Banco de Dados

Professor: Hercules Santhus

2023.1

Salvando Dados no Celular

Crie um Layout para adicionar os dados no App

TextView

TextInputLayout

Button

Criar um evento de clique

```
public class MainActivity extends AppCompatActivity {

    private Button salvar;
    private TextInputEditText nomeinputtext;
    private TextView resultado;

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

        salvar = findViewById(R.id.buttonSalvar);
        nomeinputtext = findViewById(R.id.textInputNome);
        resultado = findViewById(R.id.textViewResultado);

        salvar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
    }
}
```

SharedPreferences

A Classe SharedPreferences, cria um arquivo xml e dentro desse arquivo é guardado os dados que queremos.
Nesse arquivo é guardado pequenos dados como:

- Configurações do App
- Nomes de usuários
- Fases de Jogos
- Etc...

```
SharedPreferences dados = getSharedPreferences(ARQUIVO_DE_DADOS,  mode:  0 );
```

Modo de gravação: privado

Nome do arquivo

SALVAR DADOS

```
//sALAVAR DADOS NO CELULAR
salvar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        SharedPreferences dados = getSharedPreferences(ARQUIVO_DE_DADOS, 0);

        SharedPreferences.Editor editor = dados.edit();

        //VALIDAR NOME
        if(nomeinputtext.getText().toString().equals("")){
            Toast.makeText(MainActivity.this, "Preencha o campo", Toast.LENGTH_SHORT).show();
        }else {
            String dado = nomeinputtext.getText().toString();
            editor.putString("nome", dado);

            editor.commit(); //saalva os dados
            resultado.setText("Olá " + dado);
        }
    }
});
```

O Objeto Editor, que permite editar o nosso arquivo de preferencias

```
editor.putString("nome", dado);
```

Dados que serão guardados

Chave

RECUPERAR DADOS

```
//RECUPERAR NOME (EXIBIR)
SharedPreferences dados = getSharedPreferences(ARQUIVO_DE_DADOS, 0);

if (dados.contains("nome")){//verifica se contem a chave “nome”

    String texto = dados.getString(key: "nome", defValue:  "não definido"); //Passa a chave e um valo padrão caso o não encontre a chave “nome”
    resultado.setText("Olá " + texto);

}else {
    resultado.setText("Usuário não definido");
}
```

```
String texto = dados.getString(key: "nome", defValue: "não definido");
```

chave

Caso não consiga recuperar o nome

MainActivity.java

Declarar os Atributos

Instanciar

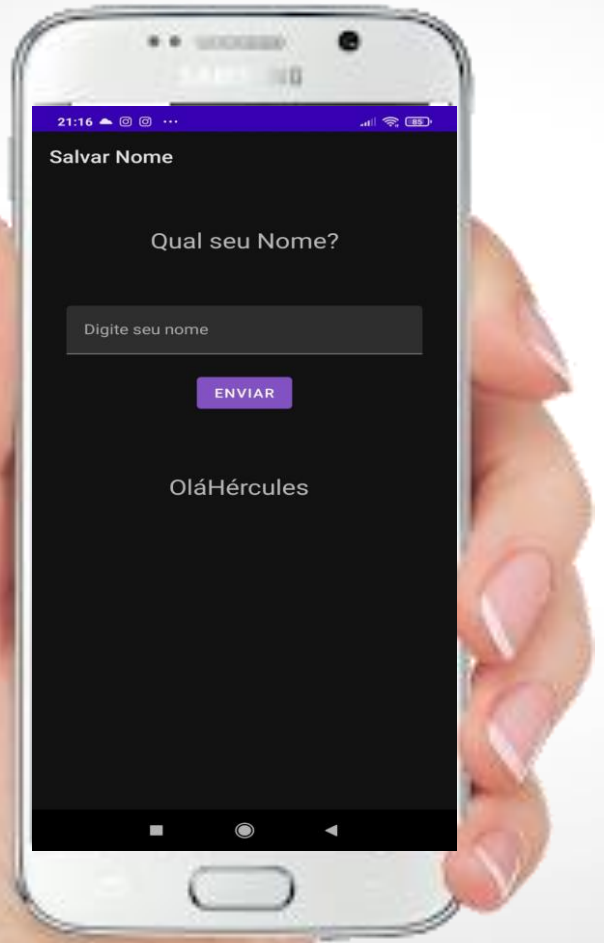
Evento de Clique para salvar dados no arquivo

