EvEMaps

Plataforma Integral de recargas en puntos de carga

Julio Schneider Estop

TFG de Informática y Comunicaciones

C.F.G.S.

Informática D.A.M

Resumen

EvEMaps es una plataforma desarrollada con Django 5.2, Django REST Framework y Android, cuyo objetivo es localizar y gestionar la recarga de vehículos eléctricos (VE).

Inicialmente, se ha diseñado como el germen de una aplicación para gestionar un sistema de reservas —de duración variable— en puntos de recarga de VE.

Problema → Ubicar y gestionar recarga VE es caótico (apps aisladas, info desactualizada).

Solución EvEMaps → Ecosistema Android + Backend Django que conecta en tiempo real puntos de carga, reservas y pagos.

Valor añadido → Todo el flujo “buscar → reservar → pagar → cargar” en una sola plataforma.

App en Android

Estructura del programa

```
EvEMaps/
├── app/                    ← módulo Android principal
│   ├── controller/         ← Controladores (lógica de negocio)
│   ├── interfaces/         ← Contratos para inyectar dependencias
│   ├── model/              ← Entidades y DTOs
│   ├── network/            ← Cliente HTTP (Ktor) y helpers de red
│   ├── repositories/       ← Abstracción de fuentes de datos
│   ├── view/               ← Pantallas Jetpack Compose + Activities
│   ├── ui/theme/           ← Colores, tipografías y estilos M3
│   └── util/               ← Utilidades (extensiones, managers…)
├── build.gradle.kts        ← Configuración raíz de Gradle
├── gradle/ & settings      ← Catálogo de versiones y rutas de módulos
└── README.md               ← Documentación técnica
```

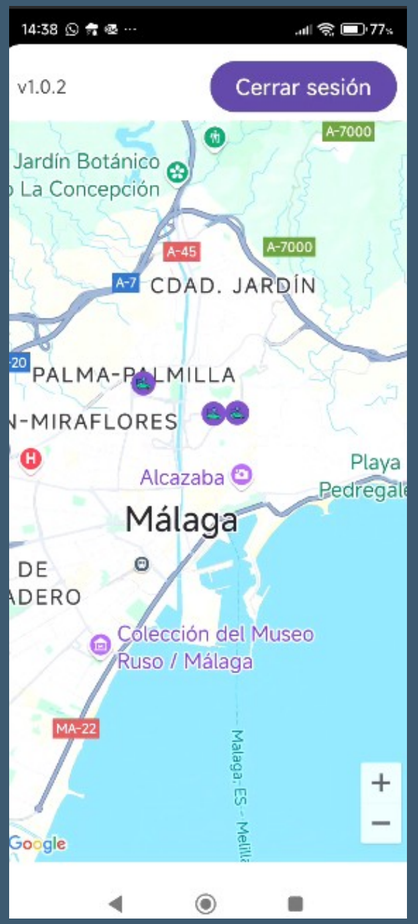
Qué hacen las aplicaciones de cada carpeta

