

SEVILLANAS MANERAS



Autor: Martín Izquierdo Cuevas

Tutor: Javier Martin

Fecha de entrega:

Convocatoria: 2024 2025

ÍNDICE

Introducción.....	1
Motivación.....	2
Abstract.....	3
Objetivo General.....	3
Objetivos Específicos	3
Metodología Utilizada.....	4
Ciclo de Vida del Proyecto.....	5
Backlog del Proyecto	6
Tecnologías y Herramientas Utilizadas en el Proyecto	7
Estimación de Recursos y Planificación	9
Fases del Proyecto y Estimación de Tiempo.....	9
Descripción de las Fases	10
Diagrama de Gantt	11
ANÁLISIS	12
 Requisitos Funcionales	12
 Requisitos No Funcionales.....	13
 Diagrama de Entidad-Relación (ER).....	13
 Casos de Uso	14
 Diagrama de Clases.....	17
 Diseño	18
 Despliegue y pruebas	23
 Conclusiones.....	24
 Vías futuras.....	25
 Glosario.....	26
 Bibliografía (formato APA 7º edición).....	27
 Diagrama de Gantt FINAL	27
 Anexos	28

Introducción

En el marco del Trabajo de Fin de Grado titulado "**Sevillanas Maneras**", presentado por **Martín Izquierdo Cuevas**, se plantea el desarrollo de una aplicación móvil interactiva para Android, cuyo propósito es acercar al usuario a la riqueza cultural e histórica de Sevilla de una manera innovadora y participativa. La ciudad, conocida por su **Feria de Abril**, su **Semana Santa** y sus monumentos emblemáticos, posee una identidad única que merece ser explorada de forma más accesible y dinámica.

El objetivo principal de "**Sevillanas Maneras**" es proporcionar una experiencia inmersiva que permita descubrir los eventos y lugares más representativos de la ciudad, como la **Catedral de Sevilla**, la **Torre del Oro** o el **Barrio de Santa Cruz**, no solo mediante información textual, sino a través de mapas interactivos, imágenes y curiosidades históricas presentadas de manera atractiva. A diferencia de una guía tradicional, esta aplicación pretende que el usuario no solo consulte datos, sino que interactúe con ellos, favoreciendo un aprendizaje experiencial que refuerce la conexión con la cultura sevillana.

Para garantizar un acceso fluido y actualizado a la información, "**Sevillanas Maneras**" empleará almacenamiento en la nube a través de **Firebase Firestore**, lo que permitirá la sincronización en tiempo real y la consulta de contenido sin necesidad de almacenamiento local. Además, se integrará **Google Maps API** para facilitar la localización de puntos de interés y diseñar rutas personalizadas.

El desarrollo de la aplicación se llevará a cabo en **Kotlin**, utilizando **Android Studio** como entorno de trabajo y aplicando la metodología ágil **SCRUM**, lo que permitirá una mejora continua a lo largo del proceso de desarrollo.

Este documento presenta la planificación, el análisis, el diseño y las tecnologías que sustentan el proyecto, estableciendo las bases para su correcta implementación y futura optimización.

Motivación

La idea de desarrollar "**Sevillanas Maneras**" surge de la observación de cómo, a pesar del enorme atractivo cultural de Sevilla, la forma en que se transmite esta información sigue siendo, en muchos casos, tradicional y poco interactiva. Si bien existen guías y recursos digitales, la mayoría se limitan a ofrecer listados de lugares y eventos sin un componente dinámico que invite al usuario a sumergirse realmente en la experiencia.

Este proyecto busca **romper con ese enfoque pasivo**, brindando a locales y turistas una herramienta que les permita **vivir Sevilla** de una manera diferente, a través de rutas temáticas, curiosidades y datos históricos accesibles desde cualquier dispositivo. La aplicación combinará elementos educativos y de entretenimiento para que el usuario no solo aprenda sobre la ciudad, sino que también disfrute del proceso de exploración.

Otro factor clave en la motivación de este proyecto es la **necesidad de aprovechar las tecnologías actuales para mejorar el acceso a la información cultural**. Mediante **almacenamiento en la nube con Firebase Firestore**, la aplicación ofrecerá datos en tiempo real, evitando la necesidad de actualizaciones manuales o almacenamiento local. Asimismo, la integración de **Google Maps API** permitirá que el usuario localice fácilmente los puntos de interés y diseñe recorridos personalizados en función de sus preferencias.

Finalmente, "**Sevillanas Maneras**" representa un reto tanto académico como profesional, ya que abarca múltiples áreas del desarrollo de software, desde la programación en **Kotlin** hasta la gestión de bases de datos en la nube y el diseño de interfaces intuitivas. Este proyecto no solo busca facilitar el acceso a la historia y cultura sevillana, sino también contribuir a la digitalización del turismo cultural, alineándose con las nuevas tendencias tecnológicas en el sector.

Abstract

This project is motivated by the desire to preserve and promote the rich cultural heritage of Seville, especially among younger audiences and tourists. The application, Sevillanas Maneras, is an interactive Android app designed to teach users about iconic traditions, locations, and festivities of the city, such as the Feria de Abril, Semana Santa, the Cathedral, and the Torre del Oro. The app offers a dynamic and engaging experience through curated content, images, curiosities, and interactive elements, encouraging users to explore Seville's culture in a playful yet educational way. The goal is not only to transmit knowledge but also to spark interest and appreciation for local customs. Additionally, the project provides a technological and creative challenge, combining Firebase for user authentication and data storage, custom interfaces for improved usability, and Kotlin-based development following modern Android standards. By merging tradition and innovation, this project highlights how mobile technology can serve as a powerful tool for cultural education and digital tourism, bringing history and identity to users' fingertips in a way that is both accessible and appealing.

Objetivo General

Desarrollar una aplicación móvil interactiva que permita a los usuarios explorar la historia, tradiciones y lugares emblemáticos de Sevilla mediante contenido multimedia, mapas dinámicos y curiosidades, optimizando la accesibilidad y la experiencia de aprendizaje.

Objetivos Específicos

Facilitar el acceso a la cultura sevillana: Ofrecer información detallada y organizada sobre los eventos más importantes de la ciudad, como la **Feria de Abril** y la **Semana Santa**, así como sobre sus monumentos y rincones históricos.

Integrar herramientas interactivas: Implementar funcionalidades como mapas dinámicos con **Google Maps API**, rutas temáticas y acceso a contenido multimedia que enriquezcan la experiencia del usuario.

Optimizar la accesibilidad a la información: Almacenar los datos en la nube mediante **Firebase Firestore**, garantizando un acceso rápido y actualizado sin depender del almacenamiento local del dispositivo.

Diseñar una interfaz intuitiva y atractiva: Desarrollar una experiencia de usuario fluida y accesible, utilizando buenas prácticas de **UX/UI** para asegurar una navegación sencilla y adaptada a distintos perfiles de usuarios.

Fomentar la conexión emocional con Sevilla: Presentar la historia y las tradiciones a través de un enfoque narrativo, con anécdotas y curiosidades que hagan que el usuario se sienta parte de la ciudad.

Garantizar una experiencia de uso eficiente y moderna: Aplicar metodologías ágiles como **SCRUM**, permitiendo la evolución continua del proyecto y la implementación progresiva de mejoras en la aplicación.