Recuperar dados e exibir para o usuário

```
public class MainActivity extends AppCompatActivity {

    private Button salvar;
    private TextInputEditText nomeinputtext;
    //private TextInputLayout nome;
    private TextView resultado;
    private static final String ARQUIVO_DE_DADOS = "ArquivodeDados";

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

        salvar = findViewById(R.id.buttonSalvar);
        nomeinputtext = findViewById(R.id.editNome);
        resultado = findViewById(R.id.textViewResultado);


        //sALAVAR DADOS NO CELULAR
        salvar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                SharedPreferences dados = getSharedPreferences(ARQUIVO_DE_DADOS, 0);
                SharedPreferences.Editor editor = dados.edit();

                //VALIDAR NOME
                if(nomeinputtext.getText().toString().equals("")){
                    Toast.makeText(MainActivity.this, "Preencha o campo", Toast.LENGTH_SHORT).show();
                }else {
                    String dado = nomeinputtext.getText().toString();
                    editor.putString("nome", dado);
                    editor.commit(); //saalva os dados
                    resultado.setText("Olá " + dado);
                }
            }
        });

        //RECUPERAR NOME (EXIBIR)
        SharedPreferences dados = getSharedPreferences(ARQUIVO_DE_DADOS, 0);
        if (dados.contains("nome")){ //verifica s econtem essa chave ("nome")
            String nomes = dados.getString("nome", "não definido");
            resultado.setText("Olá " + nomes);
        }else {
            resultado.setText("Usuário não definido");
        }
    }
}
```

Você executar e salvar

Feche seu App e abra novamente para verificar se o texto ficou salvo



Banco de Dados

Tabela

Código do Cliente	Nome do Cliente	CPF	Endereço
1	Fulando de Tal	123.456.789-00	Avenida Brasil, 1000
2	Ciclano de Souza	234.567.890-11	Rua São Paulo, 500
3	Beltrano da Silva	345.678.901-22	Praça Sulamerica, 100

