de los botones inscribirse y cancelar inscripción dependiendo de la acción que ejecute el usuario.

(a) Realizar Inscripción (b) Cancelar Inscripción

Figura 3.14: Pantalla de manejo de inscripciones

Cada vez que un usuario se inscribe o cancela la inscripción en una carrera esto se ve reflejado dentro de la pantalla de Próximas Carreras a la cual se accede desde el menú principal.

La figura 3.15 muestra un ejemplo de como se visualizará una carrera dentro de la lista de

carreras en las que el usuario se ha inscrito o ha participado, desde estas pantallas se puede acceder a la información de cada carrera y a los resultados de la participación del usuario en la carrera.

Figura 3.15: Ejemplo de visualización de carrera inscrita o carrera realizada

3.1.3. PRUEBAS DE FUNCIONALIDAD DE MONITOREO DE CARRERAS

Las pruebas de monitoreo de carreras del prototipo consisten en probar el envío de información de la actividad física de los atletas desde la aplicación móvil y la monitorización de los atletas participantes desde el módulo de administración de la aplicación web. Además, en estas pruebas se mostrarán los resultados de los atletas y de la carrera en la aplicación web y aplicación móvil.

3.1.3.1. PROCESO DE MONITOREO DE CARRERAS

Una vez configurado el código de la carrera en la opción editar carreras se comparte el código generado con los atletas inscritos. Después el administrador ingresa a la carrera a ser monitorizada mediante la opción monitorear carreras del menú de administración, esta pantalla mostrará a los usuarios que han ingresado al monitoreo de la carrera con el código proporcionado.

La figura 3.16 muestra la pantalla de monitoreo de carrera con los usuarios que han ingresado en ella y sus correspondientes ubicaciones.

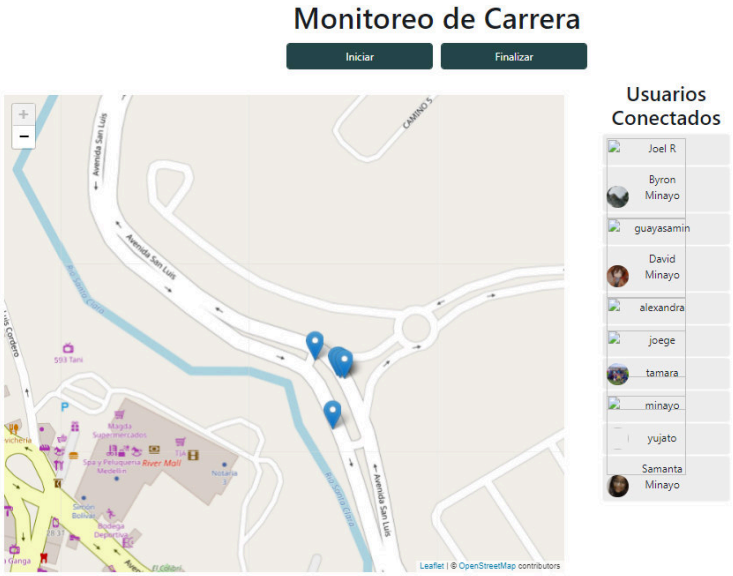


Figura 3.16: Pantalla de monitoreo de trayectoria

Desde la aplicación móvil los usuarios acceden a la pantalla de información de la carrera mostrada en la figura 3.13, esta pantalla permite ingresar el código de la carrera a los usuarios inscritos y dirigirse a la pantalla de monitoreo mostrada en la figura 3.17, esta pantalla mostrara un mapa con la trayectoria recorrida por el atleta en tiempo real y los parámetros de actividad física como velocidad, distancia, cantidad de pasos y el tiempo.

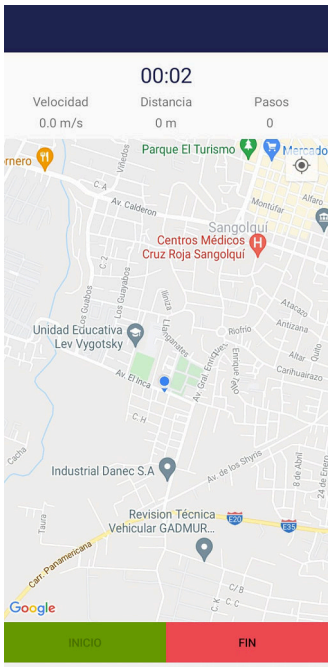


Figura 3.17: Pantalla de monitoreo de trayectoria

Desde la pantalla de monitoreo mostrada en la figura 3.17 el atleta puede dar inicio a la monitorización de la carrera y a su vez el administrador de la carrera habilita el ingreso de datos de los usuarios en la carrera mediante el botón inicio de la pantalla mostrada en la figura 3.16.
Una vez que el atleta a termina la carrera finaliza la transmisión de datos al prototipo de plataforma mediante el botón fin de la pantalla de monitoreo. El administrador también finaliza la carrera desde su interfaz de monitoreo para deshabilitar el ingreso de datos de los usuarios en la carrera.

3.1.3.2. RESULTADOS DE MONITOREO DE CARRERAS

Finalmente se visualizarán los resultados de la monitorización de carreras en la aplicación web y la aplicación móvil.

APLICACIÓN MÓVIL

Una vez que el usuario a finalizado la monitorización de la carrera en la aplicación móvil, la carrera se elimina de carreras inscritas y se muestra en la pantalla de carreras realizadas.