Promover la digitalización del turismo cultural: Convertir la aplicación en una herramienta útil para locales y visitantes, contribuyendo a la modernización del acceso a la información patrimonial de la ciudad.

Estos objetivos guiarán el desarrollo de "**Sevillanas Maneras**", asegurando que la aplicación no solo sea funcional y accesible, sino también innovadora y alineada con las necesidades del usuario.

Metodología Utilizada

He optado por utilizar la metodología ágil **SCRUM**, debido a su flexibilidad y eficiencia en proyectos de software. SCRUM permite organizar el trabajo en iteraciones llamadas **sprints**, facilitando la adaptación a posibles cambios y la mejora continua del producto. Esta metodología proporciona un marco de trabajo estructurado, dividiendo el desarrollo en fases bien definidas y promoviendo una entrega progresiva de funcionalidades.

La elección de **SCRUM** responde a la necesidad de gestionar el desarrollo de la aplicación de manera iterativa, permitiendo realizar ajustes a medida que se avanza en la implementación. Dado que el proyecto requiere la integración de múltiples tecnologías, como **Firebase Firestore** para almacenamiento en la nube y **Google Maps API** para la geolocalización de puntos de interés, es fundamental contar con una metodología que favorezca la retroalimentación constante y la optimización de recursos.

Ciclo de Vida del Proyecto

El desarrollo de "**Sevillanas Maneras**" se ha estructurado en varias fases, cada una de ellas con entregables específicos que aseguran la progresión del proyecto de manera ordenada:

1.- Análisis y planificación

- Definición de los requisitos funcionales y no funcionales.
- Estudio de las tecnologías más adecuadas para la implementación.
- Diseño del backlog inicial con las tareas prioritarias del desarrollo.

2.- Diseño de la interfaz y arquitectura del sistema

- Creación de prototipos en **Figma**.
- Diseño del flujo de navegación de la aplicación.
- Modelado de la base de datos en **Firebase Firestore**.

3.- Desarrollo e integración de funcionalidades

- Implementación de la interfaz gráfica en **Android Studio** con **Kotlin**.
- Conexión de la aplicación con la base de datos en la nube.
- Desarrollo de la integración con **Google Maps API**.

4.- Pruebas y optimización

- Pruebas unitarias y de integración.
- Optimización del rendimiento y corrección de errores.
- Validación de la usabilidad y accesibilidad de la aplicación.

5.- Despliegue y documentación

- Elaboración de la documentación técnica y del manual de usuario.
- Implementación de posibles mejoras basadas en pruebas y retroalimentación.

Backlog del Proyecto

El backlog del proyecto está compuesto por un conjunto de tareas organizadas en sprints, siguiendo un enfoque incremental. A continuación, se detallan las tareas principales que se han diseñado para la gestión y desarrollo del proyecto:

Sprint 1: Análisis y Definición

- Investigación sobre las tecnologías a utilizar.
- Definición de los requisitos funcionales y no funcionales.
- Diseño inicial de la base de datos en **Firebase Firestore**.

Sprint 2: Diseño UI/UX

- Creación de prototipos en **Figma**.
- Definición del flujo de navegación y estructura de la aplicación.
- Diseño de la interfaz de usuario en **Android Studio**.

Sprint 3: Desarrollo del Backend y Conexión con Firebase

- Implementación de la base de datos en la nube.
- Desarrollo del sistema de consulta y actualización de datos.
- Pruebas de conexión y verificación del almacenamiento en la nube.

Sprint 4: Integración de Funcionalidades Clave

- Implementación de **Google Maps API** para geolocalización de puntos de interés.
- Desarrollo del sistema de visualización de información de eventos y lugares.
- Pruebas de integración con la base de datos y corrección de errores.

Sprint 5: Optimización y Pruebas

- Pruebas unitarias y de integración.
- Corrección de errores y mejora de la experiencia de usuario.
- Validación de accesibilidad y rendimiento en distintos dispositivos.

Sprint 6: Documentación y Despliegue

- Elaboración de la documentación técnica.
- Creación del manual de usuario.
- Publicación y evaluación final del proyecto.

Tecnologías y Herramientas Utilizadas en el Proyecto

Para el desarrollo de "**Sevillanas Maneras**", se han seleccionado tecnologías y herramientas que garantizan un **rendimiento óptimo, accesibilidad, almacenamiento en la nube** y una **experiencia de usuario fluida**. A continuación, se detallan los componentes clave del proyecto y la justificación de su elección.

Lenguaje de Programación

- **Kotlin**

- Es el lenguaje oficial para el desarrollo de aplicaciones Android, recomendado por Google.
- Ofrece mayor seguridad, mejor rendimiento y menos código en comparación con Java.
- Permite una integración fluida con **Android Jetpack**, facilitando el desarrollo de interfaces modernas y eficientes.



Entorno de Desarrollo (IDE)

- **Android Studio**

- Es el entorno de desarrollo oficial de Android.
- Proporciona herramientas avanzadas para depuración, simulación y optimización del rendimiento.



- Permite el uso de **Emuladores Android** para probar la aplicación en diferentes dispositivos y resoluciones.

Base de Datos y Almacenamiento en la Nube



- **Firebase Firestore**

- Base de datos NoSQL en la nube, que permite la sincronización en tiempo real.
- Facilita el acceso a la información sin depender del almacenamiento local del dispositivo.
- Mejora la escalabilidad y evita la necesidad de gestionar servidores propios.

API y Servicios Externos



- **Google Maps API**

- Permite mostrar mapas interactivos dentro de la aplicación.
- Ayuda a ubicar los puntos históricos y culturales de Sevilla con precisión.
- Facilita la creación de rutas personalizadas para los usuarios.

Diseño UI/UX



- **Figma**

- Herramienta de diseño utilizada para la creación de **prototipos** y **mockups**.
- Permite visualizar la estructura de la interfaz antes de su implementación.
- Facilita la colaboración y revisión de la experiencia de usuario (UX).

Control de Versiones y Gestión del Código



- **Git y GitHub**

- Git permite un control eficiente del código fuente, asegurando estables y seguras del proyecto.

- GitHub facilita la colaboración, almacenamiento y gestión del código de forma remota.

Justificación de la Selección de Tecnologías

- ✓ **Eficiencia y rendimiento:** Kotlin y Firebase permiten una ejecución rápida y optimizada.
- ✓ **Escalabilidad:** Firebase Firestore permite almacenar y actualizar datos sin necesidad de infraestructura propia.
- ✓ **Accesibilidad y experiencia de usuario:** Google Maps y Figma facilitan la interacción con la aplicación.
- ✓ **Seguridad y mantenimiento:** GitHub garantiza un desarrollo seguro y sin pérdidas de código.

Estimación de Recursos y Planificación

Para la correcta planificación y gestión del desarrollo de "**Sevillanas Maneras**", se ha establecido un cronograma de trabajo basado en la metodología **SCRUM**, distribuyendo las tareas en diferentes fases y asegurando una evolución progresiva del proyecto.

La planificación ha sido diseñada teniendo en cuenta los **plazos establecidos por la normativa**, siendo la **primera entrega** de documentación el **31 de marzo de 2025** y la **entrega final el 21 de mayo de 2025**.

