



UNIVERSIDADE DA CORUÑA

PROGRAMACIÓN DE SISTEMAS 23/24 Q2



Aplicación de Gestor de viajes en Android: TripTracks

Autores: Nicolás Rivas Rodriguez

Xoel González Pereira

Fecha: *A Coruña, 15 Marzo 2024*

Índice

Capítulos	Página
1. Introducción	1
1.1. Objetivos	1
1.2. Motivación	1
1.3. Trabajo relacionado	1
2. Análisis de requisitos	2
2.1. Funcionalidades	2
2.2. Dependencias entre las funcionalidades implementadas	2
2.3. Prioridades	3
3. Planificación inicial	4
3.1. Iteraciones	4
3.2. Responsabilidades	4
3.3. Hitos	5
3.4. Incidencias	5
4. Diseño	6
4.1. Arquitectura	6
4.2. Diseño e implementación de la 3a iteración	6
4.3. Persistencia	6
4.4. Vista	8
4.5. Comunicaciones	8
4.6. Sensores	8
4.7. Trabajo en Background	8
4.8. Pruebas y resultados finales	8
4.9. Componentes Android	10
4.10. Librerías	11
5. Conclusiones y Trabajo Futuro	12
6. Bibliografía	13

Cuadro 1: Tabla de versiones.

Versión	Fecha	Autor
1	15/03/2024	Xoel - Nicolás
2	19/04/2024	Xoel - Nicolás
3	30/05/2024	Xoel - Nicolás

1. Introducción

Este proyecto consiste en la creación de una app móvil usando Android Studio, que es un gestor de viajes. Las funcionalidades básicas de la app permitirían al usuario crear, modificar y compartir itinerarios de viaje con otros usuarios. Dentro de cada itinerario, se podrán planificar las actividades que se desean realizar, así como su visualización en un calendario integrado. Dentro de este calendario, el usuario podrá marcar como realizadas las actividades aún no realizadas para ese itinerario, siguiendo un buen progreso de viaje. A nivel de itinerario también habrá una sección que le permitirá al usuario hacer una gestión de documentos, teniendo así de forma organizada los documentos necesarios para ese viaje, esto sería independiente para cada usuario. Una vez haya pasado la fecha de finalización del viaje, el itinerario se marcaría como finalizado, por lo que no se podrían seguir añadiendo actividades ni destinos, quedando guardado y sirviendo solo para visualización.

1.1. Objetivos

Durante el desarrollo del proyecto, se tendrán en cuenta tanto objetivos principales como secundarios. El objetivo principal de la aplicación será el implementar una aplicación que será un gestor de viajes, que ofrezca al usuario la posibilidad de crear itinerarios, gestionar documentos importantes para el viaje y poder compartir con otros usuarios los itinerarios creados, además de poder gestionar dentro de cada itinerario los archivos multimedia, sea compartido o no. Como objetivos secundarios se buscaría el implementar una función de recomendación de nuevos destinos que el usuario podría visitar basado en los itinerarios realizados por el usuario, así como el poder extender su implementación de una forma más global, permitiendo que se convierta en una red social en la que los usuarios podrían poner calificaciones o indicar qué itinerarios les gustan.

1.2. Motivación

La idea nace debido al interés general por viajar, sumado a la idea de poder compartir las experiencias vividas con otras personas, de la misma forma que se hace en las redes sociales. Al fin y al cabo, cualquier usuario busca la comodidad por encima de todo, por lo que juntar estas dos ideas en una misma aplicación, facilitaría los dos aspectos, creando un ecosistema similar al de cualquier red social, pero con el añadido de poder gestionar todos los aspectos necesarios para realizar un viaje, ya sea en solitario o en grupo.

