

# MERCAELX



TOURISM IN ELCHE

**Proyecto final de grado**  
**Desarrollo de Aplicaciones Multiplataforma**

**Autor: Pablo López Trives**  
**Tutora: Patricia Martí Gran**  
**Centro: IES Severo Ochoa**

## Sumario

1. Introduction.....	3
1. Introducción.....	4
1.1 Justificación.....	4
1.2 Objetivos.....	5
2. Casos de uso.....	6
Caso de Uso 1: Consultar puntos turísticos cercanos.....	6
Caso de Uso 2: Ver información detallada de un lugar.....	6
Caso de Uso 3: Seguir una ruta turística autoguiada.....	7
Caso de Uso 4: Descubrir recomendaciones por categoría.....	8
3. Diagrama de Gantt.....	9
4. Diagrama Base de datos.....	10
4.1 Estructura general.....	10
4.2 Modelo de datos.....	10
4.3 Ventajas del diseño actual.....	11
5. Tecnologías usadas.....	12
1. Android Studio + Kotlin + KAPT.....	12
2. Google Maps SDK.....	12
3. Firebase Firestore.....	12
4. Firefoo.....	13
5. Multilenguaje (strings.xml).....	13
6. Material Design y diseño responsivo.....	13
6. Viabilidad del Proyecto MERCAELX.....	14
6.1 Viabilidad Técnica.....	14
6.2 Viabilidad Económica.....	14
6.3 Viabilidad Legal y de Privacidad.....	15
6.4 Viabilidad Social y Medioambiental.....	15
6.5 Sostenibilidad y Futuro.....	15
7. Aspectos Transversales y Medioambientales.....	17
7.1 Igualdad, inclusión y accesibilidad.....	17
7.2 Desarrollo sostenible y medio ambiente.....	17
7.3 Formación y competencias digitales.....	18
8. Dificultades encontradas y soluciones aplicadas.....	19
8.1 Implementación del multilenguaje con soporte dinámico.....	19
8.2 Mostrar marcadores multilingües desde Firebase Firestore.....	20
8.3 Sincronización de la API de Google Maps con Firebase y filtrado por categorías.....	21
8.4 Integración de la API de direcciones de Google para trazado de rutas.....	22
8.5 Optimización del rendimiento al manejar muchos marcadores en tiempo real.....	23
8.6. Gestión de permisos de ubicación y experiencia de usuario.....	24
8.7 Herramientas de apoyo utilizadas.....	25
9 Arquitectura de la aplicación MercaElx.....	26
9.1 Capa de Presentación (UI / Frontend).....	26
9.2 Capa de Lógica de Negocio.....	26
9.3 Capa de Datos.....	26
9.4 Servicios externos.....	27
9.5 Gestión de idioma (Internacionalización).....	27

9.6 Flujo general.....	27
10 Pruebas Realizadas.....	29
10.1 Pruebas en el emulador de Android Studio.....	29
10.2 Pruebas en dispositivos reales.....	29
Localización.....	29
Acceso a internet y Firebase.....	30
Cambio de idioma.....	30
Interacciones.....	30
11 Visuales.....	31
12. Bibliografía.....	35

# 1. Introduction

The MERCAELX project is an Android application designed to promote the cultural and gastronomic richness of the city of Elche (Spain). The goal is to provide both tourists and local residents with an easy-to-use and multilingual platform where they can explore cultural landmarks, natural sites, and a wide variety of restaurants tailored for all budgets.

This app emerged from the idea of bringing together technology and local heritage to make discovering Elche more accessible, modern, and engaging. Users can view points of interest on a map, filter by category (religious, cultural, natural, or gastronomy), obtain detailed information in several languages, and even trace walking routes to their selected destinations.

MERCAELX aims to be not only a practical tool for visitors, but also a digital showcase of what the city has to offer. It reflects the importance of innovation applied to tourism, combining geolocation, Firebase technology, and user-friendly design with a real-world application that could be extended to other cities in the future.

# 1.Introducción

## 1.1 Justificación

Esta es una idea propuesta por el centro para crear una aplicación de turismo sobre Elche porque siento que esta ciudad tiene muchísimo potencial turístico que muchas veces pasa desapercibido, además de que me han dado toda la libertad creativa. Desde pequeño he vivido rodeado de lugares que tienen un valor histórico, cultural y natural enorme, pero me he dado cuenta de que no siempre se les da la promoción o el reconocimiento que merecen. Por eso, me gustaría aportar mi granito de arena y utilizar la tecnología para acercar Elche tanto a los visitantes como a los propios habitantes.

Elche tiene elementos únicos en el mundo, como el Palmeral, que es Patrimonio de la Humanidad, o el Misteri d'Elx, que también lo es, y que forma parte de nuestra identidad cultural. Además, hay muchísimos otros rincones interesantes: museos, parques, rutas naturales, gastronomía típica, fiestas populares... Sin embargo, no siempre es fácil encontrar toda esta información de forma clara, accesible y bien organizada. Muchas veces está repartida en webs poco intuitivas y con sobrecarga de anuncios o en redes sociales desactualizadas.

Por eso, quiero que esta aplicación sea simple, directa y útil. Nada de registros, cuentas ni procesos largos. La mayoría de las personas que están de viaje no quieren complicarse con contraseñas o correos de confirmación; lo que quieren es abrir la app y saber qué pueden hacer ese día, cerca de donde están. Al no requerir una cuenta, la experiencia será más rápida, más respetuosa con la privacidad del usuario y mucho más amigable. Que cualquiera, incluso alguien mayor o con poca experiencia tecnológica, pueda usarla sin problemas.

Además, lo hago también porque me gustaría crear algo que tenga impacto real en mi entorno. Es fácil desarrollar ideas pensando en grandes ciudades o en modas, pero a veces lo más valioso es mirar alrededor y ver qué se puede mejorar o aportar en tu propio entorno. Esta aplicación sería mi forma de unir mi interés por la tecnología con el cariño que siento por Elche. Me encantaría que sirviera para que más gente descubra sus encantos, disfrute su cultura y valore todo lo que la ciudad ofrece.

Y aunque parezca un proyecto sencillo, creo que tiene mucho potencial. Podría incluir rutas personalizadas, geolocalización, información útil para cada época del año, y sobre todo, una experiencia cuidada y accesible para todos.

En definitiva, quiero hacer esta aplicación porque creo en el valor de lo local, en el poder de la tecnología sencilla y bien pensada, y porque siento que Elche se merece una herramienta moderna que la muestre al mundo como el lugar tan especial que es.

## 1.2 Objetivos

Mi objetivo principal es crear una app de turismo para Elche que cualquier persona pueda usar de forma rápida y sin complicaciones, sin necesidad de registrarse ni dar datos personales. Quiero que sea algo útil de verdad, que ayude tanto a turistas como a gente de aquí a descubrir lo mejor de la ciudad.

- Hacer una aplicación sencilla y directa. No quiero que la gente se frustre con menús complicados o pasos innecesarios. Mi idea es que entres, veas qué hay para hacer cerca de ti, y salgas a disfrutar.
- Aprovechar la ubicación del usuario para hacer recomendaciones útiles. Por ejemplo, si alguien está en el centro, que pueda ver qué lugares interesantes tiene a pocos minutos caminando.
- Tener toda la información práctica al alcance. Horarios, precios, fotos, consejos... todo lo necesario para que el visitante pueda organizar su visita sin perder tiempo buscando en distintas webs.
- Añadir rutas y mapas interactivos. Me gustaría que la app permita seguir recorridos turísticos sencillos, con explicaciones breves y curiosas, sin necesidad de guía o conexión constante.
- Diseñar una app que sea visual y fácil de usar para todo el mundo. No solo para jóvenes acostumbrados a la tecnología, sino también para familias o personas mayores que quieran moverse por la ciudad sin complicaciones.
- Crear una base sólida para seguir mejorando. Me interesa que la app esté bien hecha desde el principio, con una estructura que me permita seguir añadiendo

cosas en el futuro, como nuevos idiomas, modo sin conexión o integración con transporte público.