Registro

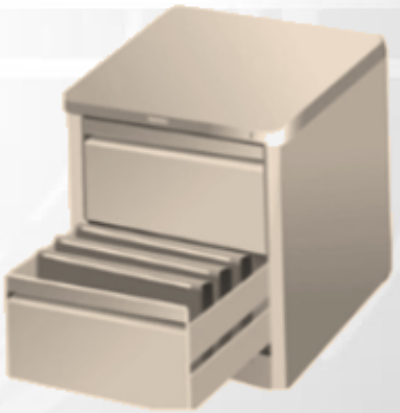
Chave Primária

Campo

papel

pasta

arquivo

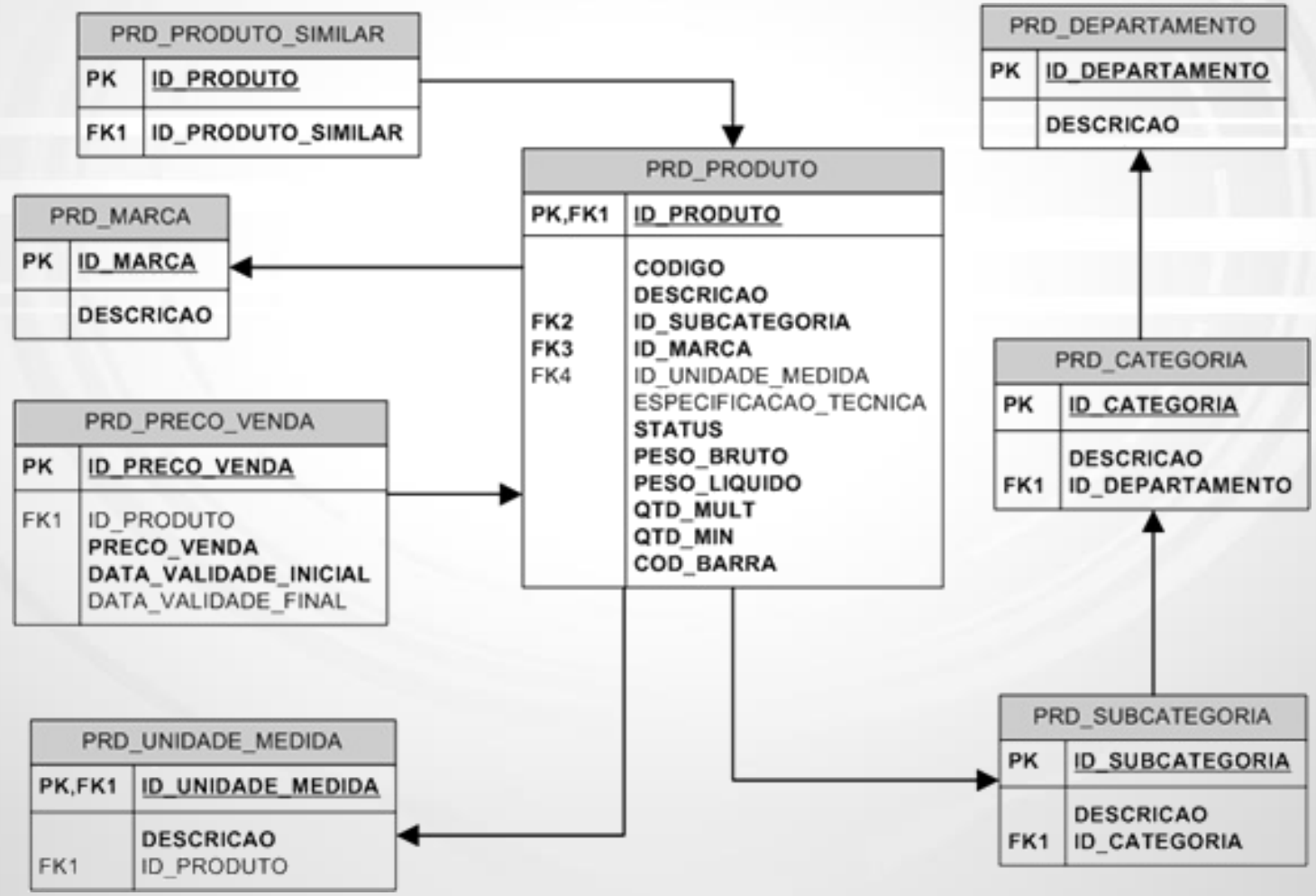


Registros

Tabelas

Banco de dados

Relacionamento de Tabelas



Linguagem SQL

Structured Query Language

CRIAR TABELA

```
CREATE TABLE usuário (

codigoUser INT
Nome VARCAR(50),
Email VARCHAR(50),
SENHA INT
 )
```

INSERIR REGISTRO:

```
INSERT INTO usuário (codigoUser, Nome, Email, Senha )

VALUES ( “001”, “Hercules”, “hercules@ufrr.br”, “12345”)
```

CodUser	Nome	Email	Senha
1	Hercules	hercules@ufrr.br	12345
2	João	joao@ufrr.br	4321
3	Maria	maria@ufrr.br	98765

SELECIONAR REGISTRO:

```
SELECT  *  FROM usuário

WHERE codigoUser = 3
```

CodUser	Nome	Email	Senha
3	Maria	maria@ufrr.br	98765

ATUALIZAR REGISTRO:

```
UPDATE  usuário
SET email = “novoemail@ufrr.br”
WHERE codigoUser = 3
```

CodUser	Nome	Email	Senha
3	Maria	**novoemial@ufrr.br**	98765

DELETAR REGISTRO:

```
DELETE * FROM usuário

WHERE codigoUser = 3
```

CodUser	Nome	Email	Senha
1	Hercules	hercules@ufrr.br	12345
2	João	joao@ufrr.br	4321

Comandos Básicos:

- Criar
- Inserir
- Selecionar
- Atualizar
- Deletar

SGBDs

SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Diferentemente de outros SGBDs tradicionais, o SQLite não funciona como um processo em execução separado que aceita conexões de clientes.

Em vez disso, o SQLite lê e grava diretamente em arquivos de banco de dados no sistema de arquivos do computador em que está sendo usado.

Essa característica torna o SQLite extremamente leve e fácil de usar, ideal para aplicativos de desktop, dispositivos móveis ou qualquer cenário em que um banco de dados simples e sem necessidade de configuração seja necessário.

