

Desafío - Weather Latam Consumo de API REST (I)

En este desafío validaremos nuestros conocimientos de Consumo de API, Retrofit y StateFlow. Para lograrlo, necesitarás aplicar durante el curso, especialmente lo relacionado a consumo de API, Retrofit, StateFlow y MVVM, utilizando de apoyo el archivo:

Apoyo Desafío evaluado - Consumo de API REST (I)

Lee todo el documento antes de comenzar el desarrollo **individual**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Este Desafío es la continuación del desafío de la Unidad anterior, y será requerido para trabajar con el siguiente desafío de la próxima unidad, por lo que se recomienda terminarlo, ya que será ocupado más adelante.

Nuestro cliente, Weather Latam, nos ha pedido desarrollar una funcionalidad que permita consumir una API en Internet para así poder tener las últimas actualizaciones online.

Antes de empezar, deberás completar los siguientes pasos:

- Para obtener esta información usaremos el servicio de <https://openweathermap.org/api>
- Puedes seleccionar una de las siguientes key:
 - fcd51b9342252e2bb7daa90b7f20c2e7
 - 000477936bfcd7a4b6c887a3a149a0a5
 - 5938e2f76c5c22ef4b2f9471841e7f0c.
- El endpoint que usaremos es el siguiente: <https://openweathermap.org/current>
- Una vez selecciones tu Key, reemplaza los valores en el endpoint `https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}`,
- Los valores a ser reemplazados dentro los "{ }", son los siguientes:
 - Para lat y lon, utiliza -33.43107, -70.64666 respectivamente
 - Para API Key, reemplaza con tu developer key

- Ejemplo:
<https://api.openweathermap.org/data/2.5/weather?lat=-33.43107&lon=-70.64666&appid=02735b3ab7109f96e982d9a42ae2a22e>

Una vez realizados estos pasos, ¡Empecemos!

Usando el archivo de ayuda:

- Crea una interfaz que contenga el servicio.
- En la clase Repository transforma los datos recibidos desde la API y conviértelos en una clase DTO usando un mapper.
- Recuerda siempre usar MVVM, ya que posteriormente lo fusionamos con Room.
- Usando StateFlow, muestra los datos en pantalla:
 - Si el usuario cambia de Metrics a Imperial o viceversa, se debe mostrar el cambio en los valores en pantalla, sin necesidad de refrescar.
 - Recuerda manejar los posibles errores.

Requerimientos

1. Crear una interfaz que contenga el endpoint de la API.
(2 Puntos)
2. Transformar los datos recibidos desde la API a una clase DTO.
(2 Puntos)
3. Ocupar correctamente la clase RepositoryImp y ViewModel.
(2 Puntos)
4. Usar la clase DTO para mostrar la información al usuario.
(2 Puntos)
5. Usar StateFlow para mostrar la información al usuario de forma reactiva.
(2 Puntos)



¡Mucho éxito!

Consideraciones y recomendaciones

- Para transformar los datos desde la API a una clase DTO, usa un mapper.
- Usar correctamente RepositoryImp y ViewMode significa, por ejemplo, no hacer cálculos o transformaciones de datos en los fragments.
- Apoyate en las guías de ejercicios.