- Apoyar el turismo local de una forma más consciente y equilibrada. Me gustaría que la app también sirviera para que la gente descubra lugares menos conocidos y para que los negocios locales tengan más visibilidad.

## 2.Casos de uso

### **Caso de Uso 1: Consultar puntos turísticos cercanos**

#### **Actor principal:**

Usuario (turista o local)

#### **Descripción:**

El usuario abre la aplicación y visualiza los lugares turísticos cercanos gracias a la geolocalización.

#### **Flujo principal:**

- El usuario abre la app.
- La aplicación solicita permiso para usar la ubicación.
- Se muestran en un mapa o lista los puntos turísticos más cercanos.
- El usuario selecciona uno para ver más detalles.

#### **Resultado esperado:**

El usuario encuentra lugares interesantes cerca y sabe cómo llegar a ellos.

### **Caso de Uso 2: Ver información detallada de un lugar**

#### **Actor principal:**

Usuario

**Descripción:**

El usuario selecciona un punto de interés y accede a una ficha con toda la información útil.

**Flujo principal:**

- El usuario selecciona un lugar desde el mapa o la lista.
- Se muestra una ficha con:
  - Nombre del lugar
  - Descripción
  - Horarios
  - Precios
  - Fotografías
  - Cómo llegar

**Resultado esperado:**

El usuario dispone de toda la información necesaria para decidir si desea visitar el sitio.

**Caso de Uso 3: Seguir una ruta turística autoguiada****Actor principal:**

Usuario

**Descripción:**

El usuario elige una ruta turística sugerida por la app y la recorre a su ritmo, sin necesidad de guía.

**Flujo principal:**

- El usuario entra en la sección “Rutas”.
- Selecciona una ruta (por ejemplo, “Ruta del centro histórico”).
- Aparece un mapa interactivo con los puntos marcados.
- Al llegar a cada punto, el usuario puede leer una explicación corta y visual.

**Resultado esperado:**

El usuario disfruta de una visita completa y guiada sin depender de terceros.

**Caso de Uso 4: Descubrir recomendaciones por categoría****Actor principal:**

Usuario

**Descripción:**

El usuario desea explorar lugares según sus intereses (naturaleza, cultura, gastronomía...).

**Flujo principal:**

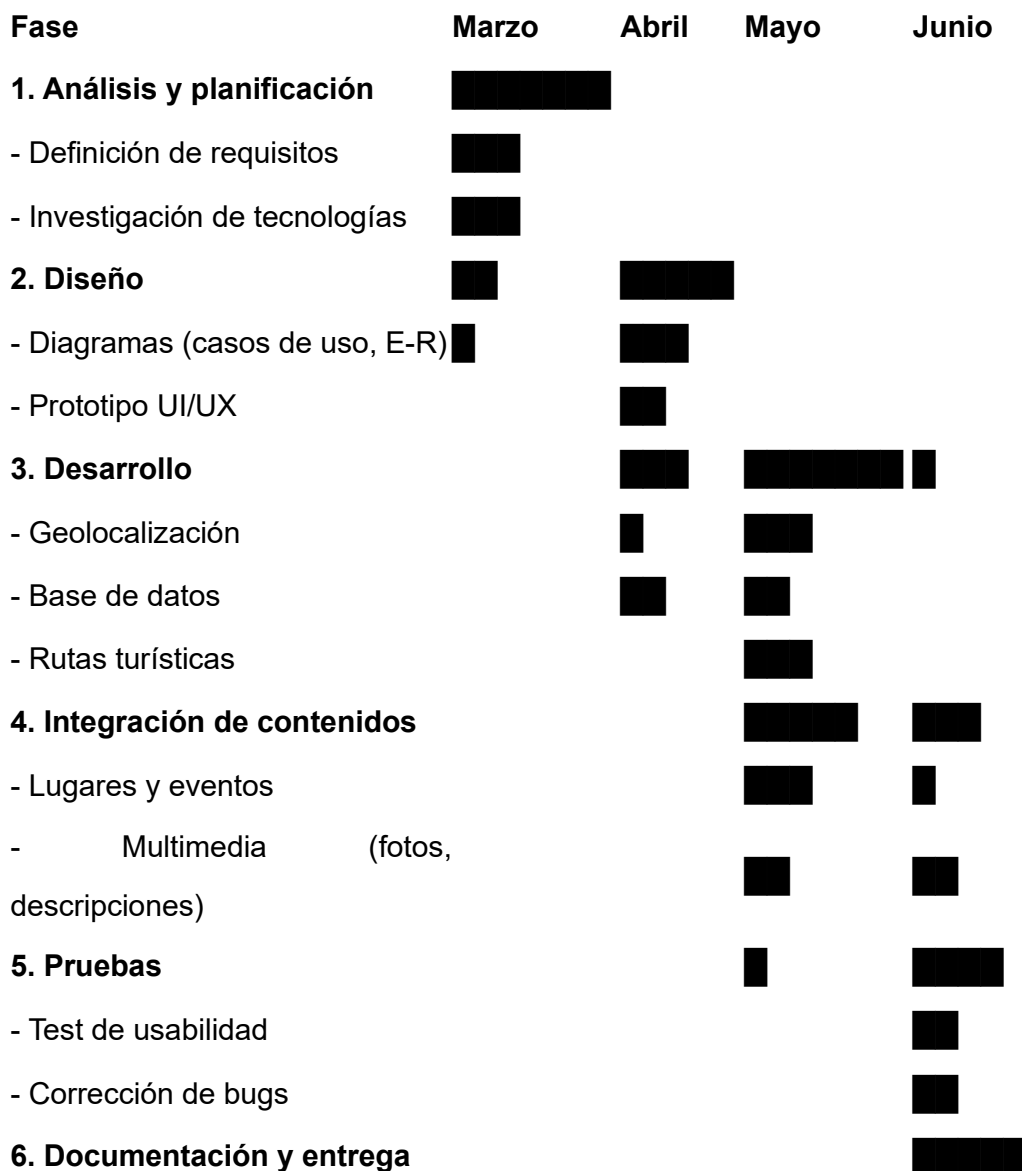
- El usuario accede a la sección de categorías.
- Escoge una (por ejemplo, “Restaurantes”).
- La aplicación muestra los lugares que coinciden con esa temática.
- El usuario selecciona uno para ver su información detallada.

**Resultado esperado:**

El usuario descubre lugares adaptados a lo que más le interesa hacer en ese momento.