La figura 3.18 muestra las carreras monitorizadas por el usuario.
Desde la lista de carreras realizadas se accede a los resultados de cada carrera. La figura 3.19(a) muestra la pantalla de resultados donde se observa los resultados de la participación

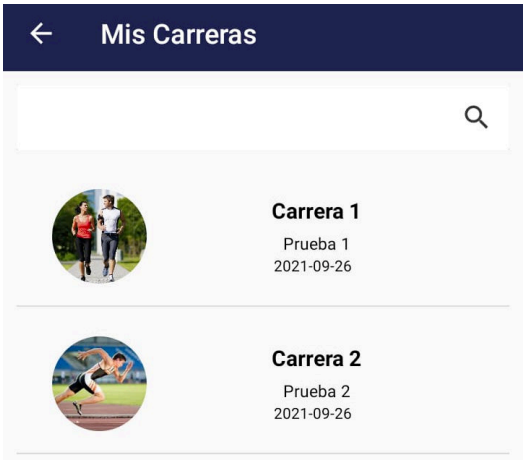
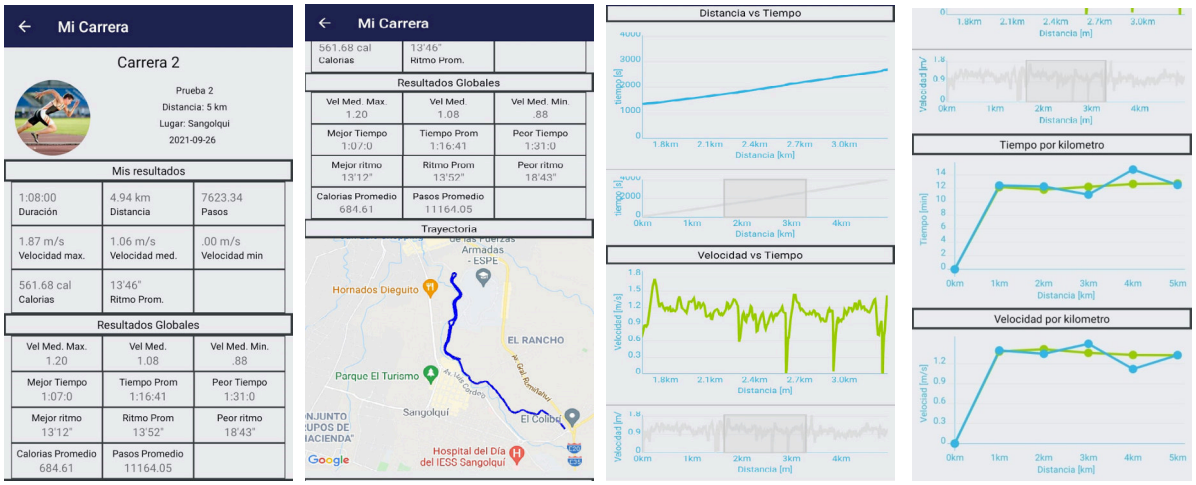


Figura 3.18: Pantalla de carreras realizadas

del atleta en el bloque mis resultados y los resultados promedios de la participación de los atletas en la carrera en el bloque Resultados Globales. La figura 3.19(b) muestra la trayectoria recorrida por el atleta. Además, la figura 3.19(c) muestra las gráficas distancia y velocidad en función del tiempo de la participación del atleta en la carrera. Finalmente,la figura 3.19(d) muestra las gráficas de velocidad y tiempo por kilómetro del atleta y el tiempo y velocidad promedio por kilómetro de los participantes de la carrera.



(a) Resultados de carrera (b) Trayectoria de carrera (c) Gráficas de resultados de carrera (d) Gráficas de resultados por kilómetro

Figura 3.19: Pantalla de resultados de carrera

En el perfil de usuario adicional a su información personal se muestra los resultados de su participación en todas las carreras.

La figura 3.20 muestra el perfil de usuario incluyendo los resultados de las carreras realizadas. El bloque Información de Carreras muestra los resultados promedios de las carreras en las que el usuario a participado incluyendo la información de la última carrera en la que participo. El bloque Avance por carreras muestras gráficas de tiempo, velocidad, calorías

Figura 3.20: Pantalla perfil de usuario con historial de carreras

quemadas y distancia en función de las carreras realizadas por el atleta. Además, cabe recalcar que para el cálculo de calorías quemadas se considero las características físicas del usuario como su peso y la actividad realizada por el usuario.

APLICACIÓN WEB

Los usuarios autenticados en la aplicación web pueden acceder a la información ingresada desde la aplicación móvil.

La figura 3.21 muestra la lista de carreras en las que el usuario a participado dentro del perfil del usuario en la aplicación web.

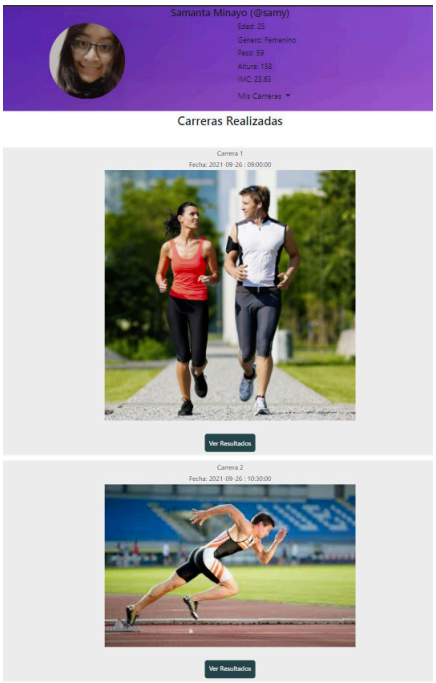


Figura 3.21: Pantalla de carreras realizadas

Al acceder a una carrera en la que el usuario a participado se muestra inicialmente la información de la carrera incluyendo su trayectoria como se observa en la figura 3.22.