1.3. Trabajo relacionado

Se han encontrado diversas aplicaciones similares al Gestor de Viajes TripTracks, entre las que se pueden destacar TripIt o TripCase. Estas aplicaciones se caracterizan

por también permitir al usuario crear, visualizar y gestionar los itinerarios realizados para un viaje, además de que estas aplicaciones también ofrecen al usuario la posibilidad de poder compartir con otros usuarios el itinerario creado. La principal diferencia encontrada con el gestor de viajes a implementar es que estas dos aplicaciones automatizan el proceso de la creación del itinerario al recopilar detalles de los vuelos, hoteles y otros servicios de forma automática, mientras que TripTracks está enfocado a que el usuario pueda crear itinerarios personalizados desde cero, permitiendo agregar actividades específicas y gestionar documentos de forma manual.

2. Análisis de requisitos

2.1. Funcionalidades

El proyecto se basa en 4 funcionalidades principales:

- 1-Creación y Organización de Itinerarios:
El usuario podrá crear y modificar itinerarios y marcarlos en un mapa.
- 2-Calendario de Actividades y Reservas:
El usuario podrá acceder a un calendario donde visualizar las actividades y las reservas de un itinerario, además, podrá marcarlas como realizadas una vez se completen.
- 3-Gestión de Documentos:
El usuario podrá guardar y acceder a todos los documentos necesarios para completar un itinerario desde la propia aplicación.
- 4-Gestión de Multimedia: Los usuarios podrán compartir diferentes archivos multimedia, como imágenes o vídeos, con el resto de personas que están realizando el mismo itinerario.

2.2. Dependencias entre las funcionalidades implementadas

En la primera iteración, se ha mantenido la misma prioridad a la hora de implementar las funcionalidades, ya que durante el desarrollo de esta se han seguido los puntos a implementar en el orden que fueron marcados. De momento, han sido implementadas las funcionalidades de creación y organización de itinerarios, ya que el usuario una vez inicie sesión podrá tener acceso a los itinerarios que tiene creados, o poder crear más. Además, el usuario tiene la posibilidad de entrar en los detalles del itinerario seleccionado, pudiendo modificar tanto el título del itinerario como el sitio en el que se planea hacer el viaje. También se ha implementado en esta iteración la posibilidad de poder ver en un mapa el destino en el que se va a realizar el viaje, y también se ha añadido la segunda funcionalidad propuesta, que es la de poder gestionar un calendario de actividades para cada itinerario, en la que aparecerá un calendario con los días desde la fecha de inicio del viaje hasta el fin de este, y en

este rango de días se podrán marcar actividades, pudiendo elegir entre 3 categorías: exploración, gastronomía y entretenimiento. A cada actividad se le podrá poner un nombre, y estas actividades se guardarán para el itinerario asociado. Cabe destacar que también existe la posibilidad de en el propio calendario elegir nuevamente las fechas de inicio y fin de viaje, pulsando el botón editar en los detalles del itinerario. Para la segunda iteración, también se ha implementado la posibilidad de compartir a otras cuentas registradas los itinerarios, y en el caso de que el itinerario creado también tenga actividades asociadas, el usuario al que se le comparta también podrá ver esas actividades marcadas. Finalmente, en la tercera iteración, se añadieron dos nuevas funcionalidades. Ahora el usuario puede guardar y visualizar documentos y subir y compartir imágenes asociadas a itinerarios.

Por lo tanto, en cuanto a las dependencias entre funcionalidades se pueden identificar las siguientes:

- 1- Creación de cuentas
Para poder tener acceso a la app es necesario que el usuario se registre en la aplicación con un usuario y contraseña válidos.
- 1- Creación de itinerarios:
Para poder crear itinerarios, primero el usuario se tiene que haber registrado en la aplicación, indicando un correo electrónico y una contraseña válida. Después, podrá iniciar sesión y proceder a la creación.
- 2-Edición de itinerarios,visualización en el mapa y calendario de actividades:
Para poder ver los detalles, editar los itinerarios,visualizar el destino en el mapa y gestionar actividades el itinerario primero tiene que haber sido creado.
- 3-Compartir itinerarios:
Para poder compartir itinerarios, es obligatorio que el itinerario que se quiere compartir haya sido creado, pero además que el destinatario que lo vaya a recibir sea una cuenta registrada en la aplicación.
- 4-Guardar documentos:
Para poder guardar documentos, es necesario que el usuario inicie sesión con una cuenta existente o cree una nueva.
- 5-Compartir imágenes:
Para poder compartir imágenes, es necesario que exista un itinerario previo al cual asociarlas.