## 3. Diagrama de Gantt



## 4. Diagrama Base de datos

Key	Value	Type
MercaDix		Collection
Lugares		Document
LugaresArray		Array
0	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Garten mit Tausenden von Palmen, zum Nationalen K...	Map
1	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Alte Festung, Sitz des MAHE, ein Ancient fortress, ho...	Map
2	{categoria: {de: Religios, en: Religious, es: Religioso, fr: Religieux, ru: Религиозный, zh: 宗教}, descripcion: {de: Barocker Tempel, in dem das Misteri d'Elx gefeie...	Map
3	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Ethnologisches Museum über das ländliche Leben de...	Map
4	{categoria: {de: Natürlich, en: Natural, es: Natural, fr: Naturel, ru: Природный, zh: 自然}, descripcion: {de: Großer Palmenhain, UNESCO-Weltkulturerbe, ein Lar...	Map
5	{categoria: {de: Natürlich, en: Natural, es: Natural, fr: Naturel, ru: Природный, zh: 自然}, descripcion: {de: Naturschutzgebiet mit Wanderwegen und Vogelbeob...	Map
6	{categoria: {de: Religios, en: Religious, es: Religioso, fr: Religieux, ru: Религиозный, zh: 宗教}, descripcion: {de: Historische Kirche mit barocker Architektur, en: ...	Map
7	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Museum über die Geschichte des Palmerals, ein Mus...	Map
8	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Archäologischer Park mit bedeutenden iberischen un...	Map
9	{categoria: {de: Natürlich, en: Natural, es: Natural, fr: Naturel, ru: Природный, zh: 自然}, descripcion: {de: Naturschutzgebiet mit Dünen, Feuchtgebieten und V...	Map
10	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Museum mit einer Sammlung paläontologischer Foss...	Map
11	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Museum für zeitgenössische Kunst mit Werken span...	Map
12	{categoria: {de: Cultural, en: Cultural, es: Cultural, fr: Cultural, ru: Культурный, zh: 文化}, descripcion: {de: Historisches Gebäude mit einer Sammlung von Doku...	Map
13	{categoria: {de: Restaurant, en: Restaurant, es: Restaurante, fr: Restaurant, ru: Ресторан, zh: 餐厅}, descripcion: {de: Traditionelle spanische Küche mit Tapas-Spe...	Map
14	{categoria: {de: Restaurant, en: Restaurant, es: Restaurante, fr: Restaurant, ru: Ресторан, zh: 餐厅}, descripcion: {de: Veganer Bistro mit frischen, saisonalen Zuta...	Map
15	{categoria: {de: Restaurant, en: Restaurant, es: Restaurante, fr: Restaurant, ru: Ресторан, zh: 餐厅}, descripcion: {de: Indisches Restaurant mit Tandoori-Gerichte...	Map
16	{categoria: {de: Restaurant, en: Restaurant, es: Restaurante, fr: Restaurant, ru: Ресторан, zh: 餐厅}, descripcion: {de: Modernes Fusion-Restaurant mit internatio...	Map
17	{categoria: {de: Fast Food, en: Fast Food, es: Comida rápida, fr: Restauration rapide, ru: Фастфуд, zh: 快餐}, descripcion: {de: Beliebter Ort für Döner und Falafel ...	Map
18	{categoria: {de: Spanisch, en: Spanish, es: Española, fr: Espagnole, ru: Испанская, zh: 西班牙菜}, descripcion: {de: Authentisches Tapas-Restaurant mit lokalen S...	Map
19	{categoria: {de: Vegetarisch, en: Vegetarian, es: Vegetariana, fr: Végétarienne, ru: Вегетарианская, zh: 素食}, descripcion: {de: Gesundes vegetarisches Essen mit...	Map

### 4.1 Estructura general

La base de datos utilizada es NoSQL (Firebase Firestore) y está orientada a documentos. La colección principal se llama "Lugares" y contiene un único documento que posee el array lugaresArray, en el que cada entrada representa un lugar turístico o restaurante.

### 4.2 Modelo de datos

Cada objeto del array lugaresArray incluye los siguientes campos:

- **nombre:** objeto multilingüe (es, en, fr, de, ru, zh) con el nombre del lugar.
- **descripcion:** objeto multilingüe con la descripción del lugar.
- **direccion:** objeto multilingüe con la dirección física del lugar.
- **horario:** string que indica las horas de apertura.
- **precio:** objeto multilingüe con el precio o indicación de gratuidad.
- **categoria:** objeto multilingüe con el tipo de lugar (Cultural, Religioso, Restaurante, Natural...).
- **imageUrl:** string con la URL de una imagen representativa.

- **latitud:** objeto con campos `__lat__` y `__lon__`, que representan las coordenadas geográficas.

### 4.3 Ventajas del diseño actual

- **Multilingüismo integrado:** todos los campos relevantes están localizados, lo cual permite mostrar la información automáticamente en el idioma del usuario.
- **Flexibilidad:** al estar en Firestore, es posible modificar o añadir lugares dinámicamente sin necesidad de reestructurar la base.
- **Escalabilidad:** se pueden añadir fácilmente nuevas categorías o idiomas.
- **Uso geográfico:** las coordenadas permiten representar los lugares en Google Maps y calcular rutas.

## **5. Tecnologías usadas**

El desarrollo de MERCAELX se ha apoyado en una combinación de tecnologías modernas que permiten construir una aplicación robusta, fluida, escalable y mantenible. A continuación, se detallan las herramientas y librerías utilizadas, así como las razones por las que han sido seleccionadas:

### **1. Android Studio + Kotlin + KAPT**

La aplicación se ha desarrollado con Android Studio, el entorno oficial para apps Android, utilizando el lenguaje Kotlin, recomendado por Google por su sintaxis concisa, mayor seguridad frente a errores y compatibilidad total con Java. Además, se ha usado kapt (Kotlin Annotation Processing Tool) para procesar anotaciones necesarias en algunas bibliotecas como Firebase o Room, mejorando la integración y automatización en tiempo de compilación.

### **2. Google Maps SDK**

Para integrar el mapa y mostrar marcadores interactivos, se ha utilizado el Google Maps SDK para Android, lo que permite al usuario visualizar su ubicación, los lugares turísticos, restaurantes y obtener rutas. Su versatilidad y soporte oficial garantizan una experiencia fluida y fiable para la navegación en tiempo real.

### **3. Firebase Firestore**

El almacenamiento de los lugares (turísticos y gastronómicos) se gestiona a través de Firebase Firestore, una base de datos NoSQL en la nube, escalable y en tiempo real. Permite almacenar datos estructurados con soporte multilenguaje, acceso instantáneo y actualizaciones sin necesidad de servidores backend.

## 4. Firefoo

Para el manejo avanzado de la base de datos Firestore, se ha utilizado Firefoo, una herramienta de escritorio que permite visualizar, editar y exportar datos fácilmente desde Firestore. Esta herramienta ha sido clave para organizar los datos por categoría, introducir multilingüismo, y mantener una estructura clara y limpia sin errores de formato.

## 5. Multilenguaje (strings.xml)

Se ha implementado soporte multilingüe completo (español, inglés, alemán, francés, ruso y chino) utilizando los archivos de recursos strings.xml, lo que permite que tanto la interfaz como los datos adaptables (nombre, descripción, categoría...) se muestren automáticamente en el idioma del dispositivo del usuario.

## 6. Material Design y diseño responsivo

La aplicación sigue los principios de Material Design, utilizando componentes como CardView, FloatingActionButton, AlertDialog y menús desplegables. Esto ofrece una experiencia moderna, intuitiva y coherente en todos los dispositivos Android.

# 6. Viabilidad del Proyecto MERCAELX

## 6.1 Viabilidad Técnica

La aplicación MERCAELX ha demostrado una alta viabilidad técnica desde su concepción hasta su implementación final. Está desarrollada para la plataforma Android, el sistema operativo móvil más utilizado en España y en el mundo, lo que garantiza una amplia base de usuarios potenciales. El uso de tecnologías modernas, como Firebase Firestore, Google Maps SDK, FusedLocationProviderClient y Kotlin, ha permitido construir una solución robusta, escalable y fácilmente mantenible.

