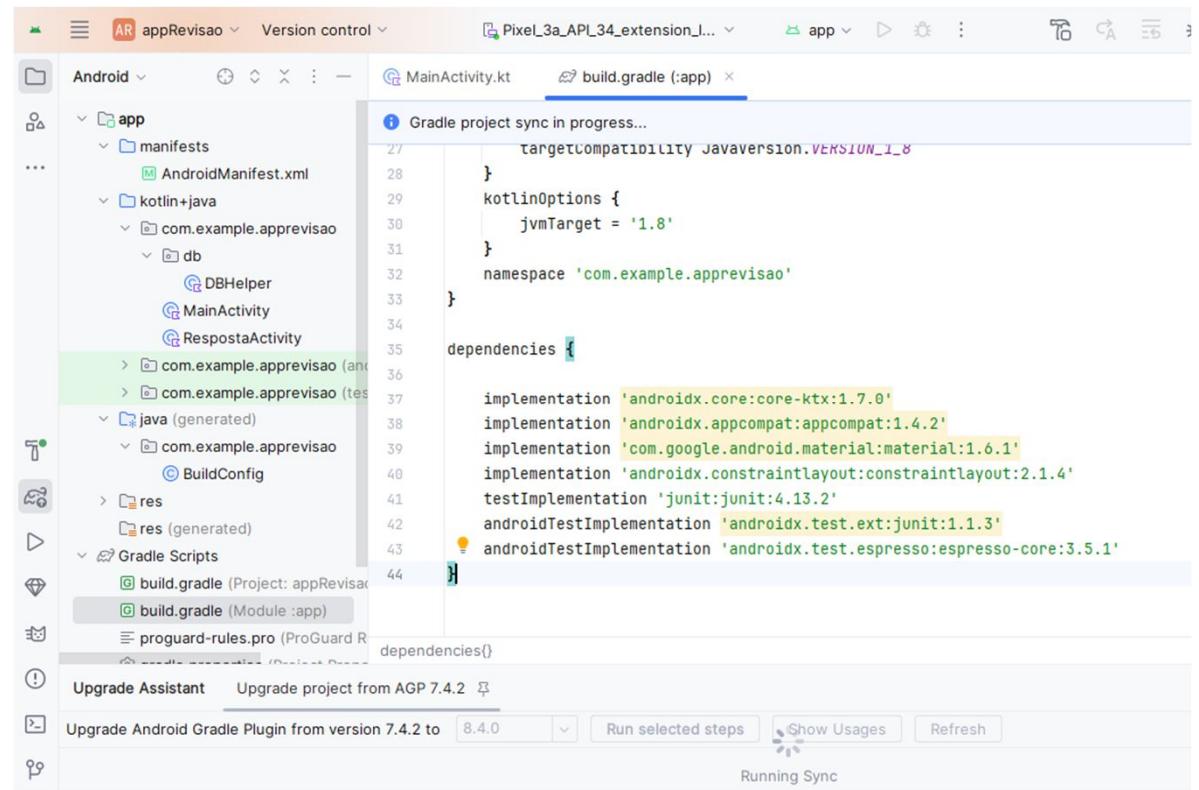


Sql lite - android studio

# APP SQL LITE

- Iremos criar uma aplicação no Jet Pack Composer:
- Incluir as dependências:



```
targetCompatibility JavaVersion.VERSION_1_8
kotlinOptions {
    jvmTarget = '1.8'
}
namespace 'com.example.apprevisao'

dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.4.2'
    implementation 'com.google.android.material:material:1.6.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
}
```

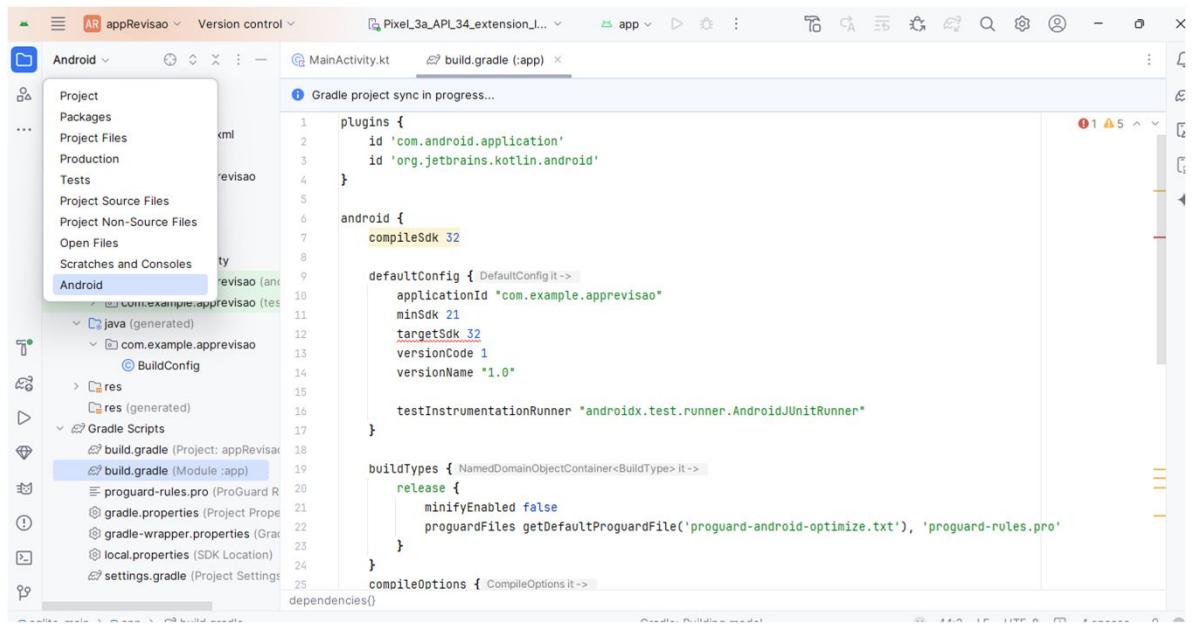
The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `build.gradle` file for the `:app` module.

```
sourceCompatibility JavaVersion.VERSION_1_8  
targetCompatibility JavaVersion.VERSION_1_8  
}  
kotlinOptions {  
    jvmTarget = '1.8'  
}  
namespace 'com.example.apprevisao'  
}  
  
dependencies {  
  
    implementation 'androidx.core:core-ktx:1.7.0'  
    implementation 'androidx.appcompat:appcompat:1.4.2'  
    implementation 'com.google.android.material:material:1.6.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'  
}
```

A red arrow points to the `build.gradle` file in the project tree, highlighting it. The status bar at the bottom shows the path `salite-main > app > build.gradle` and the time `44:2`.

# APP SQL LITE

- Para o funcionamento das classes e métodos no Jet Pack Composer devemos Inserir os plugins no projeto:



The screenshot shows the Android Studio interface with the project 'appRevisao' open. The left sidebar displays the project structure, including 'Project', 'Packages', 'Project Files', 'Production', 'Tests', 'Project Source Files', 'Project Non-Source Files', 'Open Files', 'Scratches and Consoles', and 'Android'. The 'Android' section is selected. The right pane shows the 'build.gradle (:app)' file with the following code:

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}

android {
    compileSdk 32

    defaultConfig {
        applicationId "com.example.apprevisao"
        minSdk 21
        targetSdk 32
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies{}
```

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}

android {
    compileSdk 32

    defaultConfig { DefaultConfig it ->
        applicationId "com.example.apprevisao"
        minSdk 21
        targetSdk 32
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes { NamedDomainObjectContainer<BuildType> it ->
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

compileOptions { CompileOptions it ->
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
```

# APP SQL LITE

- No MainActivity devemos importar as bibliotecas widget para utilizar o TexFild e Button

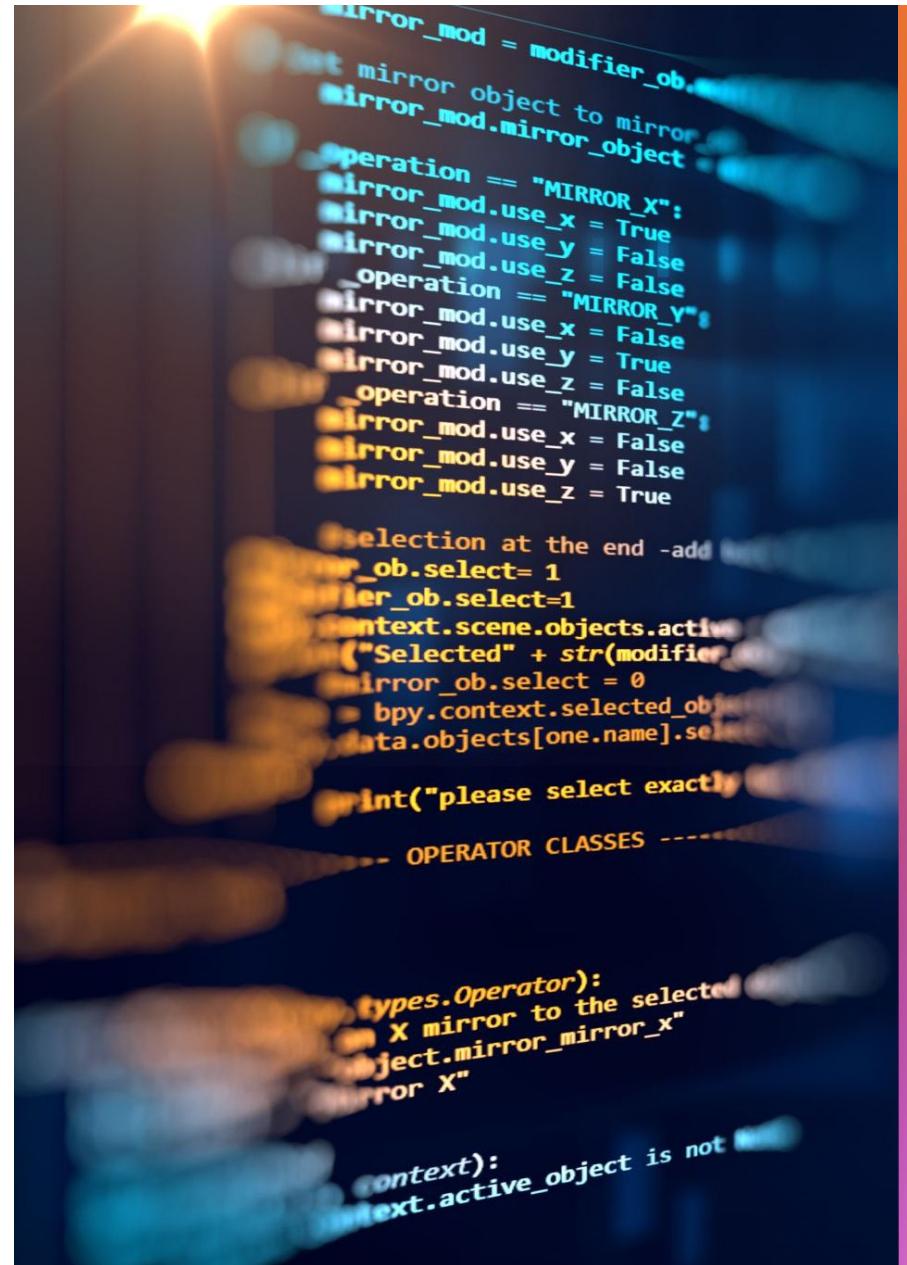
```
1 package com.example.apprevisao
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.widget.Button
7 import android.widget.EditText
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val edtNome:EditText = findViewById(R.id.edtNome)
15         val edtEndereco:EditText = findViewById(R.id.edtEndereco)
16         val edtBairro:EditText = findViewById(R.id.edtBairro)
17         val edtCep:EditText = findViewById(R.id.edtCep)
18
19         val btnCadastrar:Button = findViewById(R.id.btnCadastrar)
20
21         btnCadastrar.setOnClickListener { it:View!
22             val intent = Intent(packageContext: this, RespostaActivity::class.java)
23             intent.putExtra(name: "nome", edtNome.text.toString())
24             intent.putExtra(name: "endereco", edtEndereco.text.toString())
25             intent.putExtra(name: "bairro", edtBairro.text.toString())
26             intent.putExtra(name: "cep", edtCep.text.toString())
27             startActivity(intent)
28         }
29     }
30 }
```