2.3. Prioridades

Debido al propósito principal de la aplicación, las funcionalidades más prioritarias son las que están directamente relacionadas con la gestión de viajes, es decir, crear y organizar itinerarios, crear un calendario de actividades y gestionar los documentos relacionados con el viaje. La funcionalidad con menos prioridad sería la Gestión de multimedia.

3. Planificación inicial

3.1. Iteraciones

Para la planificación inicial se ha decidido que se establecerán tres iteraciones:

- 1-Primera iteración:
Implementación de la funcionalidad de creación de itinerarios, incluyendo la adición de destinos y actividades, además de establecer la base para la gestión de itinerarios utilizando Firebase Realtime Database.
- 2-Segunda iteración:
Desarrollar el calendario de actividades y reservas, permitiendo a los usuarios visualizar y marcar actividades como realizadas. Además, se implementará la posibilidad de poder ver el itinerario a realizar en un mapa. Se integrará la modificación en tiempo real de actividades utilizando Firebase Realtime Database.
- 3-Tercera iteración:
Se implementará la gestión de documentos, proporcionando al usuario un espacio privado para almacenar y gestionar documentos de viaje, esto mediante el uso de Firebase Storage. Además, se implementará la gestión de multimedia usando Firebase Realtime Database, permitiendo la recopilación y visualización de archivos multimedia asociados a cada itinerario, ya sea de forma compartida o no.

3.2. Responsabilidades

Para la división de responsabilidades se han ido asignando funcionalidades a cada uno de los miembros del grupo, para la primera iteración se implementaron 6 repartidas de la siguiente forma:

Feature	Funcionalidad	Realizada por
1	Creación básica de itinerarios	Xoel
2	Inicio de sesión y gestión de usuarios	Nicolás
3	Visualización de un mapa del itinerario	Xoel
4	Integración de Firebase Realtime para almacenar itinerarios	Xoel
5	Calendario de actividades	Xoel
6	Posibilidad de compartir itinerarios entre Usuarios	Nicolás
12	Integración de Firebase Storage para almacenar documentos	Xoel
13	Integración de Firebase Storage para almacenar multimedia	Nicolás
16	Integración de sensor de cámara para realizar fotografías	Nicolás

Las features 7, 8, 9, 10, 11, 14 y 15 se centran en la corrección de errores y reestructuración del código, realizadas entre los dos integrantes.

3.3. Hitos

Para el desarrollo se fijan 4 hitos. En el primero de ellos, se espera tener una aplicación capaz de almacenar un itinerario de viaje, ya sea creado de forma individual o conjunta, y poder ver los diferentes destinos de este. En el segundo, se espera que la aplicación cuente con un calendario en el cual marcar las diferentes actividades que se van a realizar, así como las reservas que se han hecho previamente. En el siguiente, es decir el tercero, se espera que la aplicación permita almacenar y gestionar documentos importantes para el itinerario, como entradas o billetes, y que el usuario pueda acceder a ellos desde la misma. Finalmente, en el cuarto hito, se espera que los diferentes usuarios que formen parte de un mismo itinerario, puedan compartir entre ellos las fotos o vídeos que se han ido sacando durante el viaje.