Firebase proporciona una infraestructura en la nube completamente gestionada que simplifica el almacenamiento, la autenticación (si se desea ampliar en el futuro) y la sincronización en tiempo real. Además, el uso de Firefoo ha optimizado la gestión avanzada de datos en Firestore, lo que ha permitido mantener un control preciso y rápido sobre la base de datos durante el desarrollo.

Google Maps ofrece una API potente para la representación geográfica y la interacción con el usuario mediante marcadores, rutas y geolocalización. Estas herramientas han sido ampliamente probadas, tienen documentación actualizada y una comunidad activa, lo que respalda su fiabilidad a largo plazo.

La arquitectura de la aplicación está basada en principios de modularidad, mantenimiento y escalabilidad, permitiendo añadir nuevas funciones sin rehacer el núcleo. Está preparada para ampliarse, por ejemplo, incluyendo funciones como favoritos, reseñas de usuarios, rutas personalizadas, accesibilidad, entre otras.

## 6.2 Viabilidad Económica

Desde el punto de vista económico, MERCAELX también presenta una viabilidad muy positiva, sobre todo en una primera fase de lanzamiento:

- **Costes iniciales:** El desarrollo se ha realizado con herramientas gratuitas y de código abierto (Android Studio, Firebase plan gratuito, Google Maps con uso limitado sin coste).
- **Escalabilidad económica:** Si el número de usuarios aumenta, Firebase permite ampliar su plan progresivamente en función del uso (pago por uso), por lo que el gasto se adapta al crecimiento.

- **Rentabilidad:** La aplicación podría monetizarse mediante publicidad local, acuerdos con restaurantes y tiendas o versiones premium con funcionalidades exclusivas. También es posible optar a subvenciones o ayudas locales al turismo o la digitalización de negocios.

## 6.3 Viabilidad Legal y de Privacidad

La aplicación no recopila información personal del usuario, ni realiza registro o autenticación, lo que simplifica enormemente su cumplimiento con el Reglamento General de Protección de Datos (RGPD). Todos los datos mostrados son públicos (nombre del lugar, ubicación, imagen, categoría) y no se almacena ninguna acción del usuario.

En caso de futuras ampliaciones con cuentas, valoraciones o historial, ya se ha considerado el uso de Firebase Authentication y el cifrado seguro de datos sensibles.

Además, la información sobre lugares ha sido recopilada de fuentes abiertas o generada manualmente para el proyecto, asegurando el cumplimiento de derechos de autor y licencias de uso de imágenes, con prioridad por contenidos de dominio público o con licencia Creative Commons.

## 6.4 Viabilidad Social y Medioambiental

Desde un punto de vista social, MERCAELX fomenta el turismo local, el consumo responsable en restaurantes variados (de todos los bolsillos) y el descubrimiento del patrimonio histórico y natural de Elche, promoviendo una ciudad más conectada y accesible tanto para residentes como visitantes.

Además, al evitar el uso de recursos físicos (folletos, mapas impresos, cartelería), contribuye a la reducción del impacto medioambiental, alineándose con los objetivos de desarrollo sostenible y las políticas digitales del entorno municipal.

También se ha desarrollado con una interfaz accesible, con soporte multilingüe en seis idiomas (español, inglés, francés, alemán, ruso y chino), lo cual mejora la inclusión turística y social.

## 6.5 Sostenibilidad y Futuro

MERCAELX ha sido diseñada para ser autosostenible y evolutiva. Gracias a su estructura modular, las actualizaciones son fáciles de implementar. El uso de servicios cloud

garantiza su mantenimiento sin necesidad de infraestructura propia, y el coste de servidor se adapta al tráfico.

Su valor diferencial es combinar geolocalización, turismo, gastronomía y multilingüismo en una app sencilla, visual y accesible. Con apoyo institucional, tiene potencial para convertirse en una herramienta oficial de promoción local.

En resumen, MERCAELX es una aplicación técnicamente sólida, económicamente viable, socialmente útil y respetuosa con la privacidad y el medio ambiente, con margen claro de expansión y evolución. Todo esto respalda su viabilidad como proyecto real y aplicable en un contexto urbano como el de Elche.



## 7. Aspectos Transversales y Medioambientales

### 7.1 Igualdad, inclusión y accesibilidad

MERCAELX ha sido concebida como una aplicación abierta, inclusiva y accesible para todo tipo de usuarios, sin distinción de edad, género, nacionalidad o capacidades.

Uno de los pilares del proyecto es su fuerte compromiso con la diversidad lingüística. La aplicación cuenta con soporte multilingüe completo en seis idiomas: español, inglés, francés, alemán, ruso y chino, lo que la hace accesible tanto para la población local como para turistas internacionales.

Gracias a este enfoque, MERCAELX se adapta automáticamente al idioma configurado por el dispositivo o elegido en la configuración interna, facilitando su uso a personas que no hablan español y promoviendo la integración y participación de comunidades extranjeras en la ciudad.

Además, la interfaz es intuitiva, con buen contraste visual, textos claros y botones accesibles, lo que contribuye a la inclusión de personas con menor experiencia digital o limitaciones visuales. En futuras versiones se prevé incorporar características adicionales como navegación por voz o compatibilidad con lectores de pantalla.

### 7.2 Desarrollo sostenible y medio ambiente

La app contribuye a los Objetivos de Desarrollo Sostenible (ODS) promoviendo un modelo de turismo más respetuoso con el entorno y más eficiente:

- Disminución del uso de papel: elimina la necesidad de mapas impresos, folletos o guías físicas.
- Fomento del comercio local: visibiliza restaurantes y espacios culturales del municipio.
- Optimización del transporte: gracias al trazado de rutas a pie y la geolocalización, reduce desplazamientos innecesarios o el uso de vehículos contaminantes.
- Uso de servicios cloud eficientes: el almacenamiento y gestión de datos mediante Firebase evita infraestructuras locales con alto consumo energético.

## 7.3 Formación y competencias digitales

El desarrollo del proyecto ha potenciado competencias clave del siglo XXI como:

- El uso responsable de la tecnología.
- La gestión autónoma de proyectos.
- La resolución de problemas reales mediante herramientas digitales.
- La conciencia ambiental en la toma de decisiones técnicas.

En resumen, MERCAELX es un proyecto transversal que aúna inclusividad, sostenibilidad, tecnología y utilidad social, alineándose con los valores de la formación profesional moderna.

## 8. Dificultades encontradas y soluciones aplicadas

Durante el desarrollo de MERCAELX, surgieron diversos retos técnicos que requirieron análisis, pruebas y soluciones específicas para garantizar el correcto funcionamiento de la aplicación. A continuación, se describen las principales dificultades y las estrategias adoptadas para resolverlas:

### 8.1 Implementación del multilenguaje con soporte dinámico

Uno de los principales objetivos del proyecto era ofrecer una experiencia inclusiva y accesible para usuarios de diferentes nacionalidades. Para ello, se planteó una arquitectura multilenguaje con soporte para seis idiomas: español, inglés, alemán, francés, ruso y chino. Sin embargo, surgieron varias dificultades:

- **Problema:** Aunque se configuraban correctamente los strings.xml en cada idioma, al cambiar de idioma en la actividad de ajustes (SettingsActivity) y volver a MainActivity, el idioma no se mantenía.
- **Solución:** Se implementó una función personalizada `applyAppLocale(context: Context)` que se invoca tanto en `attachBaseContext()` como en `onResume()` para forzar la configuración del idioma seleccionado cada vez que se inicia o se reanuda la actividad. Se usa `SharedPreferences` para almacenar la preferencia del idioma de forma persistente.

```
fun applyAppLocale(context: Context): Context {  
    val prefs = context.getSharedPreferences("settings", Context.MODE_PRIVATE)  
    val lang = prefs.getString("language", "es") ?: "es"  
    val locale = Locale(lang)  
    Locale.setDefault(locale)  
  
    val config = Configuration()  
    config.setLocale(locale)  
    return context.createConfigurationContext(config)  
}
```

Además, se aplicó esta lógica también en los fragments y en otras actividades secundarias como `LugarDetalleActivity`, lo cual evitó comportamientos inconsistentes en el cambio de idioma.