Carpeta
controller/
interfaces/
model/

Pantallas de la aplicación

Login

Mapa

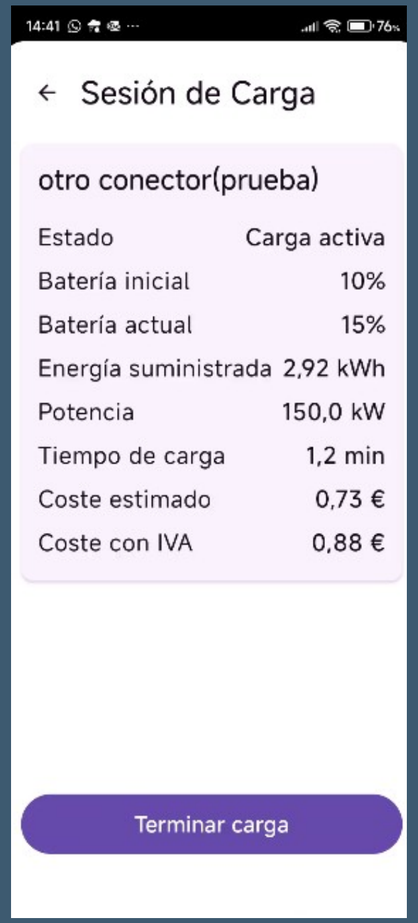


Reserva

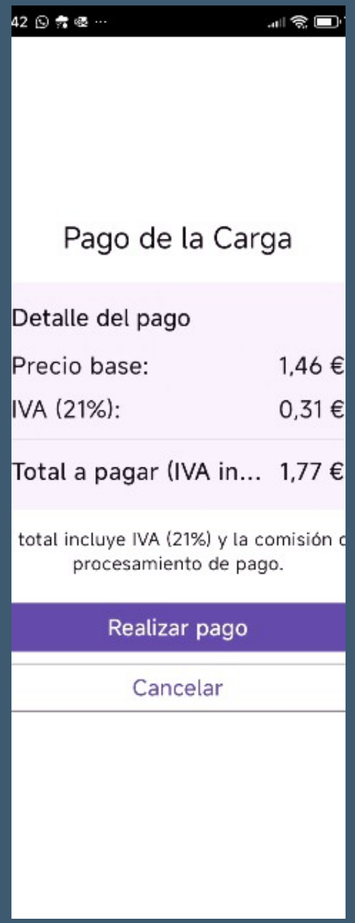
Pantallas de la aplicación

Informe

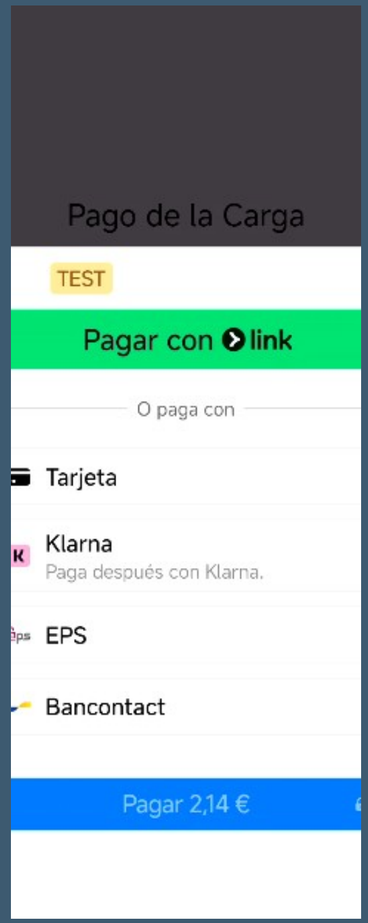
S. Carga



Pago



Card



Mejoras

- No uso Room, pero tengo previsto implementarlo (porque lo único que guardo localmente es el token JWT y las preferencias, que se guardan con SharedPreferences)

- No he usado DataStore porque guardo pocas cosas, el token y la session.

- No uso Retrofit porque he implementado Ktor que está 100% escrita en Kotlin y viene preparada para coroutines, trae ya de serie implementada la conversión a/desde JSON, además, es multiplataforma por si quisiera hacer una versión IOS con Ktolin Multiplatform.

- Esta versión es MVC simplificado porque en esta versión sólo quería probar el funcionamiento con la integración con Django y próximamente migraré a MVVM.

Aplicación EvEMaps - Django

Resumen de la aplicación Web

El backend de la aplicación lo he realizado con Django, debido a que estaba estudiando un curso de Django, además me pareció una apuesta interesante.

La aplicación la he alojado en Pythonanywhere, porque en 15 minutos tenía un servidor Django operativo con HTTPS sin tocar ni nginx ni certbot.

ÁRBOL DEL PROYECTO

Carpeta / Archivo	Función principal
electrolineras_project/	*Root* del proyecto Django
└─ manage.py	CLI de Django (migraciones, servidor, etc.)
└─ electrolineras_project/	Núcleo: settings.py, urls.py, WSGI/ASGI
└─ core/	Utilidades y componentes transversales
└─ electrolineras/	Modelos y API de puntos, reservas y sesiones
└─ payments/	Integración Stripe / facturación
└─ messenger/	Chat AJAX entre usuarios
└─ pages/	Mini-CMS: blog & páginas estáticas
└─ profiles/	Extiende el modelo User con datos extra

CARACTERÍSTICAS QUE CONVIENE REMARCAR

- Separación por responsabilidad.
- Cada carpeta es una app Django autónoma, así se puede escalar sin dificultad.
- Fácil mantenimiento.
- Actualizar Stripe o cambiar la lógica de reservas no toca perfiles ni chat.
- Preparada para multi-usuario
- El API usa JWT (JSON web token) que identifica de forma exacta al usuario en cada request, filtros por request.user que actúa como llave para filtrar datos y controlar permisos y transacciones; varias personas pueden reservar en paralelo sin colisiones.

Gracias por su atención

Para contactar, consulte mi portfolio

https://portfolio.juliomalaga.me

julio@juliomalaga.me