3.4. Incidencias

- 1- Falta de experiencia en desarrollo de aplicaciones móviles y Firebase:
Aunque se tiene poca experiencia en desarrollo de aplicaciones móviles y no se tiene conocimiento sobre Firebase, se irán adquiriendo los conocimientos necesarios a medida que se desarrolle el proyecto.
- 2- Qué se ha probado que puede funcionar:
A medida que se desarrolle el proyecto, se explorarán diferentes opciones a la hora de realizar la implementación en Android Studio, además de valorar diferentes recursos de Firebase.
- 3- Qué hay que testear/validar antes de empezar cada iteración:
Antes de empezar cada iteración se realizarán pruebas exhaustivas para garantizar el correcto funcionamiento de las funcionalidades desarrolladas.
 - Para la primera iteración se comprobará la capacidad de crear itinerarios nuevos, la verificación de la adición de destinos y actividades dentro de cada itinerario, además del testeo de la sincronización de datos con Firebase Realtime Database.
 - Para la segunda iteración se comprobará la correcta visualización de las actividades en el calendario, además de verificar la capacidad de marcar actividades como realizadas y el testeo de la integración con Firebase Realtime Database para la modificación en tiempo real de actividades.
 - Para la tercera iteración se comprobará la capacidad de almacenar y recuperar documentos de viaje en el espacio privado del usuario, además de verificar de la recopilación y visualización correcta de archivos multimedia asociados a cada itinerario. Se hará un testeo de la integración con Firebase Storage para la gestión de documentos y otro de la sincronización en tiempo real de los archivos multimedia utilizando Firebase Realtime Database.

- 4- Alternativas en caso de problemas

Si surgen dificultades debido a la falta de experiencia, se buscarán recursos como documentación oficial para el desarrollo de aplicaciones móviles y Fire-base.

4. Diseño

4.1. Arquitectura

Para asegurar cierta independencia de las funcionalidades, se implemento una arquitectura en la cual cada componente tenga responsabilidades propias, de manera similar que en la arquitectura CLEAN. En este caso se separan las siguientes capas:

- Presenter: Agrupa las vistas de la aplicación, se limita a mostrar la información y cargar las diferentes actividades.
- Domain: Se divide en 4 subdirectorios, cuenta con las entidades, la lógica de negocio, los repositorios y los servicios.
- Data: Agrupa las clases encargadas de acceder directamente a la base de datos.

4.2. Diseño e implementación de la 3a iteración

En la tercera iteración se han cumplido todos los hitos marcados, consiguiendo una aplicación que permite a un usuario, previamente identificado por FirebaseAuthentication, crear, ver, modificar y compartir itinerarios. A mayores, como se marca en el segundo hito, el usuario puede visualizar tanto un mapa del itinerario como un calendario de actividades en el cual puede añadir y editar eventos en fechas concretas. En el tercer hito se ha cumplido con lo propuesto, que es que la aplicación permita almacenar y gestionar documentos importantes para el usuario, como tarjetas de embarque, pasaportes o billetes. Este es un punto necesario, ya que el usuario tiene que tener a disposición documentación importante para poder usarla mientras realiza el viaje. En cuanto al cuarto hito, también se ha cumplido lo propuesto, que era que los diferentes usuarios que formen parte de un mismo itinerario puedan compartir entre ellos multimedia que se ha ido realizando durante el viaje, ya sea cargada o realizada en el momento. Así, un usuario que no haya compartido el itinerario con ningún otro usuario registrado en la plataforma, podrá subir imágenes pero también habrá la posibilidad de compartirlo con otros y de que estos suban multimedia junto a este usuario. Así, se marcan como finalizados los cuatro hitos establecidos en el desarrollo de esta aplicación.

