

INFORME PRÁCTICA EXPLORATORIA - Uso de REST en aplicaciones Android

Carles Salvador Bertran, Rubén Sanz García

Aplicacions Distribuïdes
20/10/2019

Índice

1. Objetivos	p. 2
2. Tecnologías	p. 2
3. Desarrollo	p. 3

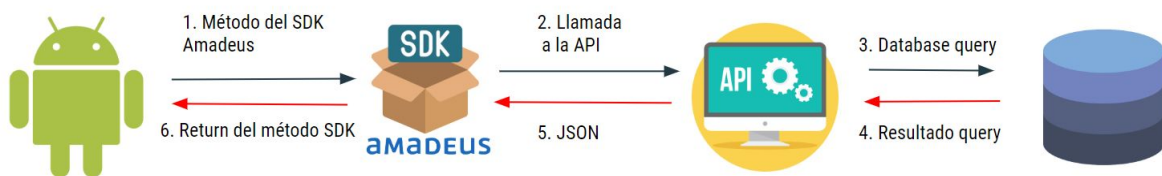
1. Objetivos

El objetivo para esta práctica era desarrollar una aplicación móvil que utilizase los servicios REST para conectarse con alguna API. Para este proyecto se ha desarrollado una aplicación cuya funcionalidad nos permite ver una lista de las mejores ofertas de vuelo entre dos localizaciones para una fecha determinada.

2. Tecnologías

Las tecnologías utilizadas para el desarrollo de esta práctica han sido las siguientes:

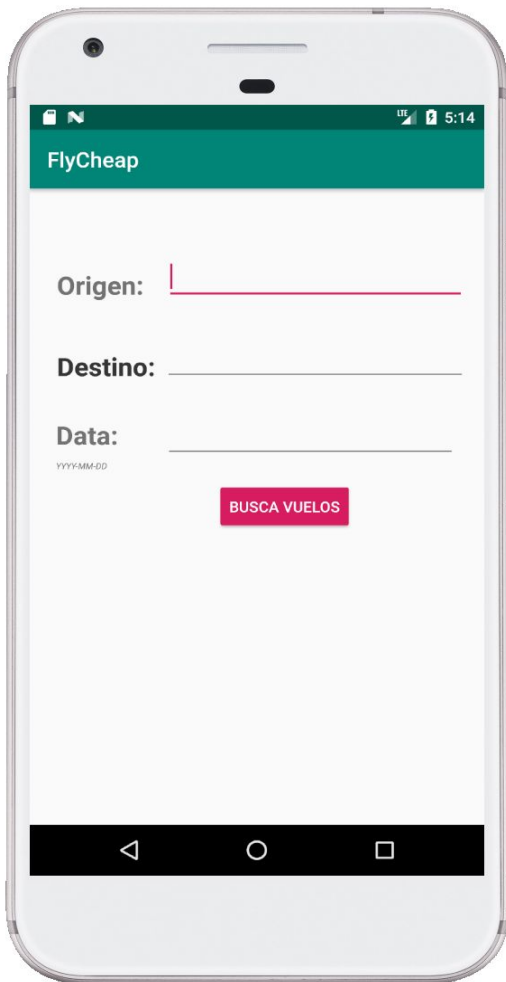
- **Android Studio:** es el entorno oficial de desarrollo de la plataforma Android. Ofrece muchas herramientas para desarrollar aplicaciones de forma nativa y existe un gran cantidad de documentación tanto desde la propia página oficial para desarrolladores como en diversas páginas de foros y consultas en Internet. Está basado en el software IntelliJ IDEA de JetBrains.
- **Amadeus API:** Para las conseguir los precios de los vuelos a distintos destinos usamos la API de Amadeus, más concretamente una nombrada “Flight low-fare search”. Esta es una API Rest que dados los parámetros de: origen, destino y fecha de vuelo, nos devuelve una lista con los precios más baratos para esos criterios de búsqueda. Una ventaja de esta API es que Amadeus nos proporciona un SDK para Java que contiene diferentes métodos para realizar de forma más sencilla las llamadas a la API y también tratar con más facilidad los resultados de las llamadas.