The screenshot shows the Android Studio interface with the file `MainActivity.kt` open in the editor. The code defines a `MainActivity` that extends `AppCompatActivity`. It sets the content view to `activity_main` and finds views for `edtNome`, `edtEndereco`, `edtBairro`, and `edtCep`. It also finds a button `btnCadastrar`. A red brace groups the four EditText variable declarations. A red arrow points from the brace to the `btnCadastrar` declaration. Another red arrow points from the `btnCadastrar` declaration back to the brace.

```
1 package com.example.apprevisao
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.widget.Button
7 import android.widget.EditText
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val edtNome:EditText = findViewById(R.id.edtNome)
15         val edtEndereco:EditText = findViewById(R.id.edtEndereco)
16         val edtBairro:EditText = findViewById(R.id.edtBairro)
17         val edtCep:EditText = findViewById(R.id.edtCep)
18
19         val btnCadastrar:Button = findViewById(R.id.btnCadastrar)
20
21         btnCadastrar.setOnClickListener { it:View!
22             val intent = Intent( packageContext: this, RespostaActivity::class.java)
23             intent.putExtra( name: "nome", edtNome.text.toString())
24             intent.putExtra( name: "endereco", edtEndereco.text.toString())
25             intent.putExtra( name: "bairro", edtBairro.text.toString())
26             intent.putExtra( name: "cep", edtCep.text.toString())
27             startActivity(intent)
28 }
```

```
setContentView(R.layout.activity_main
```

- O método `setContentView(R.layout.activity\_main)` é uma linha de código comum no desenvolvimento Android, normalmente usada em uma classe `Activity` para definir o layout da interface do usuário para essa atividade.
  - Aqui está uma breve visão geral de seu propósito e uso: ### Finalidade `setContentView` é um método da classe `Activity` que define o layout da atividade.
  - Ao passar `R.layout.activity\_main` como argumento, você especifica qual arquivo de layout XML deve ser usado para exibir a UI para esta atividade.



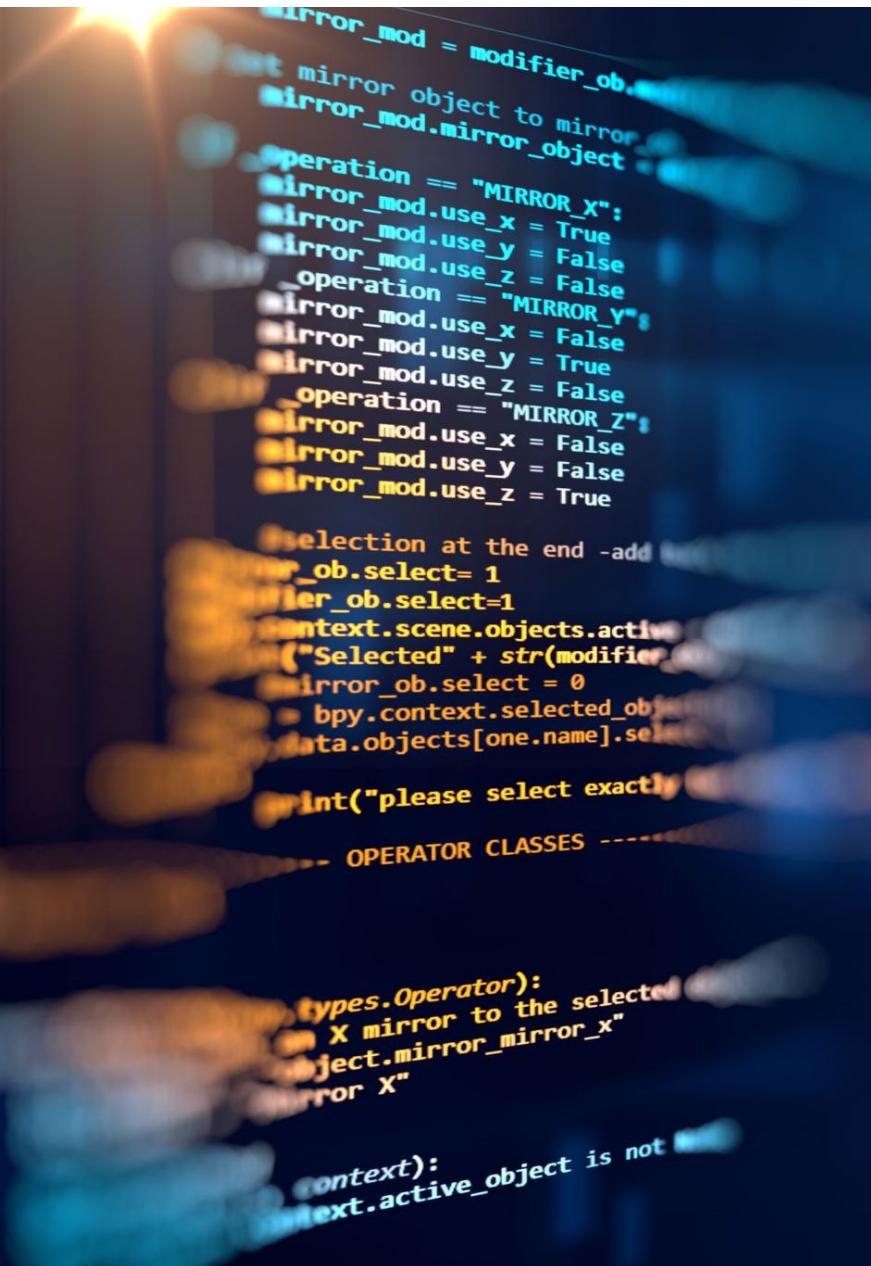
The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MainActivity.kt` file.

```
1 package com.example.apprevisao
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.widget.Button
7 import android.widget.EditText
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val edtNome:EditText = findViewById(R.id.edtNome)
15         val edtEndereco:EditText = findViewById(R.id.edtEndereco)
16         val edtBairro:EditText = findViewById(R.id.edtBairro)
17         val edtCep:EditText = findViewById(R.id.edtCep)
18
19         val btnCadastrar:Button = findViewById(R.id.btnCadastrar)
20
21         btnCadastrar.setOnClickListener { it:View!
22             val intent = Intent( packageName: this, RespostaActivity::class.java)
23             intent.putExtra( name: "nome",edtNome.text.toString())
24             intent.putExtra( name: "endereco",edtEndereco.text.toString())
25             intent.putExtra( name: "bairro",edtBairro.text.toString())
26             intent.putExtra( name: "cep",edtCep.text.toString())
27         }
28     }
29 }
```

A red oval highlights the entire class definition from line 9 to line 28. The code defines a new activity named `MainActivity` that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `activity_main`. Inside the `onCreate` method, it finds views by their IDs (`edtNome`, `edtEndereco`, `edtBairro`, `edtCep`) and a button (`btnCadastrar`). It then sets an `onClick` listener on the button. The listener creates an `Intent` to the `RespostaActivity` and puts extra data into it, including the text from the `edtNome`, `edtEndereco`, `edtBairro`, and `edtCep` EditTexts.

## findViewById(R.id.edtNome)

- No desenvolvimento Android usando Kotlin, a linha `findViewById(R.id.edtNome)` é usada para vincular um elemento de UI de seu arquivo de layout XML a uma variável em sua atividade ou código de fragmento.
- Isso permite que você interaja com o elemento da UI de forma programática.
- Declara uma variável somente leitura.
- Uma vez atribuído, seu valor não pode ser alterado.
- O nome da variável que conterá a referência à view EditText.
- Especifica o tipo da variável.
- `EditText` é uma visualização no Android que permite ao usuário inserir e editar texto.
- `findViewById(R.id.edtNome)`



# Android.sqlite

- Iremos criar variáveis vinculados aos atributos dos parâmetros enviados dos textos que serão inseridos nos textfields.

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. A red arrow points from the project tree to the code editor. A red circle highlights a section of the code in the MainActivity.kt file.

```
import android.widget.EditText

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

        val edtNome:EditText = findViewById(R.id.edtNome)
        val edtEndereco:EditText = findViewById(R.id.edtEndereco)
        val edtBairro:EditText = findViewById(R.id.edtBairro)
        val edtCep:EditText = findViewById(R.id.edtCep)

        btnCadastrar.setOnClickListener {
            val intent = Intent(this, RespostaActivity::class.java)
            intent.putExtra("name",edtNome.text.toString())
            intent.putExtra("endereco",edtEndereco.text.toString())
            intent.putExtra("bairro",edtBairro.text.toString())
            intent.putExtra("cep",edtCep.text.toString())
            startActivity(intent)
        }
    }
}
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the "Android" tab. It includes the `app` module with its sub-directories: `manifests`, `kotlin+java`, and `res`. The `build.gradle` file for the `app` module is currently selected.
- Code Editor:** The main editor window contains the `MainActivity.kt` file. The code defines the `MainActivity` class which extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `R.layout.activity_main`. Inside this method, it finds views by ID for `edtNome`, `edtEndereco`, `edtBairro`, and `edtCep`. It also finds a button `btnCadastrar`. When the button is clicked, it creates an intent to start the `RespostaActivity` and puts extra data for the name, address, neighborhood, and cep into the intent's extra field.
- Toolbars and Status Bar:** The top bar shows the project name "Pixel\_3a\_API\_34\_extension\_L..." and the module "app". The bottom status bar indicates the code has 30.2 / 1 char, 1 F, 1 HFS+, and 4 spaces.

```
import android.widget.EditText

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

        val edtNome:EditText = findViewById(R.id.edtNome)
        val edtEndereco:EditText = findViewById(R.id.edtEndereco)
        val edtBairro:EditText = findViewById(R.id.edtBairro)
        val edtCep:EditText = findViewById(R.id.edtCep)

        val btnCadastrar:Button = findViewById(R.id.btnCadastrar)

        btnCadastrar.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, RespostaActivity::class.java)
            intent.putExtra( name: "nome", edtNome.text.toString())
            intent.putExtra( name: "endereco", edtEndereco.text.toString())
            intent.putExtra( name: "bairro", edtBairro.text.toString())
            intent.putExtra( name: "cep", edtCep.text.toString())
            startActivity(intent)
        }
    }
}
```

# Android.sqlite

- Iremos criar variável que implementa o método criado para o botão inserir os dados armazenados nos textos digitados.

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The project structure includes an 'app' folder containing 'manifests', 'kotlin+java', and 'res' folders. The 'kotlin+java' folder contains 'com.example.appRevisao' and 'com.example.appRevisao (generated)' packages. The 'MainActivity' class is located in the first package, and 'ExampleInstrumentedTest' and 'ExampleUnitTest' are in the second. The code editor displays the 'MainActivity.kt' file. A red oval highlights the line 'val btnCadastrar:Button = findViewById(R.id.btnCadastrar)'. The code implements an onClickListener for this button, reading text from four EditText fields and starting an activity with the intent.

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

        val edtNome:EditText = findViewById(R.id.edtNome)
        val edtEndereco:EditText = findViewById(R.id.edtEndereco)
        val edtBairro:EditText = findViewById(R.id.edtBairro)
        val edtCep:EditText = findViewById(R.id.edtCep)

        val btnCadastrar:Button = findViewById(R.id.btnCadastrar)

        btnCadastrar.setOnClickListener { it:View!
            val intent = Intent( packageName: this, RespostaActivity::class.java)
            intent.putExtra( name: "nome", edtNome.text.toString())
            intent.putExtra( name: "endereco", edtEndereco.text.toString())
            intent.putExtra( name: "bairro", edtBairro.text.toString())
            intent.putExtra( name: "cep", edtCep.text.toString())
            startActivity(intent)
        }
    }
}
```

The screenshot shows the Android Studio interface with the following details:

- Project Bar:** AR appRevisao, Version control, Pixel\_3a\_API\_34\_extension\_I..., app, navigation icons.
- Toolbars:** Standard Android Studio toolbars for search, code navigation, and file operations.
- Left Sidebar:** Project tree showing the project structure:
  - app
    - manifests: AndroidManifest.xml
    - kotlin+java
      - com.example.apprevisao
        - MainActivity
        - RespostaActivity
      - com.example.apprevisao (and)
      - ExampleInstrumentedTest
      - com.example.apprevisao (tes)
      - ExampleUnitTest
    - java (generated)
    - res
    - res (generated)
  - Gradle Scripts
    - build.gradle (Project: appRevisao)
    - build.gradle (Module :app) - selected
    - proguard-rules.pro (ProGuard R)
    - gradle.properties (Project Properties)
    - gradle-wrapper.properties (Gradle)
    - local.properties (SDK Location)
    - settings.gradle (Project Settings)
- Central Area:** Editor showing the MainActivity.kt file content.

```
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val edtNome:EditText = findViewById(R.id.edtNome)
15         val edtEndereco:EditText = findViewById(R.id.edtEndereco)
16         val edtBairro:EditText = findViewById(R.id.edtBairro)
17         val edtCep:EditText = findViewById(R.id.edtCep)
18
19         val btnCadastrar:Button = findViewById(R.id.btnCadastrar)
20
21         btnCadastrar.setOnClickListener { it:View!
22             val intent = Intent( packageContext: this, RespostaActivity::class.java)
23             intent.putExtra( name: "nome", edtNome.text.toString())
24             intent.putExtra( name: "endereco", edtEndereco.text.toString())
25             intent.putExtra( name: "bairro", edtBairro.text.toString())
26             intent.putExtra( name: "cep", edtCep.text.toString())
27             startActivity(intent)
28         }
29     }
30 }
```
- Bottom Status Bar:** sqlite-main > app > src > main > java > com > example > apprevisao > MainActivity, 30:2 LF, UTF-8, 4 spaces.

```
8
9    class MainActivity : AppCompatActivity() {
10        override fun onCreate(savedInstanceState: Bundle?) {
11            super.onCreate(savedInstanceState)
12            setContentView(R.layout.activity_main)
13
14            val edtNome:EditText = findViewById(R.id.edtNome)
15            val edtEndereco:EditText = findViewById(R.id.edtEndereco)
16            val edtBairro:EditText = findViewById(R.id.edtBairro)
17            val edtCep:EditText = findViewById(R.id.edtCep)
18
19            val btnCadastrar:Button = findViewById(R.id.btnCadastrar)
20
21            btnCadastrar.setOnClickListener { it:View!
22                val intent = Intent( packageContext: this, RespostaActivity::class.java )
23                intent.putExtra( name: "nome", edtNome.text.toString() )
24                intent.putExtra( name: "endereco", edtEndereco.text.toString() )
25                intent.putExtra( name: "bairro", edtBairro.text.toString() )
26                intent.putExtra( name: "cep", edtCep.text.toString() )
27                startActivity(intent)
28            }
29        }
30    }
```

## Android.sqlite

- A linha `startActivity(intent)` é usada no desenvolvimento Android para iniciar uma nova atividade. Aqui está uma visão detalhada do que esta linha faz e como ela se encaixa no processo geral de navegação entre atividades:

The screenshot shows the Android Studio interface with the following details:

- Toolbar:** AR appRevisao, Version control, Pixel\_3a\_API\_34\_extension\_l..., app, navigation icons.
- Side Navigation Bar:** Icons for File, Project, Build, Run, and others.
- Project Structure:** Shows the project tree under "Android".
  - app:** manifests (AndroidManifest.xml), kotlin+java (com.example.apprevisao (MainActivity, RespostaActivity)), com.example.apprevisao (ExampleInstrumentedTest, ExampleUnitTest).
  - java (generated):** com.example.apprevisao (BuildConfig).
  - res:**
  - Gradle Scripts:** build.gradle (Project: appRevisao), build.gradle (Module :app), proguard-rules.pro, gradle.properties, gradle-wrapper.properties, local.properties, settings.gradle.
- Code Editor:** MainActivity.kt (selected tab). The code defines the MainActivity class which overrides onCreate and sets the content view to activity\_main. It also initializes EditTexts for name, address, neighborhood, and cep, and a button for saving. The button's onClickListener starts an Intent to RespostaActivity, passing the text values from the EditTexts as extras.

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

        val edtNome:EditText = findViewById(R.id.edtNome)
        val edtEndereco:EditText = findViewById(R.id.edtEndereco)
        val edtBairro:EditText = findViewById(R.id.edtBairro)
        val edtCep:EditText = findViewById(R.id.edtCep)

        val btnCadastrar:Button = findViewById(R.id.btnCadastrar)

        btnCadastrar.setOnClickListener { it:View!
            val intent = Intent( packageContext: this, RespostaActivity::class.java)
            intent.putExtra( name: "nome",edtNome.text.toString())
            intent.putExtra( name: "endereco",edtEndereco.text.toString())
            intent.putExtra( name: "bairro",edtBairro.text.toString())
            intent.putExtra( name: "cep",edtCep.text.toString())
            startActivity(intent)
        }
    }
}
```
- Bottom Navigation:** sqlite-main > app > src > main > java > com > example > apprevisao > MainActivity. Status bar: 30:2, LF, UTF-8, 4 spaces.

---

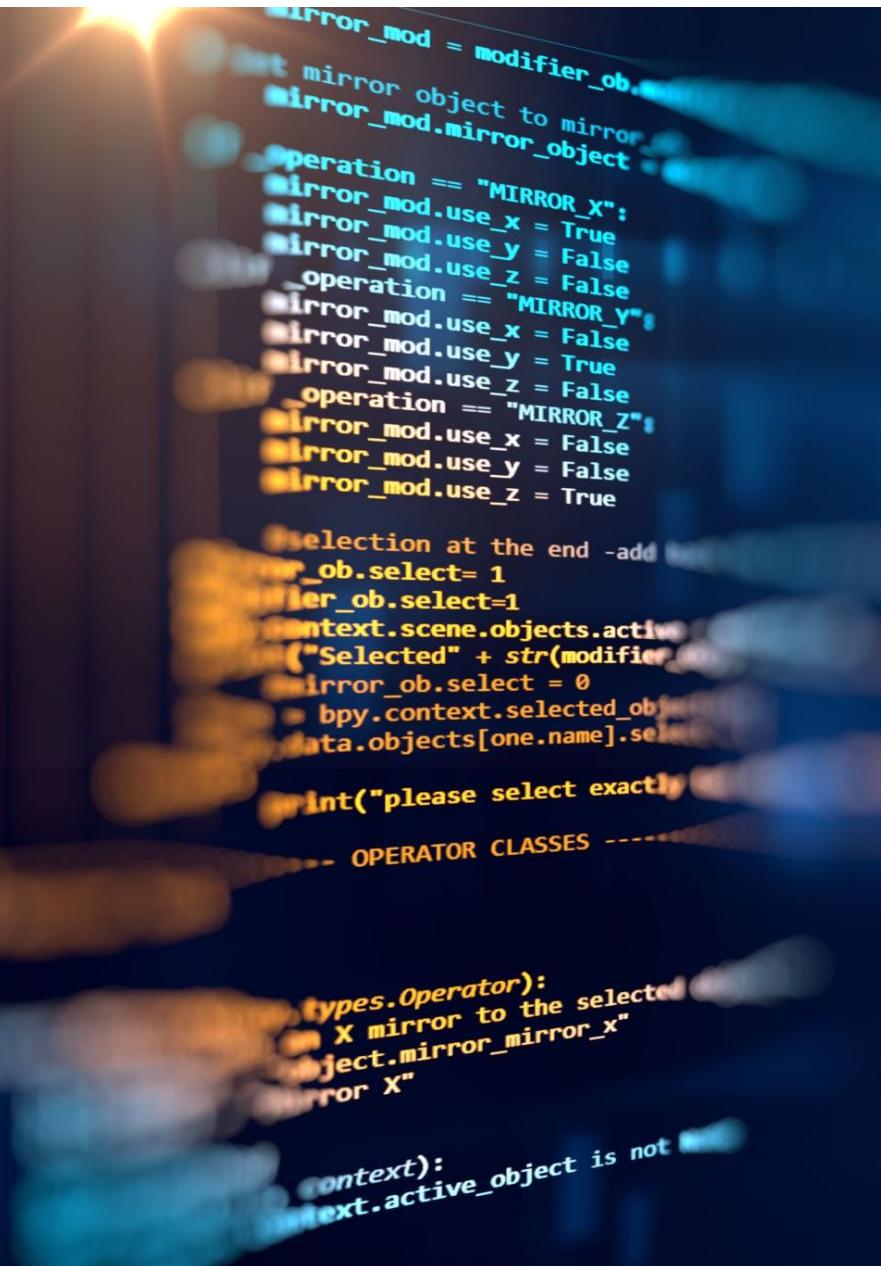
## RespostaActivity.kt

- Salvando as informações do MainActivity o método SetText envia as informações para o banco de dados.
- Ao clicar o botão Cadastrar é acionado o método SetOnClickListener;



## Android.sqlite

- O db.addPessoa(edtNome.text.toString(),  
edtEndereco.text.toString(),  
edtBairro.text.toString(), edtCep.text.toString())  
finish()
- Envia as informações setadas nas variáveis e  
armazena no SQLite nativo do Android Studio.
- A biblioteca androidx.sqlite contém interfaces  
abstratas com implementações básicas que  
podem ser usadas para criar bibliotecas que  
acessam o SQLite



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the "Android" tab. It includes the `app` module with its sub-directories: `manifests`, `kotlin+java`, and `db`. Inside `db`, the `DBHelper` file is selected and highlighted in grey. Other files like `MainActivity` and `RespostaActivity` are also visible.
- Code Editor:** The main window shows the `DBHelper.kt` file content. The code defines a class `DBHelper` that extends `SQLiteOpenHelper`. It contains an `onCreate` method which generates a SQL query to create a table named `TABLE_NAME` with columns `ID_COL` (INTEGER PRIMARY KEY), `NAME_COL` (TEXT), `END_COL` (TEXT), `BAR_COL` (TEXT), and `CEP_COL` (TEXT).
- Toolbars and Status:** The top bar shows the current device as "Pixel\_3a\_API\_34\_extension\_l..." and the app flavor as "app". The bottom navigation bar includes tabs for "Upgrade Assistant" and "Upgrade project from AGP 8.4.0". A message in the status bar indicates that the project was successfully upgraded from AGP 8.4.0 to 8.4.1.

The screenshot shows the Android Studio interface with the following details:

- Title Bar:** AR appRevisão Version control Pixel\_3a\_API\_34\_extension\_l... app
- Side Navigation Bar:** Shows project structure under "Android". The "app" module is expanded, showing "manifests", "kotlin+java", "res", and "java (generated)". "kotlin+java" contains "com.example.apprevisao" which has "DBHelper", "MainActivity", and "RespostaActivity". "res" contains "drawable". "java (generated)" contains "com.example.apprevisao" with "BuildConfig". "Bookmarks" section has "sqlite-main" selected.
- Code Editor:** Displays the content of `DBHelper.kt`. The code is as follows:

```
// below is the method for creating a database by a sqlite query
override fun onCreate(db: SQLiteDatabase) {
    // below is a sqlite query, where column names
    // along with their data types is given
    val query = ("CREATE TABLE " + TABLE_NAME + " (" +
        + ID_COL + " INTEGER PRIMARY KEY, " +
        NAME_COL + " TEXT, " +
        END_COL + " TEXT, " +
        BAR_COL + " TEXT, " +
        CEP_COL + " TEXT" + ")")

    // we are calling sqlite
    // method for executing our query
    db.execSQL(query)
}

override fun onUpgrade(db: SQLiteDatabase, p1: Int, p2: Int) {
    // this method is to check if table already exists
    db.execSQL(sql: "DROP TABLE IF EXISTS " + TABLE_NAME)
    onCreate(db)
}

// This method is for adding data in our database
fun addPessoa(name : String, endereco : String, bairro : String, cep : String){

    // below we are creating
    // a content values variable
    val values = ContentValues()
```

- Bottom Status Bar:** sqlite-main > app > src > main > java > com > example > apprevisao > db > DBHelper > onUpgrade 30:21 LF UTF-8 4 spaces

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the "app" module. Key components shown include:
  - manifests:** Contains `AndroidManifest.xml`.
  - kotlin+java:** Contains the package `com.example.apprevisao`, which includes files `DBHelper`, `MainActivity`, and `RespostaActivity`. `DBHelper` is currently selected.
  - com.example.apprevisao (androidTest):** Contains `ExampleInstrumentedTest`.
  - com.example.apprevisao (test):** Contains `ExampleUnitTest`.
  - java (generated):** Contains the file `BuildConfig`.
  - res:** Contains the folder `drawable`.
- Code Editor:** The main editor window displays the `DBHelper.kt` file. The code implements a database helper class with methods for executing SQL queries, upgrading the database, and adding new people to the database.
- Toolbars and Status Bar:** The top bar includes standard Android Studio icons for file operations, navigation, and search. The bottom status bar shows the current time (48:21), file encoding (UTF-8), and code style settings (4 spaces).

```
// we are calling sqlite
// method for executing our query
db.execSQL(query)

}

override fun onUpgrade(db: SQLiteDatabase, p1: Int, p2: Int) {
    // this method is to check if table already exists
    db.execSQL( sql: "DROP TABLE IF EXISTS " + TABLE_NAME)
    onCreate(db)
}

// This method is for adding data in our database
fun addPessoa(name : String, endereco : String, bairro : String, cep : String){

    // below we are creating
    // a content values variable
    val values = ContentValues()

    // we are inserting our values
    // in the form of key-value pair
    values.put(NAME_COL, name)
    values.put(END_COL, endereco)
    values.put(BAR_COL, bairro)
    values.put(CEP_COL, cep)

    // here we are creating a
    // writable variable of
    // our database as we want to
}
```

The screenshot shows the Android Studio interface with the following details:

- Title Bar:** AR appRevisao Version control Pixel\_3a\_API\_34\_extension\_l... app
- Side Navigation Bar:** Shows the project structure under "Android". The "app" module is expanded, showing "manifests", "kotlin+java", "res", and "java (generated)". "kotlin+java" contains "com.example.apprevisao" which has "db", "MainActivity", "RespostaActivity", and "ExampleInstrumentedTest". "res" contains "drawable". "java (generated)" contains "com.example.apprevisao" which has "BuildConfig". "Bookmarks" lists "sqlite-main".
- Central Editor Area:** Displays the code for `DBHelper.kt`. The code is as follows:

```
38     val values = ContentValues()
39
40     // we are inserting our values
41     // in the form of key-value pair
42     values.put(NAME_COL, name)
43     values.put(END_COL, endereco)
44     values.put(BAR_COL, bairro)
45     values.put(CEP_COL, cep)
46
47     // here we are creating a
48     // writable variable of
49     // our database as we want to
50     // insert value in our database
51     val db = this.writableDatabase
52
53     // all values are inserted into database
54     db.insert(TABLE_NAME, nullColumnHack: null, values)
55
56     // at last we are
57     // closing our database
58     db.close()
59 }
60
61 companion object{
62     // here we have defined variables for our database
63
64     // below is variable for database name
65     private val DATABASE_NAME = "appSQLite"
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "Android". The "app" module is expanded, showing "manifests", "kotlin+java", and "res" directories. "kotlin+java" contains "com.example.apprevisao" which has "db", "MainActivity", and "RespostaActivity". "res" contains "drawable".
- Code Editor:** The main window shows the content of the `DBHelper.kt` file. The code defines a database helper class with variables for database name, version, and table names.
- Status Bar:** At the top, there are tabs for "Pixel\_3a\_API\_34\_extension\_l..." and "app". To the right of the tabs are various icons for navigating between files, switching between tabs, and performing other operations.
- Right Sidebar:** On the far right, there is a vertical toolbar with icons for navigation, search, and other development tools.

```
// closing our database
db.close()

}

companion object{
    // here we have defined variables for our database

    // below is variable for database name
private val DATABASE_NAME = "appSQLite"

    // below is the variable for database version
private val DATABASE_VERSION = 1

    // below is the variable for table name
val TABLE_NAME = "CadastroPessoa"

    // below is the variable for id column
val ID_COL = "id"

    // below is the variable for name column
val NAME_COL = "name"

    // below is the variable for age column
val END_COL = "endereco"

    // below is the variable for age column
val BAR_COL = "bairro"
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "Android". The "app" module is expanded, showing "manifests", "kotlin+java", and "res". "kotlin+java" contains "com.example.apprevisao" which has "DBHelper", "MainActivity", and "RespostaActivity". "com.example.apprevisao" is also selected in the "Bookmarks" section.
- Code Editor:** The main window shows the content of the "DBHelper.kt" file. The code defines variables for database name, version, table name, and columns.
- Status Bar:** At the bottom, the status bar shows the file path "sqlite-main > app > src > main > java > com > example > apprevisao > db > DBHelper", and the settings "88:2 LF UTF-8 4 spaces".

```
// here we have defined variables for our database
private val DATABASE_NAME = "appSQLite"

// below is the variable for database version
private val DATABASE_VERSION = 1

// below is the variable for table name
val TABLE_NAME = "CadastroPessoa"

// below is the variable for id column
val ID_COL = "id"

// below is the variable for name column
val NAME_COL = "name"

// below is the variable for age column
val END_COL = "endereco"

// below is the variable for age column
val BAR_COL = "bairro"

// below is the variable for age column
val CEP_COL = "cep"
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the `app` module. It includes the `manifests`, `kotlin+java`, `res`, and `java (generated)` sections. The `MainActivity` file is selected in the `kotlin+java` section.
- Code Editor:** The main window shows the `MainActivity.kt` file. The code defines an `MainActivity` class that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `R.layout.activity_main`. Inside the `onCreate` method, it finds views by ID: `edtNome`, `edtEndereco`, `edtBairro`, `edtCep`, and `btnCadastrar`. It then sets an `onClickListener` for the `btnCadastrar` button. The listener checks the text of the `edtNome`, `edtEndereco`, `edtBairro`, and `edtCep` EditTexts and creates an Intent to start the `RespostaActivity`.
- Toolbars and Status Bar:** The top bar shows the project name `Pixel_3a_API_34_extension_l...`, the build variant `app`, and various icons for navigation and search. The bottom status bar indicates the screen resolution is 3-2, the language is English (US), and the battery level is at 100%.

```
package com.example.apprevisao

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText

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

        val edtNome:EditText = findViewById(R.id.edtNome)
        val edtEndereco:EditText = findViewById(R.id.edtEndereco)
        val edtBairro:EditText = findViewById(R.id.edtBairro)
        val edtCep:EditText = findViewById(R.id.edtCep)

        val btnCadastrar:Button = findViewById(R.id.btnCadastrar)

        btnCadastrar.setOnClickListener { it:View!
            val intent = Intent( packageContext: this, RespostaActivity::class.java)
            intent.putExtra( name: "nome",edtNome.text.toString())
            intent.putExtra( name: "endereco",edtEndereco.text.toString())
            intent.putExtra( name: "bairro",edtBairro.text.toString())
            intent.putExtra( name: "cep",edtCep.text.toString())
            startActivity(intent)
        }
    }
}
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the "Android" tab. The "app" module is expanded, showing "manifests", "kotlin+java", and "res". "kotlin+java" contains "com.example.apprevisao" which includes "MainActivity", "DBHelper", "RespostaActivity", and "ExampleInstrumentedTest". "res" contains "drawable".
- Code Editor:** The main editor window shows the "MainActivity.kt" file. The code implements an `AppCompatActivity` and overrides the `onCreate` method to set the content view and find views by ID. It also defines several variables for `EditText` and a `Button`, and sets an `onClick` listener for the button.
- Toolbars and Status Bar:** The top bar shows the project name "Pixel\_3a\_API\_34\_extension\_l..." and the module "app". The status bar indicates the device is a "Pixel 3a API 34" and the battery level is at 5%.

```
import android.widget.EditText

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

        val edtNome:EditText = findViewById(R.id.edtNome)
        val edtEndereco:EditText = findViewById(R.id.edtEndereco)
        val edtBairro:EditText = findViewById(R.id.edtBairro)
        val edtCep:EditText = findViewById(R.id.edtCep)

        val btnCadastrar:Button = findViewById(R.id.btnCadastrar)

        btnCadastrar.setOnClickListener { it:View!
            val intent = Intent( packageName: this, RespostaActivity::class.java)
            intent.putExtra( name: "nome",edtNome.text.toString())
            intent.putExtra( name: "endereco",edtEndereco.text.toString())
            intent.putExtra( name: "bairro",edtBairro.text.toString())
            intent.putExtra( name: "cep",edtCep.text.toString())
            startActivity(intent)
        }
    }
}
```

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** AR appRevisao, Version control, Pixel\_3a\_API\_34\_extension\_l..., app, C, E, S, T, A, E, F, M, Q, G.
- Left Sidebar:** Android, app, manifests (AndroidManifest.xml), kotlin+java (com.example.apprevisao: MainActivity, RespostaActivity), com.example.apprevisao (androidTest, ExampleInstrumentedTest), com.example.apprevisao (test, ExampleUnitTest), java (generated) (com.example.apprevisao: BuildConfig), res, res (generated), Gradle Scripts (build.gradle, build.gradle), proguard-rules.pro.
- Right Sidebar:** Bookmarks (sqlite-main).
- Code Editor:** RespostaActivity.kt

```
1 package com.example.apprevisao
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.widget.Button
6 import android.widget.EditText
7 import com.example.apprevisao.db.DBHelper
8
9 class RespostaActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_resposta)
13
14         val edtNome: EditText = findViewById(R.id.edtNome)
15         val edtEndereco: EditText = findViewById(R.id.edtEndereco)
16         val edtBairro: EditText = findViewById(R.id.edtBairro)
17         val edtCep: EditText = findViewById(R.id.edtCep)
18         val btnConfirmar: Button = findViewById(R.id.btnConfirmar)
19
20         edtNome.setText(intent.getStringExtra(name: "nome"))
21         edtEndereco.setText(intent.getStringExtra(name: "endereco"))
22         edtBairro.setText(intent.getStringExtra(name: "bairro"))
23         edtCep.setText(intent.getStringExtra(name: "cep"))
24
25         // SQLite
26         val db = DBHelper(context: this, factory: null)
27
28         btnConfirmar.setOnClickListener { it: View! }
```

- Bottom Navigation:** sqlite-main > app > src > main > java > com > example > apprevisao > db, 3:8, LF, UTF-8.

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under the "Android" tab. The "app" module is expanded, showing "manifests", "kotlin+java", and "res" directories. "kotlin+java" contains "com.example.apprevisao" package with "MainActivity" and "RespostaActivity" files, both of which are selected and highlighted in grey.
- Code Editor:** The main editor area displays the "RespostaActivity.kt" file. The code implements an AppCompatActivity and overrides the onCreate method to set the content view and initialize four EditText fields (edtNome, edtEndereco, edtBairro, edtCep) with string extra values from the intent. It also initializes a DBHelper and adds a new Pessoa to the database before finishing the activity.
- Toolbars and Status Bar:** The top bar includes standard Android Studio icons for file operations, navigation, and search. The status bar at the bottom shows the current screen resolution (28x18), battery level (15%), signal strength (LTE-R), and other system information.

```
8
9 class RespostaActivity : AppCompatActivity() {
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_resposta)
13
14        val edtNome: EditText = findViewById(R.id.edtNome)
15        val edtEndereco: EditText = findViewById(R.id.edtEndereco)
16        val edtBairro: EditText = findViewById(R.id.edtBairro)
17        val edtCep: EditText = findViewById(R.id.edtCep)
18        val btnConfirmar: Button = findViewById(R.id.btnConfirmar)
19
20        edtNome.setText(intent.getStringExtra("nome"))
21        edtEndereco.setText(intent.getStringExtra("endereco"))
22        edtBairro.setText(intent.getStringExtra("bairro"))
23        edtCep.setText(intent.getStringExtra("cep"))
24
25        // SQLite
26        val db = DBHelper(context, null)
27
28        btnConfirmar.setOnClickListener { v: View! -
29            db.addPessoa(edtNome.text.toString(), edtEndereco.text.toString(), edtBairro.text.toString(),
30                         edtCep.text.toString())
31            finish()
32        }
33    }
34}
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the "Android" tab. The "app" module is expanded, showing "manifests", "kotlin+java", and "res". "kotlin+java" contains "com.example.apprevisao" which includes "MainActivity", "RespostaActivity", "ExampleInstrumentedTest", and "ExampleUnitTest".
- MainActivity.kt:** The code defines four EditTexts (edtNome, edtEndereco, edtBairro, edtCep) and a Button (btnConfirmar). It sets their text based on intent extras and adds them to a SQLite database via DBHelper.
- RespostaActivity.kt:** The code sets the content view to activity\_resposta and handles a button click to add a person to the database and finish the activity.
- Build Variants:** The top bar shows "Pixel\_3a\_API\_34\_extension\_l..." and "app".
- Toolbar:** The top right features standard Android Studio icons for save, run, and search.
- Bottom Bar:** Shows the current file path: "sqlite-main > app > src > main > java > com > example > apprevisao > db".

```
setContentView(R.layout.activity_resposta)

val edtNome: EditText = findViewById(R.id.edtNome)
val edtEndereco: EditText = findViewById(R.id.edtEndereco)
val edtBairro: EditText = findViewById(R.id.edtBairro)
val edtCep: EditText = findViewById(R.id.edtCep)
val btnConfirmar: Button = findViewById(R.id.btnConfirmar)

edtNome.setText(intent.getStringExtra("nome"))
edtEndereco.setText(intent.getStringExtra("endereco"))
edtBairro.setText(intent.getStringExtra("bairro"))
edtCep.setText(intent.getStringExtra("cep"))

// SQLite
val db = DBHelper(context, null)

btnConfirmar.setOnClickListener { it: View!
    db.addPessoa(edtNome.text.toString(), edtEndereco.text.toString(), edtBairro.text.toString(),
        edtCep.text.toString())
    finish()
}
```

**xml**

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right.

**Project Structure:**

- Root: appRevisao
- Version control: Pixel\_3a\_API\_34\_extension\_l...
- Module: app
- File: DBHelper.kt
- File: strings.xml (selected)
- File: colors.xml
- File: backup\_rules.xml
- File: data\_extraction\_rules.xml
- Folder: res (generated)

**Code Editor (strings.xml):**

```
<resources>
    <string name="app_name">appRevisao</string>
    <string name="txtNome">Nome:</string>
    <string name="txtEndereco">Endereço:</string>
    <string name="txtBairro">Bairro:</string>
    <string name="txtCep">Cep:</string>
    <string name="btnCadastrar">Cadastrar</string>
    <string name="txtResposta">Confirmação dos Dados</string>
    <string name="btnConfirmar">Confirmar</string>
</resources>
```

**Bottom Bar:**

- Upgrade Assistant: Upgrade project from AGP 8.4.0
- Upgrade Android Gradle Plugin from version 8.4.0 to 8.4.1
- Run selected steps
- Show Usages
- Refresh

**Status Bar:**

- 1:1 LF UTF-8
- 4 spaces

AR appRevisao Version control Pixel\_3a\_API\_34\_extension\_l... app : To A S F Q G X

Android

- com.example.apprevisao
  - BuildConfig
  - res
    - drawable
    - layout
      - activity\_main.xml
      - activity\_resposta.xml
    - mipmap
    - values
      - colors.xml
      - strings.xml
    - themes (2)
  - xml
    - backup\_rules.xml
    - data\_extraction\_rules.xml
  - res (generated)
- Gradle Scripts
  - build.gradle (Project: appRevisao)

Bookmarks

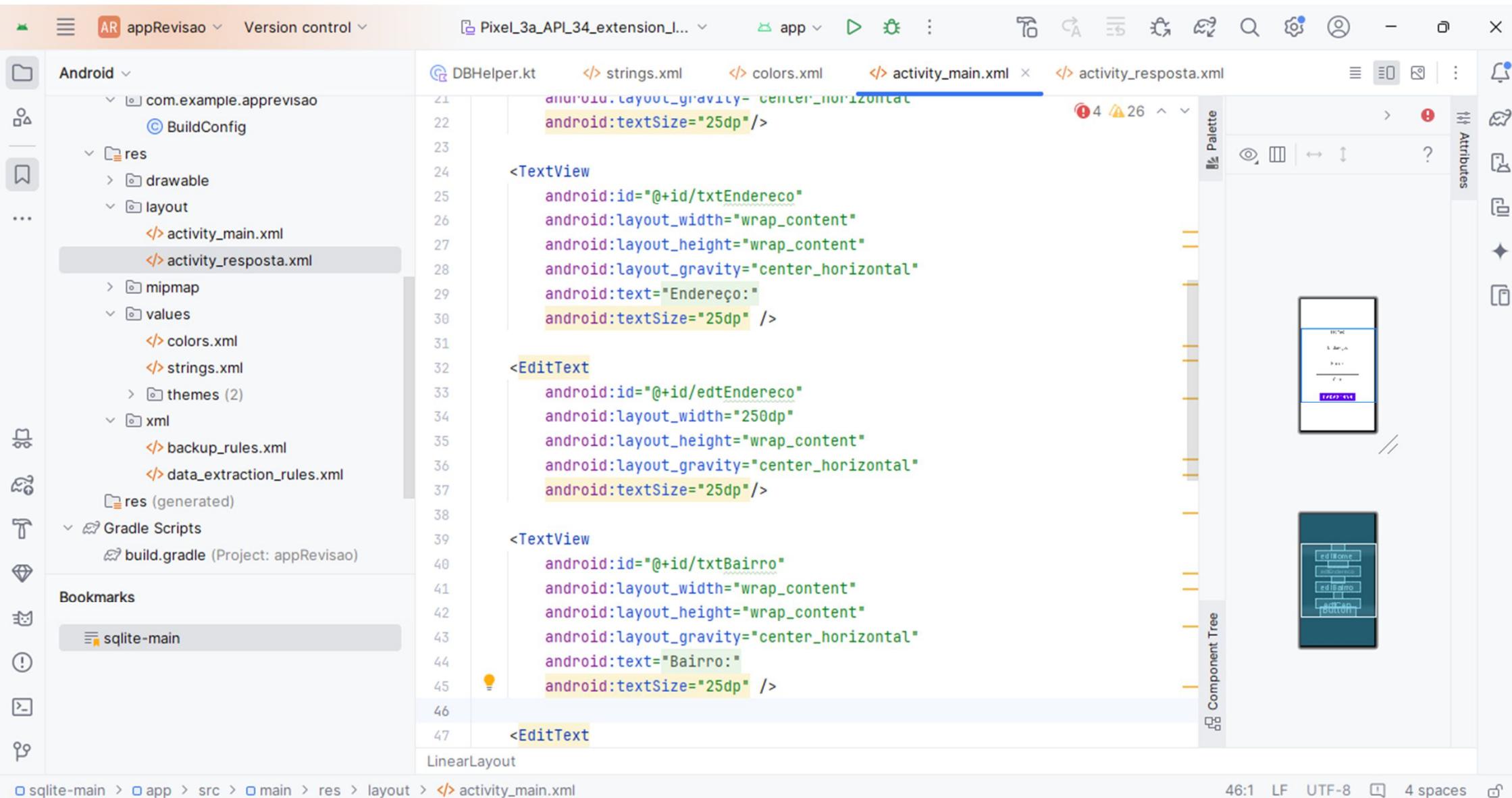
- sqlite-main

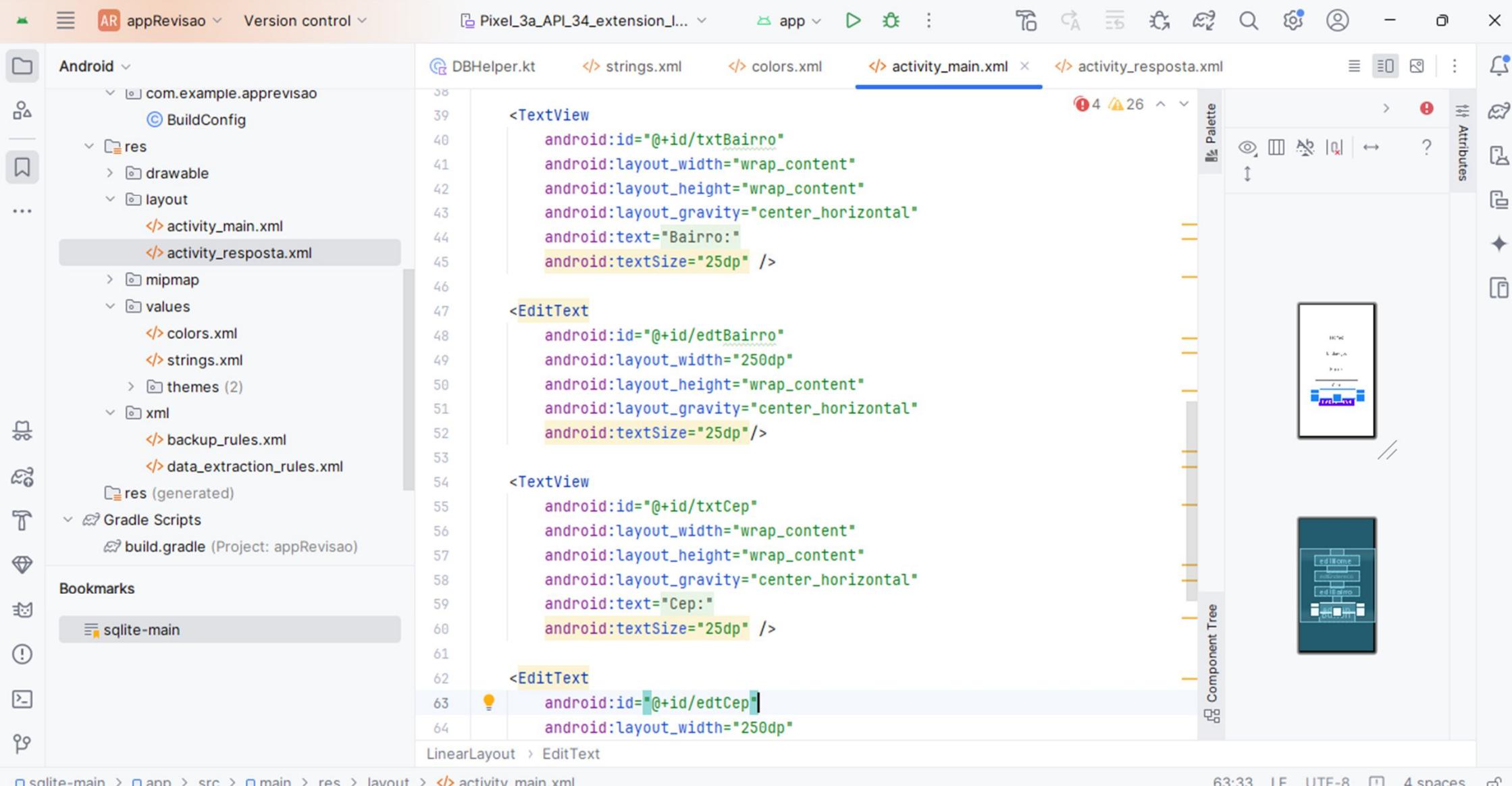
DBHelper.kt strings.xml colors.xml activity\_main.xml activity\_resposta.xml

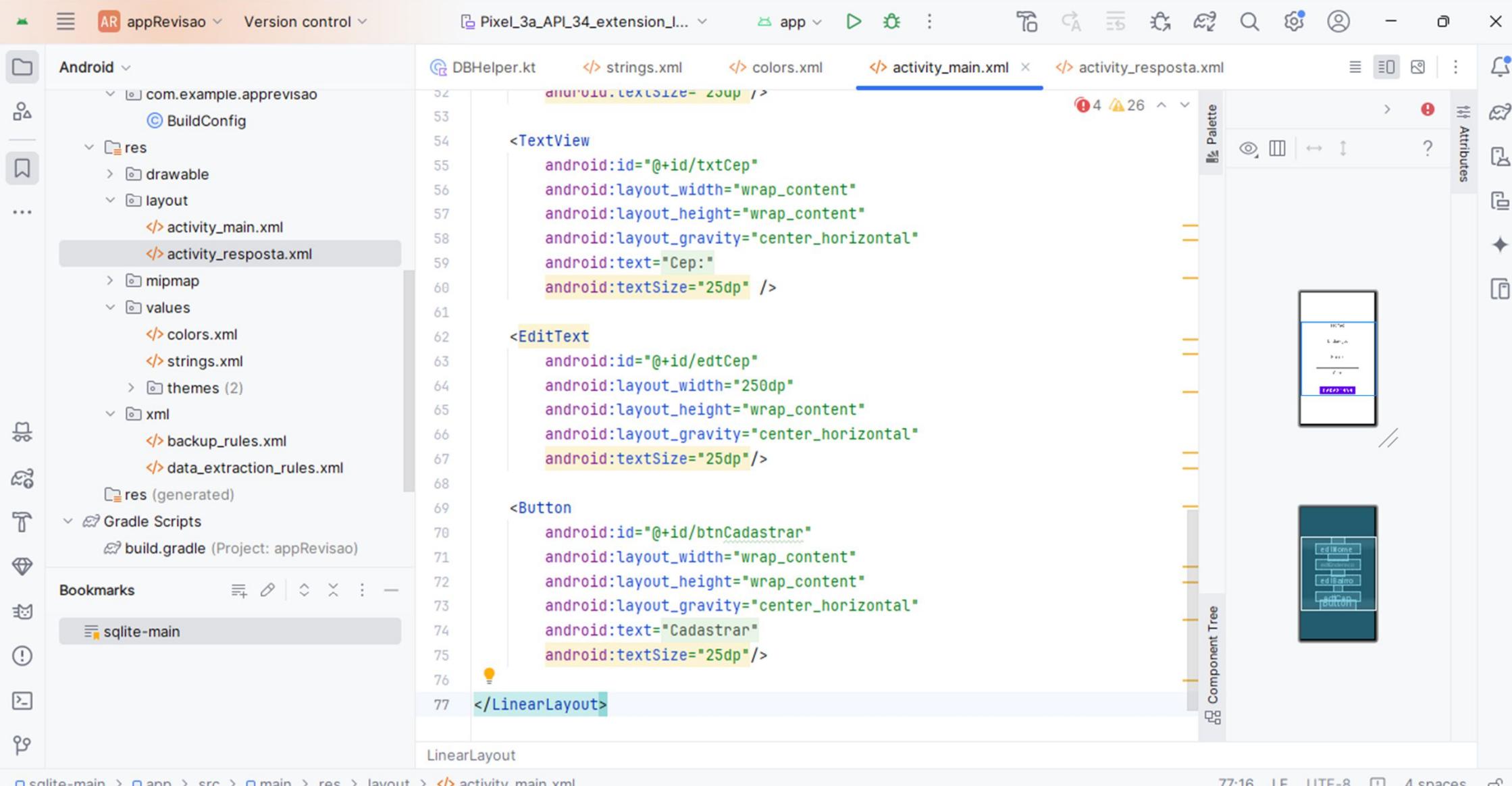
1 <?xml version="1.0" encoding="utf-8"?>  
2 <LinearLayout  
3 xmlns:android="http://schemas.android.com/apk/res/android"  
4 android:layout\_width="match\_parent"  
5 android:layout\_height="wrap\_content"  
6 android:layout\_gravity="center\_vertical"  
7 android:orientation="vertical">  
8  
9 <TextView  
10 android:id="@+id/txtNome"  
11 android:layout\_width="wrap\_content"  
12 android:layout\_height="wrap\_content"  
13 android:layout\_gravity="center\_horizontal"  
14 android:text="Nome:"  
15 android:textSize="25dp" />  
16  
17 <EditText  
18 android:id="@+id/editNome"  
19 android:layout\_width="250dp"  
20 android:layout\_height="wrap\_content"  
21 android:layout\_gravity="center\_horizontal"  
22 android:textSize="25dp" />  
23  
24 <TextView  
25 android:id="@+id/txtEndereco"  
26 android:layout\_width="wrap\_content"  
27 android:layout\_height="wrap\_content"