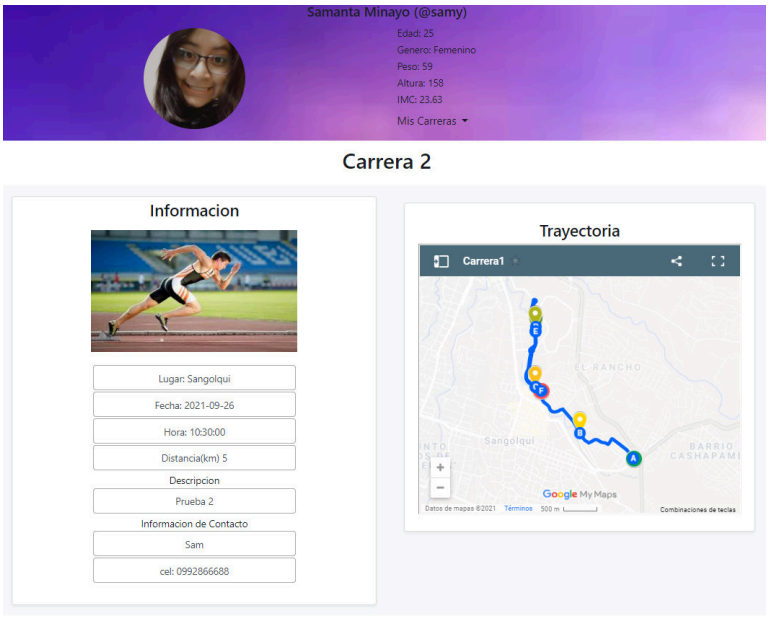


Figura 3.22: Pantalla de información de carrera

La aplicación web muestra los resultados de la participación del atleta en la carrera, además muestra los resultados generales de todos los atletas participantes en la carrera. Finalmente se puede filtrar los resultados en base al género y edad de los participantes, también es posible filtrar los resultados en base a los parámetros tiempo y velocidad empleados por los atletas participantes de la carrera. En la figura 3.23 se observa el bloque de resultados de la carrera.

Figura 3.23: Pantalla de resultados por carrera

La información mostrada en los resultados de la carrera incluye gráficas de edad y género en

función del tiempo empleado para completar la carrera y la velocidad media empleada en la carrera por los participantes. La figura 3.24 muestra las gráficas obtenidas de la participación de los atletas en la carrera.

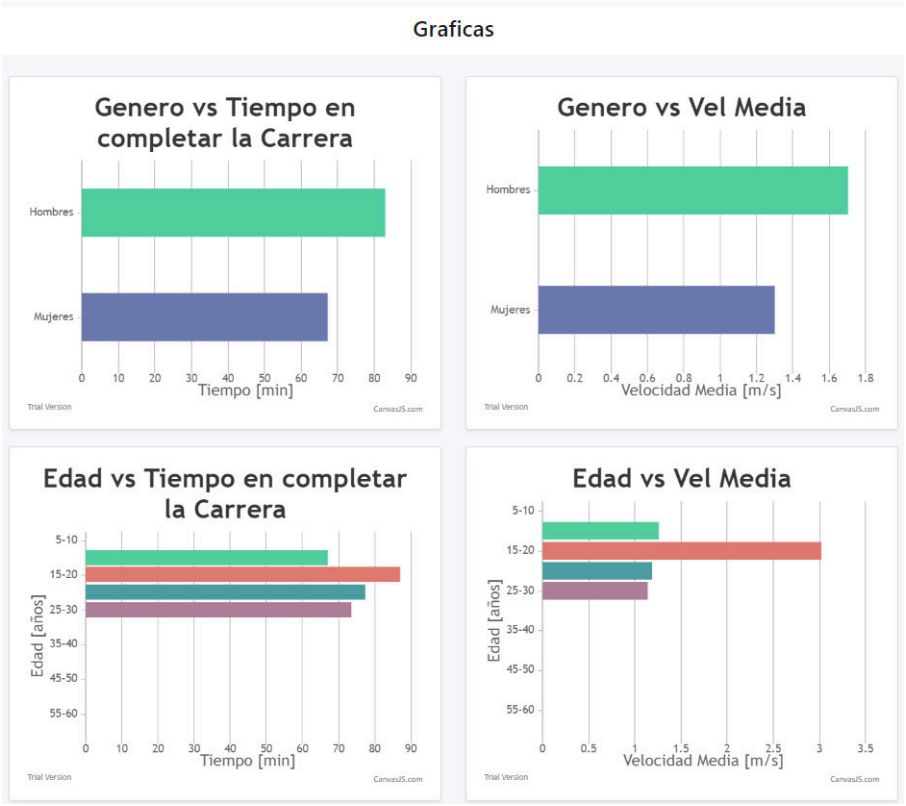


Figura 3.24: Pantalla de gráficas de carreras

Finalmente, la figura 3.25 muestras las gráficas de velocidad y tiempo por kilómetro del atleta y el tiempo y velocidad promedio por kilómetro de los participantes de la carrera.

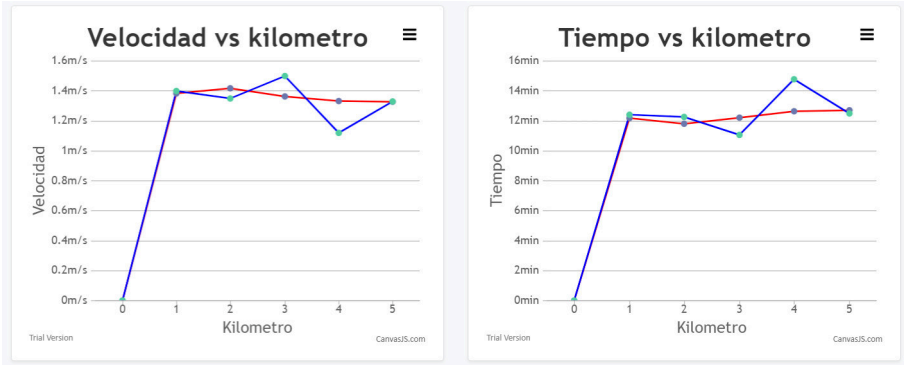


Figura 3.25: Gráficas de resultados por kilometro

La pantalla perfil de usuario de la aplicación web muestra los resultados de todas las carreras en las que el usuario a participado incluyendo gráficas de los parámetros físicos de la participación del atleta en función de la carrera realizada.

La figura 3.26 muestra el bloque de resultados promedios y totales de la participación del

atleta, además muestra los resultados de la última carrera en la que el usuario participo.