Criando o Banco

```
public class MainActivity extends AppCompatActivity {

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

        try{

            //Cria o banco de Dados
            SQLiteDatabase bancoDados = openOrCreateDatabase("bancoapp", MODE_PRIVATE, null);

        }catch (Exception ex){
            ex.getStackTrace();
        }
    }
}
```

Para criar um Banco de Dados, vamos usar a classe SQLiteDatabase

É preciso está dentro do try catch

Private: apenas o meu App use esse banco de dados

Nome do banco

Método que abre um banco que já exista ou cria um novo

Criar Tabela

IF NOT EXISTS

A tabela será criada se ela não não existir no banco de dados

```
public class MainActivity extends AppCompatActivity {

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

        try{
            //Cria o banco de Dados
            SQLiteDatabase bancoDados = openOrCreateDatabase("bancoapp", MODE_PRIVATE, null);

            //Cria as Tabelas
            bancoDados.execSQL("CREATE TABLE IF NOT EXISTS usuario (" +
                                    "nome VARCHAR," +
                                    "idade INT(3))");

        }catch (Exception ex){
            ex.getStackTrace();
        }
    }
}
```

usuario	
nome	idade
Hercules	30
Maria	25

```
public class MainActivity extends AppCompatActivity {

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

        try{

            //Inserir dados na Tabelas
            bancoDados.execSQL("INSERT INTO usuario (nome, idade) VALUES ('Hercules', 30)");
            bancoDados.execSQL("INSERT INTO usuario (nome, idade) VALUES ('Maria', 25)");


        }catch (Exception ex){
            ex.getStackTrace();
        }
    }
}
```

Índice [0]

Índice [1]

usuario	
nome	idade
Hercules	30
Maria	25

Cursor percorre todos os registros da tabela

```
public class MainActivity extends AppCompatActivity {

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

        try{

            //Recuperar Dados
            Cursor cursor =  bancoDados.rawQuery("SELECT * FROM usuario", null);

            //indice das Colunas da Tabela
            int indiceNome = cursor.getColumnIndex("nome");
            int indiceIdade = cursor.getColumnIndex("idade");

            cursor.moveToFirst(); //começa do primeiro

            while (cursor !=null){
                Log.i("RESULTADO - nome",cursor.getString(indiceNome) );
                Log.i("RESULTADO - idade",cursor.getString(indiceIdade) );
                cursor.moveToNext();
            }


        }catch (Exception ex){
            ex.getStackTrace();
        }
    }
}
```

Perceba que aparece os nomes dos registros inseridos

```
D/ForceDarkHelper: updateByCheckExcludeList: pkg: com.example.bancodec
I/RESULTADO - nome: Hercules
I/RESULTADO - idade: 30
I/RESULTADO - nome: Maria
I/RESULTADO - idade: 25
W/Activity: Slow Operation: Activity com.example.bancodedados/.MainAct
W/Looper: Slow Looper main: Long Msg: seq=3 plan=23:05:47.601  late=5
W/Looper: Slow Looper main: Activity com.example.bancodedados/.MainAct
```

Version Control | Run | TODO | Problems | Terminal | App Inspection | Logcat

Launch succeeded (26 minutes ago)

```
try{
    //Cria o banco de Dados
    SQLiteDatabase bancoDados = openOrCreateDatabase("bancoapp", MODE_PRIVATE, null);

    //Cria as Tabelas
    bancoDados.execSQL("CREATE TABLE IF NOT EXISTS usuario (" +
            "nome VARCHAR," +
            "idade INT(3))");

    //Inserir Tabelas
    bancoDados.execSQL("INSERT INTO usuario (nome, idade) VALUES ('Hercules', 30)");
    bancoDados.execSQL("INSERT INTO usuario (nome, idade) VALUES ('Maria', 25)");

    //Recuperar Dados
    Cursor cursor =  bancoDados.rawQuery("SELECT * FROM usuario", null);

    //indice das Colunas da Tabela
    int indiceNome = cursor.getColumnIndex("nome");
    int indiceIdade = cursor.getColumnIndex("idade");

    cursor.moveToFirst(); //começa do primeiro

    while (cursor !=null){
        Log.i("RESULTADO - nome",cursor.getString(indiceNome) );
        Log.i("RESULTADO - idade",cursor.getString(indiceIdade) );
        cursor.moveToNext();
    }


}catch (Exception ex){
    ex.getStackTrace();
}
```