Fases del Proyecto y Estimación de Tiempo

A continuación, se detallan las fases principales del desarrollo de la aplicación junto con la duración estimada para cada una de ellas:

Fase del Proyecto	Duración Estimada	Plazo de Ejecución
Fase 1: Análisis y planificación	1 semana	14 marzo – 21 marzo 2025
Fase 2: Diseño UI/UX y base de datos	2 semanas	21 marzo – 31 marzo 2025
Fase 3: Desarrollo del backend	3 semanas	1 abril – 28 abril 2025
Fase 4: Implementación de funcionalidades clave	2 semanas	29 abril – 6 mayo 2025
Fase 5: Pruebas y optimización	1 semana	7 mayo – 13 mayo 2025
Fase 6: Documentación y entrega final	1 semana	14 mayo – 21 mayo 2025

Descripción de las Fases

Fase 1: Análisis y planificación (14 marzo – 21 marzo 2025)

- Identificación de requisitos funcionales y no funcionales.
- Análisis de tecnologías y herramientas a utilizar.
- Diseño del modelo de datos en **Firebase Firestore**.
- Elaboración del backlog con tareas iniciales.

Fase 2: Diseño UI/UX y base de datos (21 marzo – 31 marzo 2025)

- Creación de **wireframes y prototipos en Figma**.
- Diseño del flujo de navegación de la aplicación.
- Definición final de la base de datos en **Firebase Firestore**.
- **Entrega de la primera documentación del proyecto (31 marzo)**.

Fase 3: Desarrollo del backend (1 abril – 28 abril 2025)

- Implementación de la base de datos en la nube.
- Desarrollo del sistema de consulta y actualización de datos.
- Configuración de autenticación.

Fase 4: Implementación de funcionalidades clave (29 abril – 6 mayo 2025)

- Integración de **Google Maps API** para geolocalización.
- Desarrollo de las pantallas principales de la aplicación.
- Implementación del sistema de búsqueda y filtrado de lugares.

Fase 5: Pruebas y optimización (7 mayo – 13 mayo 2025)

- Corrección de errores y mejora del rendimiento.
- Validación de la accesibilidad y experiencia de usuario.

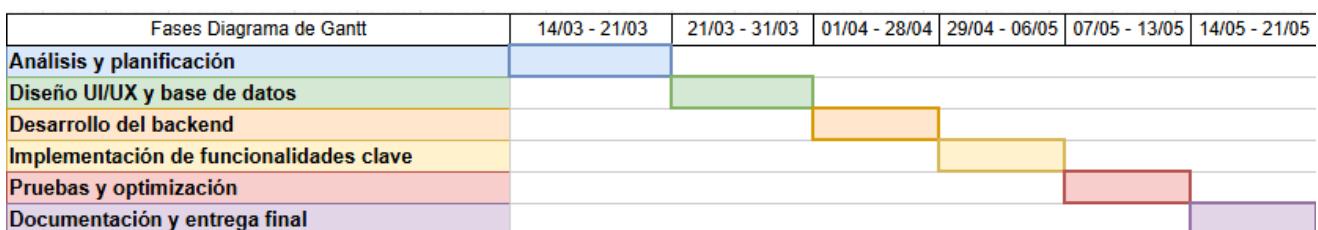
Fase 6: Documentación y entrega final (14 mayo – 21 mayo 2025)

- Elaboración de la **documentación técnica y manual de usuario**.
- Preparación de la **presentación del proyecto**.
- **Entrega final del TFG (21 mayo)**.

Diagrama de Gantt

Como representación de la estimación inicial del proyecto, elaboré un diagrama de Gantt desde la segunda quincena del mes de marzo hasta la tercera semana de junio, estableciendo una planificación estructurada para completar cada fase de desarrollo dentro de los plazos previstos. Si bien la planificación inicial considera un desarrollo escalonado y progresivo, es posible que se realicen ajustes en función de los avances y dificultades encontradas durante la implementación.

Por tanto, la representación de mi planificación inicial del proyecto mediante un Diagrama de Gantt fue la siguiente:



Tras la planificación inicial, el proyecto se irá ajustando conforme avance el desarrollo, asegurando que se cumplan los objetivos y plazos establecidos para la entrega final.

ANÁLISIS

El propósito de esta aplicación es ofrecer una experiencia interactiva para que los usuarios aprendan sobre las tradiciones y lugares icónicos de Sevilla (Semana Santa, Feria de Abril, Catedral, Torre del Oro, etc.). Antes de usar la app, los usuarios deben **registrarse e iniciar sesión**.

Requisitos Funcionales

1. Registro de Usuarios

- Los usuarios deben poder registrarse con correo electrónico y contraseña.
- Se debe validar el correo electrónico antes de activar la cuenta.
- Debe haber una opción para recuperar la contraseña.

2. Autenticación

- Los usuarios pueden iniciar y cerrar sesión.
- Deben poder modificar su perfil y eliminar su cuenta.

3. Exploración de Lugares y Tradiciones

- La aplicación debe permitir visualizar información sobre los lugares históricos de Sevilla.
- Se mostrarán imágenes, descripciones y datos históricos de cada sitio.

4. Ubicación y Geolocalización

- Se podrá acceder a información de cada lugar mediante un mapa interactivo.

5. Favoritos y Seguimiento

- Los usuarios pueden guardar lugares de interés en su lista de favoritos.
- Se podrá consultar un historial de lugares visitados.

Requisitos No Funcionales

- **Usabilidad:** La interfaz debe ser intuitiva y fácil de usar.
- **Seguridad:** Los datos de los usuarios deben ser almacenados de forma segura (hash de contraseñas, autenticación segura).
- **Compatibilidad:** La aplicación debe funcionar en la mayoría de dispositivos Android (mínimo Android 8.0).
- **Escalabilidad:** La base de datos debe permitir la adición de nuevos lugares sin afectar el rendimiento.
- **Rendimiento:** La aplicación debe cargar imágenes y datos de manera rápida, optimizando las consultas a la base de datos.

Diagrama de Entidad-Relación (ER)

1. Usuarios

- id_usuario (PK)
- nombre
- email
- contraseña

2. Elementos

- id_elemento (PK)
- nombre
- descripción
- tipo (lugar,cultura, monumento, tradición...)
- imagen (nombre del recurso en drawable o enlace)
- ubicacionMaps (URL a Google Maps)
- recomendaciones