Figura 3.26: Pantalla perfil de usuario con historial de carrera

En la figura 3.27 se observa las gráficas de tiempo, velocidad media, calorías quemadas y distancia en función de la carrera realizadas.

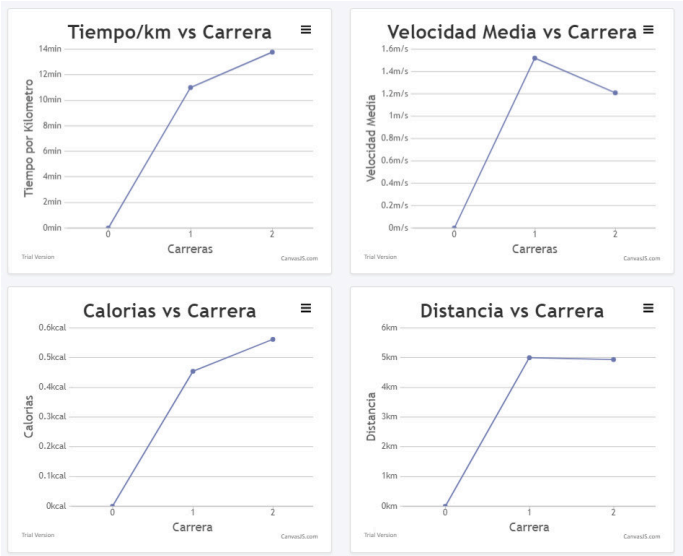


Figura 3.27: Pantalla de monitoreo de trayectoria

3.2. PRUEBAS DE INTEGRACIÓN

En esta sección se realizarán pruebas de integración entre la aplicación web y móvil con los servicios de Firebase. Para realizar estas pruebas, se realizaron acciones desde cualquiera de las dos aplicaciones y los resultados se ven reflejados desde la otra aplicación, así como en los servicios de Firebase.

Autenticación en Firebase

Inicialmente se mostrará el servicio de Autenticación de Firebase. La figura 3.28 muestra la lista de usuarios registrados en Firebase, los usuarios subrayados en amarillo corresponden

al usuario de administración y a los usuarios atletas registrados en el prototipo de plataforma con los diferentes proveedores de autenticación proporcionados por Firebase.

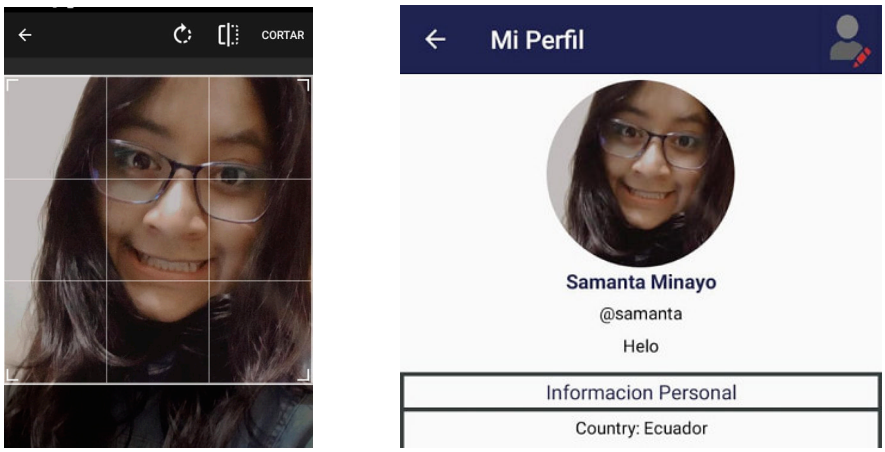
Figura 3.28: Usuarios registrados en Firebase Authentication

Dentro de las pruebas funcionales se realizaron pruebas para cada aplicación del prototipo de plataforma, a continuación, se realizarán dos ejemplos para verificar la integración entre las aplicaciones web y móvil con los servicios de Realtime Database y Storage de Firebase.

Almacenamiento en Firebase

Para probar el servicio de Storage se ingresará la foto de perfil desde la aplicación móvil y se verificará su correcto ingreso desde la consola de Firebase, finalmente se accederá a la aplicación web para visualizar la actualización de foto de perfil del usuario.

La figura 3.29 muestra la actualización de foto de perfil desde la aplicación móvil. La figura 3.29(a) muestra el recorte de la imagen una vez que se ha seleccionado desde la galería o se ha tomado una foto desde la cámara. La figura 3.29(b) muestra la imagen actualizada en el perfil de usuario.



(a) Seleccionar y Recortar imagen de perfil (b) Visualización de imagen de perfil

Figura 3.29: Actualización de foto de perfil desde aplicación móvil

La figura 3.30 muestra el almacenamiento de la imagen en la consola de Firebase en el servicio de Storage.

Figura 3.30: Almacenamiento de la imagen en Storage de Firebase

Finalmente, la figura 3.31 muestra el perfil de usuario desde la aplicación web donde se visualiza la imagen de perfil actualizada.

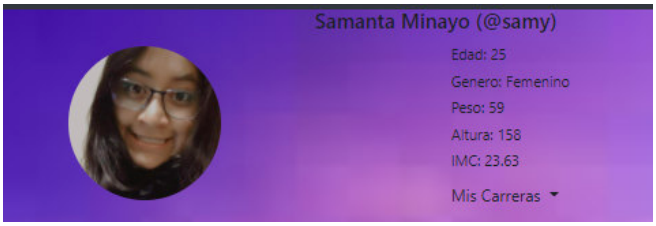


Figura 3.31: Visualización de foto de perfil en aplicación web

Base de datos en Firebase

Para probar el servicio de Realtime Database se creará una nueva carrera desde el módulo de administración de la aplicación web y se verificará el ingreso de la información de la carrera en la base de datos de Firebase y el acceso a esta información desde la aplicación móvil.

La figura 3.32 muestra el nodo de la base de datos de Firebase donde se ha registrado la información de la nueva carrera creada por el administrador.
La figura 3.33 muestra la nueva carrera creada por el administrador desde la aplicación web en la lista de próximas carreras.
Finalmente, en la figura 3.34 se observa la lista de carreras incluyendo la nueva carrera creada por el administrador desde la pantalla buscar carreras de la aplicación móvil.

