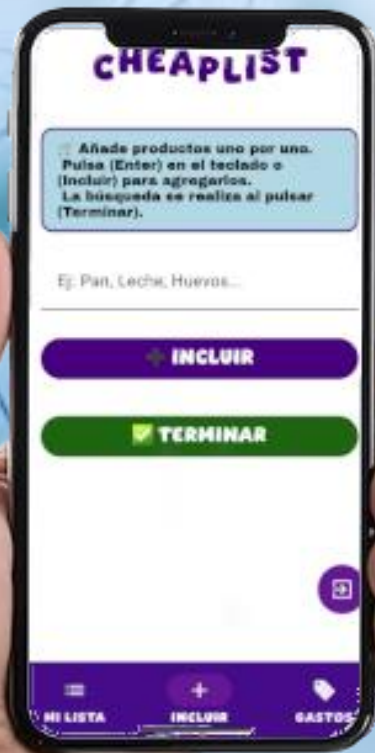


CHEAPLIST

AUTOR: Jose Manuel Moreno Valenzuela CICLO
FORMATIVO: (DAM)
CENTRO EDUCATIVO: Cesur
TUTOR DEL PROYECTO: Francisco Romero Guillén
CURSO: 2023/2025



¿Qué es CheapList y para qué sirve?



- CheapList es una aplicación móvil que permite comparar los precios de productos entre dos supermercados reales: **Mercadona** y **Dia**.
- La app está pensada para personas que quieren **ahorrar en su compra diaria** de forma rápida, sencilla y desde el móvil. El usuario solo tiene que escribir qué quiere comprar, y CheapList le dice **dónde está más barato**.
- Además, CheapList permite **añadir tickets de compra manualmente**, asignándoles un supermercado y un precio total, para llevar un **control mensual de gastos**. De esta forma, el usuario puede ver en qué supermercado gasta más, gracias a un gráfico visual muy claro.
- Funciona con una interfaz moderna e intuitiva, conectada a una base de datos actualizada con los precios reales de ambos supermercados.

¿Qué tecnologías he utilizado?




Para desarrollar CheapList he utilizado una combinación de tecnologías modernas tanto para la parte del usuario (frontend) como para el funcionamiento interno (backend):

◆ Frontend (App Android)

- **Android Studio** con **Jetpack Compose** para diseñar la interfaz.
- **Kotlin** como lenguaje de programación.
- **Room** para la base de datos local.
- **Firebase Auth** para el inicio de sesión (correo y Google).
- **Lottie** para animaciones modernas.

◆ Backend (Servidor y scraping)

- **Flask** (Python) para crear la API.
 - **PostgreSQL** como base de datos en el servidor.
 - **Selenium + Tor** para hacer scraping de productos en tiempo real.
 - **PgBouncer** y **Redis** para mejorar el rendimiento con muchos usuarios.
 - **Gunicorn** para servir la API y **Render/VPS** para alojarla.
- 

¿Cómo lo he hecho?



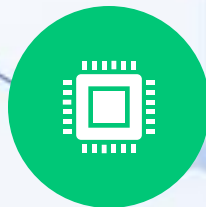
✿ 1. Diseño de la idea y objetivos

Pensé en un problema real: el precio de la compra diaria. Quería una app que ayudara a comparar precios de verdad, no con datos inventados.



💡 3. Creación del backend (API Flask)

Monté la API en Python con Flask para conectar la app a una base de datos real. Añadí registro, login, rutas protegidas con JWT y conexión a PostgreSQL.



💻 5. Implementación en servidor VPS

Instalé todo en un VPS real: PostgreSQL, Gunicorn, PgBouncer, Redis... Dejé el backend funcionando 24/7 con dominio propio: <https://api.cheaplistapp.com>.



📱 2. Desarrollo del frontend (App Android)

Empecé creando la interfaz con Jetpack Compose. Fui creando las pantallas de login, añadir producto, mi lista y tickets. Usé Room para guardar productos localmente al principio.



🕷️ 4. Scraping con Selenium + Tor

Desarrollé scripts para obtener precios reales desde Mercadona y Dia. Usé Tor para evitar bloqueos por IP. Los datos se actualizan en el servidor.



📊 6. Pruebas, correcciones y mejoras


Hice muchas pruebas con usuarios, mejoré el rendimiento, añadí animaciones y validaciones, y conecté todo para que funcione de forma fluida.

¿Qué problemas tuve y cómo los resolví?




Scraping con Mercadona

Mercadona bloqueaba el acceso por IP al hacer scraping.

 **Solución:** Usé Selenium con el navegador real y conecté todo por Tor Tor para que cambiara la IP en cada sesión.


Gestión de usuarios y seguridad

No quería que cualquiera accediera a la API ni a los datos de otros usuarios.

 **Solución:** Implementé autenticación con JWT y protegí todas las rutas sensibles del backend.


Rendimiento lento con muchos usuarios

Al hacer muchas peticiones, la app se ralentizaba.

 **Solución:** Instalé PgBouncer para optimizar las conexiones a PostgreSQL y Redis para cachear las búsquedas por 5 minutos.


Animaciones y diseño en la app

Al principio la interfaz era muy simple y poco visual.

