

Instituto Tecnológico Superior De Lerdo



Programación móvil

Practica: 4.1

Profesor: Jesús Salas Marín

Alumno: Luis Andres Rodriguez Campos

Carrera: Ing. Informática

Sección: A

Grado: 8

Numero de control: 17231573

Primero modificaremos el objeto creado con anterioridad denominado “Constants”

```
package com.hackaprende.earthquakemonitor

object Constants {
    const val BASE_URL = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/"
    const val GET_LAST_HOUR_URL = "summary/all_hour.geojson"
}
```

Ahora modificaremos la conexión a la base de datos para utilizar sus valores

```
@Parcelize
@Entity(tableName = "eq_table")
data class Earthquake(@PrimaryKey val id: String,
    val place: String,
    val magnitude: Double,
    val time: Long,
    val latitude: Double,
    val longitude: Double) : Parcelable
```

También el archivo de la base de datos lo modificaremos de la siguiente manera

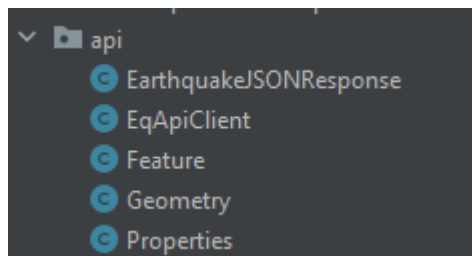
```
@Database(entities = {Earthquake.class}, version = 1)
public abstract class EqDatabase extends RoomDatabase {

    public abstract EqDao eqDao();

    private static final int NUMBER_OF_THREADS = 4;
    public static final ExecutorService databaseWriteExecutor =
        Executors.newFixedThreadPool(NUMBER_OF_THREADS);

    private static volatile EqDatabase INSTANCE;
    public static EqDatabase getDatabase(final Context context) {
        if (INSTANCE == null) {
            synchronized (EqDatabase.class) {
                if (INSTANCE == null) {
                    INSTANCE = Room.databaseBuilder(context.getApplicationContext(),
                        EqDatabase.class, name: "earthquakes_db")
                        .build();
                }
            }
        }
        return INSTANCE;
    }
}
```

Modificaremos todos estos archivos con los siguientes codigos



```
package com.hackaprende.earthquakemonitor.api;

import java.util.List;

public class EarthquakeJSONResponse {
    private List<Feature> features;

    public List<Feature> getFeatures() { return features; }
}
```

```
package com.hackaprende.earthquakemonitor.api;

import ...

public class EqApiClient {
    public interface EqService {
        @GET("all_hour.geojson")
        Call<EarthquakeJSONResponse> getEarthquakes();
    }

    private final Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://earthquake.usgs.gov/earthquakes/feed/v1.0/su
        .addConverterFactory(MoshiConverterFactory.create())
        .build();

    private EqService service;

    private static final EqApiClient ourInstance = new EqApiClient();

    public static EqApiClient getInstance() { return ourInstance; }

    private EqApiClient() {
    }

    public EqService getService() {
        if (service == null) {
            service = retrofit.create(EqService.class);
        }
        return service;
    }
}
```

```
package com.hackaprende.earthquakemonitor.api;

public class Feature {
    private String id;
    private Properties properties;
    private Geometry geometry;

    public String getId() { return id; }

    public Properties getProperties() { return properties; }

    public Geometry getGeometry() { return geometry; }
}
```

```
package com.hackaprende.earthquakemonitor.api;

public class Geometry {
    private double[] coordinates;

    public double getLongitude() { return coordinates[0]; }

    public double getLatitude() { return coordinates[1]; }
}
```

```
Feature.java x eq_list_item.xml x activity_main.xml x Properties.java x Geometry.java x v
package com.hackaprende.earthquakemonitor.api;

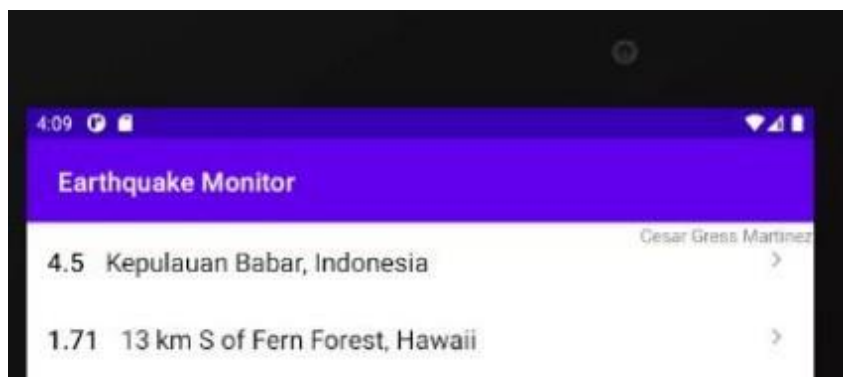
public class Properties {
    private double mag;
    private String place;
    private long time;

    public double getMag() { return mag; }

    public String getPlace() { return place; }

    public long getTime() { return time; }
}
```

