Buscando registros

Para realizarmos uma busca de registros em uma tabela de nosso banco de dados utilizaremos a função *query*, essa função recebe os seguintes parâmetros:

- ***table* (Obrigatório):** Nome da tabela na qual o registro deve ser inserido;
- ***columns*:** Lista de *Strings*. Define quais colunas desejamos retornar na busca.
- ***distinct:*** Objeto *bool*. Define se possíveis duplicatas de registros devem ser removidas da busca.
- ***groupBy:*** *String*. Define por qual coluna desejamos agrupar o resultado da busca.
- ***having:*** *String*. Define uma condição de filtro quando houver o uso de funções agregadas.
- ***limit:*** *int*. Define o número máximo de registros que podem ser tornados na busca.
- ***offset:*** *int*. Define o ponto inicial da busca, ou seja, quantos registros devem ser ignorados antes de selecionar os registros a serem retornados pela busca.
- ***orderBy:*** *String*. Define por qual coluna desejamos classificar o resultado da busca.
- ***where:*** *String*. Define a cláusula de filtro da busca. Normalmente utilizada em conjunto com o parâmetro *whereArgs*.
- ***whereArgs:*** Lista de *Strings*. Define os argumentos que serão utilizados para realizar o filtro na busca.

Para nosso exemplo implementamos o método **buscar** e o método **listar**. O primeiro método irá realizar uma busca com filtro, já o segundo trará todos os registros presentes na base de dados. Vejamos como ficou nosso código final.

```dart
@override
  Future<Usuario> buscar(int id) async {
    Database banco = await dbLocal.getConexao();
    var dados = await banco.query(
      dbLocal.tabela,
      where: "id=?",
      whereArgs: [id],
    );
    return Usuario.fromMap(dados.first);
  }

  @override
  Future<List<Usuario>> listar() async {
    Database banco = await dbLocal.getConexao();
    var dados = await banco.query(
      dbLocal.tabela,
    );
    return dados.map((mapa) => Usuario.fromMap(mapa)).toList();
  }
```

Modificando registros

Para modificarmos um registro em uma tabela de nosso banco de dados utilizaremos a função update, essa função recebe quatro parâmetros:

- ***String:*** Nome da tabela na qual o registro deve ser inserido;
- ***Map<String, dynamic>:*** Objeto Map contendo os dados a serem armazenados.
- ***where:*** String. Define a cláusula de filtro de registros aos quais desejamos atualizar. Normalmente utilizada em conjunto com o parâmetro whereArgs.
- ***whereArgs:*** Lista de Strings. Define os argumentos que serão utilizados para realizar o filtro dos registros.

Para nosso exemplo implementamos o método **atualizar**, o qual recebe como parâmetro o objeto **Usuario**, a **condição** e o **valor da condição** de atualização. Esses dados serão repassados para a função update. Vejamos como ficou nosso código final.

```dart
@override
  Future<int?> atualizar({
    required Usuario entidade,
    required String condicao,
    required List valoresCondicao,
  }) async {
    Database banco = await dbLocal.getConexao();
    return await banco.update(
      dbLocal.tabela,
      entidade.toMap(),
      where: condicao,
      whereArgs: valoresCondicao,
    );
  }
```

Removendo registros

Para removermos um registro em uma tabela de nosso banco de dados utilizaremos a função delete, essa função recebe três parâmetros:

- ***String:*** Nome da tabela na qual o registro deve ser inserido;
- ***where:*** String. Define a cláusula de filtro de registros aos quais desejamos remover. Normalmente utilizada em conjunto com o parâmetro whereArgs.

- ***whereArgs:*** Lista de Strings. Define os argumentos que serão utilizados para realizar o filtro dos registros.

Para nosso exemplo implementamos o método **remover**, o qual recebe como parâmetro a **condição** e os **valores** para a remoção. Esses dados serão repassados para a função *delete*. Vejamos como ficou nosso código final.

```dart
@override
  Future<int?> remover({
    required String condicao,
    required List valoresCondicao,
  }) async {
    Database banco = await dbLocal.getConexao();
    return await banco.delete(
      dbLocal.tabela,
      where: condicao,
      whereArgs: valoresCondicao,
    );
  }
```