3. Favoritos (subcolección dentro de Usuarios)

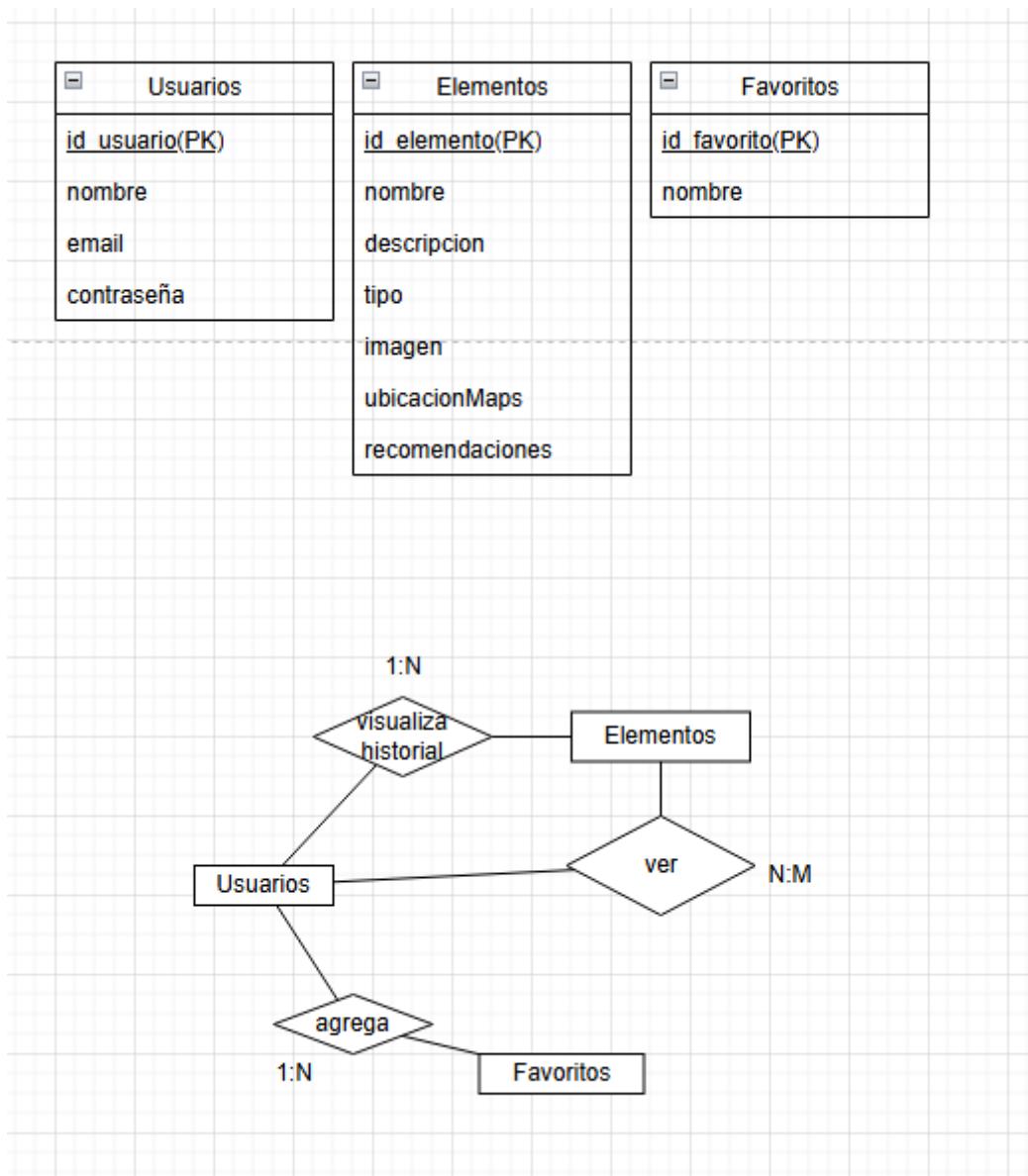
- id_favorito (PK)
- nombre (nombre del elemento marcado como favorito)

Relaciones:

Usuarios 1 --- N Favoritos

Usuarios 1 --- N Visualiza (historial)

Usuarios N --- M Elementos



Casos de Uso

Actores del sistema:

-
1. **Usuario**
 2. **Administrador**

Casos de uso principales:

- **Registro e inicio de sesión** (usuario)
- **Explorar lugares** (usuario)
- **Ver detalles de un lugar** (usuario)
- **Añadir/quitar de favoritos** (usuario)
- **Actualizar información de lugares** (administrador)
- **Gestionar base de datos de lugares** (administrador)
- **Eliminar Usuarios** (administrador)

Caso de uso	Actor	Descripción breve
Registro e inicio de sesión	Usuario	Permite al usuario crear una cuenta o acceder a una ya existente.
Explorar lugares	Usuario	Muestra una lista de sitios culturales y turísticos relevantes de Sevilla.
Ver detalles de un lugar	Usuario	Al seleccionar un lugar, se muestran curiosidades, historia y multimedia asociada.
Añadir/quitar de favoritos	Usuario	Permite guardar o eliminar lugares para acceso rápido en el futuro.
Actualizar información de lugares	Administrador	Edita los datos de los lugares (nombre, historia, imágenes, etc.).
Gestionar base de datos de lugares	Administrador	Puede añadir, eliminar o modificar entradas en la base de datos.
Eliminar usuarios	Administrador	Permite borrar cuentas de usuarios que incumplan normas o por mantenimiento.