4.3. Persistencia

El sistema utiliza el servicio de Firebase para poder almacenar los datos necesarios, ya sea en la nube, la cual se usará para datos compartidos como los itinerarios

o los archivos multimedia, o en local para almacenar los documentos personales de cada usuario. Se almacenan, para cada usuario, itinerarios, documentos y eventos, los cuales estan asociados a Itinerarios. Un itinerario está compuesto por:

- Un String id para diferenciarlo
- Un String Admin para saber quien ha creado el itinerario
- Un String itineraryTitle para almacenar el nombre del itinerario
- Un String country para almacenar el país del itinerario
- Un String state para almacenar la región del itinerario
- Un String city para almacenar la ciudad del itinerario
- Un String startDate para almacenar la fecha de inicio
- Un String endDate para almacenar la fecha final
- Una lista de Strings Collaborators para almacenar los usuarios con acceso
- Una lista de Strings de uris que referencian a las fotografías de la base de datos

Un Evento está formado por:

- Un String id para diferenciarlo
- Un String date para almacenar la fecha
- Un String category para determinar el tipo de evento
- Un String description que almacena una descripción breve del evento

Un Documento está formado por:

- Un String documentId para diferenciarlo
- Un String name para almacenar el nombre
- Un String imageUrl para almacenar la url de la imagen que se va a cargar
- Un long timestamp para almacenar la hora de creación

4.4. Vista

La aplicación cuenta con una actividad principal que permite al usuario ver sus itinerarios y crear nuevos. Además, cuenta con una segunda actividad que permite visualizar la información de cada uno de estos. Esta puede generar un fragmento para visualizar un mapa con los destinos o un calendario de actividades. Finalmente, hay otras tres actividades más: la primera para acceder a los documentos del usuario, que se podrá acceder a ella desde la actividad principal y permitirá la visualización de los documentos personales del usuario. Desde ella, se podrá acceder a otra actividad en la que se producirá la carga de nuevos documentos personales. La tercera actividad estará dedicada a la carga y visualización de multimedia desde itinerarios individuales o compartidos, y estará accesible desde la segunda actividad principal de la aplicación, que es la del detalle de itinerario.

4.5. Comunicaciones

Se ha implementado la comunicación con el servicio de Firebase para la sincronización de datos en tiempo real, así como para el almacenamiento y recuperación de archivos multimedia y documentos de viaje. Se garantiza una conexión segura y confiable con los servidores de Firebase para garantizar la integridad de los datos y la disponibilidad de la aplicación.

4.6. Sensores

Se ha integrado el sensor de la cámara para permitir la realización de fotografías y así proveer una mejor experiencia de usuario, permitiendo al cliente captar momentos sin tener que salir de la propia aplicación, y así poder compartirlo con otros.

4.7. Trabajo en Background

Se ha implementado una tarea en segundo plano, que carga el API de países, estados y ciudades necesarias para que estén disponibles al usuario en el momento en el que este proceda a la creación del itinerario, asegurando así que estén disponibles al usuario cuando se requiere.

4.8. Pruebas y resultados finales

Para la realización de las pruebas y la verificación de que el gestor de viajes implementado funciona correctamente, se llevaron a cabo diversas pruebas que involucraron la creación de diferentes usuarios.

- **Creación de usuarios e inicio de sesión:** Se comprobó mediante Firebase Authentication que al registrar un usuario este se almacenara correctamente, permitiendo así que solo inicien sesión los usuarios registrados en la plataforma.

