

Documentación asociada a la aplicación multiplataforma de visualización de restaurantes en la ciudad de Gijón

Contenido

1. Introducción	2
2. Detalle sobre el conjunto de datos	2
3. Detalles del desarrollo	2
4. Arquitectura de la aplicación	3
5. Mapa de navegación	4
6. Capturas de pantalla.....	5

Sergio Losada González
UO271744

Nota: el código fuente correspondiente a la aplicación desarrollada se puede encontrar en un repositorio público de GitHub, cuyo enlace figura a continuación:

<https://github.com/sergio-losada/flutter-restaurants-app>

1. Introducción

Para la entrega correspondiente al trabajo individual de la asignatura Informática Móvil, se presenta una aplicación destinada a la visualización de un listado de restaurantes en la ciudad de Gijón. La información de dichos establecimientos ha sido obtenido de un archivo de datos abiertos del propio Ayuntamiento de Gijón, y consumido mediante una petición GET, siguiendo el estándar REST y bajo el protocolo HTTP.

Es de mención que, durante el propio desarrollo de esta aplicación, y por causas ajenas a mí, el Ayuntamiento de Gijón sufrió un ciberataque, que dejó inutilizados varios de sus servicios. Entre los afectados, se encontraba el servicio de datos abiertos que estaba utilizando para suministrar datos a mi interfaz de usuario. Es por ello que, para no depender de esta institución, pero manteniendo la filosofía del trabajo de consumir una API REST, aprovechando que tenía el fichero JSON descargado decidí subirlo a un servicio de hosting, como es GitHub pages, y realizar la petición a dicha dirección.

Es por ello que, tal y como se indica en el código fuente de la aplicación, la URL utilizada por la petición GET <https://opendata.gijon.es/descargar.php?id=810&tipo=JSON>, hoy día todavía caída, se sustituyó por <https://sergio-losada.github.io/restaurants.json>.

2. Detalle sobre el conjunto de datos

En ese último enlace encontramos un array en formato JSON con 50 elementos. Cada uno de estos items corresponde a un restaurante de la ciudad, del cual se especifican datos de diferentes tipos, tales como el nombre, la dirección y el código postal, el teléfono y correo electrónico, las redes sociales o página web, las coordenadas (latitud, longitud), una imagen del establecimiento y otros datos, tales como las líneas de bus mediante las cuales se puede llegar o algunas etiquetas que identifican al lugar.

Fue precisamente esta gran variedad de información la que hizo que me decantase por este conjunto de datos, pues ofrecía gran versatilidad a la hora de mostrar la información, pudiendo esta ser presentada mediante botones, tablas, mapas y otros componentes.