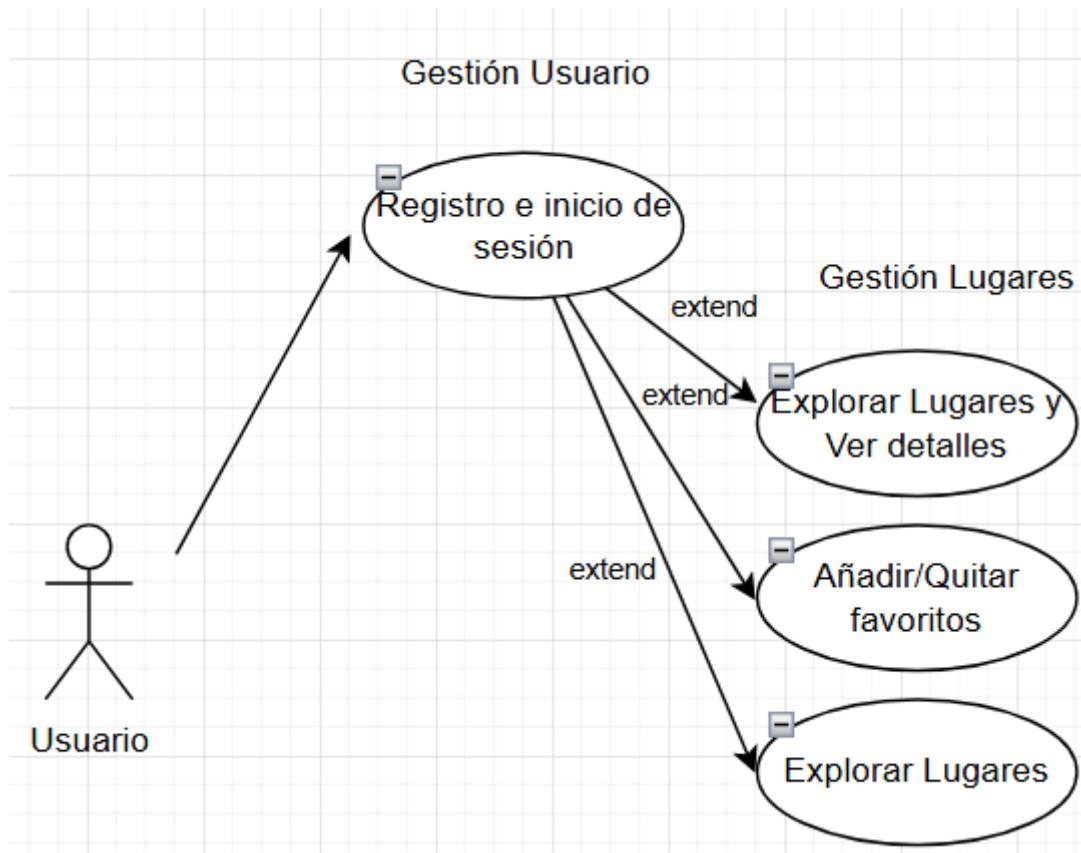
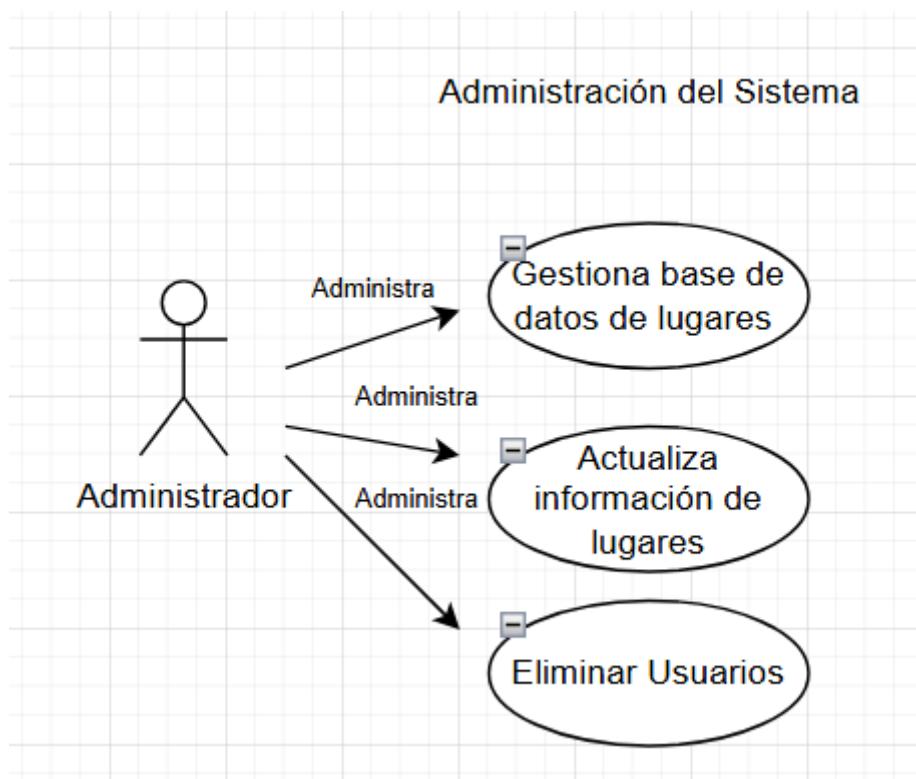
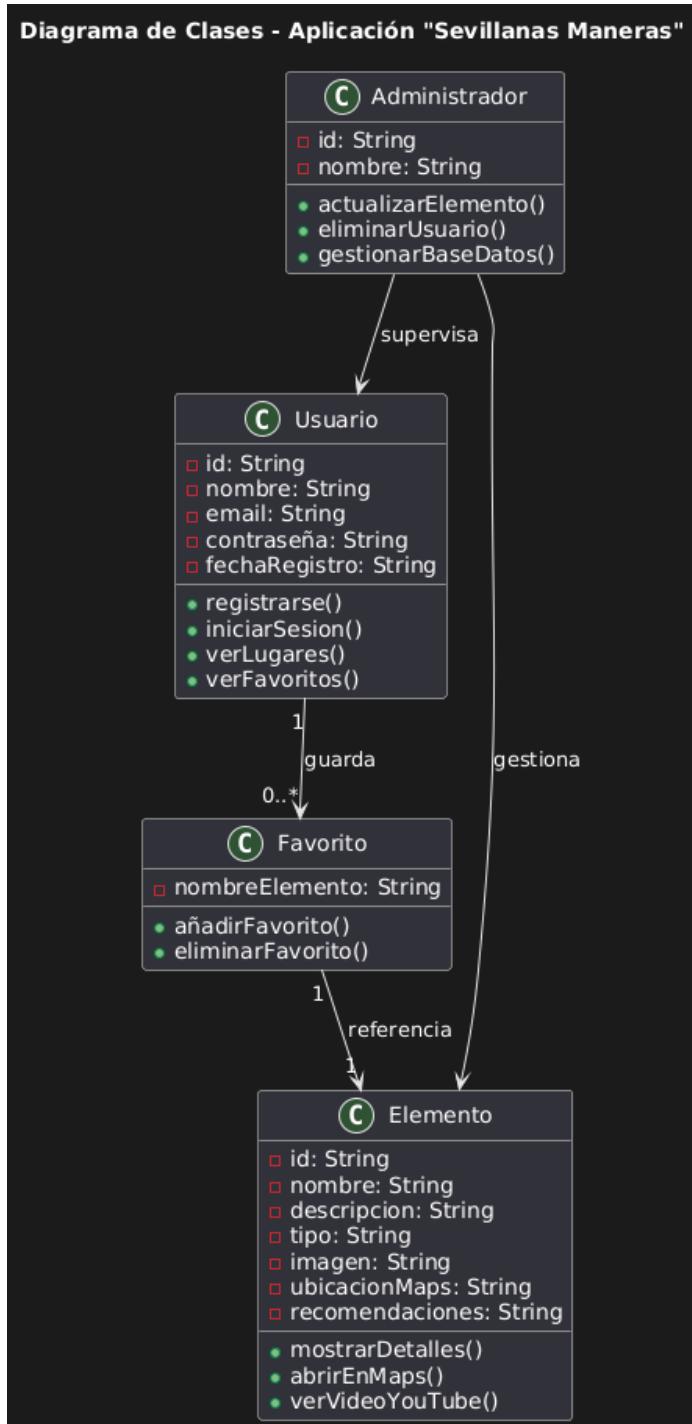


Diagrama de Clases



Diseño

El mockup del proyecto ha sido diseñado en **Figma** y representa la estructura visual de la aplicación, mostrando las principales pantallas y su flujo de navegación. El diseño sigue una estética inspirada en la cultura sevillana, utilizando una combinación de colores rojo, dorado y negro para reforzar la identidad visual.

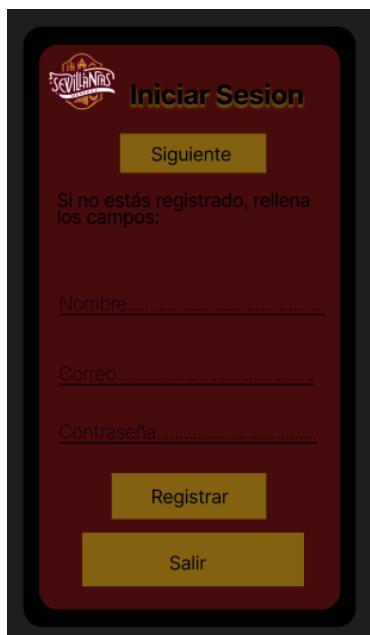
El mockup incluye pantallas clave como el inicio de sesión, registro de usuarios, exploración de lugares, perfil del usuario y detalles de ubicación. La navegación se basa en botones intuitivos que permiten acceder fácilmente a las diferentes secciones de la app.

Es importante destacar que este diseño es un **boceto inicial** y puede estar sujeto a **modificaciones y mejoras** en función de pruebas de usabilidad y feedback recibido durante el desarrollo.