3.3. ENTREVISTAS

Las pruebas funcionales y de integración mostraron que la aplicación web obtiene y muestra correctamente la información registrada desde la aplicación móvil por lo que esta etapa de entrevistas se la realizará a los usuarios de la aplicación móvil. Además, el prototipo de plataforma es manejado por un solo administrador por lo que los errores registrados en el módulo de administración se los obtuvieron mediante las pruebas funcionales y de integración de este capítulo.

La dinámica de la entrevista consistió en verificar la funcionalidad de la aplicación móvil en los diferentes dispositivos android de los usuarios, estos usuarios corresponden al perfil

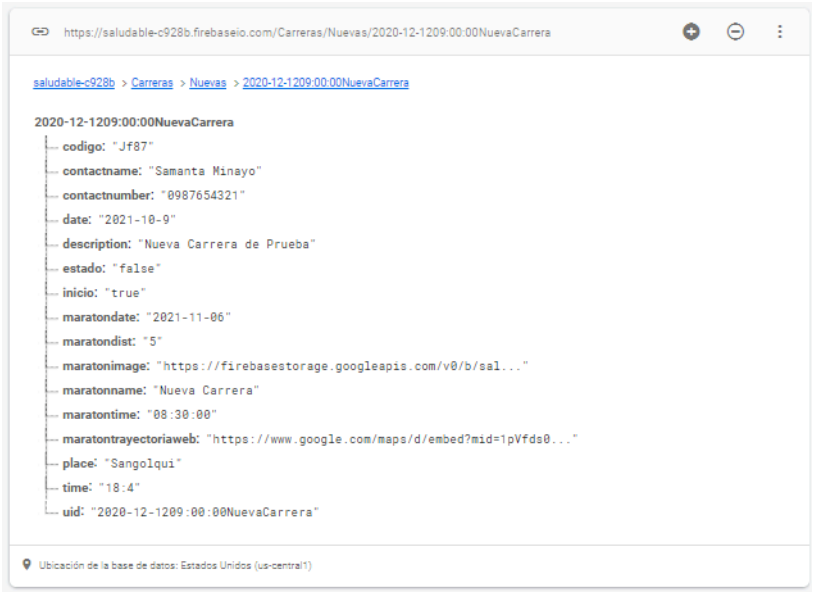


Figura 3.32: Almacenamiento de información de carrera en Realtime Database de Firebase

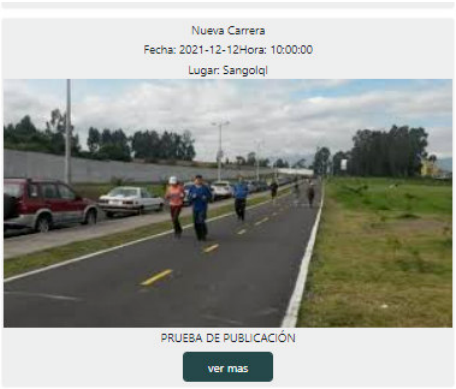


Figura 3.33: Visualización de nueva carrera en la aplicación web

atleta y son los diez usuarios con los cuales se realizo la entrevista inicial en el proceso de diseño del prototipo de plataforma. La entrevista fue diseñada principalmente con preguntas cerradas con las opciones si o no, en caso de detectar problemas de funcionalidad se diseñaron preguntas adicionales con la finalidad de obtener información acerca del error.

El modelo de la entrevista se encuentra en el ANEXO E.

Mediante las entrevistas realizadas a diez usuarios de la aplicación móvil se identificaron los errores existentes en la aplicación móvil con el propósito de corregirlos.

A continuación, en la figura 3.35 se presentan los resultados a las preguntadas cerradas de la entrevista.

La pregunta uno: ¿Existió algún problema al registrarse o iniciar sesión? sirve para comprobar si el usuario pudo registrarse en el prototipo de plataforma, donde el 90 % tuvo problemas al registrarse en el prototipo de plataforma y un 10 % se registró exitosamente. De esta pregunta surge una pregunta adicional ¿Qué proveedor de autenticación y registro fue usada por los usuarios? de la cual se encontró que los usuarios con problemas se registraron y

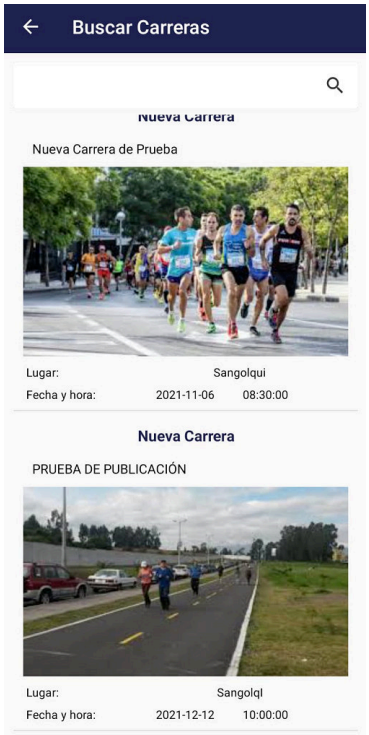


Figura 3.34: Visualización de nueva carrera en la aplicación móvil

autenticaron mediante el proveedor de Google-Gmail.

La pregunta dos: ¿La aplicación te permitió agregar la foto de perfil en el registro? permite comprobar si el usuario pudo agregar su foto de perfil. El resultado muestra que el 60 % de los entrevistados pudo agregar su foto de perfil.

En la pregunta tres: ¿Se mostró la información de carreras en la pantalla buscar carreras? Los resultados muestran que el 100 % de los entrevistados tuvo acceso a las carreras existentes en el prototipo de plataforma.

En la pregunta cuatro: ¿Se pudo registrar y eliminar inscripciones en las carreras? Los resultados muestran que el 100 % de los entrevistados pudo realizar las acciones de ingresar y eliminar inscripción sobre las carreras.