## 8.2 Mostrar marcadores multilingües desde Firebase Firestore

MERCAELX almacena los datos de los lugares en Firestore con estructura multilingüe (campos nombre, descripcion, categoria, etc. son mapas de claves con idioma). Integrar esta estructura en el mapa fue un reto:

- **Problema:** Al recuperar los marcadores desde Firestore y mostrarlos en el mapa, solo se mostraba el valor por defecto (normalmente en español), sin tener en cuenta el idioma del sistema o seleccionado.
- **Solución:** En la función loadMarkers(), se identificó el idioma del dispositivo mediante Locale.getDefault().language y se utilizó para extraer el contenido localizado de cada campo multilingüe. En caso de no existir traducción específica, se usaba el valor en español como fallback.

```
val nombreMap = lugar["nombre"] as? Map<String, String>
val nombre = nombreMap?.get(lang) ?: nombreMap?.get("es") ?: getString(R.string.nombre_no_disponible)
```

Esto permitió que los marcadores y los datos que aparecen en las ventanas de detalle se adaptaran al idioma del usuario en tiempo real.

```
{
  "categoria": {
    "de": "Kulturell",
    "en": "Cultural",
    "es": "Cultural",
    "fr": "Culturel",
    "ru": "Культурный",
    "zh": "文化"
  },
}
```

## 8.3 Sincronización de la API de Google Maps con Firebase y filtrado por categorías

Uno de los valores añadidos de la app era mostrar los lugares geolocalizados en un mapa, integrando Firebase y Google Maps. Los principales retos fueron:

- **Problema:** Inicialmente, los cambios en la base de datos de Firebase no se reflejaban automáticamente en la app, o se mostraban datos obsoletos en caché.
- **Solución:** Se rediseñó el método `loadMarkers()` para eliminar los marcadores existentes y recargar la información en cada `onResume()`, lo que garantiza una sincronización actualizada con Firestore.

```
for (marker in todosLosMarkers) {
    marker.remove()
}
todosLosMarkers.clear()
loadMarkers()
}
```

- **Problema adicional:** Implementar un sistema para filtrar por categoría (por ejemplo, mostrar solo lugares religiosos, culturales o restaurantes) a partir de la estructura JSON en Firebase.
- **Solución:** Se desarrolló una función `filtrarMarkersPorCategoria()` que, mediante un botón y un diálogo con categorías, oculta o muestra los marcadores según su campo `categoria` traducido dinámicamente. Esto se adapta automáticamente al idioma actual del sistema o seleccionado por el usuario.

```
private fun filtrarMarkersPorCategoria(categoria: String) {
    todosLosMarkers.forEach { it.isVisible = false }

    if (categoria == "todos") {
        todosLosMarkers.forEach { it.isVisible = true }
    } else {
        todosLosMarkers.forEach { marker ->
            val lugar = marker.lugar as? Map<*, *>
            val categoriaMap = lugar?.get("categoria") as? Map<*, *>
            val categoriaMarker = categoriaMap?.get(Locale.getDefault().language) as? String
            ?: categoriaMap?.get("es") as? String

            if (categoriaMarker?.lowercase()?.contains(categoria) == true) {
                marker.isVisible = true
            }
        }
    }
}
```

## 8.4 Integración de la API de direcciones de Google para trazado de rutas

- **Problema:** Obtener rutas entre la ubicación actual y un destino implicaba conectarse a la API de Directions de Google, decodificar la polilínea y manejar errores de red.
- **Solución:** Se creó la función `drawRouteTo()` que realiza una petición HTTP en un hilo separado, parsea la respuesta JSON, decodifica la polilínea y la dibuja sobre el mapa. Se contemplan casos de error con try-catch y se muestra feedback al usuario.

```
if (status == "OK") {
    val points = json.getJSONArray("routes")
        .getJSONObject(0)
        .getJSONObject("overview_polyline")
        .getString("points")

    val decodedPath = decodePoly(points)
    runOnUiThread {
        currentPolyline?.remove()
        currentPolyline = map.addPolyline(
            PolylineOptions()
                .addAll(decodedPath)
                .color(ContextCompat.getColor(this, R.color.elche_green))
                .width(10f)
        )
    }
} else {
    runOnUiThread {
        Toast.makeText(this, "Error en la respuesta de la API: $status", Toast.LENGTH_LONG).show()
    }
}
```

## 8.5 Optimización del rendimiento al manejar muchos marcadores en tiempo real

- **Problema:** Al cargar gran cantidad de marcadores (más de 40 lugares en total), la interacción con el mapa se volvía más lenta, y el rendimiento en dispositivos de gama baja era notablemente afectado.
- **Solución:** Se adoptaron varias estrategias para mejorar la eficiencia:
  - Los marcadores solo se cargan una vez en `onResume()` y se almacenan en la lista `todosLosMarkers`, evitando recargas innecesarias.
  - Se implementó el filtrado por categorías mediante visibilidad (`marker.isVisible = false/true`) en lugar de destruir y recrear marcadores, lo cual es mucho más eficiente.
  - Se limitó el área visible del mapa a los límites geográficos de Elche mediante `LatLngBounds`, lo que reduce la carga innecesaria de tiles y gestos fuera de contexto.

```
override fun onResume() {  
    super.onResume()  
  
    applyAppLocale(this)  
  
    for (marker in todosLosMarkers) {  
        marker.remove()  
    }  
    todosLosMarkers.clear()  
    loadMarkers()  
}
```

