

# Desenvolvimento Para Dispositivos móveis



Prof. Me. Clênio Silva  
E-mail: [clenio.silva@uniube.br](mailto:clenio.silva@uniube.br)

# Conexão com banco de dados SQLite

Para realizar a persistência de dados em banco de dados SQLite no AndroidStudio, você precisará seguir alguns passos básicos:

- Configurar o Banco de Dados SQLite:
  - No Android Studio, crie uma classe que estenda SQLiteOpenHelper. Esta classe será responsável por criar e atualizar o banco de dados, além de fornecer métodos para manipular os dados.
  - No método onCreate() dessa classe, você criará as tabelas necessárias para armazenar as informações do usuário, como nome de usuário e senha.

# Conexão com banco de dadosSQLite

- Definir uma Classe de Contrato de Banco de Dados:
  - Crie uma classe que defina constantes para o nome das tabelas, colunas e outras informações relacionadas ao banco de dados.
- Implementar Métodos para Manipular os Dados:
  - Na classe que estende SQLiteOpenHelper, implemente métodos para inserir, atualizar, excluir e recuperar informações do banco de dados.
- Manipular Usuários:
  - Crie métodos para adicionar usuários ao banco de dados, verificar credenciais de login, etc.
  -

# Conexão com banco de dadosSQLite

- Gerenciar a Conexão com o Banco de Dados:
  - Lembre-se de abrir e fechar a conexão com o banco de dados conforme necessário para evitar vazamentos de recursos.
- Implementar Métodos para Manipular os Dados:
  - Na classe que estende SQLiteOpenHelper, implemente métodos para inserir, atualizar, excluir e recuperar informações do banco de dados.
- Manipular Usuários:
  - Crie métodos para adicionar usuários ao banco de dados, verificar credenciais de login, etc.

# Conexão com banco de dadosSQLite

Para o nosso exemplo vamos criar uma aplicação para validar o usuário senha de um usuário da aplicação. Para acessar a tela principal da aplicação dever-se estar logado no sistema. Você também terá a opção de cadastrar um novo usuário para acessar o sistema.

# Conexão com banco de dadosSQLite

- 1) Crie um projeto no AndroidStudio com o nome **“conexaoBancoSqlite”**
- 2) Na estrutura de pastas do projeto tem uma pasta com o nome “java” e dentro desta pasta tem um pacote com o nome **“com.example.conexaobancosqlite”**, vamos criar nossas classes e activitys nesse pacote.

# Conexão com banco de dadosSQLite

3). Crie uma classe com o nome **DBHelper**, essa classe será responsável por gerenciar o banco de dados:

```
package com.example.conexaobancosqlite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

9 usages
public class DBHelper extends SQLiteOpenHelper {

    1 usage
    private static final int DATABASE_VERSION = 1;
    1 usage
    private static final String DATABASE_NAME = "UsersDB";
    3 usages
    private static final String TABLE_USERS = "users";
    2 usages
    private static final String KEY_ID = "id";
    2 usages
    private static final String KEY_USERNAME = "username";
    2 usages
    private static final String KEY_PASSWORD = "password";
```

# Conexão com banco de dadosSQLite

3). Crie uma classe com o nome **DBHelper**, essa classe será responsável por gerenciar o banco de dados:

```
1 usage
public DBHelper(Context context) { super(context, DATABASE_NAME, factory: null, DATABASE_VERSION); }

@Override
public void onCreate(SQLiteDatabase db) {
    String CREATE_USERS_TABLE = "CREATE TABLE " + TABLE_USERS + "("
        + KEY_ID + " INTEGER PRIMARY KEY," + KEY_USERNAME + " TEXT,"
        + KEY_PASSWORD + " TEXT" + ")";
    db.execSQL(CREATE_USERS_TABLE);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);
    onCreate(db);
}

2 usages
public static String getKeyUsername() { return KEY_USERNAME; }

2 usages
public static String getKeyPassword() { return KEY_PASSWORD; }

2 usages
public static String getTableUsers() { return TABLE_USERS; }
```



