

PETAGRAM

Introducción

Este documento presenta el desarrollo de la actividad solicitada en el curso “Desarrollo de aplicaciones avanzadas en Android”, cuyo objetivo es integrar la librería Retrofit para consumir una API simulada y mostrar un timeline tipo Instagram en una aplicación móvil Android.

Las funcionalidades implementadas incluyen:

- Integración de Retrofit como cliente HTTP.
- Creación de modelos y deserializadores con GSON.
- Simulación del timeline de un usuario.
- Configuración del usuario principal desde un menú dedicado.
- Consumo y visualización de datos en un RecyclerView.
- Estructuración adecuada del proyecto en Android Studio.

Integración de Retrofit

Retrofit se integró para:

- Hacer peticiones HTTP a endpoints simulados.
- Convertir respuestas JSON en objetos Java/Kotlin con GSON.
- Facilitar la comunicación entre la app y el servidor.

La configuración básica incluye:

- RetrofitClient
- ApiService
- Uso de GsonConverterFactory

Endpoints Utilizados

A continuación, se documentan los endpoints implementados en la aplicación para simular la funcionalidad de Instagram.

Endpoint – Obtener publicaciones recientes del usuario

Método: GET

URL:

<https://api.instagram.com/v1/users/{user-id}/media/recent/>

Descripción:

Devuelve la lista de publicaciones más recientes del usuario configurado como principal.

Parámetros:

- user-id (obligatorio)
- access_token (Sandbox)

Ejemplo de respuesta JSON:

```
{  
  "data": [  
    {  
      "id": "125487",  
      "images": {  
        "standard_resolution": {  
          "url": "https://imagen-ejemplo.jpg"  
        }  
      },  
      "caption": {  
        "text": "Esta es una fotografía de prueba"  
      }  
    }  
  ]  
}
```

Endpoint – Obtener información del usuario**Método:** GET**URL:**

<https://api.instagram.com/v1/users/{user-id}>

Descripción:

Obtiene información sobre el usuario principal:

- nombre

- foto de perfil
- ID
- nombre completo

Ejemplo de respuesta JSON:

```
{
  "data": {
    "id": "12345",
    "username": "usuario_prueba",
    "profile_picture": "https://foto-perfil.jpg",
    "full_name": "Usuario de Prueba"
  }
}
```

Implementación en Código

Se muestran las clases más relevantes para la integración:

RetrofitClient

```
public class RetrofitClient {

  private static final String BASE_URL = "https://api.instagram.com/v1/";
  private static Retrofit retrofit = null;

  public static Retrofit getClient() {
    if (retrofit == null) {
      retrofit = new Retrofit.Builder()
        .baseUrl(BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build();
    }
    return retrofit;
  }
}
```

```

        }

    }

ApiService

public interface ApiService {

    @GET("users/{user-id}/media/recent/")
    Call<PostResponse> getRecentMedia(
        @Path("user-id") String userId,
        @Query("access_token") String token
    );

    @GET("users/{user-id}")
    Call<InstagramResponse> getUserInfo(
        @Path("user-id") String userId,
        @Query("access_token") String token
    );
}

```

Deserialización con GSON

Ejemplo de modelo recibido desde el endpoint:

```

public class Foto {

    private String id;
    private Imagenes images;
    private Caption caption;

    public Imagenes getImages() { return images; }
    public Caption getCaption() { return caption; }
}

```

Los modelos permiten que el Adapter reciba objetos listos para mostrar.

Presentación de datos en la Aplicación

Los datos obtenidos desde la API se muestran usando:

- Un RecyclerView
- Un Adapter (TimelineAdapter)
- Un item personalizado (item_post.xml)
- La pantalla principal que consume Retrofit (TimelineActivity)

Flujo general:

1. El usuario abre el Timeline.
2. Se consultan las publicaciones mediante Retrofit.
3. Se recibe un objeto PostResponse.
4. El Adapter procesa cada publicación.
5. Las imágenes y texto se muestran en la UI.

Configurar Usuario Principal

La aplicación incluye un ítem de menú que lleva a la pantalla:

ConfigurarCuentaActivity

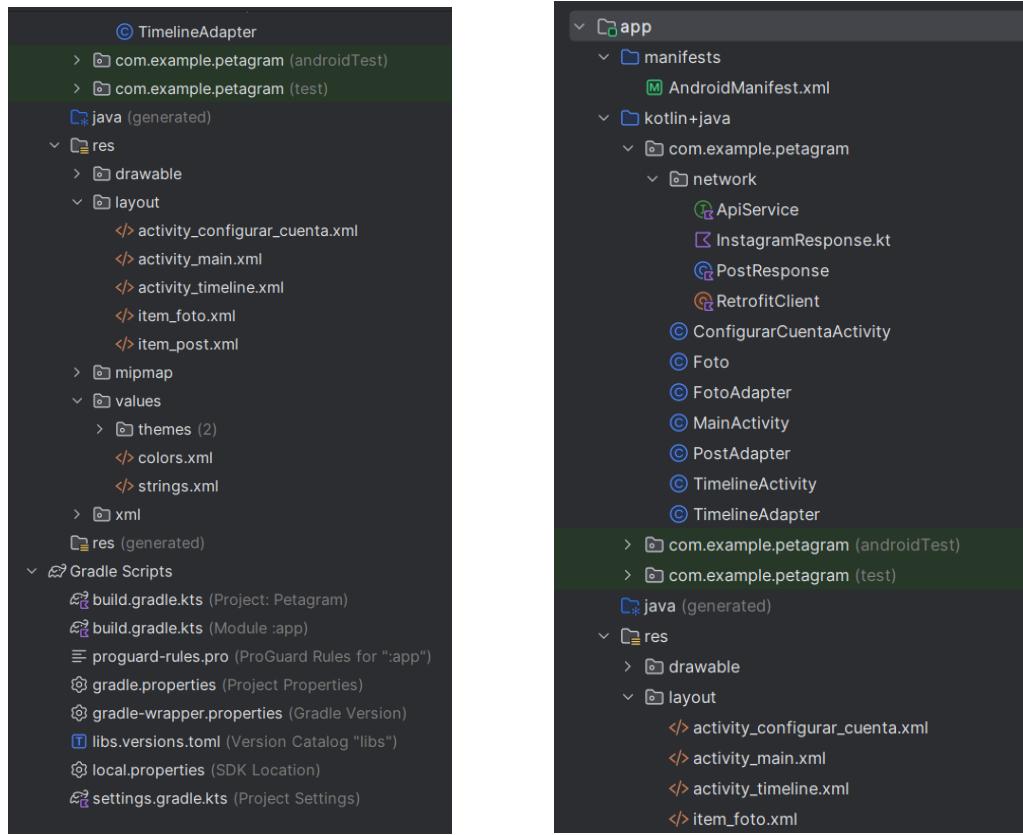
Aquí el usuario selecciona su cuenta principal.

