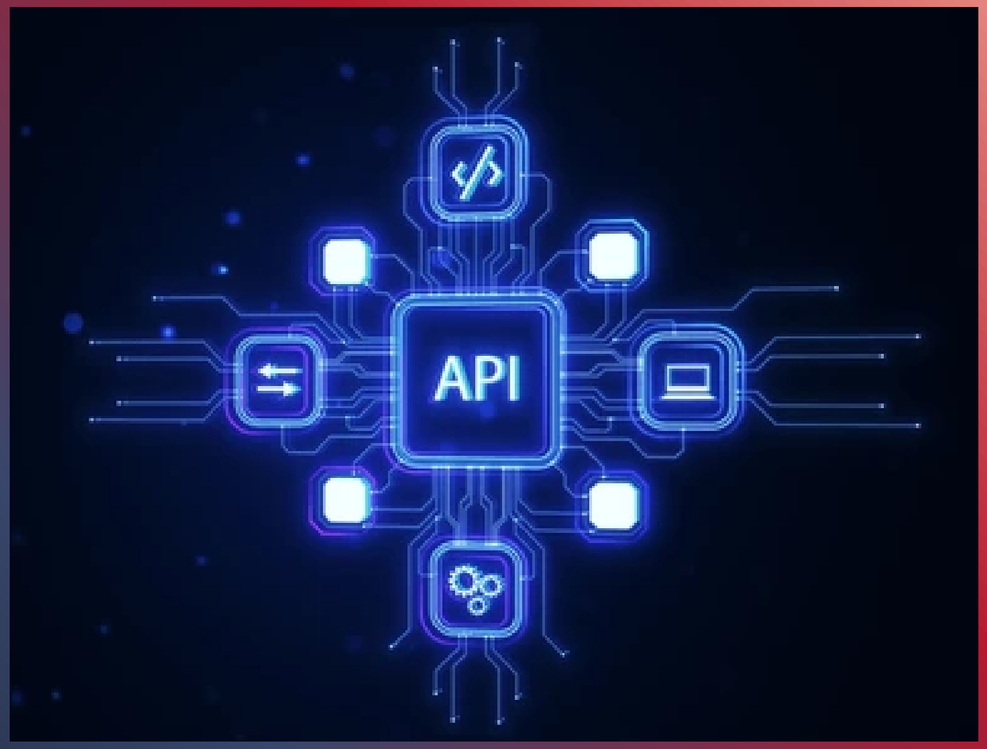
APIs remotas (REST/GraphQL)



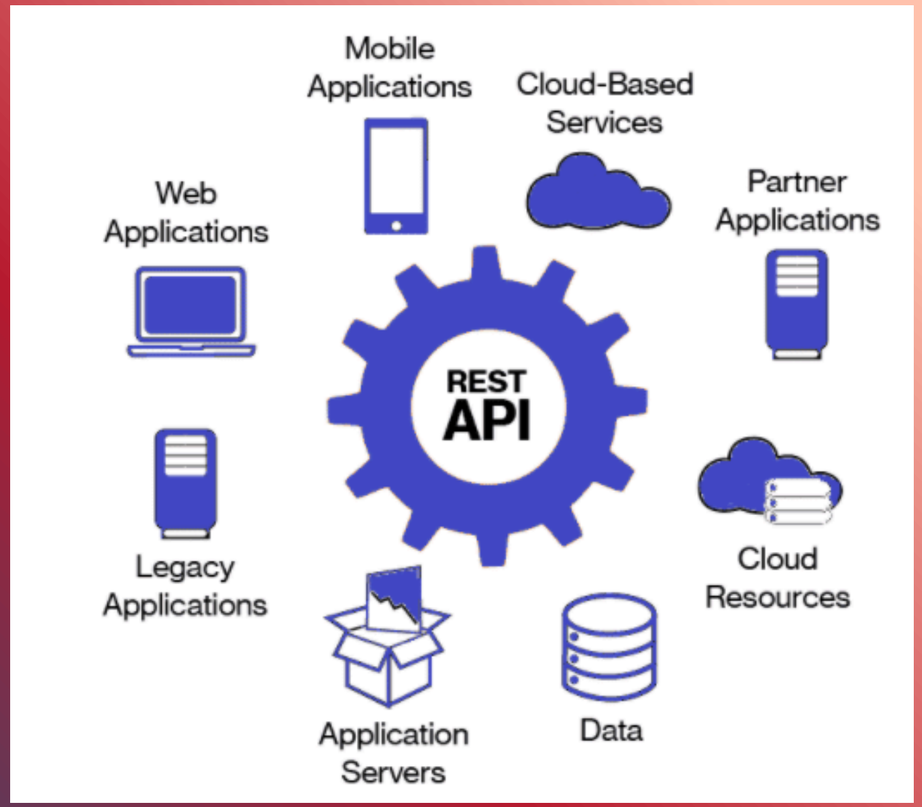
Elementos:

- Álvaro Frias - 49400;
- Eduardo Mesquita - 49507;
- Ari Jesus - 49903;
- Tiago Valério - 52334.

Apresentação Coletiva - Programação de Dispositivos Móveis

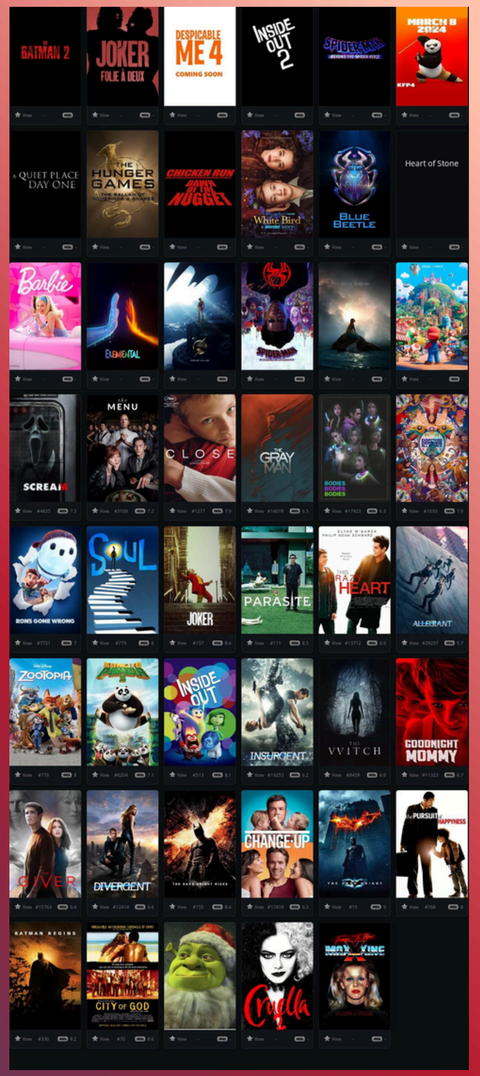
Introdução:

API REST é um conjunto de regras que permite a comunicação entre sistemas via HTTP, sem manter estado entre solicitações.



Objetivo do Trabalho:

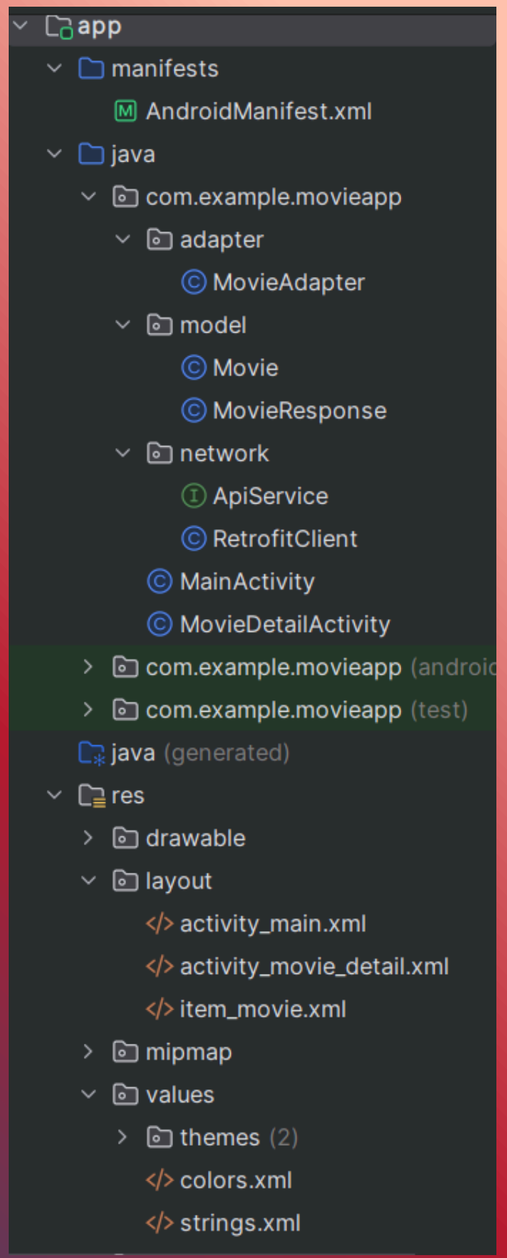
- Criar uma aplicação Android em Java que apresente uma lista de filmes;
- Consumir dados de uma API remota (TMDB);
- Apresentar os dados em tempo real;
- Demonstrar integração RESTful.