Pantalla de Iniciar de Sesión

Esta pantalla permite a los usuarios registrarse o acceder a la aplicación si ya tienen una cuenta.

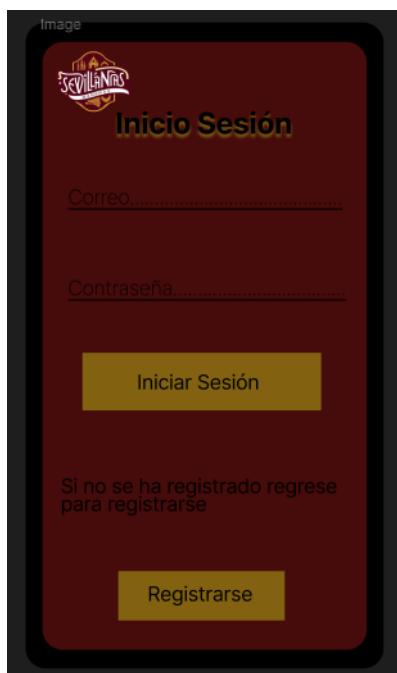
Funciones disponibles:



1. **Iniciar sesión:**
 - Si el usuario ya está registrado, puede presionar el botón "**Siguiente**" para dirigirse a la pantalla de inicio de sesión.
2. **Registro de nuevos usuarios:**
 - Si el usuario no tiene una cuenta, puede llenar los campos:
 - **Nombre**
 - **Correo electrónico**
 - **Contraseña**
 - Una vez completados, debe pulsar el botón "**Registrar**" para crear su cuenta.
3. **Salir de la aplicación:**
 - El botón "**Salir**" permite cerrar la aplicación o regresar a una pantalla anterior.

Pantalla de Inicio de Sesión

Esta pantalla permite a los usuarios acceder a la aplicación introduciendo sus credenciales o registrarse si aún no tienen cuenta.



Funciones disponibles:

1. **Iniciar sesión:**
 - El usuario debe ingresar su **correo electrónico** y **contraseña** en los campos correspondientes.
 - Al presionar el botón "**Iniciar Sesión**", se verificará la información y, si es correcta, accederá a la aplicación.
2. **Registrarse:**
 - Si el usuario aún no tiene cuenta y llegó a esta pantalla por error, se muestra un mensaje indicándole que debe registrarse.
 - Puede hacerlo fácilmente presionando el botón "**Registrarse**", que lo llevará a la pantalla de registro.

Pantalla Principal

La pantalla muestra una interfaz de usuario bien organizada con los siguientes elementos:

Funciones disponibles:



1. Barra de navegación superior:

- Menú desplegable para seleccionar categorías (Lugares, Cultura, Tradiciones, Monumentos)
- Nombre de la categoría seleccionada actualmente (ej. "Lugares")
- Nombre del usuario que ha iniciado sesión y si lo pulsas te lleva a tu perfil.

2. Carrusel de imágenes:

- Muestra imágenes representativas del lugar seleccionado
- Contiene botones con flechas para navegar entre diferentes lugares

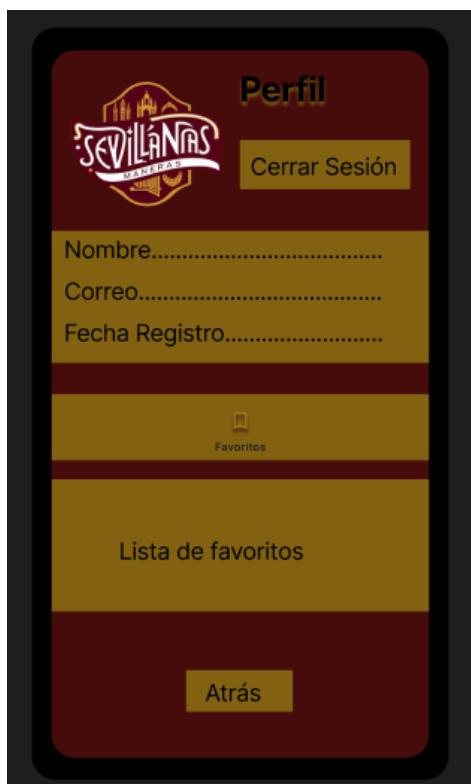
3. Área de información:

- Nombre del lugar
- Descripción/Curiosidades sobre el lugar
- Botón para abrir Google Maps con la ubicación del lugar
- Botón para guardar el lugar en favoritos del perfil.

Pantalla de Perfil

La pantalla de perfil está diseñada para que los usuarios puedan ver su información personal, acceder a sus lugares favoritos y cerrar sesión:

Funciones disponibles:



1. Barra Superior

- Icono de la App.
- Título "Perfil".
- Botón "Cerrar Sesión" y redirige a la pantalla de inicio de sesión.

2. Información del Usuario

- Nombre del usuario.
- Correo electrónico.
- Fecha de registro.

3. Sección de Favoritos

- Título "Favoritos".
- Lista de lugares guardados.

- Si no hay favoritos, muestra: "Aún no tienes lugares favoritos".

4. Botón de Navegación

- Botón "Atrás" vuelve a la pantalla anterior (Lugares).

Pantalla de Ubicación

Esta pantalla se activa cuando el usuario selecciona "Ver en mapa" desde la pantalla de detalles de un lugar. Muestra la ubicación exacta e información relevante para visitarlo.



1. Barra Superior

- Título "Ubicación" (centrado)
- Nombre del lugar destacado (ej: "Real Alcázar de Sevilla")

2. Mapa Interactivo (Google Maps API)

- Vista completa del mapa centrada en las coordenadas del lugar
- Controles de zoom estándar de Google Maps
- Botón "Cómo llegar" (abre Google Maps con ruta desde la posición actual)

3. Sección de Recomendaciones

Encabezado: "Recomendaciones para tu visita"

Contenido:

- Horario óptimo de visita (ej: "Mejor horario: 9:00-11:00 para evitar colas")
- Transporte recomendado (metro, autobús líneas específicas)
- Lugares cercanos destacados (ej: "A 5 min: Catedral de Sevilla")
- Consejos prácticos (ej: "Llevar calzado cómodo - suelo empedrado")

4. Botón de Navegación

- Botón "Atrás" vuelve a la pantalla anterior (Lugar donde pulsaste Maps en la pantalla principal).