Así quedaría nuestra app en la versión visual



```
<TextView
    android:id="@+id/eq_list_item_magnitude"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/text_black"
    android:textSize="18sp"
    android:gravity="center"
    android:textStyle="bold"
    tools:text="4.6" />

<ImageView
    android:id="@+id/eq_list_item_arrow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:src="@drawable/ic_chevron_right_gray" />

<TextView
    android:id="@+id/eq_list_item_title"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:textColor="@color/text_black"
    android:textSize="18sp"
    android:layout_toEndOf="@id/eq_list_item_magnitude"
    android:layout_toStartOf="@id/eq_list_item_arrow"
```

```

<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <data>

    </data>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="16dp">

        <TextView
            android:id="@+id/magnitude_text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:fontFamily="sans-serif-medium"
            android:textColor="@color/black"
            tools:text="5.0" />

        <ImageView
            android:id="@+id/arrow_image"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/black"
            android:layout_alignParentEnd="true"
            android:src="@drawable/ic_chevron_gray" />

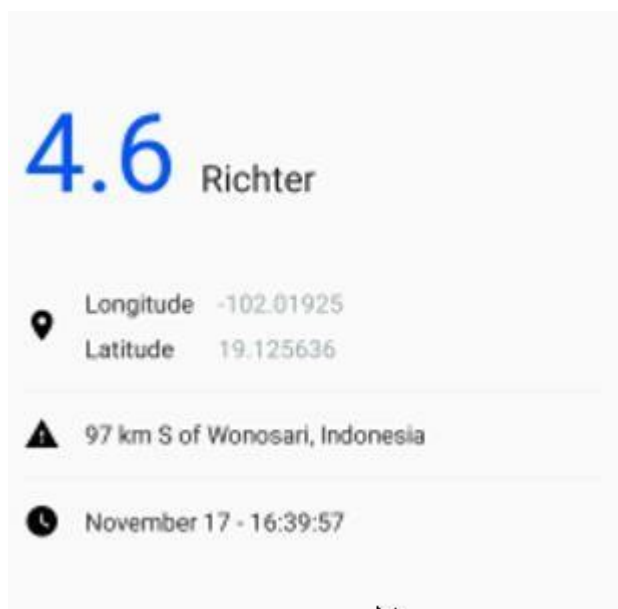
        <TextView
            android:id="@+id/place_text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toEndOf="@+id/magnitude_text"
            android:layout_toStartOf="@+id/arrow_image"
            android:paddingStart="16dp"
            android:paddingEnd="16dp"
            android:textSize="20sp"
            android:textColor="@color/black"
            tools:text="Honolulu Hawaii" />

    </RelativeLayout>
</layout>

```

I

Si damos click se nos abrirá el terremoto



Conclusiones

Esta practica presento más problemas ya que al usar bases de datos y apis, se dificulta un poco más en investigar como se usa cada una de estas cosas, pero al saber dominar estas tecnologías es mucho más fácil el manejo de datos y el uso de estos, para mostrarlos y de más.