Apresentação Coletiva - Programação de Dispositivos Móveis

Estrutura da Aplicação

- model/ → Estrutura dos dados (JSON → objetos Java)
- network/ → Comunicação REST (Retrofit + API Key)
- adapter/ → Ligação entre dados e interface
- ui/ e res/layout/ → Interface visual e interação



Código da Integração REST (Imagens de exemplo)

```java
package com.example.movieapp.network;

import com.example.movieapp.model.MovieResponse;
import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Query;

3 usages
public interface ApiService {
    1 usage
    @GET("movie/popular")
    Call<MovieResponse> getPopularMovies(@Query("api_key") String apiKey, @Query("page") int page);
}
```

Interface APIService define os endpoints

```java
package com.example.movieapp.network;

import okhttp3.HttpUrl;
import okhttp3.Interceptor;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

import java.io.IOException;

2 usages
public class RetrofitClient {
    1 usage
    private static final String BASE_URL = "https://api.themoviedb.org/3/";
    1 usage
    private static final String API_KEY = "938f7d1bef2541960231fabd155a12ee";
    3 usages
    private static Retrofit retrofit;

    public static Retrofit getClient() {
        if (retrofit == null) {
            OkHttpClient client = new OkHttpClient.Builder()
                    .addInterceptor(new Interceptor() {
                        @Override
                        public Response intercept(Chain chain) throws IOException {
                            Request original = chain.request();
                            HttpUrl originalHttpUrl = original.url();

                            HttpUrl url = originalHttpUrl.newBuilder()
                                    .addQueryParameter( name: "api_key", API_KEY)
                                    .build();

                            Request.Builder requestBuilder = original.newBuilder().url(url);
                            Request request = requestBuilder.build();
                            return chain.proceed(request);
                        }
                    })
                    .build();

            retrofit = new Retrofit.Builder()
                    .baseUrl(BASE_URL)
                    .addConverterFactory(GsonConverterFactory.create())
                    .client(client)
                    .build();
        }
        return retrofit;
    }
}
```

Retrofit configura base URL e API key

Apresentação Coletiva - Programação de Dispositivos Móveis

Código da Interface (Imagens de Exemplo)

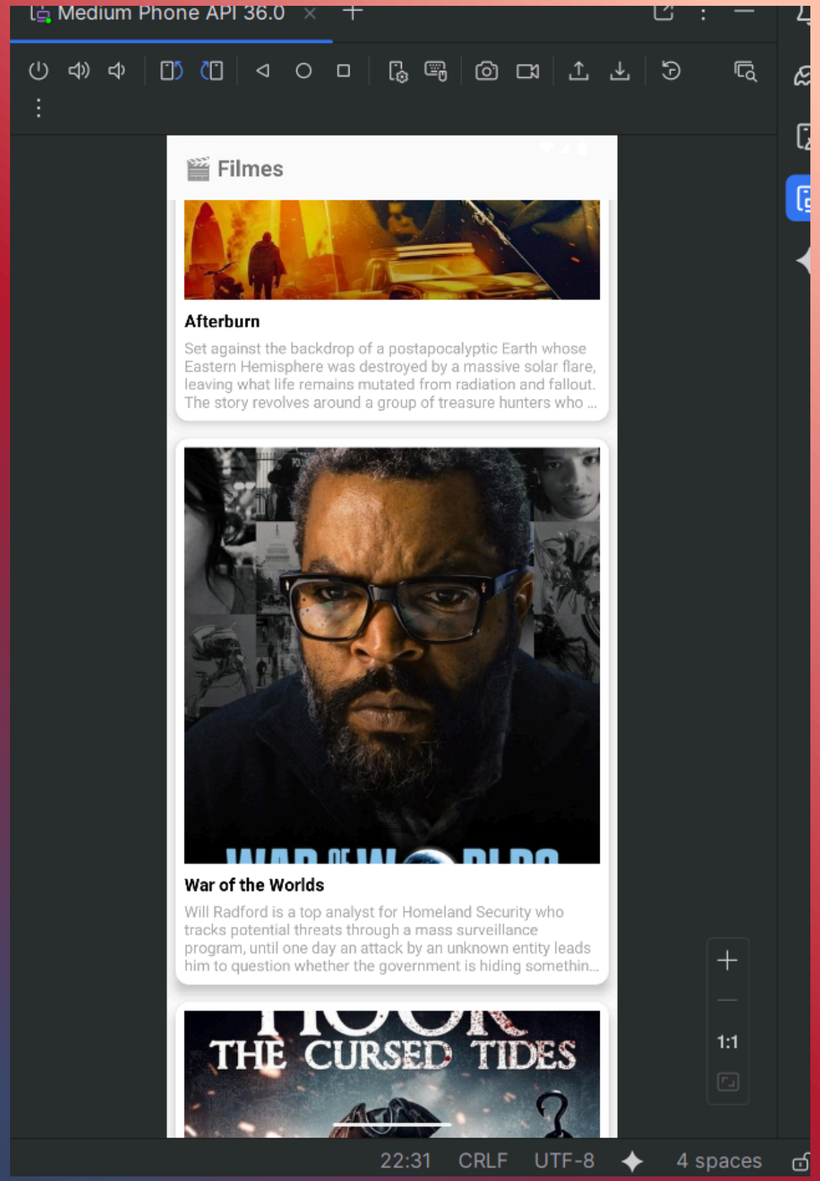
MainActivity recebe os dados da API

MovieAdapter mostra cada filme no RecyclerView

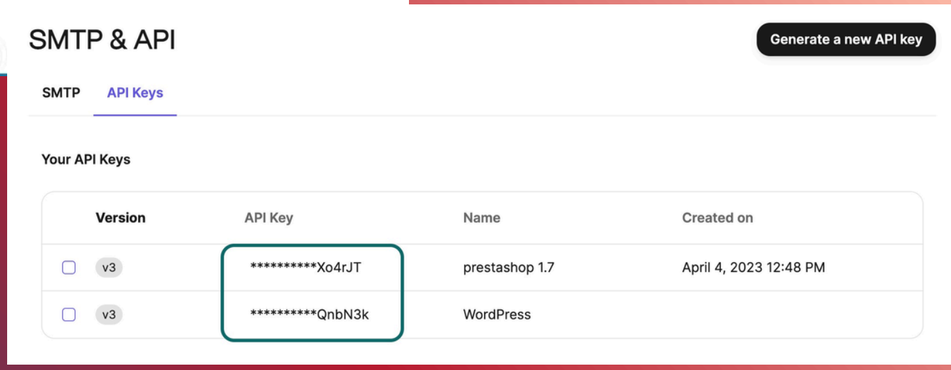
Interface da Aplicação (tipos de dados exemplo)

Resultados Obtidos:

- Lista dinâmica de filmes;
- Carregamento de imagens e descrições;
- Detalhes de cada filme ao clicar.



Conclusão



Conclusões e Aprendizagens:

- Integração com API REST implementada com sucesso;
- Comunicação com servidor TMDB via Retrofit;
- Manipulação de JSON e exibição dinâmica;
- Aprendizagem: arquitetura em camadas, consumo de dados online;
- Possível melhoria: sistema de favoritos ou pesquisa.

Objetivos:

Objetivos e Implementações Futuras:

- API do Mapbox para representação dos locais
- MediaWiki API para recolha de informações para o uso no nosso trabalho coletivo
- Entre outros...