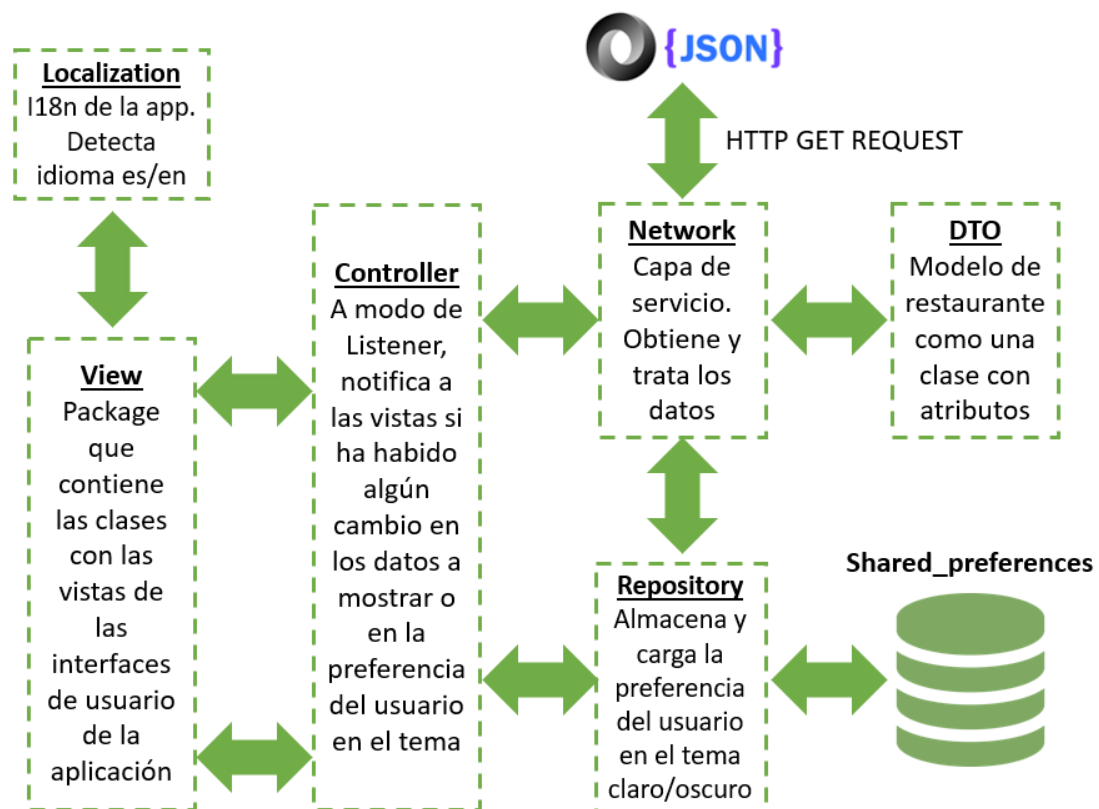
3. Detalles del desarrollo

Para el desarrollo de esta aplicación se ha optado por el empleo de la tecnología Flutter, un *framework* del lenguaje de programación Dart cuyo principal punto diferenciador frente a otros lenguajes de programación móvil es que gracias a su potente compilador es capaz de ejecutarse tanto en Android como en iOS, además de sobre un navegador web e incluso de forma nativa en los sistemas operativos Windows y Linux, aunque en esta ocasión solamente nos ocupa la plataforma móvil.

Se garantiza que el desarrollo de esta aplicación ha sido de forma personal y original, respetando la nomenclatura de clases y archivos de Flutter, comentando debidamente el código y habiendo probado su correcta funcionalidad mediante pruebas de interfaz de usuario, implementadas dentro del directorio /test. Además, se cuenta con internacionalización a 2 idiomas, inglés y español, que la aplicación detecta del idioma del dispositivo, y 2 temas, claro u oscuro, recordando la preferencia del usuario al cerrar.

4. Arquitectura de la aplicación

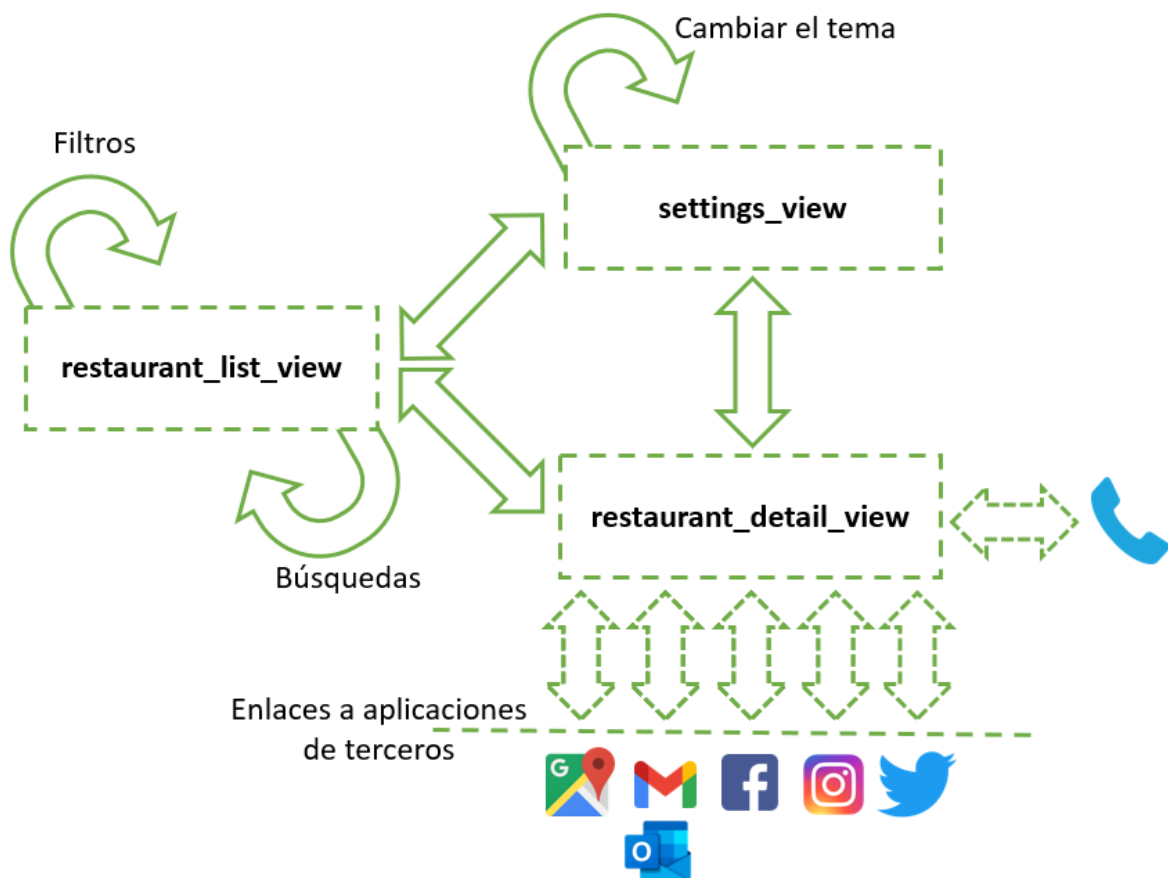
La aplicación se ha desarrollado siguiendo el modelo arquitectónico Model-View-ViewModel (MVVM), desacoplando la lógica y persistencia de la interfaz de usuario. A continuación, se presenta un diagrama a fin de explicar la interrelación de los diferentes packages que contienen las clases que conforman la arquitectura de la aplicación:



5. Mapa de navegación

En este penúltimo apartado, se incluye un diagrama que representa el mapa de navegación a través de las diferentes pantallas de la app, las cuales hacen referencia a las clases:

- `Restaurant_list_view`: página principal de la app, que contiene un listado con todos los restaurantes obtenidos vía REST a través de la petición HTTP GET. El JSON obtenido podrá ser filtrado y realizar búsquedas sobre él, a fin de actualizar el listado de restaurantes a razón de los criterios seleccionados.
- `Restaurant_detail_view`: página a la que se accede al pulsar sobre un elemento de la lista. Al iniciarse, cargará los datos del restaurante seleccionado y los mostrará de forma ordenada, además de contener enlaces tanto a la marcación del teléfono como a apps de terceros (aplicaciones de mapas, de correo electrónico o redes sociales).
- `Settings_view`: página accesible desde cualquiera de las dos vistas anteriores a través del icono de la rueda dentada de la esquina superior derecha de la barra de navegación. Desde ella, permite al usuario cambiar el tema (claro u oscuro) de la app, recordando su preferencia cuando la actividad vuelva a iniciarse.

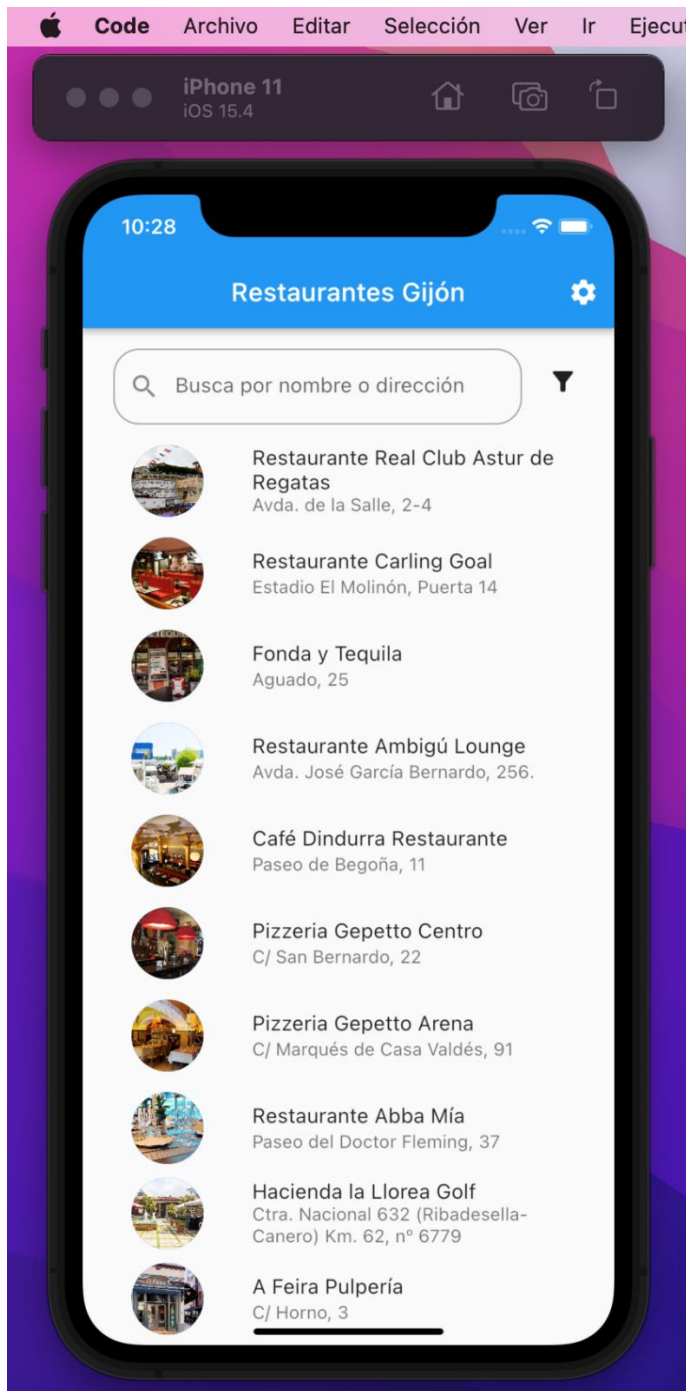


6. Capturas de pantalla

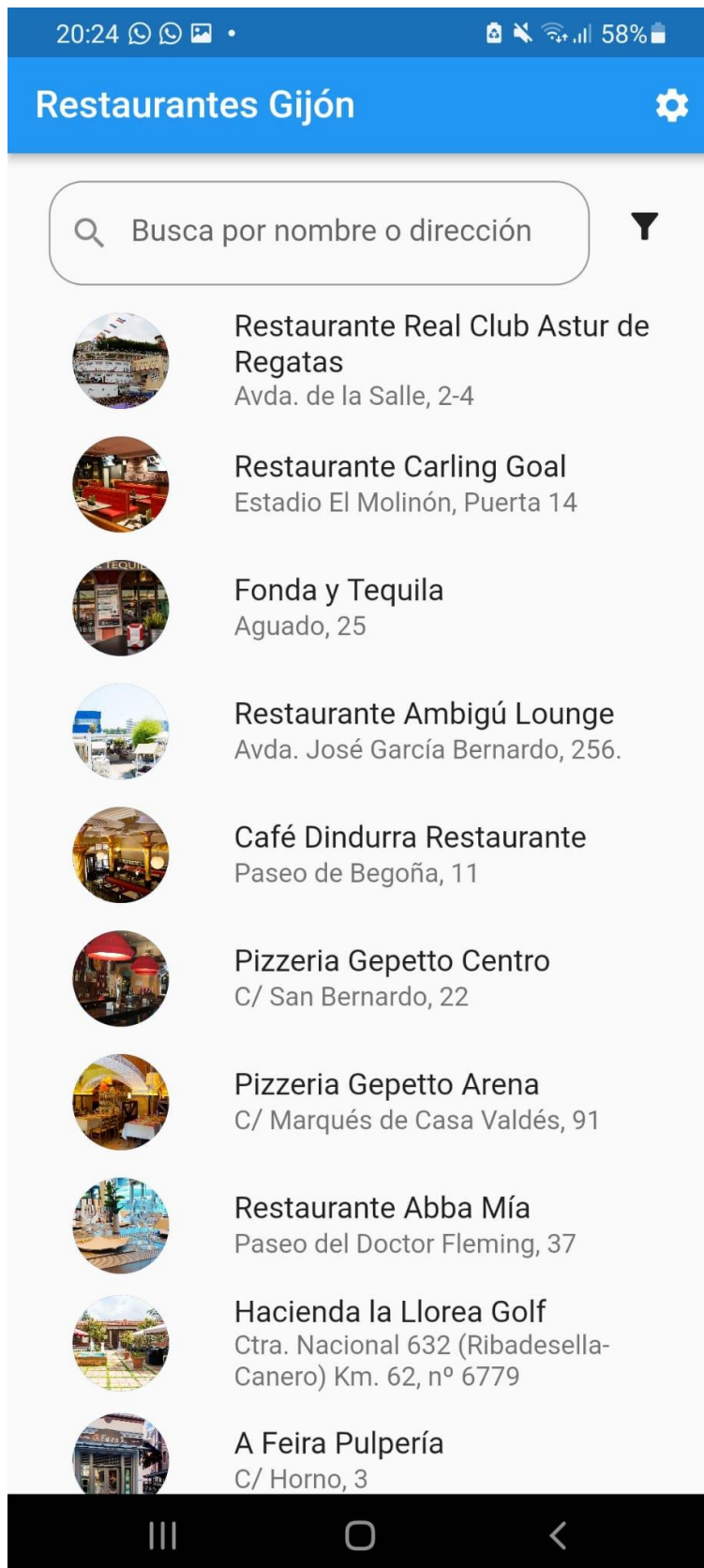
Por último, se adjuntan en este apartado algunas de las capturas de pantalla tomadas tanto en un dispositivo real Android, como desde el emulador de un iPhone 11 con sistema operativo iOS. La finalidad es demostrar que la aplicación es funcional sobre ambas plataformas, manteniendo un diseño uniforme.