## 8.6. Gestión de permisos de ubicación y experiencia de usuario

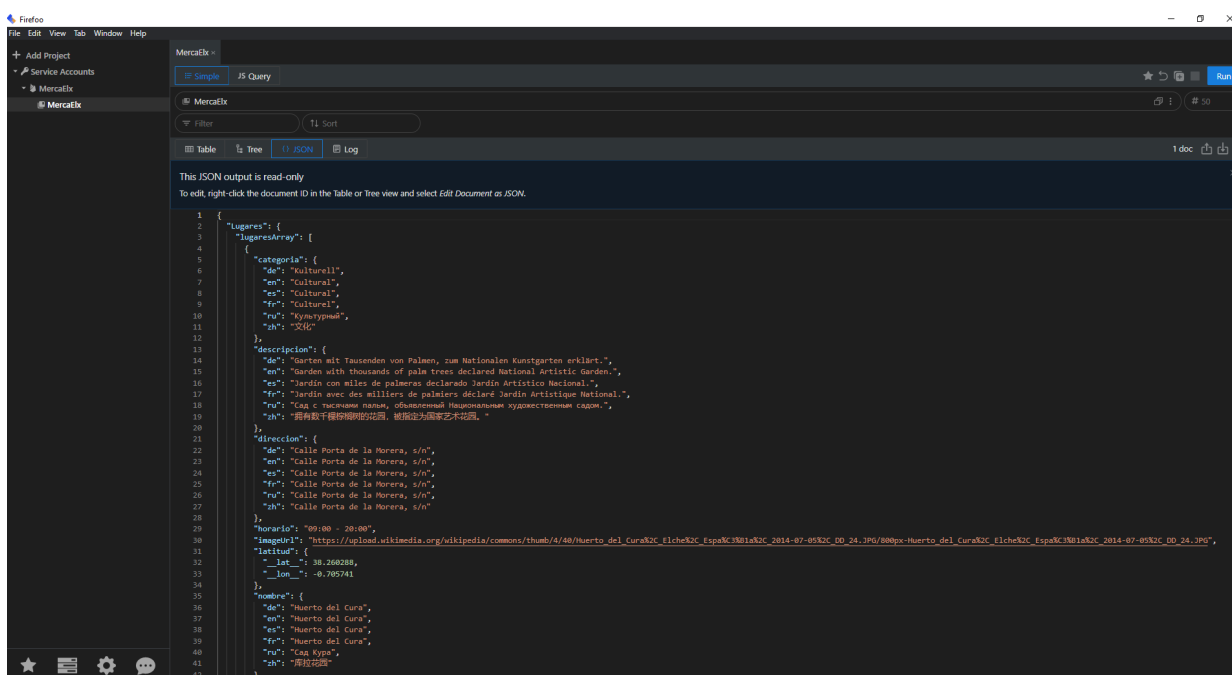
- **Problema:** Al iniciar la app, si el usuario denegaba los permisos de ubicación, algunas funciones quedaban inutilizables (como mostrar la ruta, lugares cercanos o centrar el mapa). Esto afectaba negativamente a la usabilidad.
- **Solución:**
  - Se implementó una comprobación exhaustiva de permisos con `ActivityCompat.requestPermissions`, mostrando mensajes claros cuando el permiso es denegado o debe activarse desde ajustes.
  - Se añadieron mensajes de Toast explicativos, y se diseñó la app para seguir funcionando parcialmente incluso sin permisos (por ejemplo, se puede seguir viendo el mapa y los marcadores, pero no trazar rutas).
  - También se reorganizó la lógica para que `enableMyLocation()` sea robusta y condicional, evitando errores por falta de permisos.

```
@SuppressWarnings("MissingPermission")
private fun enableMyLocation() {
    if (!::map.isInitialized) return
    if (isPermissionsGranted()) {
        map.isMyLocationEnabled = true
        centerMapOnMyLocation()
        startLocationUpdates()
    } else {
        requestLocationPermission()
    }
}
```



## 8.7 Herramientas de apoyo utilizadas

- Firefoo fue clave para gestionar y visualizar de forma eficiente los datos en Firestore, sobre todo en fase de desarrollo y pruebas.
- KAPT (Kotlin Annotation Processing Tool) se utilizó para habilitar características de procesamiento de anotaciones en bibliotecas como Glide o Room, aunque en este proyecto fue usado principalmente con las dependencias de Firebase y para la preparación de futuras extensiones.



# 9 Arquitectura de la aplicación MercaElx

La arquitectura de MERCAELX está diseñada siguiendo una estructura modular, escalable y clara que combina varios componentes clave para garantizar el correcto funcionamiento de la aplicación. A continuación, se describen los bloques principales:

## 9.1 Capa de Presentación (UI / Frontend)

- **Actividades de Android (MainActivity, LugarDetalleActivity, SettingsActivity...)**
  - Controlan la interacción del usuario y muestran los datos recibidos.
  - Se gestionan eventos como clics en marcadores, cambios de idioma y botones de navegación.
- **Layouts XML**
  - Definen la apariencia visual de cada pantalla.
  - Se adaptan a múltiples idiomas dinámicamente mediante strings.xml.

## 9.2 Capa de Lógica de Negocio

- **Controladores y lógica en Kotlin**
  - Gestionan el filtrado de lugares por categoría.
  - Cargan y muestran marcadores desde Firestore en tiempo real.
  - Controlan rutas con la API de Google Directions.
  - Gestionan el idioma seleccionado desde SharedPreferences con `applyAppLocale()`.

## 9.3 Capa de Datos

- **Firebase Firestore**
  - Contiene la colección MercaElx > Lugares > lugaresArray.
  - Cada lugar incluye nombre, descripción, precio, dirección, categoría, idioma y coordenadas.
  - La app accede a los datos usando la librería `FirebaseFirestore.getInstance()`.

- **Firestore Realtime Updates (FirestoreLocationProviderClient)**
  - Usa la localización en tiempo real para mostrar la posición actual del usuario y calcular distancias.

## 9.4 Servicios externos

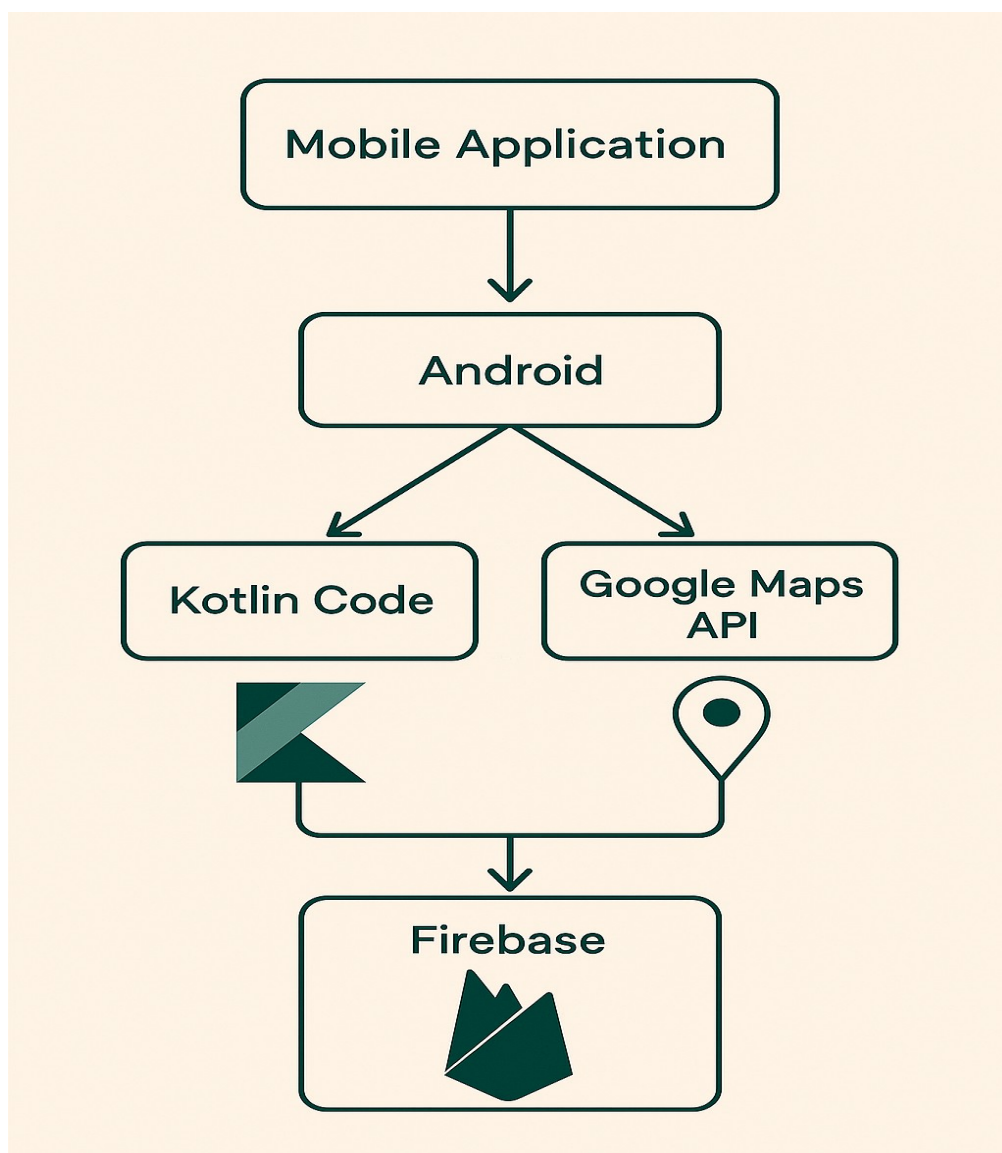
- **Google Maps SDK**
  - Muestra el mapa de Elche, marcadores personalizados y control de cámara.
- **Google Directions API**
  - Trazado de rutas entre la ubicación del usuario y el lugar seleccionado.
- **Firestore**
  - Herramienta usada durante el desarrollo para gestionar mejor la base de datos Firestore.