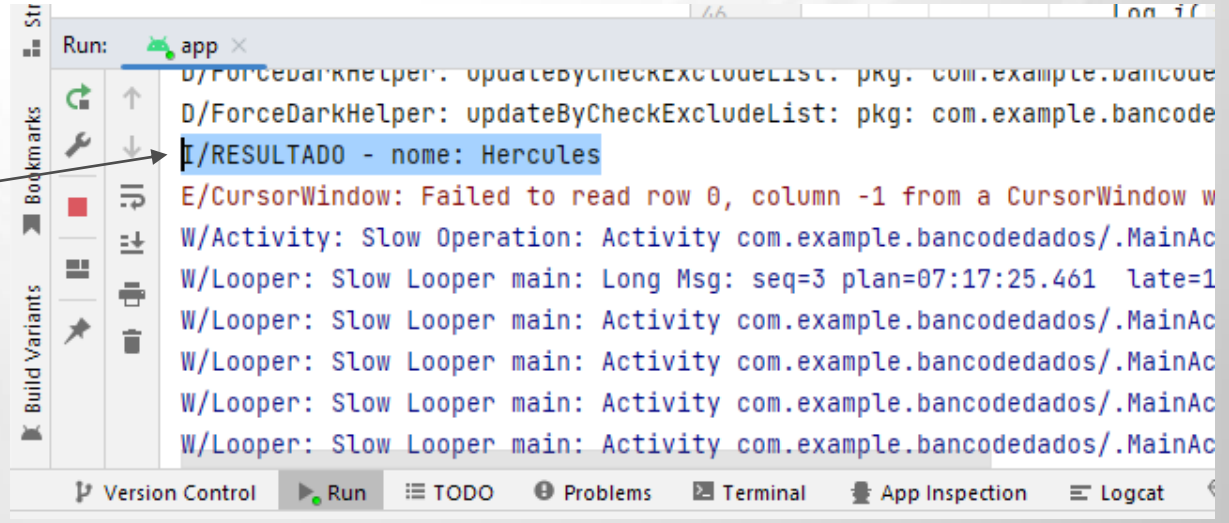
Filtros

```
Cursor cursor =  bancoDados.rawQuery (sql:"SELECT * FROM usuario", selectionArgs: null);
```

```
String consulta = “SELECT * FROM usuario”
```

```
Cursor cursor =  bancoDados.rawQuery (consulta, selectionArgs: null);
```

```
String consulta =
        "SELECT nome " +
        "FROM usuario " +
        "WHERE idade = 30";
```



Chave Primaria

```
bancoDados.execSQL("CREATE TABLE IF NOT EXISTS usuario (" +
        "id INT PRIMARY KEY AUTOINCREMENT  ,"+
        "nome VARCHAR," +
        "idade INT(3))");
```

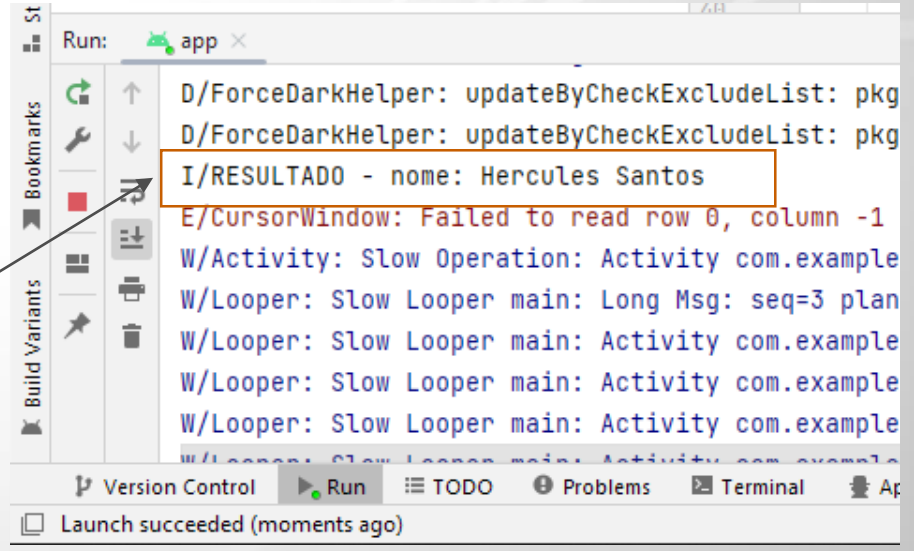
Uma chave é o código de um registro, ele é único e não se repete.

Esse atributo é incrementado automaticamente pelo banco de dados, sem a necessidade do usuário preencher esse campo.

UPDATE

```
String atualiza =
        "UPDATE usuario " +
        "SET nome = 'Hercules Santos‘ " +
        "WHERE idade = 30";
```

```
Cursor cursor =  bancoDados.rawQuery (atualiza, selectionArgs: null);
```




```
String consulta =
        "SELECT nome " +
        "FROM usuario " +
        "WHERE idade = 30";
```

```
Cursor cursor =  bancoDados.rawQuery (consulta, selectionArgs: null);
```