La pregunta cinco: ¿Se mostró la información correspondiente a las inscripciones en la opción mis próximas carreras? permite comprobar que la información de las carreras se organice correctamente en función de la interacción con el usuario de la aplicación móvil. El resultado muestra que el 100 % de los entrevistados visualizó la información de carreras inscritas en la pantalla mis próximas carreras.

En la pregunta seis: ¿Se registraron problemas al ingresar en la pantalla de monitoreo mediante el código de la carrera? Los resultados muestran que el 100 % de los entrevistados no tuvo problemas al ingresar en la pantalla de monitoreo de la carrera.

La pregunta siete: ¿Recibió la notificación de nueva carrera disponible? permite comprobar si se está enviando la notificación desde la aplicación web o si la aplicación móvil esta con-

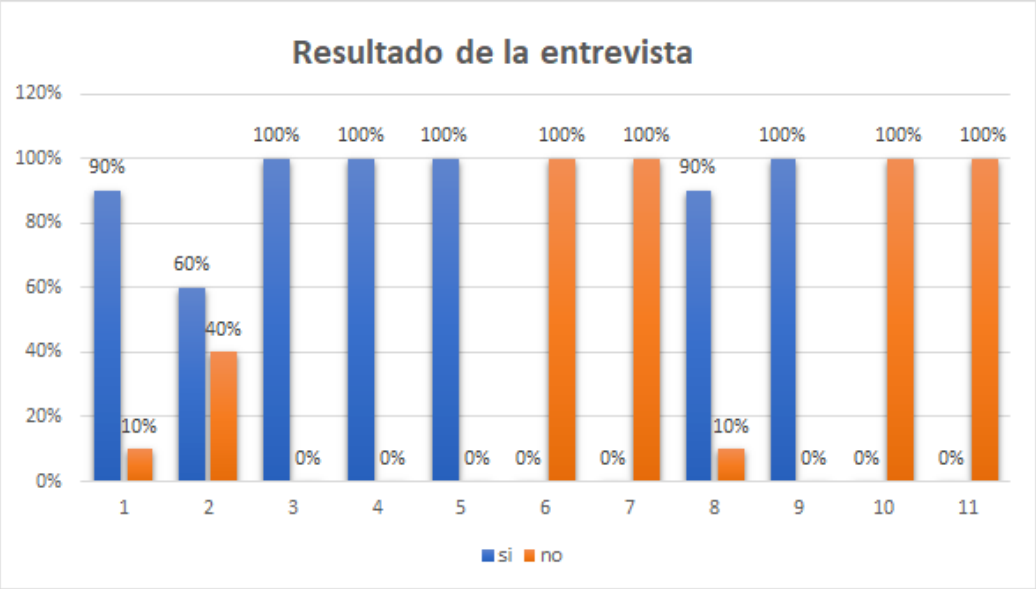


Figura 3.35: Visualización de nueva carrera en la aplicación móvil

figurada correctamente para recibir notificaciones. El 100 % de los entrevistados no recibió la notificación.

La pregunta ocho: ¿Se graficó correctamente la ruta recorrida por el atleta al monitorizar la carrera? permite comprobar la precisión del gps al obtener la ubicación. El 90 % de los entrevistados confirmó que la gráfica de la trayectoria fue correcta.

La pregunta nueve: ¿Se registro problemas al finalizar la monitorización de la carrera? permite comprobar si se envía correctamente la información final de la participación del atleta en la carrera. El 100 % de los entrevistados manifestó que se generó un error al finalizar la carrera.

En la pregunta diez: ¿Se actualizó la información de carreras inscritas y carreras realizadas al finalizar la monitorización de una carrera? Los resultados muestran que el 100 % de los usuarios registraron errores en la actualización de información de carreras.

En la pregunta once: ¿Se mostraron los resultados de la carrera realizada? Los resultados muestran que el 100 % de los usuarios no tuvo acceso a los resultados de la carrera.

Finalmente, en el análisis de la entrevista se observó que existen problemas generales en la aplicación móvil para los usuarios a excepción de la pregunta ocho donde un solo usuario registro problemas, por lo que a ese usuario se le realizó la pregunta adicional ¿Qué modelo de dispositivo android usa?

Esta entrevista se ejecutó tras la realización de la monitorización de la primera carrera. Una vez corregido los errores se realizó la prueba con una segunda carrera para verificar la corrección de errores.

3.4. CORRECCIÓN DE ERRORES

En los resultados de las entrevistas, el proceso de desarrollo y las pruebas funcionales y de integración del prototipo de plataforma se encontraron múltiples errores, los cuales fueron corregidos a lo largo del desarrollo de este trabajo. La tabla 3.1 muestra la lista de errores que se presentaron y la solución empleada para cada uno de ellos.