Todas las capturas de pantalla tomadas, así como grabaciones de pantalla desde iOS, se encuentran en el repositorio del proyecto, en el directorio /documentation, junto con una copia de este documento en formato PDF.

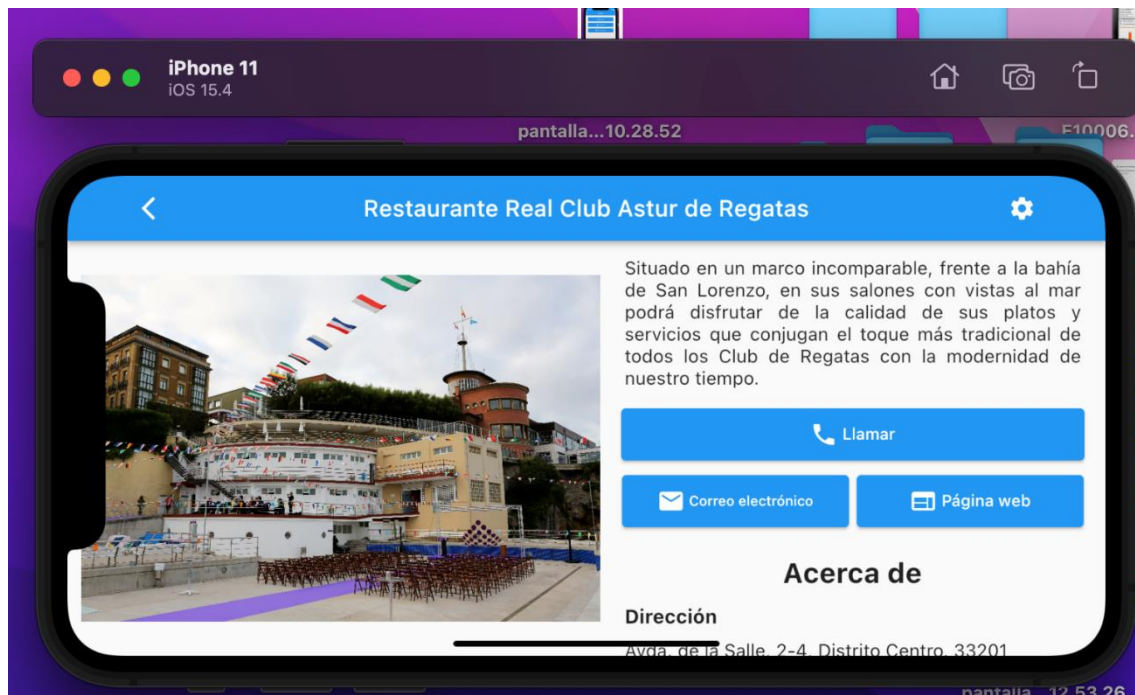
Captura de pantalla de restaurant_list_view en iOS:



Captura de pantalla de restaurant_list_view en Android:



Captura de pantalla de restaurant_detail_view en horizontal, en iOS:



Captura de pantalla de restaurant_detail_view en horizontal, en Android:

