Multimedia y bases de datos

Programación Multimedia y Dispositivos Móviles

2º Técnico en Desarrollo de Aplicaciones Multiplataforma

¿Qué vamos a ver?

1

Reproducción de sonido

2

Bases de datos

3

Insertar

4

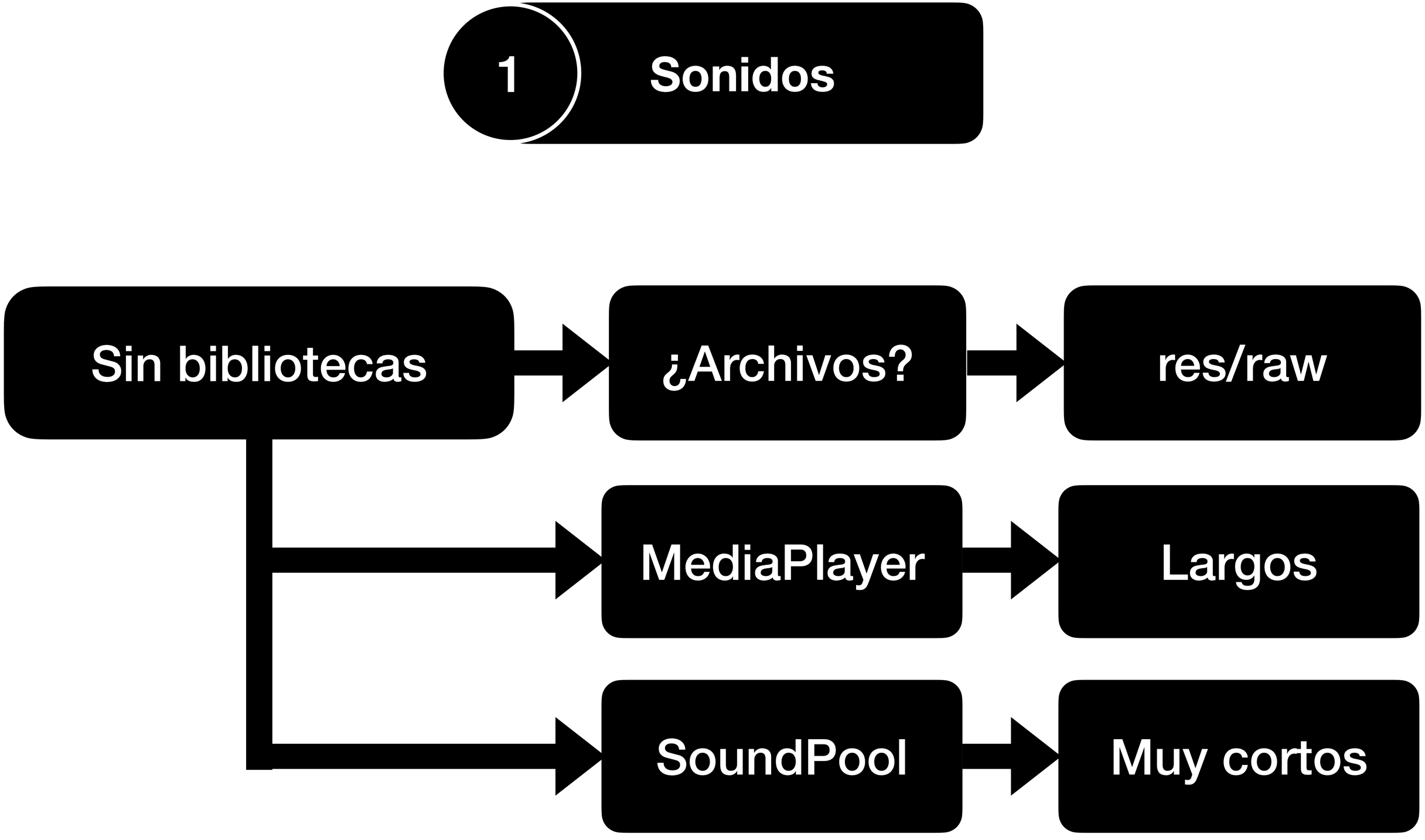
Borrar

5