- **Creación y Visualización de Itinerarios:** Se comprobó reiteradamente que al crear un itinerario, este se guardase correctamente mediante Firebase Realtime Database, y que el usuario pudiera visualizar correctamente el lugar correspondiente a la localización introducida. También se verificó que el usuario pudiera modificar el destino en detalle posteriormente, y que al hacerlo, el mapa se actualizara automáticamente mostrando el nuevo destino seleccionado para ese itinerario. Además, el usuario podrá editar en el detalle del itinerario otros campos como el nombre que le ha dado a este, además de la localización.
- **Edición de Fechas:** También se verificó la funcionalidad de edición de la fecha del itinerario introducida por el usuario. Se confirmó que si la fecha de inicio es superior a la de finalización, el sistema no actualiza la información, evitando así errores en la planificación del itinerario.
- **Creación y Visualización de Eventos:** Se hicieron pruebas también para verificar que cuando un usuario cree eventos para un itinerario planificado en el calendario, solo los pueda crear dentro del rango de fechas seleccionado inicialmente. Si se decidiese modificar el intervalo de fechas de itinerario, se borrarán todos los eventos asociados a la fecha planificada anteriormente. Cabe resaltar que se comprobó en detalle que se pudiesen crear dentro de la fecha establecida diferentes tipos de actividades con tres categorías: Exploración, Gastronomía y Entretenimiento. Los usuarios pueden darle un nombre a esas actividades, editarlas modificando el nombre o la categoría, o eliminarlas. Múltiples usuarios a los que se haya compartido el itinerario podrán añadir eventos o realizar las acciones mencionadas sobre los eventos presentes.
- **Eliminar y Compartir Itinerarios:** Se comprobó asimismo que los itinerarios pudieran eliminarse correctamente y que se pudieran compartir entre diversos usuarios registrados en la plataforma. Al compartir un itinerario, todos los usuarios con acceso a este pudieron añadir eventos de planificación de actividades junto con los demás, así como editar los eventos introducidos por otros usuarios y modificar otras características del itinerario.
- **Documentación Individual:** A cada uno de estos usuarios se les añadió documentación de viaje de forma individual, garantizando que cada usuario tenga acceso exclusivo a su documentación mediante el inicio de sesión. Esto permitirá que el usuario dentro de su cuenta pueda acceder a una preview de documentos importantes como pasaportes o tarjetas de embarque, y clickando en cada uno de ellos podrá ver información relevante para ese documento.
- **Multimedia en Itinerarios:** Además, se verificó que fuera posible introducir multimedia en los itinerarios, permitiendo a los usuarios añadir fotografías a los itinerarios individuales o compartidos. Estas funcionalidades se probaron en múltiples ocasiones, asegurando que la información se compartiera de forma coherente en Firebase.

- **Tema de la aplicación e idioma:** Se comprobó que los ajustes permitidos por la aplicación funcionan correctamente, permitiendo la selección de tema claro u oscuro para la aplicación, así como la selección de los idiomas preferidos para esta. En este caso, están soportados el inglés y el español.

Las pruebas realizadas confirmaron que el gestor de viajes cumple con los requisitos de funcionalidad y seguridad, permitiendo una gestión eficaz y colaborativa de los itinerarios de viaje.

A continuación, se presentan una serie de fotografías que ilustran el funcionamiento del gestor de viajes y los resultados de las pruebas realizadas:

4.9. Componentes Android

Se ha hecho uso de diferentes componentes, entre los que se pueden destacar:

- **Uso de adaptadores:**
 - Se han utilizado adaptadores para la implementación de los itinerarios en la actividad principal. Estos adaptadores permiten gestionar y mostrar en un RecyclerView los diferentes itinerarios que el usuario va añadiendo a su cuenta. En la vista del RecyclerView se puede ver el nombre asignado a cada itinerario y, manteniendo una pulsación larga, aparece un menú contextual que permite acceder a los detalles del itinerario.
 - Se han utilizado adaptadores para gestionar y mostrar los documentos personales en un RecyclerView. Al hacer clic en un documento, se pueden ver los detalles de ese documento. Además, una pulsación larga permite eliminar el documento seleccionado.
 - Se han utilizado adaptadores para gestionar y mostrar las imágenes de los itinerarios compartidos en un RecyclerView. Estos adaptadores permiten cargar y mostrar imágenes utilizando la biblioteca Glide. Una pulsación larga permite eliminar la imagen seleccionada.
- **Uso de Intents:**
 - Se han utilizado Intents para la navegación entre actividades dentro de la aplicación. Esto incluye la transición desde la actividad principal a la actividad de detalles de itinerarios, así como la apertura de actividades para la gestión de documentos personales y la visualización de multimedia. También se han usado para la apertura de galería y cámara.
- **Uso de Fragmentos:**
 - Se han utilizado Fragments para el uso del mapa en el detalle de cada itinerario, así como para poder mostrar al usuario el calendario de actividades.