# Conexão com banco de dadosSQLite

3). Crie uma classe com o nome **DBHelper**, essa classe será responsável por gerenciar o banco de dados:

```
1 usage
public static String getKeyId() { return KEY_ID; }
}
```

# Conexão com banco de dadosSQLite

4). Crie uma classe **User** para representar os dados do usuário:

```
package com.example.conexaobancosqlite;

public class User {
    2 usages
    private int id;
    3 usages
    private String username;
    3 usages
    private String password;

    public User() {}

    1 usage
    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

    // Getters and setters

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }
```

# Conexão com banco de dadosSQLite

4). Crie uma classe **User** para representar os dados do usuário:

```
package com.example.conexaobancosqlite;

public class User {
    2 usages
    private int id;
    3 usages
    private String username;
    3 usages
    private String password;

    public User() {}

    1 usage
    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

    // Getters and setters

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }
```

# Conexão com banco de dadosSQLite

4). Crie uma classe **User** para representar os dados do usuário:

```
1 usage
public String getUsername() { return username; }

public void setUsername(String username) { this.username = username; }

1 usage
public String getPassword() { return password; }

public void setPassword(String password) { this.password = password; }
}
```

# Conexão com banco de dadosSQLite

5). Crie uma classe Classe **UserRepository** para manipular operações do banco de dados:

```
package com.example.conexaobancosqlite;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

4 usages
public class UserRepository {
    3 usages
    private DBHelper dbHelper;

    2 usages
    public UserRepository(Context context) { dbHelper = new DBHelper(context); }

    1 usage
    public void addUser(User user) {
        SQLiteDatabase db = dbHelper.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(DBHelper.getKeyUsername(), user.getUsername());
        values.put(DBHelper.getKeyPassword(), user.getPassword());

        db.insert(DBHelper.getTableUsers(), nullColumnHack: null, values);
        db.close();
    }
}
```

# Conexão com banco de dadosSQLite

5). Crie uma classe Classe **UserRepository** para manipular operações do banco de dados:

```
1 usage
2
3 public boolean checkUser(String username, String password) {
4     SQLiteDatabase db = dbHelper.getReadableDatabase();
5     System.out.println("Buscando usuario: " + username + " senha: " + password);
6
7     Cursor cursor = db.query(DBHelper.getTableUsers(), new String[]{DBHelper.getKeyId()},
8                             selection: DBHelper.getKeyUsername() + "=? AND " + DBHelper.getKeyPassword() + "=?",
9                             new String[]{username, password},
10                             null, null, null, null, null);
11
12     int count = cursor.getCount();
13
14     System.out.println("COUNT: " + count);
15
16     cursor.close();
17     db.close();
18
19     return count > 0;
20 }
21 }
```

# Conexão com banco de dadosSQLite

6). No arquivo **activity\_main.xml** vamos definir o layout da nossa tela de login:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/logouber"
        android:layout_gravity="center"/>

    <EditText
        android:id="@+id/editTextUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:hint="Username" />
```

# Conexão com banco de dadosSQLite

6). No arquivo **activity\_main.xml** vamos definir o layout da nossa tela de login:

```
<EditText
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:inputType="textPassword"
    android:hint="Password" />

<Button
    android:id="@+id/buttonLogin"
    android:layout_width="350dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:layout_gravity="center"
    android:text="Login"
    android:backgroundTint="#000" />
```



# Conexão com banco de dadosSQLite

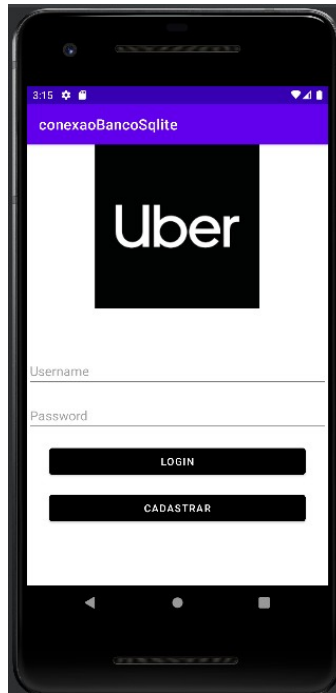
6). No arquivo **activity\_main.xml** vamos definir o layout da nossa tela de login:

```
<Button
    android:id="@+id/buttonCadastrar"
    android:layout_width="350dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Cadastrar"
    android:backgroundTint="#000"
    android:layout_gravity="center"
/>

</LinearLayout>
```

# Conexão com banco de dadosSQLite

6). No arquivo **activity\_main.xml** vamos definir o layout da nossa tela de login:



# Conexão com banco de dadosSQLite

7). No arquivo java da referente ao layout anterior, MainActivity, vamos adicionar o código para capturar as informações dos EditText e adicionar eventos nos botões de login (O qual ira usar o método checkUse da classe userRepository) para validar o se o usuário e senha estão no banco:

# Conexão com banco de dadosSQLite

7).

```
package com.example.conexaobancosqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

6 usages
public class MainActivity extends AppCompatActivity {
    2 usages
    private EditText editTextUsername, editTextPassword;
    2 usages
    private Button buttonLogin, buttonCadastrar;
    2 usages
    private UserRepository userRepository;
```

# Conexão com banco de dadosSQLite

7).

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

    editTextUsername = findViewById(R.id.editTextUsername);
    editTextPassword = findViewById(R.id.editTextPassword);
    buttonLogin = findViewById(R.id.buttonLogin);
    buttonCadastrar = findViewById(R.id.buttonCadastrar);

    userRepository = new UserRepository(context: this);

    buttonLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String username = editTextUsername.getText().toString().trim();
            String password = editTextPassword.getText().toString().trim();

            if (username.isEmpty() || password.isEmpty()) {
                Toast.makeText(context: MainActivity.this, text: "Entre com usuário e senha",
                    Toast.LENGTH_SHORT).show();
            } else {
                boolean isAuthenticated = userRepository.checkUser(username, password);
                if (isAuthenticated) {
                    Toast.makeText(context: MainActivity.this, text: "Sucesso ao realizar login",
```

# Conexão com banco de dadosSQLite

7).

```
        Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(getApplicationContext(), ActivityInicial.class);
        startActivity(intent);
    } else {
        Toast.makeText(context: MainActivity.this, text: "Usuário ou Senha inválido",
            Toast.LENGTH_SHORT).show();
    }
}

});

buttonCadastrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), CadastrarUsuario.class);
        startActivity(intent);
    }
});
}
```

# Conexão com banco de dadosSQLite

8).Vamos criar uma nova activity que será acionada quando clicarmos no botão cadastrar da activity anterior. Crie uma activity com o nome “**CadastrarUsuario**”. No arquivo “**activity\_cadastrar\_usuario.xml**” adicione o código a seguir:



# Conexão com banco de dadosSQLite

8).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".CadastroUsuario">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Cadastro de Usuário"
        android:layout_marginTop="64dp"
        android:textSize="32sp"/>

    <EditText
        android:id="@+id/usuario"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Usuário"
        android:inputType="text"
        android:layout_marginTop="64dp"/>
```



# Conexão com banco de dadosSQLite

8).

```
<EditText
    android:id="@+id/senha"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Senha"
    android:inputType="textPassword"
    android:layout_marginTop="64dp" />
```

```
<Button
    android:id="@+id/buttonSalvar"
    android:layout_width="350dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:layout_gravity="center"
    android:text="Salvar"
    android:backgroundTint="#000" />
```

```
</LinearLayout>
```

# Conexão com banco de dadosSQLite

9) Agora vamos alterar o arquivo Java “**CadastrarUsuario**”

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

6 usages
public class CadastrarUsuario extends AppCompatActivity {
    2 usages
    EditText editTextUsuario;
    2 usages
    EditText editTextSenha;
    2 usages
    Button buttonSalvar;
```

# Conexão com banco de dadosSQLite

## 9) Agora vamos alterar o arquivo Java “CadastrarUsuario”

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

    editTextUsuario = findViewById(R.id.usuario);
    editTextSenha = findViewById(R.id.senha);
    buttonSalvar = findViewById(R.id.buttonSalvar);