 **Solución:** Añadí animaciones con Lottie, mejoré los colores, usé una fuente personalizada tipo graffiti y pulí el diseño.

Guardar tickets y generar gráficas

Era complicado organizar bien los tickets y generar estadísticas visuales.

 **Solución:** Creé una base de datos local para los tickets y usé un gráfico circular que muestra en qué supermercado se gasta más.

¿Qué he aprendido durante este proyecto?

A trabajar como en el mundo real

He aprendido a desarrollar un proyecto completo de principio a fin, igual que se hace en una empresa: idea, backend, frontend, base de datos, seguridad y despliegue.

Nuevas tecnologías

He trabajado con herramientas que no se enseñaron directamente en clase, como:

- Selenium + Tor para scraping real
- Redis y PgBouncer para alto rendimiento
- JWT para seguridad con tokens
- Deploy profesional en un VPS con dominio propio

🔍 A resolver problemas reales sin rendirme

Tuve que buscar soluciones por mi cuenta, investigar documentación en inglés, probar mil veces hasta que algo funcionó... y no rendirme cuando algo fallaba.

A cuidar la experiencia del usuario

No quería una app fea o confusa. Me esforcé en hacerla atractiva, animada y sencilla. Añadí detalles como el tick verde animado, el saludo personalizado y los gráficos de gasto mensual.

A confiar en mí mismo como programador

Lo que antes me parecía imposible, ahora lo puedo hacer yo solo. He ganado confianza y estoy listo para proyectos aún más grandes.



¿Qué resultados he conseguido y qué haré después?

✅ App 100% funcional y completa

- CheapList ya está terminada y funcionando correctamente.
- Compara precios en tiempo real entre **Mercadona** y **Dia**.
- Guarda tickets, muestra gastos por mes y responde rápido.

🌐 Backend profesional desplegado

- API Flask alojada en un **servidor VPS**.

- Accesible desde cualquier móvil en:

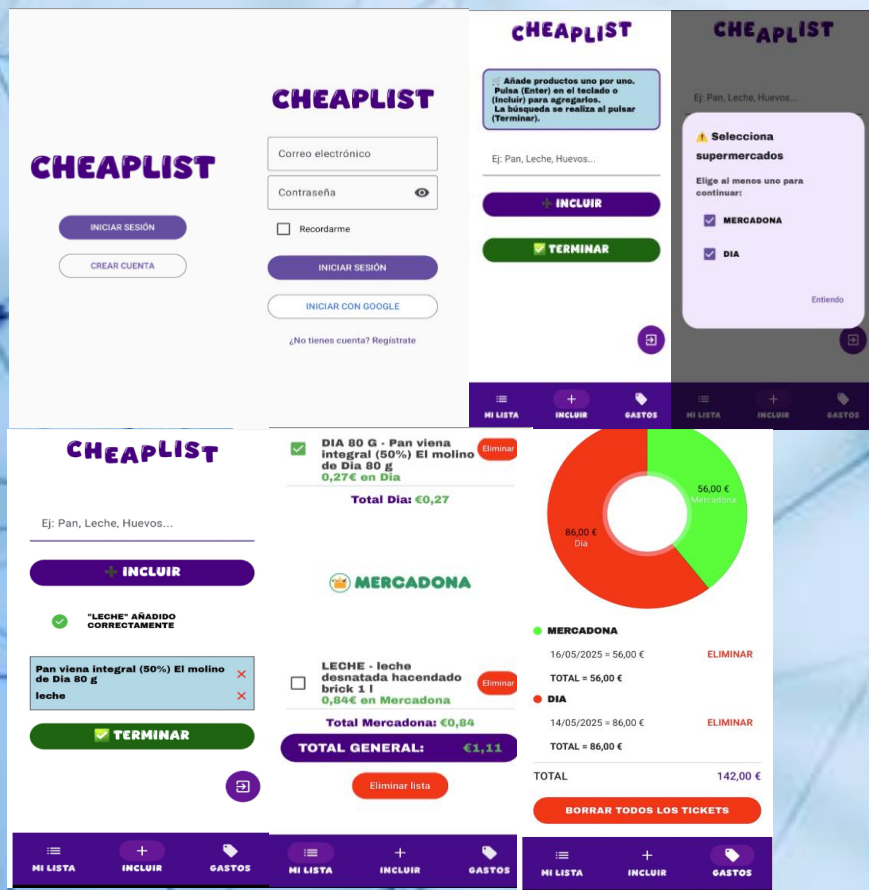
👉 <https://api.cheaplistapp.com>

🗃️ Base de datos real con productos y usuarios

- Los productos se almacenan por supermercado, con precio, fecha, y categoría.
- Cada usuario tiene su propia cuenta, tickets y productos.

🚀 Planes para el futuro

- Añadir más supermercados (Lidl, Aldi, Carrefour).
- Subir la app a **Google Play Store**.
- Crear una versión “Pro” con más estadísticas y funciones premium.
- Añadir escaneo de tickets con cámara para facilitar el control de gastos.



¡Gracias por vuestra atención!



Jose Manuel Moreno Valenzuela
Proyecto Integrado - DAM Cesur
2023/2025