■ **Uso de Layouts:**

- Los layouts se han utilizado extensivamente en la aplicación para definir la estructura y apariencia de las interfaces de usuario. Cada ítem en los RecyclerView está diseñado con un layout específico, asegurando una presentación consistente y atractiva de los datos. Además, se han usado layouts para organizar los elementos en las diferentes pantallas de la aplicación, proporcionando una experiencia de usuario intuitiva y ordenada.

■ **Uso de servicios:**

- Se ha utilizado el servicio de mapas para integrar Google Maps en la aplicación. Este servicio permite inicializar el mapa, gestionar la disponibilidad del mapa y obtener la ubicación de una dirección específica.
- Se ha utilizado la tarea en segundo plano `LoadCountriesTask` para cargar los países, estados y ciudades utilizando la biblioteca `Localisation`. Esta tarea se ejecuta en segundo plano para mejorar el rendimiento de la aplicación y actualizar la interfaz de usuario una vez que los datos están disponibles.

4.10. Librerías

Las librerías que han sido utilizadas para el desarrollo de la aplicación son:

- `androidx.appcompat:appcompat:1.6.1`
- `com.google.android.material:material:1.11.0`
- `androidx.constraintlayout:constraintlayout:2.1.4`
- `com.beastwall:localisation:1.0.5`
- `com.google.code.gson:gson:2.10`
- `com.caverock:androidsvg:1.4`
- `com.google.android.gms:play-services-maps:18.2.0`
- `com.prolificinteractive:material-calendarview:1.4.3`
- `com.applandeo:material-calendar-view:1.9.2`
- `androidx.preference:preference:1.2.1`

- junit:junit:4.13.2
- androidx.test.ext:junit:1.1.5
- androidx.test.espresso:espresso-core:3.5.1
- androidx.preference:preference-ktx:1.2.0
- com.google.android.material:material:1.12.0
- com.google.firebase:firebase-bom:32.8.0
- com.google.firebase:firebase-database:20.3.1
- com.google.firebase:firebase-storage:20.0.0
- com.github.bumptech.glide:glide:4.12.0
- com.google.firebase:firebase-analytics
- com.google.firebase:firebase-auth

5. Conclusiones y Trabajo Futuro

Con respecto a los objetivos establecidos, se han cumplido todos en el tiempo esperado y, tras las diferentes pruebas realizadas sobre la aplicación final, solo se han encontrado errores relacionados con la usabilidad, en concreto en los tiempos de espera a la hora de subir y descargar archivos de la base de datos, pero, teniendo en cuenta que la finalidad del proyecto es familiarizarse con Android Studio y, una vez hecho esto, conseguir una aplicación funcional, no suponen puntos críticos para el resultado final.

Como es lógico la aplicación es simple por lo que, si se quisiera realmente que pudiera competir con el resto de alternativas existentes en el mercado, se deberían de añadir nuevas funcionalidades, como poder establecer rutas, la posibilidad de que las actividades se puedan marcar como finalizadas si ya ha pasado la fecha de esa actividad o implementar un sistema de amistades para simplificar la compartición de itinerarios. También se podrían marcar los itinerarios como finalizados y quedar solo disponibles para la visualización una vez haya transcurrido el rango de fechas establecidas para estos. Otro punto a abordar en un futuro sería la interfaz, la cual debería de ser mas amigable e intuitiva, facilitando a los usuarios la navegación por la misma.

6. Bibliografía

Referencias

- [1] Firebase Storage. <https://firebase.google.com/docs/storage>
- [2] Firebase Realtime Database. <https://firebase.google.com/docs/database>
- [3] TripIt - Gestor de Viajes. <https://www.tripit.com>
- [4] TripCase - Gestor de Viajes. <https://www.tripcase.com>
- [5] Localisation lib. <https://github.com/rasmi-aw/Localisation-lib>
- [6] Glide lib. <https://github.com/bumptech/glide>
- [7] Andorid Developers. <https://developer.android.com/?hl=es-419>