Despliegue y pruebas

Nº	Especificación de Pruebas
1	<p>Objetivo probado: Validar que el usuario puede registrarse correctamente introduciendo datos válidos.</p> <p>Requisitos probados: El sistema debe permitir crear un nuevo usuario con nombre, email y contraseña.</p> <p>Pruebas que realizar:</p> <ul style="list-style-type: none">- Introducir todos los campos válidos y verificar que se registre correctamente.- Verificar que se guarde el usuario en Firebase Authentication y Firestore.
2	<p>Objetivo probado: Evitar el registro con campos vacíos.</p> <p>Requisitos probados: El sistema no debe permitir el registro si algún campo está vacío.</p> <p>Pruebas que realizar:</p> <ul style="list-style-type: none">- Dejar el nombre vacío.- Dejar el correo vacío.- Dejar la contraseña vacía.- Verificar que aparezca el mensaje "Por favor complete todos los campos".
3	<p>Objetivo probado: Evitar el registro con contraseñas demasiado cortas.</p> <p>Requisitos probados: La contraseña debe tener al menos 6 caracteres.</p> <p>Pruebas que realizar:</p> <ul style="list-style-type: none">- Ingresar "123" como contraseña.- Verificar que se muestre el mensaje de error correspondiente.
4	<p>Objetivo probado: Evitar el registro con un correo ya registrado.</p> <p>Requisitos probados: El sistema debe comprobar si un correo ya está en uso.</p> <p>Pruebas que realizar:</p> <ul style="list-style-type: none">- Usar un correo que ya esté registrado.- Verificar que se muestre el mensaje "El correo electrónico ya está en uso".
5	<p>Objetivo probado: Verificar la navegación desde el registro a la pantalla de inicio de sesión.</p> <p>Requisitos probados: El botón "Siguiente" debe llevar a la pantalla de inicio de sesión.</p> <p>Pruebas que realizar:</p> <ul style="list-style-type: none">- Pulsar el botón "Siguiente".- Verificar que se abra SesiónActivity.

Conclusiones

Durante el desarrollo de la aplicación Sevillanas Maneras, se ha cumplido con el objetivo principal de proporcionar una herramienta interactiva, educativa y atractiva para que los usuarios puedan conocer lugares emblemáticos y tradiciones culturales de Sevilla. Se han implementado funcionalidades clave como el registro e inicio de sesión, la visualización de lugares con imágenes y descripciones, la incorporación de vídeos explicativos y la gestión de favoritos.

Uno de los objetivos iniciales era integrar la API oficial de Google Maps para mostrar mapas incrustados directamente en la aplicación. Sin embargo, esta funcionalidad no pudo completarse debido a las restricciones impuestas por Google, que exige el registro de un método de pago para el uso de esta API. Esta barrera económica resultó inviable para el entorno educativo del proyecto.

Como alternativa eficaz, se optó por integrar la YouTube Data API v3 para mostrar automáticamente vídeos relacionados con cada lugar. Esta solución ha resultado no solo efectiva, sino también más enriquecedora para el usuario final, ofreciendo contenido visual explicativo de calidad que mejora la experiencia global.

Asimismo, se añadió un botón funcional que abre la ubicación del lugar directamente en la aplicación de Google Maps (si está instalada en el dispositivo del usuario). Esta solución externa ha permitido mantener la función de geolocalización sin depender de recursos pagos, garantizando la accesibilidad total de la app.

En cuanto a las dificultades, las más destacables han sido:

- La integración inicial fallida de la API de Google Maps.
- Problemas regionales con Firebase Storage, que se resolvieron almacenando localmente las imágenes dentro del proyecto.
- La necesidad de gestionar correctamente las restricciones de uso de la clave API de YouTube para evitar bloqueos.

A pesar de estas barreras, se ha logrado entregar una aplicación funcional, estable y preparada para futuras ampliaciones. El trabajo realizado ha sido desafiante pero gratificante. Ha permitido poner en práctica conocimientos avanzados de desarrollo Android, Firebase, consumo de APIs REST, y experiencia de usuario (UX).

A nivel personal, la realización de este proyecto ha sido una experiencia muy enriquecedora tanto a nivel técnico como profesional. Me ha permitido enfrentar problemas reales, aprender a buscar soluciones prácticas y mejorar mis habilidades de programación y planificación. El resultado final representa una base sólida sobre la que se pueden seguir construyendo nuevas funcionalidades.

Vías futuras

Durante el desarrollo de esta aplicación se propusieron varias funcionalidades que, por cuestiones técnicas, económicas o de tiempo, no han podido ser incluidas en la versión final. No obstante, se consideran elementos viables para futuras iteraciones del proyecto, ya sea en un entorno educativo o productivo.

Objetivos inicialmente propuestos y no alcanzados

- Integración de la API oficial de Google Maps: Aunque inicialmente se planteó incrustar mapas interactivos dentro de la aplicación para mostrar la ubicación exacta de cada lugar, esta funcionalidad no se implementó debido a que Google requiere un método de pago para utilizar su API, lo cual no era factible en el contexto de este proyecto académico. Se optó por una solución alternativa que redirige al usuario a Google Maps mediante un Intent.
- Carga de imágenes mediante Firebase Storage: Otra funcionalidad propuesta fue almacenar y recuperar las imágenes desde Firebase Storage. Sin embargo, las restricciones regionales en cuentas ubicadas en Europa dificultaron su implementación. Como solución, se incorporaron las imágenes de forma local dentro del proyecto.

Mejoras y funcionalidades futuras recomendadas

- Búsqueda avanzada por nombre o categoría: Implementar un sistema de filtrado que permita al usuario buscar lugares específicos según criterios como nombre, tipo (cultura, tradición, monumento, etc.), o cercanía.
- Sistema de valoraciones y comentarios: Añadir una sección en cada lugar donde los usuarios registrados puedan dejar opiniones y puntuaciones, permitiendo mejorar la experiencia colectiva.
- Integración completa de mapas interactivos: Si en el futuro se dispone de los recursos necesarios, se podría integrar de forma efectiva Google Maps o, alternativamente, Mapbox, que ofrece una API gratuita con múltiples opciones de personalización.
- Notificaciones push: Permitir enviar avisos a los usuarios cuando se añadan nuevos lugares, eventos especiales o promociones turísticas.
- Modo offline con contenido descargable: Incluir la opción de descargar descripciones, imágenes o vídeos para que los usuarios puedan consultar la información sin conexión a Internet durante sus visitas.
- Audioguías personalizadas: Incorporar audios grabados o generados por texto a voz (TTS) para ofrecer descripciones orales de cada lugar, mejorando la accesibilidad de la aplicación.
- Mejora del reproductor de vídeo: Sustituir el WebView por el YouTube Android Player API, permitiendo mayor control del vídeo (pausar, adelantar, calidad, pantalla completa, etc.) y una mejor integración nativa con la interfaz de la app.

Estas mejoras permitirán que la aplicación evolucione de una herramienta informativa a una guía cultural y turística completa, altamente funcional y adaptada a las necesidades reales de los visitantes y habitantes de Sevilla.