    UserRepository userRepository = new UserRepository(context, this);

    buttonSalvar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String username = editTextUsuario.getText().toString().trim();
            String password = editTextSenha.getText().toString().trim();

            if (username.isEmpty() || password.isEmpty()) {
                Toast.makeText(context, CadastrarUsuario.this,
                    text: "Entre com usuário e senha", Toast.LENGTH_SHORT).show();
            } else {
                try {
                    userRepository.addUser(new User(username, password));
                    Toast.makeText(context, CadastrarUsuario.this,
                        text: "Sucesso ao cadastrar usuário e senha", Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                    Toast.makeText(context, CadastrarUsuario.this,
                        text: "Erro ao cadastrar usuário e senha", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
}
```

# Conexão com banco de dadosSQLite

## 9) Agora vamos alterar o arquivo Java “CadastrarUsuario”

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

    editTextUsuario = findViewById(R.id.usuario);
    editTextSenha = findViewById(R.id.senha);
    buttonSalvar = findViewById(R.id.buttonSalvar);

    UserRepository userRepository = new UserRepository(context, this);

    buttonSalvar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String username = editTextUsuario.getText().toString().trim();
            String password = editTextSenha.getText().toString().trim();

            if (username.isEmpty() || password.isEmpty()) {
                Toast.makeText(context, CadastrarUsuario.this,
                    text: "Entre com usuário e senha", Toast.LENGTH_SHORT).show();
            } else {
                try {
                    userRepository.addUser(new User(username, password));
                    Toast.makeText(context, CadastrarUsuario.this,
                        text: "Sucesso ao cadastrar usuário e senha", Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                    Toast.makeText(context, CadastrarUsuario.this,
                        text: "Erro ao cadastrar usuário e senha", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
}
```

# Conexão com banco de dadosSQLite

9) Agora vamos alterar o arquivo Java “**CadastrarUsuario**”

```
Intent intent = new Intent(getApplicationContext(), MainActivity.class);
startActivity(intent);

} catch (Exception ex){
    Toast.makeText(context, CadastrarUsuario.this,
        text: "Erro ao cadastrar usuário e senha", Toast.LENGTH_SHORT).show();
}
```

# Conexão com banco de dadosSQLite

- 9) Por fim vamos criar uma activity com o nome **ActivityInicial** e alterar o arquivo xml (activity\_inicial.xml) com o seguinte código:

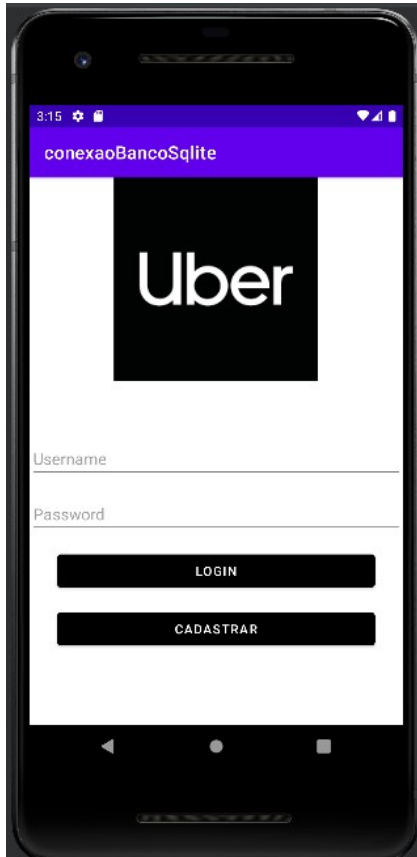
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ActivityInicial">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="128dp"
        android:gravity="center"
        android:text="@string/em_desenvolvimento"
        android:textSize="32sp"/>

</LinearLayout>
```



# Conexão com banco de dadosSQLite



Se o usuário e senha informados existirem no banco o login é realizado com sucesso e o acesso a tela inicial da aplicação é aberta

Caso o login falhe é porque o usuário e/ou senha não estão cadastrados no banco. Você pode fazer o cadastro clicando em cadastrar. Assim será redirecionado para a activity de cadastro e logo após finalizar o cadastro será redirecionado para efetuar o login.



# Conexão com banco de dadosSQLite

Segue link para o vídeo com a execução do aplicativo:

<https://youtu.be/WWet1uVWNN0>