La app carga el timeline correspondiente a dicho usuario.

Se puede guardar en:

- SharedPreferences (opcional)

Estructura del Proyecto



Descripción de la estructura:

com.example.petagram - Código fuente

Incluye:

- Activities (pantallas)
- Adapters
- Modelos
- Carpeta network con Retrofit

res - Recursos

Contiene:

- layouts (pantallas e items)
- mipmap (íconos)
- values (strings, colores, temas)
- drawable (gráficos)

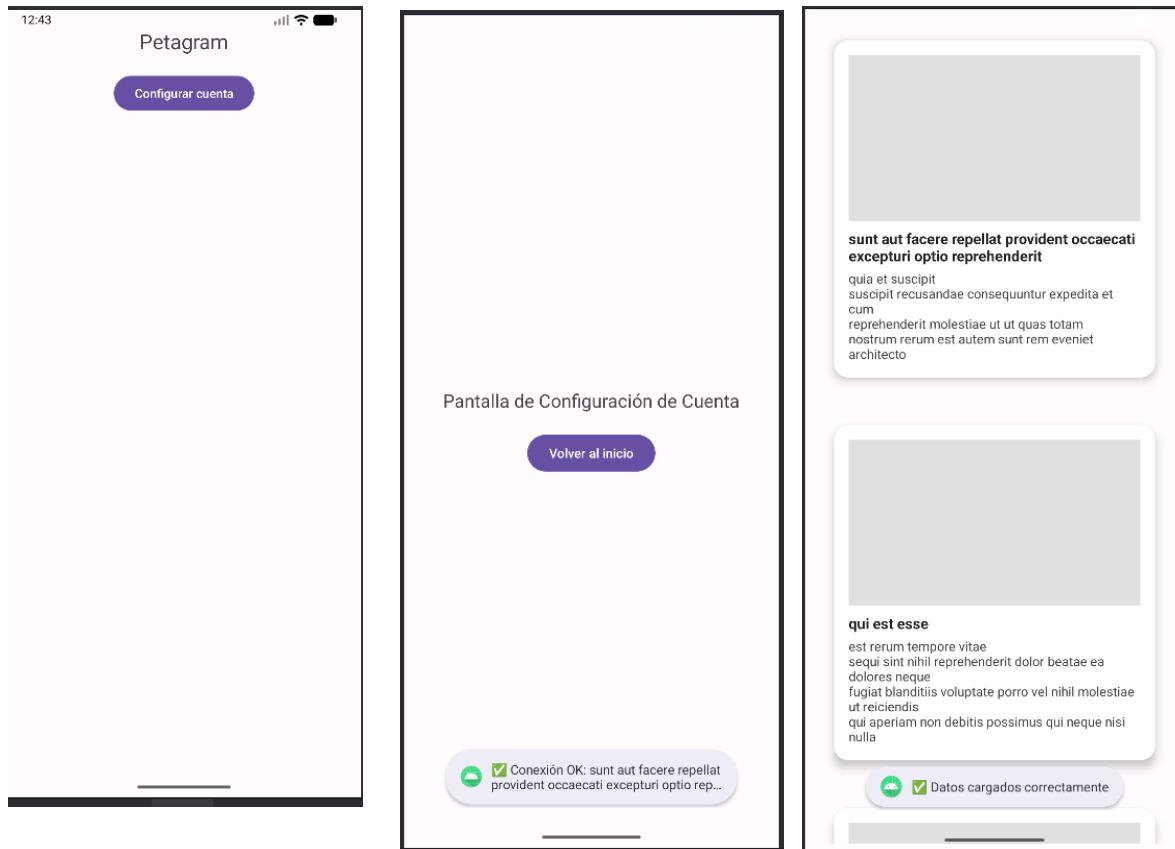
Gradle Scripts

Usado para:

- Añadir dependencias
- Configurar plugins
- Definir el SDK y versiones del proyecto

Esta estructura garantiza modularidad, orden y facilidad de mantenimiento.

Capturas de la Aplicación



Conclusiones

La integración de Retrofit permitió:

- Realizar peticiones HTTP eficientes.
- Usar GSON para convertir JSON en objetos manipulables.
- Visualizar datos dinámicos en un RecyclerView.
- Simular el comportamiento de Instagram en modo Sandbox.
- Organizar el proyecto de forma clara y escalable.