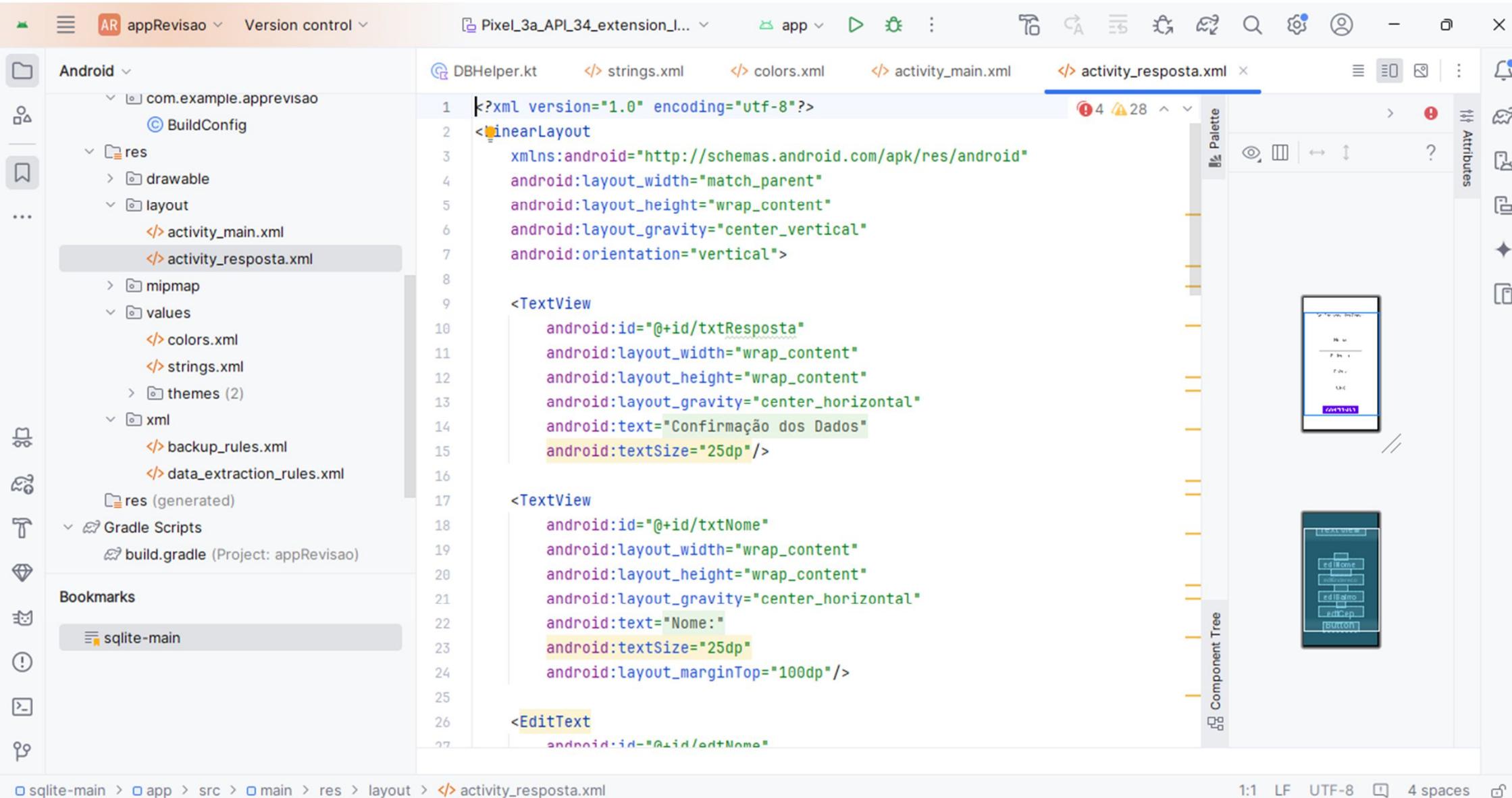
Component Tree

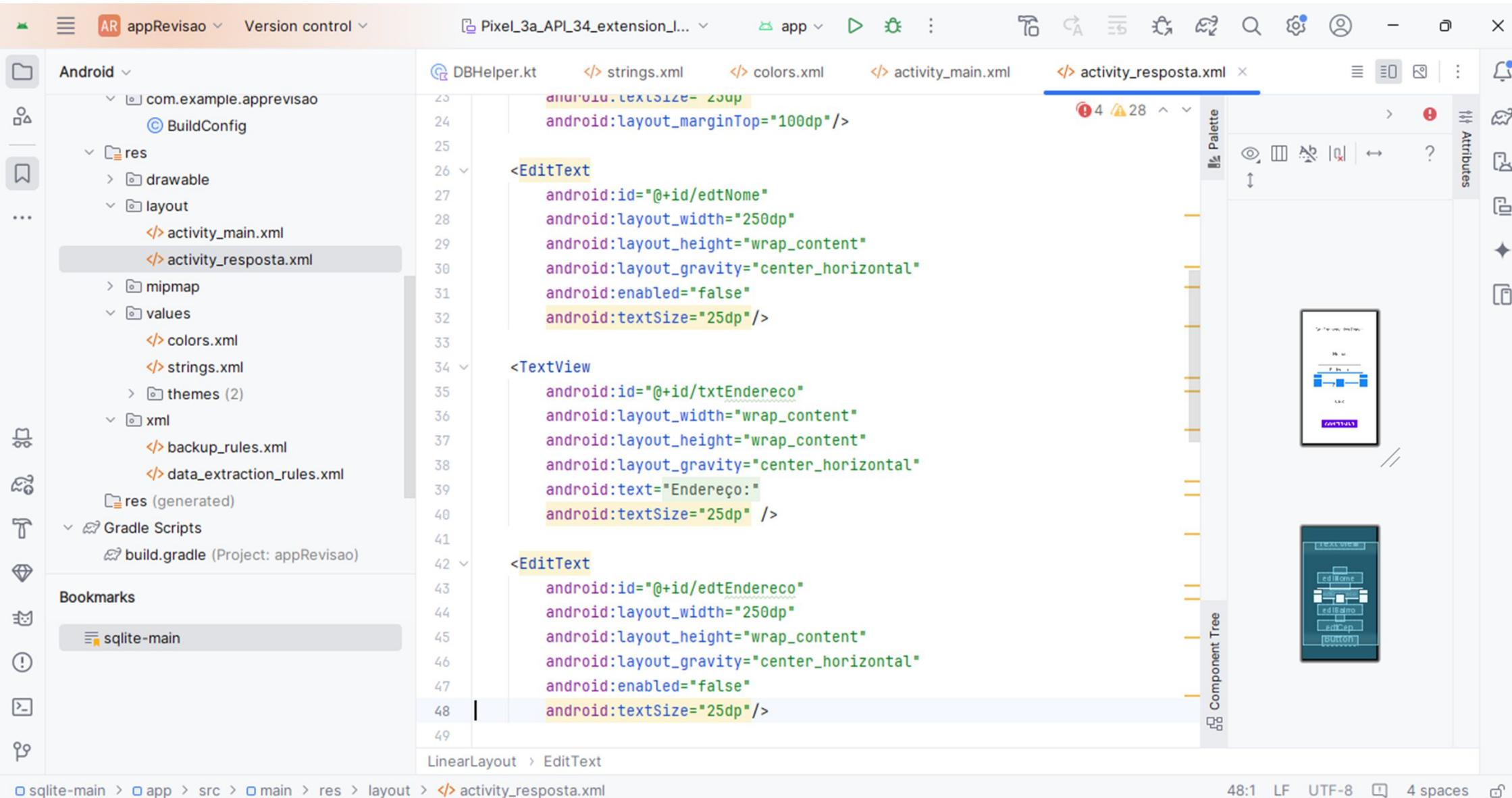
1:1 LF UTF-8 4 spaces

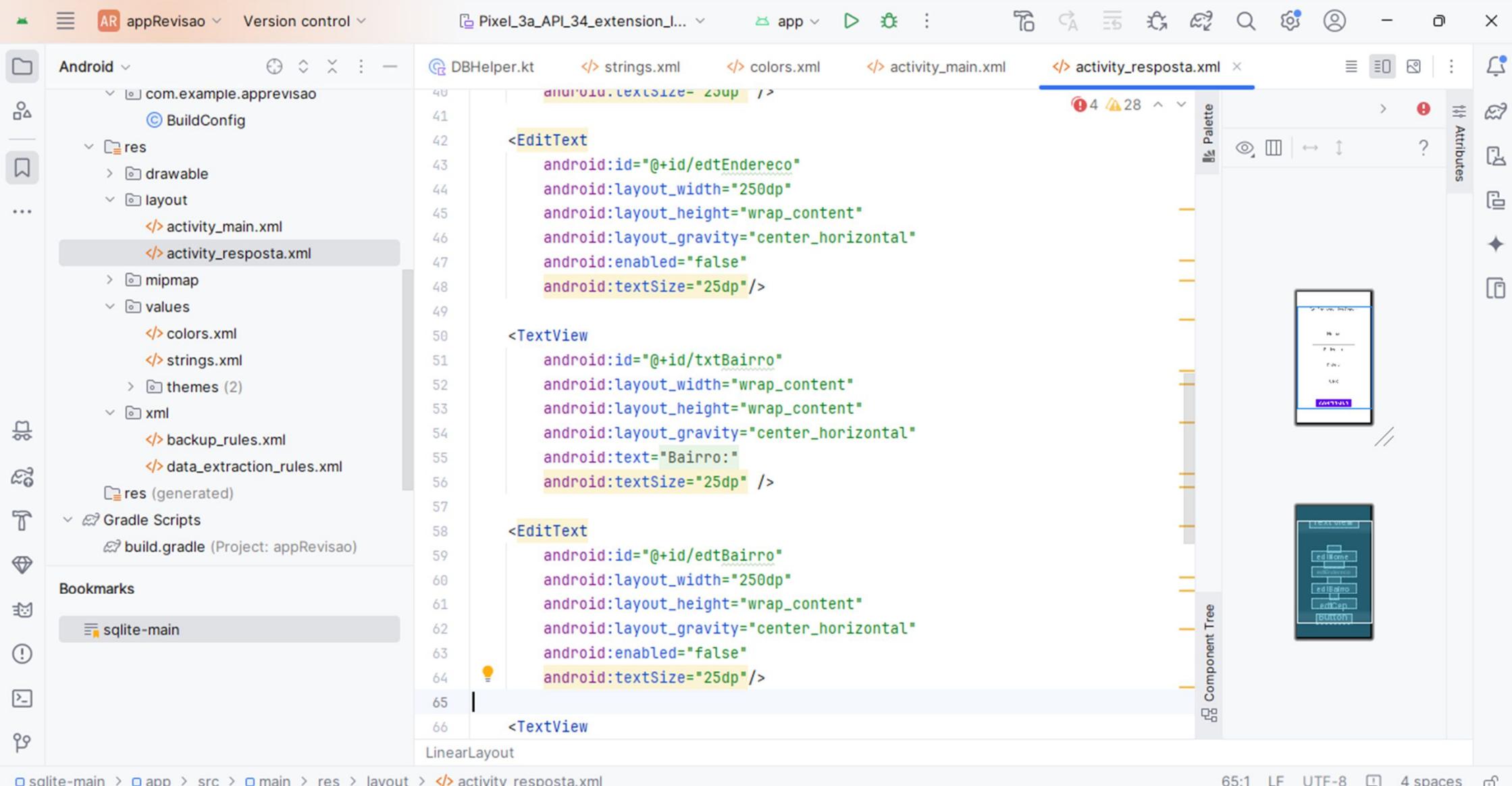


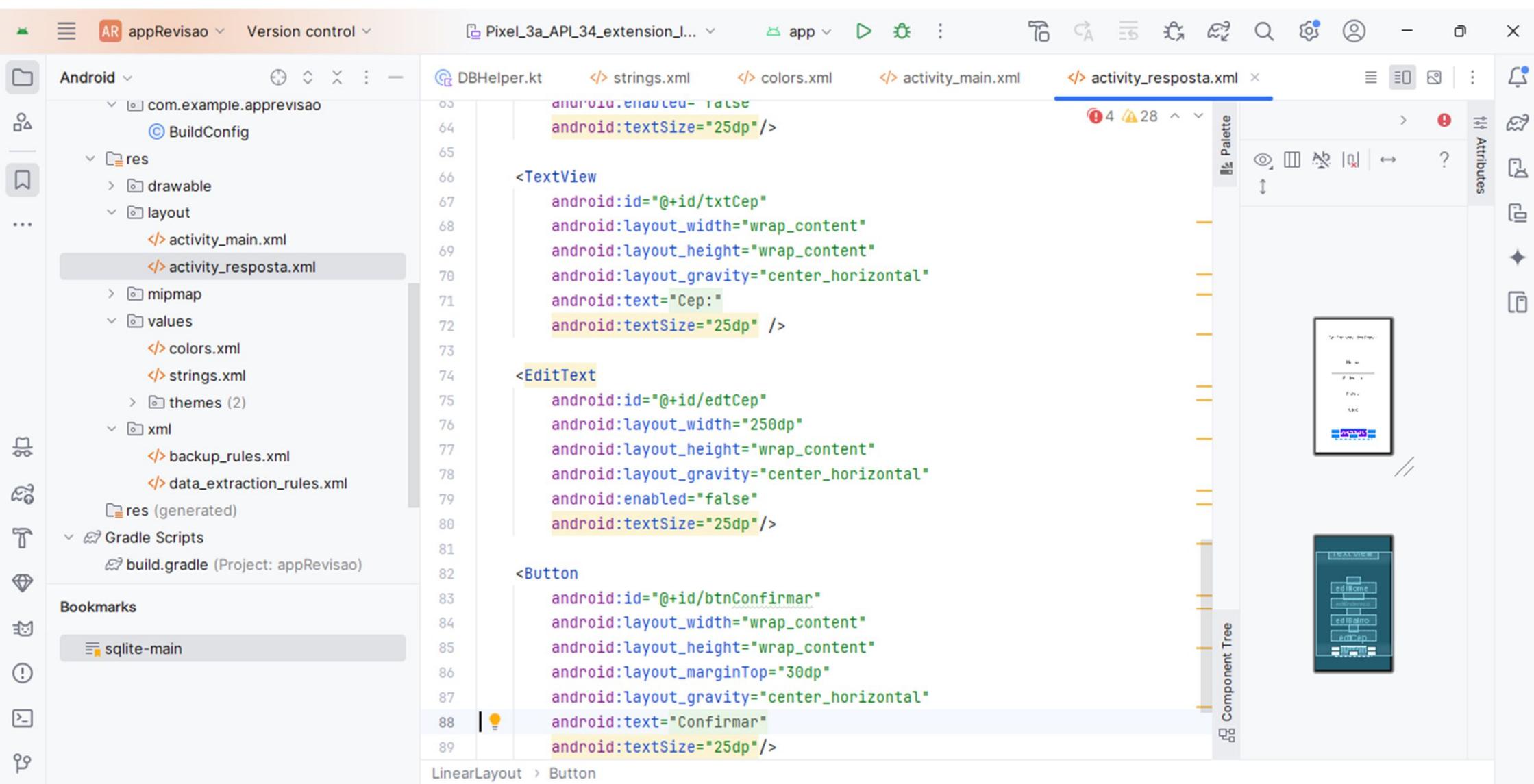












salite-main > app > src > main > res > layout > activity\_resposta.xml

88:1 LF UTF-8 4 spaces ↵