## 9.5 Gestión de idioma (Internacionalización)

- Uso de SharedPreferences para guardar el idioma seleccionado.
- Cambio de configuración regional con Configuration.setLocale(locale) y recarga del contexto.
- Soporte para 6 idiomas: español, inglés, alemán, francés, ruso y chino.
- Las descripciones, nombres, categorías y direcciones son multilingües y se cargan según el idioma actual.

## 9.6 Flujo general

1. Al abrir la app, se carga el idioma guardado.
2. Se solicita la ubicación y se inicializa el mapa.
3. Se consultan los lugares desde Firestore y se colocan los marcadores.
4. El usuario puede:
  - Ver su ubicación.
  - Filtrar por categorías.
  - Seleccionar un lugar para ver detalles.
  - Trazar rutas a cualquier lugar desde su posición.



# 10 Pruebas Realizadas

Durante el desarrollo de la aplicación MERCAELX, se llevaron a cabo numerosas pruebas tanto en el emulador de Android Studio como en dispositivos físicos para garantizar el correcto funcionamiento de la app, su rendimiento y su usabilidad.

## 10.1 Pruebas en el emulador de Android Studio

El emulador de Android Studio ha sido fundamental durante las primeras fases del desarrollo, permitiendo una validación rápida de los siguientes aspectos:

- **Carga de interfaz y navegación fluida:** Se comprobó que las actividades (MainActivity, SettingsActivity, LugarDetalleActivity) cargaban correctamente, y que los botones respondían como se esperaba.
- **Visualización del mapa de Google Maps:** Se verificó la correcta integración del SupportMapFragment, los botones de zoom, los marcadores y la funcionalidad de centrado en la ubicación simulada.
- **Visualización de datos desde Firebase:** Se probaron las consultas a Firestore usando datos reales simulados en el entorno de test.
- **Cambio de idioma:** Se testearon distintos valores de idioma (es, en, de, fr, ru, zh) configurando el idioma del emulador y mediante SharedPreferences, comprobando que los textos cambiaban correctamente.
- **Trazado de rutas:** Simulando ubicaciones, se trazaron rutas a distintos marcadores para confirmar que la API de Google Directions respondía correctamente y que los Polyline se dibujaban bien.

Se usaron dispositivos virtuales con diferentes resoluciones (HD, FullHD, tabletas) y versiones de Android (desde Android 8 a 13) para comprobar compatibilidad y comportamiento responsive.

## 10.2 Pruebas en dispositivos reales

Las pruebas en teléfonos reales se realizaron para garantizar que todos los servicios y sensores funcionaran correctamente fuera del entorno simulado:

### Localización

- Se comprobó que la aplicación solicitaba correctamente los permisos de localización y centraba el mapa en la ubicación real del usuario tras el arranque.
- Se validó la funcionalidad de mostrar los 5 lugares más cercanos con datos precisos de distancia calculada.

### **Acceso a internet y Firebase**

- Se probó el acceso a la base de datos remota de Firestore en diferentes redes (Wi-Fi, 4G) y se confirmó la carga correcta de los marcadores y su contenido.
- Se testearon escenarios con pérdida de conexión para comprobar el comportamiento de la app (mostrando Toast con mensajes de error amigables).

