



*SQLite*





---

# *SQLite*

---





- Quando precisamos persistir e consultar grandes quantidades de dados, o ideal é utilizar um **banco de dados** ao invés de um arquivo local ou *SharedPreferences*.
- Os aplicativos Flutter podem usar os bancos de dados **SQLite**.



- **SQLite** é uma biblioteca em linguagem C que implementa um banco de dados **SQL embutido**.
- Programas que usam a biblioteca **SQLite** podem ter acesso a banco de dados SQL sem executar um processo SGBD separado.

<https://www.sqlite.org/about.html>



1. Adicionar Dependências
2. Definir o modelo
3. Abrir o Banco de Dados
4. Manipular os dados no BD

# 1 - Adicionar Dependências



```
dependencies:  
  flutter:  
    sdk: flutter  
  sqflite: ^1.3.2+1  
  path: ^1.7.0
```

<https://pub.dev/packages/sqflite>

<https://pub.dev/packages/path>

## 2 - Definir o modelo de Dados



```
class Dog {  
    final int id;  
    final String name;  
    final int age;  
  
    Dog({this.id, this.name, this.age});  
  
}
```

# 3 - Abrir o Banco de Dados



```
_recuperarBancoDados() async {  
    final caminhoBancoDados = await getDatabasesPath();  
    final localBancoDados = join(caminhoBancoDados, "banco.db");  
  
    var bd = await openDatabase(localBancoDados, version: 1,  
        onCreate: (db, dbVersaoRecente) {  
        String sql =  
            "CREATE TABLE usuarios (id INTEGER PRIMARY KEY AUTOINCREMENT, nome  
VARCHAR, idade INTEGER) ";  
        db.execute(sql);  
        });  
  
    return bd;  
    //print("aberto: " + bd.isOpen.toString() );  
}
```



# 4 - Manipular os Dados



```
_salvar() async {  
    Database bd = await _recuperarBancoDados();  
  
    Map<String, dynamic> dadosUsuario = {  
        "nome": "zurich mol",  
        "idade": 42  
    };  
    int id = await bd.insert("usuarios", dadosUsuario);  
    print("Salvo: $id ");  
}
```

# 4 - Manipular os Dados



```
_atualizarUsuario(int id) async {  
    Database bd = await _recuperarBancoDados();  
  
    Map<String, dynamic> dadosUsuario = {  
        "nome": "Mariana Almeida",  
        "idade": 18  
    };  
    int retorno = await bd  
        .update("usuarios", dadosUsuario, where: "id = ?", whereArgs:  
[id]);  
  
    print("item qtde atualizada: $retorno");  
}
```

# 4 - Manipular os Dados



```
_excluirUsuario(int id) async {  
  Database bd = await _recuperarBancoDados();  
  
  int retorno = await bd.delete("usuarios",  
    /*where: "id = ?",  
    whereArgs: [id]*/  
    where: " nome = ? AND idade = ?",  
    whereArgs: ["Jamilton Damasceno", 40]);  
  
  print("item qtde removida: $retorno");  
}
```

# 4 - Manipular os Dados



```
_listarUsuarioPeloId(int id) async {  
    Database bd = await _recuperarBancoDados();  
  
    //CRUD -> Create, Read, Update and Delete  
    List usuarios = await bd.query("usuarios",  
        columns: ["id", "nome", "idade"], where: "id = ?", whereArgs:  
[id]);  
  
    for (var usuario in usuarios) {  
        print("item id: " +  
            usuario['id'].toString() +  
            " nome: " +  
            usuario['nome'] +  
            " idade: " +  
            usuario['idade'].toString());  
    }  
}
```

# Referências Bibliográficas

- **Flutter:** <https://flutter.dev/>
- **Persist data with SQLite:**  
<https://flutter.dev/docs/cookbook/persistence/sqlite>