Glosario

- **API (Application Programming Interface):** Conjunto de funciones y protocolos que permiten la comunicación entre diferentes aplicaciones. En este proyecto se ha utilizado la YouTube Data API v3 para obtener vídeos relacionados con los lugares.
- **Firebase:** Plataforma de desarrollo de aplicaciones móviles de Google que proporciona servicios como autenticación, base de datos en tiempo real y almacenamiento. Se ha utilizado Firebase Authentication para el inicio de sesión y Firebase Firestore como base de datos.
- **Firestore:** Base de datos NoSQL en la nube incluida en Firebase. Permite almacenar, sincronizar y consultar datos estructurados como documentos y colecciones.
- **Google Maps:** Servicio de mapas y geolocalización de Google. En esta app se usa para redirigir al usuario a la ubicación del lugar seleccionado, mediante enlace externo a la app instalada en el dispositivo.
- **Retrofit:** Librería de cliente HTTP usada para consumir servicios web REST. En este proyecto se emplea para realizar peticiones a la API de YouTube.
- **WebView:** Componente de Android que permite mostrar contenido web dentro de una aplicación. Se utiliza para mostrar vídeos embebidos desde YouTube.
- **SDK (Software Development Kit):** Conjunto de herramientas de desarrollo para una plataforma específica. En Android incluye librerías, emuladores y documentación necesaria para desarrollar apps.
- **ViewBinding:** Técnica de enlace entre la interfaz gráfica (XML) y el código de Kotlin que permite acceder a las vistas de manera segura y sin necesidad de findViewById.
- **YouTube Data API v3:** API de Google que permite acceder a datos de YouTube como vídeos, canales o listas de reproducción. Se usa para buscar vídeos relacionados con cada lugar desde el nombre almacenado en la base de datos.
- **Autenticación:** Proceso de verificación de la identidad de un usuario. En este caso, se realiza mediante email y contraseña con Firebase.
- **RecyclerView / ListView:** Componentes comunes en Android para mostrar listas de elementos, aunque en este proyecto no se usan directamente, sí son relevantes en proyectos similares.
- **JSON (JavaScript Object Notation):** Formato ligero para intercambio de datos, usado por la API de YouTube para enviar la respuesta con los resultados de búsqueda.
- **UI (User Interface):** Interfaz de Usuario. Hace referencia al conjunto de elementos visuales con los que interactúa el usuario.
- **UX (User Experience):** Experiencia de usuario. Engloba la percepción del usuario respecto a la facilidad y calidad de uso de la aplicación.
- **GitHub:** Plataforma de hospedaje de código que utiliza Git. Se ha usado para almacenar y versionar el código fuente del proyecto.

- **README.md:** Archivo de documentación principal de un proyecto en GitHub que contiene información esencial para instalar, ejecutar y comprender el funcionamiento de la aplicación.

Bibliografía (formato APA 7º edición)

- Google. (s.f.). *YouTube Data API v3*. Google Developers.
<https://developers.google.com/youtube/v3>
- Google. (s.f.). *Firebase Documentation*. Firebase.
<https://firebase.google.com/docs>
- Google. (s.f.). *Maps SDK for Android*. Google Developers.
<https://developers.google.com/maps/documentation/android-sdk>
- Square, Inc. (s.f.). *Retrofit*. GitHub. <https://github.com/square/retrofit>
- JetBrains. (s.f.). *Kotlin Coroutines*. Kotlin Documentation.
<https://kotlinlang.org/docs/coroutines-overview.html>
- □ Android Developers. (s.f.). *WebView*.
<https://developer.android.com/reference/android/webkit/WebView>
- OpenAI. (2025). Asistencia técnica para implementación de APIs en Android. ChatGPT.

Diagrama de Gantt FINAL

Fases Diagrama de Gantt	14/03 - 21/03	21/03 - 31/03	01/04 - 17/05	29/04 - 06/05	07/05 - 13/05	14/05 - 21/05
Análisis y planificación						
Diseño UI/UX y base de datos						
Desarrollo del backend						
Implementación de funcionalidades clave						
Pruebas y optimización						
Documentación y entrega final						

Anexos

Capturas de Pantallas de la Aplicación Final desde mi Móvil.

Se incluyen imágenes de las distintas pantallas principales de la aplicación para ilustrar la interfaz de usuario:

Pantalla de inicio y logo



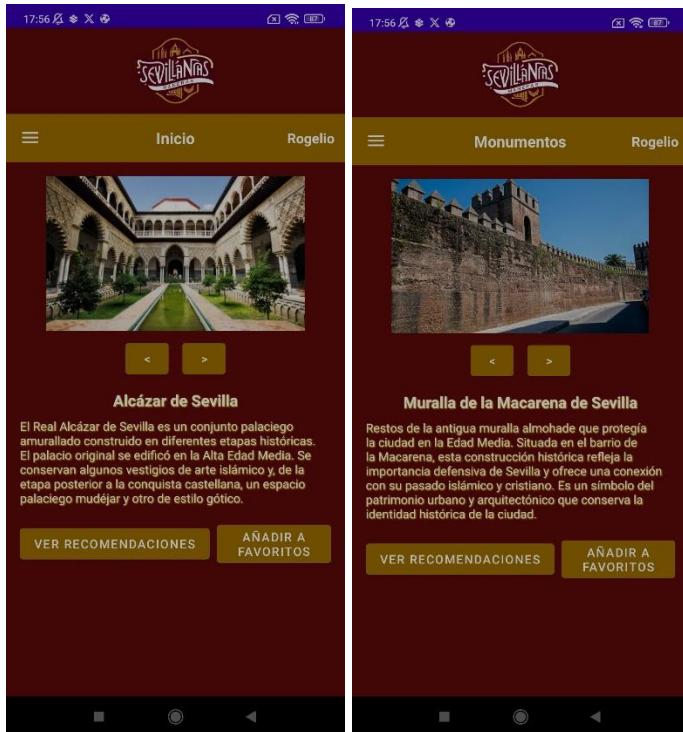
Pantalla de login y registro



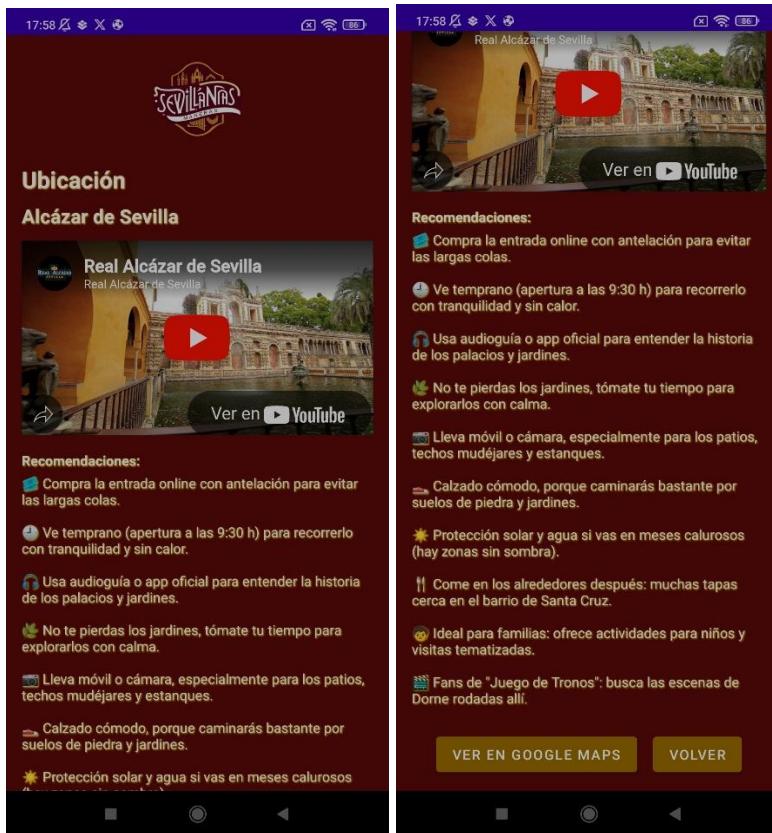
Menú lateral (drawer)



Vista principal con carrusel de lugares, detalle del lugar con imagen, descripción y botones



Pantalla de ubicación con recomendaciones y vídeo insertado desde YouTube con botón para abrir Google Maps



Perfil del usuario con datos personales y favoritos



Añade y elimina de favoritos