Actualizar

6

Consultar


1
Sonidos
Sin bibliotecas
¿Archivos?
res/raw
MediaPlayer
Largos
SoundPool
Muy cortos

1 Sonidos

MediaPlayer

```
MediaPlayer mp = MediaPlayer.create(this, R.raw.sonido)
```

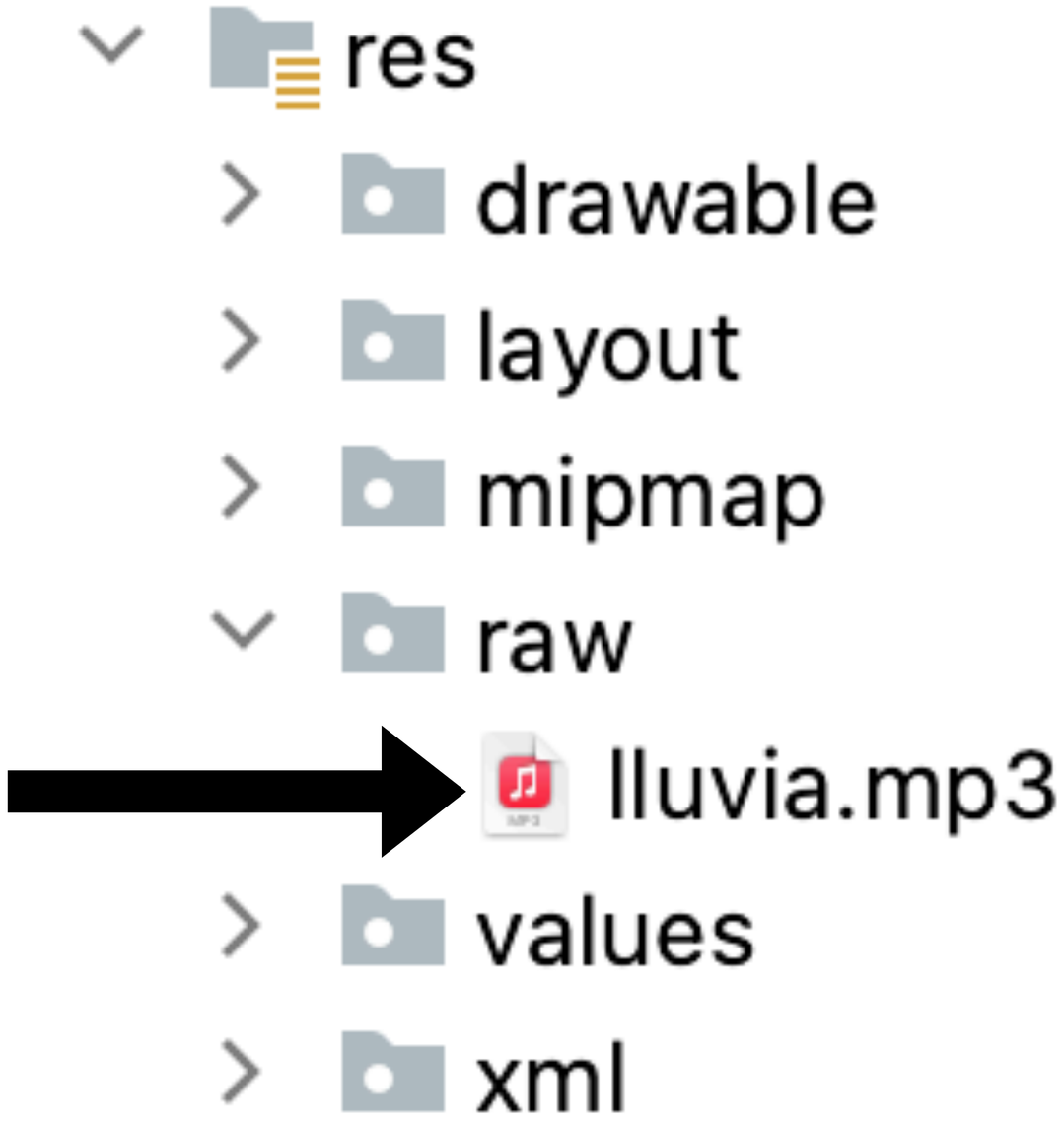
start()

pause()

stop()

release()

