Investigación Aplicada 2

INTEGRANTES:

César Alejandro Avendaño Guevara	AG230680)
Andrea Paola Calles Arias	CA222101
Lucía Milena Hernández Bonilla	HB221258
Daniel Ernesto Alvarado Roque	AR220441

DOCENTE:

Ing. Juan Carlos Menjivar Ramírez

MATERIA:

DESARROLLO DE SOFTWARE PARA MÓVILES DSM441 G05L

¿Qué es una API REST y cómo funciona?

Una API REST es una interfaz de programación de aplicaciones (API) que se ajusta a los principios de diseño del estilo arquitectónico de transferencia de estado representacional (REST), un estilo utilizado para conectar sistemas de hipermedia distribuidos. Las API REST a veces se denominan API RESTful o API web RESTful.

Las API REST se comunican a través de solicitudes HTTP para realizar funciones de base de datos estándar como crear, leer, actualizar y eliminar registros (también conocido como CRUD) dentro de un recurso.

Por ejemplo, una API REST usaría una solicitud GET para recuperar un registro. Una solicitud POST crea un nuevo registro. Una solicitud PUT actualiza un registro y una solicitud DELETE elimina uno. Todos los métodos HTTP se pueden utilizar en las llamadas a la API. Una API REST bien diseñada es similar a un sitio web que se ejecuta en un navegador web con funcionalidad HTTP integrada.

¿Qué son JSON y endpoints?

JSON (JavaScript Object Notation) es un formato de texto estándar para almacenar y transferir datos estructurados. Se basa en la sintaxis de objetos en JavaScript, pero no está ligado a él.

Un endpoint de API es una ubicación digital donde una interfaz de programación de aplicaciones (API) recibe llamadas de API, también conocidas como solicitudes de API, para recursos en su servidor. Los endpoints de API son componentes de las API y suelen tener la forma de URL o localizadores uniformes de recursos.

Librerías comunes para consumo de APIs en Android

Retrofit: Cliente HTTP seguro para Android y Java, que simplifica la comunicación con APIs REST mediante anotaciones.

Dagger 2: Inyección de dependencias que permite declarar dependencias en tiempo de compilación, facilitando el mantenimiento del código.

OkHttp: Cliente HTTP que permite realizar solicitudes HTTP de manera eficiente y sencilla.

Glide: Librería para manejo de imágenes que permite cargar y cachar imágenes, optimizando el rendimiento de la aplicación.

Firebase: Plataforma integral que proporciona servicios de autenticación, bases de datos en tiempo real y almacenamiento en la nube.

API Utilizada

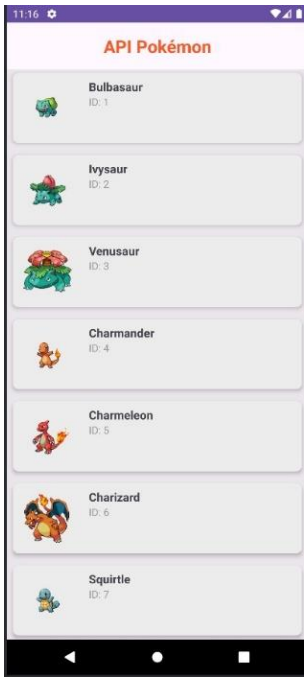
La API utilizada fue la de Pokémon RESTful

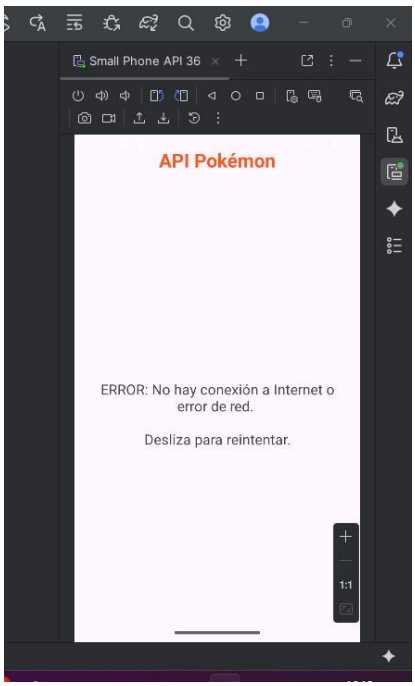
Como se puede observar se hace una petición GET con la que se muestra la lista de Pokémon

Para cargar los datos se configura el RecyclerView

```kotlin
package com.example.consumodeapi
import ...


class MainActivity : AppCompatActivity() {  2 Usages

    private lateinit var recyclerView: RecyclerView  6 Usages
    private lateinit var progressBar: ProgressBar  4 Usages
    private lateinit var textViewStatus: TextView  6 Usages
    private lateinit var swipeRefreshLayout: SwipeRefreshLayout  4 Usages
    // El adaptador debe ser de tipo PokemonAdapter
    private lateinit var pokemonAdapter: PokemonAdapter  3 Usages

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Inicialización de vistas
        recyclerView = findViewById( id = R.id.recyclerView)
        progressBar = findViewById( id = R.id.progressBar)
        textViewStatus = findViewById( id = R.id.textViewStatus)
        swipeRefreshLayout = findViewById( id = R.id.swipeRefreshLayout)

        // Configuración del RecyclerView
        // Cambiado el nombre de la variable de 'photoAdapter' a 'pokemonAdapter' para claridad
        pokemonAdapter = PokemonAdapter( pokemonList = emptyList())
        recyclerView.layoutManager = LinearLayoutManager( context = this)
        recyclerView.adapter = pokemonAdapter

        // Configuración del deslizar para refrescar
        swipeRefreshLayout.setOnRefreshListener {
            fetchPokemon() // Llamar a la función de carga de datos
        }

        // Carga inicial
        fetchPokemon()
    }

    // Función principal para la carga de datos
    private fun fetchPokemon() {  2 Usages
        // Mostrar indicadores de carga
        if (!swipeRefreshLayout.isRefreshing) {
```

Resultado reflejado en la interfaz, además de mostrar un mensaje si no hay internet o si la API no responde





Conclusiones

Una API REST facilita la creación de servicios web como: base de datos, información y aplicaciones accesibles desde diversos dispositivos y mejora las características óptimas de un sitio web o este caso de una aplicación móvil.

Una API REST funciona de forma similar a un sitio web. Un cliente (por ejemplo, un navegador web) envía una petición a un servidor (por ejemplo, el servidor web), y el servidor responde con una información y datos (por ejemplo, el sitio web).

Las APIs REST son fáciles de usar porque utilizan estructuras de URL simples y métodos estándar (por ejemplo, GET, POST, PUT, PATCH, DELETE).