Tabla 3.1: Corrección de errores

Error	Solución
Disposición irregular de componentes en las pantallas de la aplicación móvil	Manejar componentes Linear Layout para organizar la distribución de los componentes en filas y columnas. Además manejar el atributo android:layout_below de cada componente para fijar su ubicación.
No se puede autenticar con el proveedor de Google-Gmail	Exportar clave SHA1 desde android studio e ingresarla en el proyecto correspondiente al prototipo de plataforma desde la consola de Firebase
No se permite cargar y recortar imagen de perfil de usuario	Importar librería Android Image Cropper
Cierre de la aplicación móvil por atributos vacíos obtenidos desde firebase	Verificar la existencia del atributo del nodo en la base de datos antes de cargarlo en la interfaz de la aplicación móvil.
Error de actualización en tiempo real de información de carreras en la aplicación móvil	Actualizar la lectura de información de la base datos ante cualquier cambio dentro del nodo que contiene la información de próximas carreras.
Crear trayectoria desde aplicación web	Implementar ingreso de trayectoria desde la aplicación MyMaps mediante la opción compartir mapa en mi sitio web.
Ingreso de información de monitorización de los atletas fuera del tiempo de ejecución de la carrera.	Se habilitó el ingreso de la información de monitorización una vez que el administrador de inicio a la carrera hasta que el administrador finalice la carrera, la información fuera de estos parámetros será ignorada.
No se actualiza ubicación actual del usuario en la pantalla de monitoreo debido a falta de permisos	Además de solicitar permisos en la pantalla de login al instalar la aplicación verificar que se mantengan activos en las pantallas que lo requieran.
No se envía notificación de nueva carrera desde la aplicación web	Esto se produjo por un bloqueo entre la alerta de información ingresada correctamente y el envío de la notificación por lo que se configuró que se muestre la alerta tras el envío de la notificación.
Duplicación de marcadores en la pantalla de monitorización de atletas del módulo de administración.	Hacer doble lectura en la base de datos para verificar cambios, vaciar el array de marcadores y volver a cargarlos desde la base de datos.
Pérdida de información en la monitorización de la actividad física desde la aplicación móvil al suspender la pantalla del dispositivo móvil	Esto se debía a la prioridad de la notificación obligatoria para el uso de servicios en background, por lo que se configuró la notificación en PRIORITY_HIGH
No se ingresa la información del atleta al finalizar la monitorización de la carrera desde la aplicación móvil.	Transformar los datos de la carrera y los datos de resultados finales al formato correcto.
No se grafica correctamente la trayectoria	No se realizó ninguna acción debido a que el error presentado se generó únicamente en un usuario con una versión antigua del sistema operativo de android.
No se elimina la carrera de la lista de carreras inscritas una vez monitorizada la carrera.	Además de eliminar el registro en Realtime Database eliminarlo de la base de datos local del dispositivo móvil.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- El objetivo del presente Trabajo de Titulación fue desarrollar un prototipo de plataforma para la monitorización de la actividad física de atletas en una carrera. El prototipo está conformado por: aplicación móvil y aplicación web. El desarrollo se basó en una arquitectura monolítica, por lo que la aplicación móvil y aplicación web tienen integrados todos los aspectos funcionales dentro de cada programa correspondiente a las aplicaciones.

- La plataforma de Firebase proporciona servicios alojados en la nube con la facilidad de acceder a ellos mediante librerías, por lo tanto, el desarrollador solo se preocupa en consumir estos servicios. Esto ha permitido manejar las diferentes funcionalidades del prototipo de plataforma como: almacenar imágenes de carreras y perfil de usuario en el servicio Storage, manejar notificaciones desde la aplicación web hacia la aplicación móvil a través del servicio Cloud Messaging, lectura y escritura de datos en tiempo real con el servicio Realtime Database y finalmente permitió el alojamiento de la aplicación web en el servicio Hosting.

- Usar una infraestructura BaaS mediante Firebase evita que el desarrollador se encargue de la configuración, mantenimiento y aprovisionamiento de servidores, ya que para el caso de Firebase es Google quien se encarga del entorno de administración de los servidores.

- Firebase permite que el prototipo de plataforma sea escalable y de bajo costo, en caso de que la información de carreras y atletas crezca, esto debido a que Firebase ofrece varios planes de pago entre ellos un plan de consumo gratuito y planes de pago por consumo en todos sus servicios, además ofrece un límite de consumo gratuito por día independientemente del plan contratado.

- Firebase genera un dominio gratuito para el hosting, lo cual reduce costos ya que se evita la contratación de un dominio para la aplicación web, facilitando el despliegue de la aplicación web.

- Debido a los requerimientos de los usuarios, de mantener la información Offline se implementó la base de datos SqlLite dentro de la aplicación móvil, la cual permite tener una copia de la información sin conexión en la aplicación móvil.

- El desarrollo del servicio de monitoreo en la aplicación móvil se basó en las diferentes versiones del sistema operativo Android ya que dependiendo de esto las políticas de consumo de energía del dispositivo han ido avanzando y ha generado que la creación de aplicaciones que se ejecuten en segundo plano requieran de una programación más compleja en cuanto a solicitud de permisos, envió de notificaciones y manejo de servicios en segundo plano.

- Según las versiones del sistema operativo de android los dispositivos móviles manejan diferentes formas de precisión al obtener la ubicación, por lo que en versiones más actuales la precisión de la ubicación GPS obtenida desde la aplicación móvil es mejor, ya que el dispositivo usa otras herramientas para obtener la ubicación como Bluetooth.
- Una vez iniciada la sesión del usuario con Firebase Authentication desde la aplicación web, es la plataforma de firebase quien guarda el estado de sesión y desde la aplicación web se accede a este mediante una consulta, lo cual evita el uso de almacenamiento local en el navegador.
- React JS permite dividir los componentes de la aplicación web, lo cual agilizó el desarrollo, ya que permite reutilizar estas divisiones de componentes para construir nuevos más complejos y evitar extender la codificación dentro de la aplicación web.
- En base a la información recolectada en las entrevistas finales se encontraron errores funcionales dentro de las aplicaciones móvil y web los cuales fueron corregidos. Además, se encontraron errores aislados que dependieron de la versión del sistema operativo de android por lo que no se realizaron correcciones para estos errores.

4.2. RECOMENDACIONES

- El prototipo implementa dos proveedores de autenticación. Para desarrollos futuros se recomienda que se implemente otros proveedores de autenticación como Facebook y Twitter dentro del prototipo de plataforma.
- La aplicación móvil se encuentra únicamente desarrollada para el sistema operativo Android, por lo que se recomienda desarrollar una aplicación móvil para el sistema operativo iOS.
- Manejar versionamiento del código fuente de las aplicaciones web y móvil fue de suma importancia en el desarrollo de este trabajo, ya que permite organizar el código fuente de mejor manera y evitar perdidas de información. Además, permite acceder a las diferentes versiones del código facilitando el análisis y solución de errores en el desarrollo del presente trabajo.
- Las aplicaciones de este prototipo de plataforma están desarrolladas en base a una arquitectura monolítica, ya que son aplicaciones autosuficientes que incorporan dentro del código todas las funcionalidades y dependencias, lo cual dificulta futuras actualizaciones de la plataforma, por lo que se recomienda migrarla a una arquitectura basada en micro servicios.
- En la aplicación web se podría agregar la funcionalidad para manejar múltiples administradores en el prototipo de plataforma.
- En la aplicación móvil se podría agregar un módulo donde los usuarios puedan almacenar información de la actividad física sin necesidad de participar en una carrera.