1 Sonidos

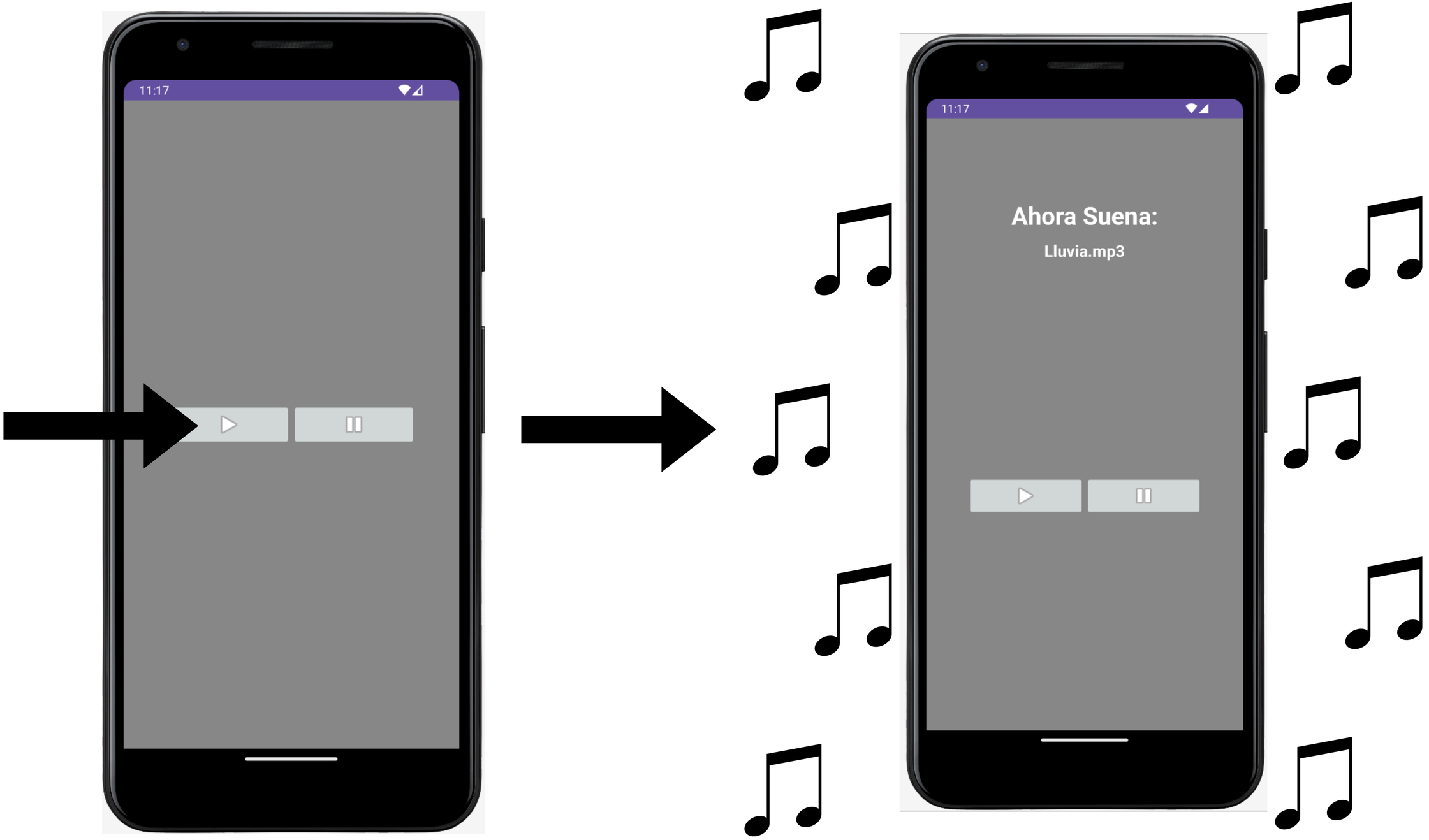


1 Sonidos

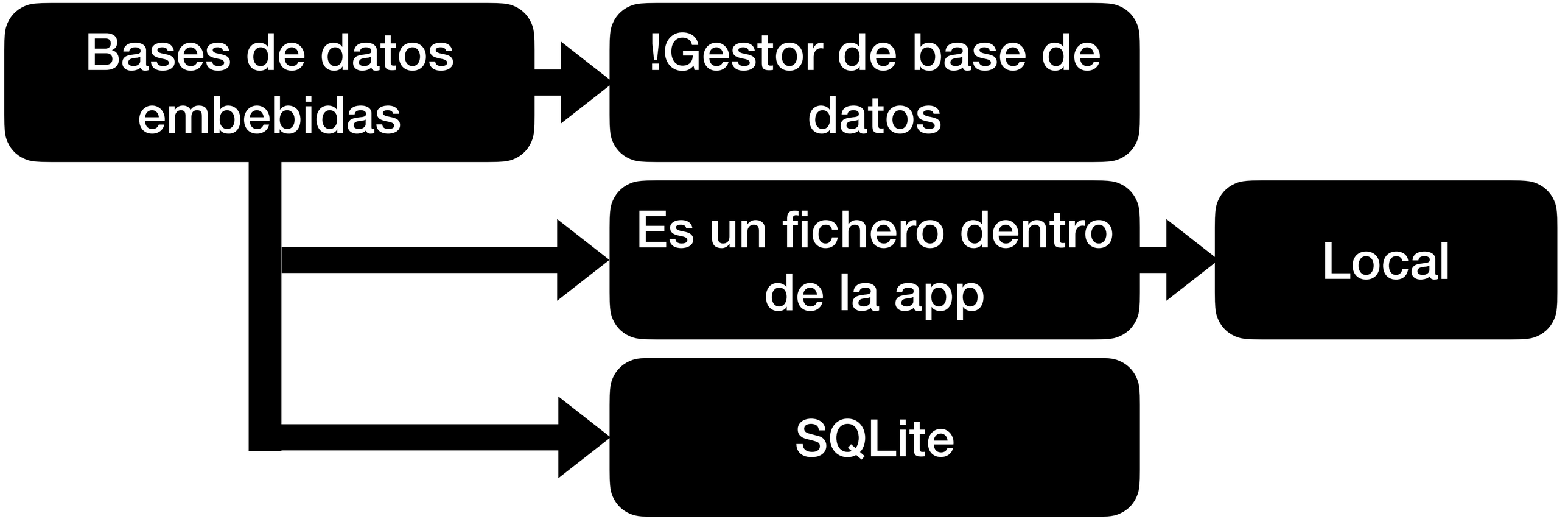
```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MediaPlayer mp = MediaPlayer.create( context: this, R.raw.lluvia);
        mp.start();
    }
}
```

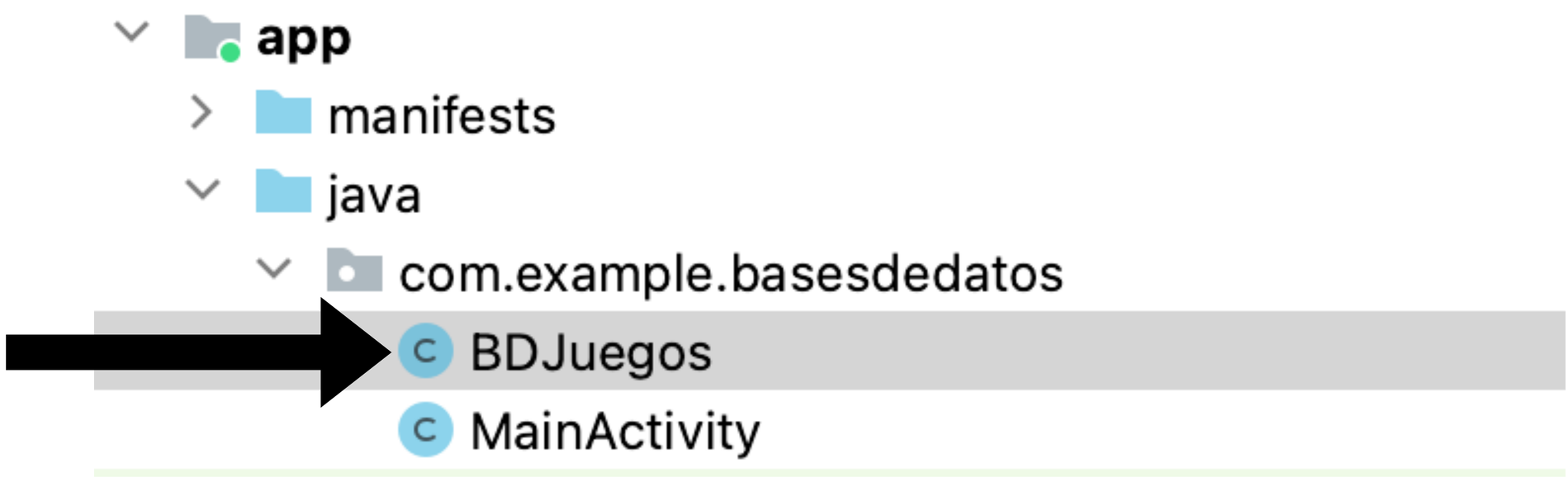

11:17
11:17
Ahora Suena:
Lluvia.mp3

2 BD


Bases de datos embebidas
!Gestor de base de datos
Es un fichero dentro de la app
Local
SQLite

2 BD

Nos creamos una clase para manejar todo

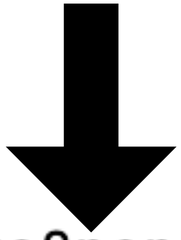


2 BD

Nos creamos una clase para manejar todo

BDJuegos.java

```
1  package com.example.basesdedatos;
2
3  import android.database.sqlite.SQLiteOpenHelper;
4
   no usages
5  public class BDJuegos extends SQLiteOpenHelper {
6
7  }
```



2 BD

Al heredar de SQLiteOpenHelper necesitamos implementar dos métodos

onCreate():
Introduciremos el código de creación de tablas

onUpgrade():
Introduciremos el código para actualizar la base de datos

También necesitamos un constructor

2 BD

Constructor

```
private static final String DATABASE_NAME = "BDJuegos.db";
1 usage
private static final int DATABASE_VERSION = 1;
1 usage
private Context contexto;
no usages
public BDJuegos(Context contexto)
{
    super(contexto, DATABASE_NAME,  factory: null, DATABASE_VERSION);
    this.contexto = contexto;
}
```

2 BD

onCreate()

```
private String SQLCREATE = "CREATE TABLE Juegos (Nombre TEXT, Plataforma TEXT)";
```

no usages

```
private SQLiteDatabase bd = null;


  @Override
  public void onCreate(SQLiteDatabase sqLiteDatabase) {
      sqLiteDatabase.execSQL(SQLCREATE);
  }
```

2 BD

onUpgrade()

```
private String SQLDROP = "DROP TABLE IF EXISTS Juegos";

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase,
                      int preVersion,
                      int NewVersion) {
    sqLiteDatabase.execSQL(SQLDROP);
    sqLiteDatabase.execSQL(SQLCREATE);
}
```

2 BD

Necesitamos un método para cerrar la conexión con la BD

```
private SQLiteDatabase bd = null;
public void cerrarBD()
{
    if(bd != null)
    {
        bd.close();
    }
}
```

3 Insertar

Obtenemos la base de datos en modo escritura

Insertamos los datos

execSQL():
Introducimos una sentencia SQL

execSQL():
Introduciremos un objeto de tipo ContentValues

Cerramos la conexión a la base de datos

3 Insertar

Obtenemos la base de datos en modo escritura

```
public void insertaVideoJuego()
{
    bd = getWritableDatabase();
}
```

3 Insertar

Insertamos los datos. execSQL(): Introducimos una sentencia SQL

```
private String SQLINSERT = "INSERT INTO Juegos (Nombre, Plataforma) VALUES ('GTA VI', 'PS5')";
no usages
public void insertaVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        bd.execSQL(SQLINSERT);
    }
}
```

3 Insertar

Insertamos los datos. execSQL(): Introduciremos un objeto de tipo ContentValues

```
public void insertaVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        ContentValues values = new ContentValues();
        values.put("Nombre", "GTA VI");
        values.put("Plataforma", "PS5");
        bd.insert( table: "Juegos",  nullColumnHack: "", values);
    }
}
```

3 Insertar

Cerramos la conexión a la base de datos

```
public void insertaVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        ContentValues values = new ContentValues();
        values.put("Nombre", "GTA VI");
        values.put("Plataforma", "PS5");
        bd.insert( table: "Juegos",  nullColumnHack: "", values);
        close();
    }
}
```

4 Borrar

Obtenemos la base de datos en modo escritura

Eliminamos los datos

execSQL():
Introducimos una sentencia SQL

delete()

Cerramos la conexión a la base de datos

4 Borrar

Obtenemos la base de datos en modo escritura

```
public void borraVideoJuego()
{
    bd = getWritableDatabase();
}
```

4 Borrar

Eliminamos los datos. execSQL(): Introducimos una sentencia SQL

```
private String SQLDELETE = "DELETE FROM Juegos WHERE Nombre='GTA VI'";
no usages
public void borraVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        bd.execSQL(SQLDELETE);
    }
}
```

4 Borrar

Eliminamos los datos. delete().

```
public void borraVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        String where = "Nombre = ?";

        String[] deleteArguments = {"GTA VI"};

        bd.delete( table: "Juegos", where, deleteArguments);
    }
}
```

4 Borrar

Cerramos la conexión a la base de datos

```
public void borraVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        String where = "Nombre = ?";

        String[] deleteArguments = {"GTA VI"};

        bd.delete( table: "Juegos", where, deleteArguments);
        close();
    }
}
```

5 Actualizar

Obtenemos la base de datos en modo escritura

Actualizamos los datos

execSQL():
Introducimos una sentencia SQL

update()

Cerramos la conexión a la base de datos

5 Actualizar

Obtenemos la base de datos en modo escritura

```
public void actualizaVideoJuego()
{
    bd = getWritableDatabase();
}
```

5 Actualizar

Actualizamos los datos. execSQL(): Introducimos una sentencia SQL

```
private String SQLUPDATE = "UPDATE Juegos SET Plataforma='PS6' WHERE Nombre='GTA VI'";
no usages
public void actualizaVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        bd.execSQL(SQLUPDATE);
    }
}
```

5 Actualizar

Actualizamos los datos. update().

```
public void actualizaVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        ContentValues values = new ContentValues();
        values.put("Nombre", "GTA VI");
        values.put("Plataforma", "PS6");

        String where = "Nombre = ? ";
        String[] updateArguments = {"GTA VI"};

        bd.update( table: "Juegos", values, where, updateArguments);
    }
}
```

5 Actualizar

Cerramos la conexión a la base de datos

```
public void actualizaVideoJuego()
{
    bd = getWritableDatabase();

    if(bd != null)
    {
        ContentValues values = new ContentValues();
        values.put("Nombre", "GTA VI");
        values.put("Plataforma", "PS6");

        String where = "Nombre = ? ";
        String[] updateArguments = {"GTA VI"};

        bd.update( table: "Juegos", values, where, updateArguments);
        close();
    }
}
```

6 Consultar

Obtenemos la base de datos en modo lectura

Leemos valores de la base de **datos**

6 Consultar

Obtenemos la base de datos en modo lectura

```
public ArrayList<Juego> obtenerTodosLosJuegos()
{
    bd = getReadableDatabase();
}
```

6 Consultar

Recursos adicionales: Clase Juego

```
public class Juego
{
    1 usage
    private String Nombre;
    1 usage
    private String Plataforma;

    1 usage
    public Juego(String nombre, String plataforma)
    {
        this.Nombre = nombre;
        this.Plataforma = plataforma;
    }
}
```

6 Consultar

Leemos valores de la base de datos

```
public ArrayList<Juego> obtenerTodosLosJuegos()
{
    bd = getReadableDatabase();

    ArrayList<Juego> juegos = new ArrayList<>();

    Cursor c = bd.query
            (
                table: "Juegos",
                    columns: null,
                    selection: null,
                    selectionArgs: null,
                    groupBy: null,
                    having: null,
                    orderBy: null,
                    limit: null
            );
```

6 Consultar

Leemos valores de la base de datos

```
c.moveToFirst();
if(c.getCount() > 0)
{
    do
    {
        juegos.add(new Juego(c.getString( i: 0), c.getString( i: 1)));
    }
    while(c.moveToNext());
}
close();
return juegos;
```