En la imagen anterior vemos la arquitectura básica de nuestra aplicación con las llamadas a la API de Amadeus.

3. Desarrollo

Creemos una aplicación vacía en la que tendríamos una única pantalla principal que nos serviría para realizar la función deseada. La apariencia de esta interfaz es la siguiente:



Consta de dos campos de texto para indicar el aeropuerto de origen y destino del vuelo junto con un campo para indicar la fecha de este en el formato indicado. El botón que indica “busca vuelos” realizará la llamada a la función que conectará con el servicio de Amadeus el cual nos devolverá una estructura de datos la cual se procesará para mostrársela al usuario en el espacio libre que podemos ver debajo del botón.

Para la llamada a la API cogemos los códigos IATA de los aeropuertos introducidos en los dos primeros campos. Para poder mostrar los diferentes aeropuertos al usuario hemos usado un fichero JSON como recurso de la aplicación dónde están listados todos los aeropuertos del mundo con su código IATA pertinente.

Cómo hemos explicado anteriormente, las llamadas a la API se realizan con los métodos proporcionados por el SDK de Amadeus:

```
FlightOffer[] flightOffers = amadeus.shopping.flightOffers.get(Params.with("origin", or).and("destination", des).and("departureDate", date));
```

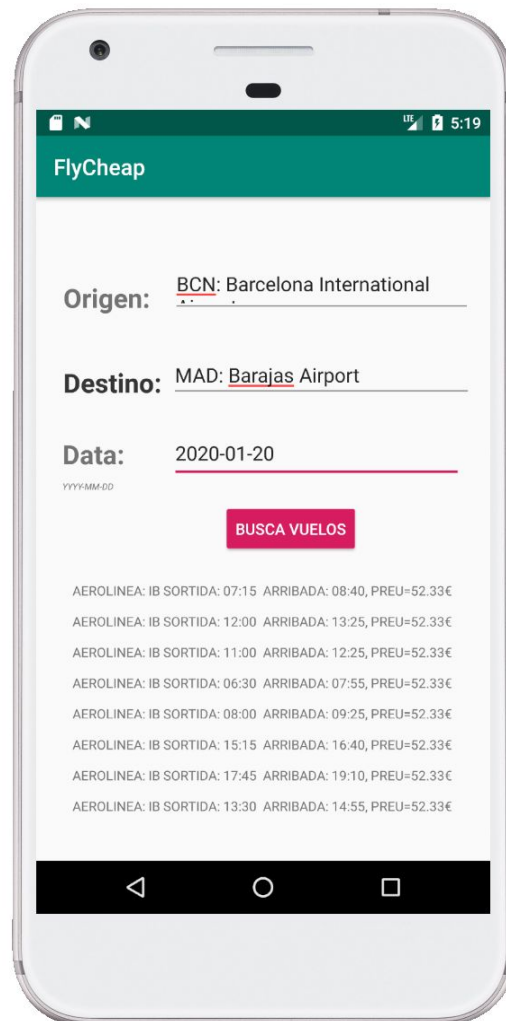
Esta sería una llamada equivalente a un

```
GET /v1/shopping/flight-offers?origin=or&destination=des&departureDate=date
```

Siendo “or”, “des” y “date” variables con origen, destino y fecha.

Es en *flightOffers* dónde se devuelve todas las ofertas de vuelo y hemos de procesar esta respuesta (con los métodos proporcionados por el SDK) para poder mostrarla como resultado de búsqueda con los campos que nos interesan, que en este caso son: aerolínea, hora de salida, hora de llegada y precio.

El resultado final después de realizar la petición es el siguiente:



Así la funcionalidad principal a la que estaba destinada está completa.