- La trayectoria de las carreras dentro del prototipo de plataforma se manejan mediante la aplicación MyMaps de Google por lo que se recomienda implementar un módulo propio de la aplicación web para la creación de trayectorias.
- La aplicación móvil se maneja de forma independiente. Para desarrollos futuros se recomienda considerar si el dispositivo maneja una aplicación propia para el monitoreo de actividad física y vincularlos para obtener información más precisa de la actividad realizada en una carrera.
- Para desarrollos futuros se recomienda vincular la aplicación a dispositivos externos como pulseras de actividad deportiva para obtener información adicional que no puede ser recolectada mediante el dispositivo móvil.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] SCRUMIZATE, “Plantilla para las historias de usuario,” 2022. [Online]. Disponible en: http://scrumizate.com/post/58/plantilla-para-las-historias-de-usuario

[2] E. Encalada, “Los atletas tienen 96 carreras para escoger en quito | el comercio,” 2016. [Online]. Disponible en: https://www.elcomercio.com/deportes/ecuador-atletas-carreras-agenda-escoger.html

[3] Bolavip, “Tips: cómo llevar el celular para correr,” 2018. [Online]. Disponible en: https://bolavip.com/otros/running-tips-gadgets-porta-celular-para-correr-cintos-brazalete-20180607-0019.html

[4] M. Deportivo, “5 aplicaciones para monitorizar tu actividad deportiva,” 2017. [Online]. Disponible en: https://www.mundodeportivo.com/elotromundo/tecnologia/20170822/43741541788/cinco-aplicaciones-monitorizar-actividad-deportiva.html

[5] M. Cordova, “La ruta de las iglesias estrenará nuevo chip para la edición de este 2018,” 2018. [Online]. Disponible en: https://www.elcomercio.com/deportes/rutadelasiglesias-estrenara-nuevo-chip-carrera.html

[6] R. Peñafiel, “La nueva piel de la quito-Últimas noticias 15k, con tecnología de punta,” 2019. [Online]. Disponible en: https://www.elcomercio.com/deportes/atletismo-camiseta-quito-ultimas-noticias.html

[7] Y. A. Rendón, “Bases de datos relacionales vs. no relacionales,” 2019. [Online]. Disponible en: https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales

[8] Firebase, “Documentación,” 2021. [Online]. Disponible en: https://firebase.google.com/docs?hl=es-419

[9] React, “Empezando,” 2021. [Online]. Disponible en: https://es.reactjs.org/docs/getting-started.html

[10] M. W. Docs, “Html: Lenguaje de etiquetas de hipertexto.” 2021. [Online]. Disponible en: https://developer.mozilla.org/es/docs/Web/HTML

[11] M. Docs, “Css básico,” 2021. [Online]. Disponible en: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics

[12] A. Moviles, “Android sdk: ¿qué es y cómo funciona?” 2020. [Online]. Disponible en: https://appmoviles.net/android-sdk-que-es-y-como-funciona

[13] A. S. developers, “Actualiza el ide y las herramientas del sdk,” 2021. [Online]. Disponible en: https://developer.android.com/studio/intro/update#sdk-manager

[14] nodeJS, “Ecmascript 2015 (es6) and beyond,” 2019. [Online]. Disponible en: https://nodejs.org/es/docs/es6/

[15] Y. Muradas, “¿qué es npm?” 2019. [Online]. Disponible en: https://openwebinars.net/blog/que-es-node-package-manager/

[16] A. S. developers, “Introducción a android studio,” 2021. [Online]. Disponible en: https://developer.android.com/studio/intro?hl=es-419

[17] Microsoft, “V. s. code, «getting started»,” 2021. [Online]. Disponible en: https://code.visualstudio.com/docs

[18] Kinsta, “Git vs github cual es la diferencia y como empezar,” 2020. [Online]. Disponible en: https://kinsta.com/es/base-de-conocimiento/git-vs-github/

[19] Kanbantool, “Kanban es parte de la metodología lean,” 2021. [Online]. Disponible en: https://kanbantool.com/es/guia-kanban/kanban-es-parte-de-la-metodologia-lean

[20] cpmformiación, “¿quÉ es lean manufacturing?” 2018. [Online]. Disponible en: https://cpmformaciongmp.com/que-es-lean-manufacturing/

[21] Kanbanize, “¿qué es una tarjeta kanban?” 2019. [Online]. Disponible en: https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tarjeta-kanban

[22] Kanbaniz, “Qué es un tablero kanban: Fundamentos,” 2019. [Online]. Disponible en: https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban

[23] IBM, “Publicación/suscripción,” 2021. [Online]. Disponible en: https://www.ibm.com/docs/es/integration-bus/10.0?topic=ssmkhh-10-0-0-com-ibm-etools-mft-doc-aq01120--htm

[24] A. M. S. y J. M. Guirao Miras, “Frecuencia y distintividad en el uso lingüístico: casos tomados de la,» universidad autónoma de madrid, universidad de granada,” 2019. [Online]. Disponible en: https://www.um.es/lacell/aelinco/contenido/pdf/14.pdf

[25] Microsoft, “Crear un diagrama de actividad uml,” 2022. [Online]. Disponible en: https://support.microsoft.com/es-es/office/crear-un-diagrama-de-actividad-uml-19745dae-2872-4455-a906-13b736f01685

5. ANEXOS

Los siguientes anexos se encuentran en formato digital:

ANEXO A. Modelo de entrevista inicial.

ANEXO B. Proyecto de aplicación móvil

ANEXO C. Proyecto de aplicación Web

ANEXO D. Manual de usuario de Administrador

ANEXO E. Modelo de entrevista final