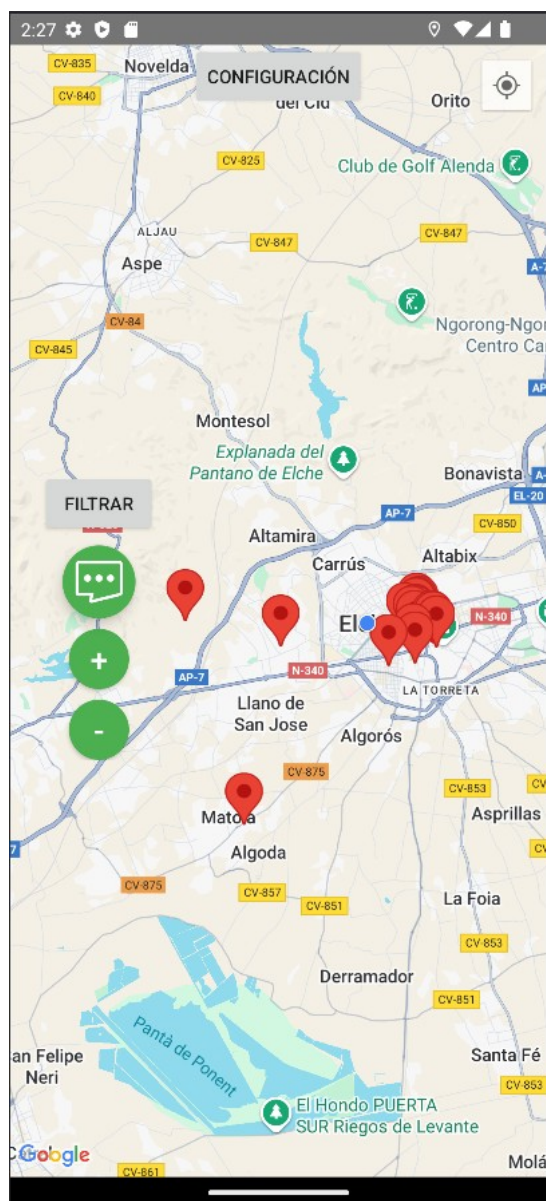
### **Cambio de idioma**

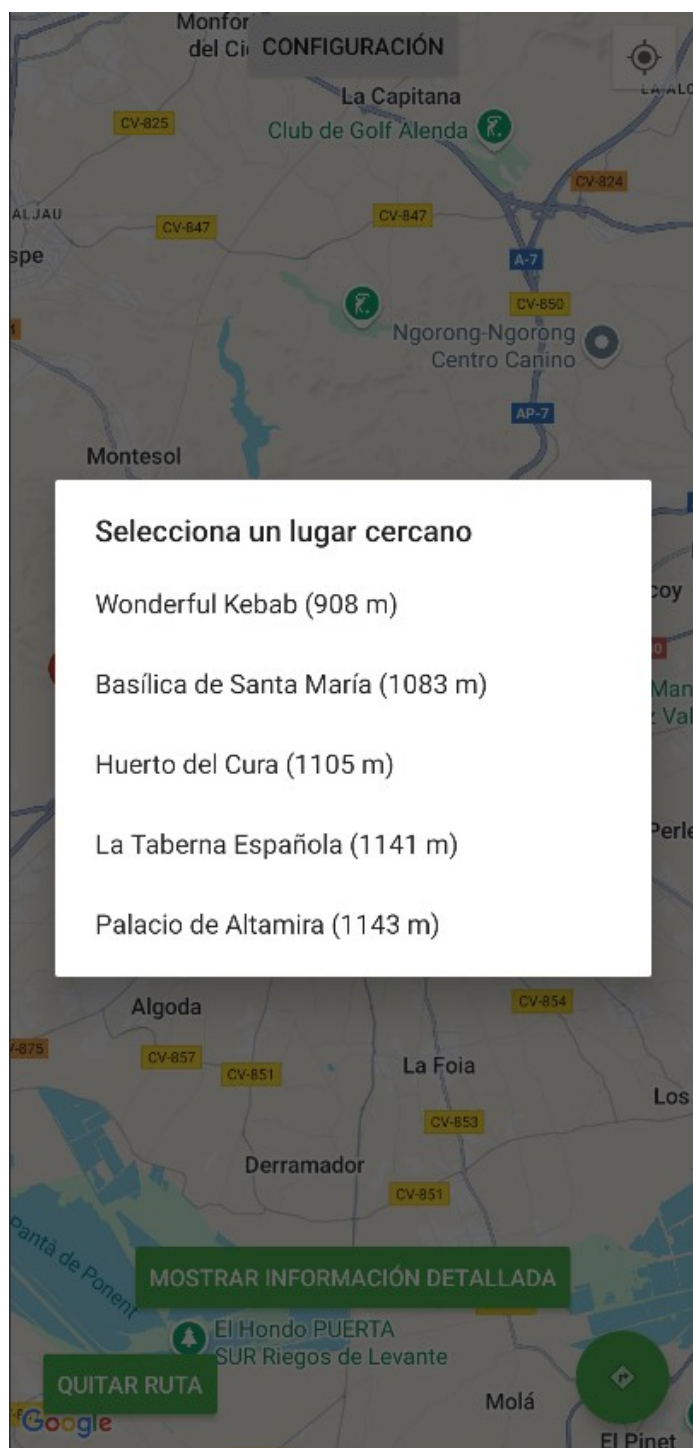
- Se probó el cambio de idioma desde SettingsActivity, confirmando que se guardaba la preferencia en SharedPreferences, que la aplicación se reiniciaba correctamente con el nuevo Locale, y que los datos multilingües desde Firebase se cargaban con la clave del idioma seleccionado.

### **Interacciones**

- Se testearon los botones de zoom, el botón de trazar ruta, el botón de filtrar por categorías y el botón de ver detalles. Todos funcionaron correctamente.
- En LugarDetalleActivity, se comprobó la correcta carga de las imágenes desde imageUrl, los textos localizados y la navegación hacia atrás.





# 11 Visuales












2:55











## Basílica de Santa María

Templo barroco donde se celebra el Misteri d'Elx.

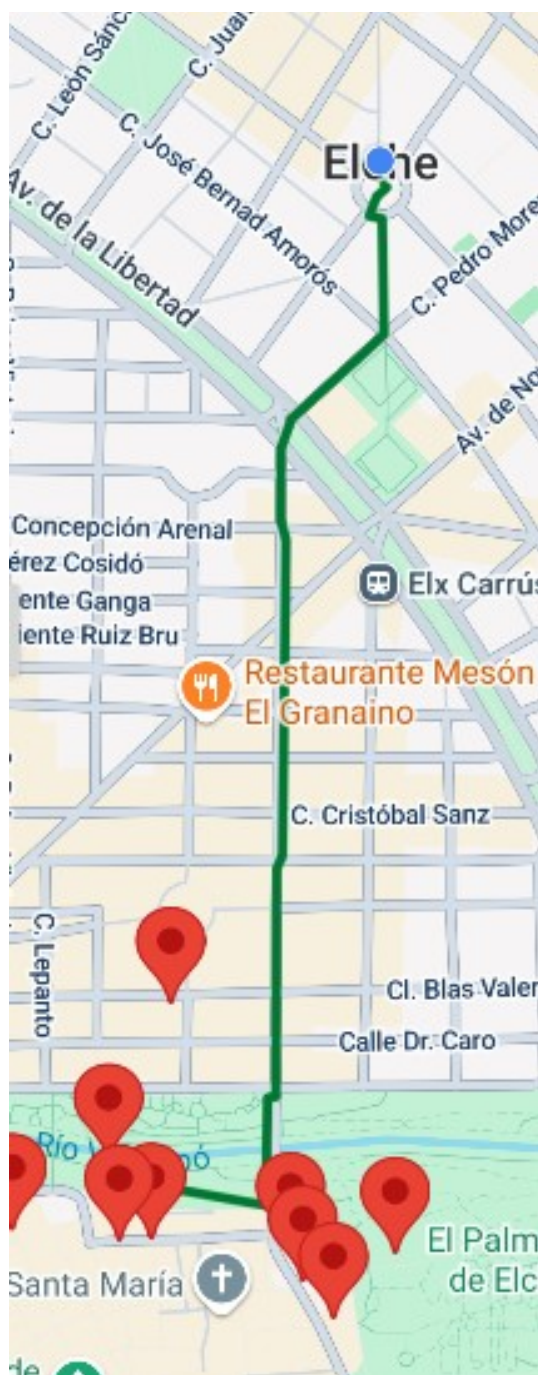
 Plaça de Santa Isabel

 09:00 - 13:00 / 17:00 - 20:00

 Precio no disponible

 Religioso

EXIT



## 12. Bibliografía

- Google. (n.d.). [Android Developers](#). Android developer official documentation.
- Google. (n.d.). Google Maps Platform Documentation.
- Firebase. (n.d.). Firebase Documentation. Google Firebase official guide.
- Android Developers. (n.d.). Material Design. Official Material Design Guidelines.
- Firefoo. (n.d.). [Firefoo – Firebase Admin Client](#). Firestore database editor and analytics tool.
- Kotlinlang.org. (n.d.). Kotlin Documentation. Official Kotlin Language Documentation.
- Android Open Source Project. (n.d.). [Localization in Android](#).
- Nielsen, J. (1994). Usability Engineering. Academic Press.
- ISO/IEC. (2011). ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE).
- European Union. (2018). General Data Protection Regulation (GDPR).
- Morin, C. (2020). Firebase Essentials for Android. Packt Publishing.
- Hussein, A. (2018). Mastering Firebase for Android Development. Packt Publishing.
- Udacity. (n.d.). Firebase in a Weekend.
- Budiu, R. & Nielsen, J. (2013). Mobile Usability. New Riders.
- Android Developers Blog. (n.d.). Google Maps and Location APIs Blog.
- Stack Overflow. (n.d.). [Community Solutions for Android & Firebase](#).
- Medium.com. (n.d.). [Firebase + Kotlin + Android articles](#).
- GitHub. (n.d.). [Open Source Android Projects](#).
- JetBrains. (n.d.). [Kotlin Plugins & Tools](#).
- Google Codelabs. (n.d.). Firebase and Maps Codelabs.
- Desarrolladores Android España. (n.d.). [Foro de desarrollo Android y Firebase](#).
- Real Academia Española. (n.d.). [Ortografía y gramática](#).