

**Docentes:**

Sónia Luz, sonia.luz@ipleiria.pt  
David Safadinho, david.safadinho@ipleiria.pt  
Cátia Ledesma, catia.ledesma@ipleiria.pt

# Ficha Prática 8

## Books – Base de Dados SQLite

### Objetivos da Ficha

- Criar e manipular Base de Dados Local SQLite

### Introdução

Nesta ficha vamos continuar com a aplicação anterior. Caso queiram manter o estado de evolução da aplicação, devem fazer uma cópia e trabalhar sobre a cópia.

Nesta ficha, pretende-se salientar o uso de base de dados locais **SQLite**, para permitir a persistência dos dados da aplicação localmente.

### Abrir a pasta do projeto da aula anterior

1. Neste projeto deve passar a considerar uma base de dados local SQLite para persistir os dados localmente.
  - 1.1. Para isso, no package **modelo**, deve criar uma nova classe **LivroBDHelper**;
    - 1.1.1. Esta classe tem de estender de **SQLiteOpenHelper**;
      - 1.1.1.1. O que obriga a que sejam implementados os métodos **onCreate** e **onUpgrade**;
      - 1.1.1.2. Para poder conseguir criar e atualizar a base de dados é necessário criar as constantes **DB\_NAME** e **TABLE\_NAME** do tipo **String** e o **DB\_VERSION** do tipo **Inteiro**;
      - 1.1.1.3. Para permitir a manipulação dos componentes **SQLite** é necessária a criação de uma instância da classe **SQLiteDatabase**;

- 1.1.1.4. Após isso devem implementar o construtor, recebendo apenas como parâmetro o **Context**;
- 1.1.1.5. Para definir permissões de escrita na base de dados, teremos de utilizar o método **getWritableDatabase()**;
- 1.1.1.6. Visto que já temos a base de dados criada, podemos proceder à criação da tabela **Livro**, no método **onCreate()**;
- 1.1.1.7. Devem também implementar o método **onUpgrade()** para permitir a atualização da versão da base de dados;
- 1.1.1.8. De seguida vamos implementar os métodos para realizar as operações de CRUD: **adicionarLivroBD**, **editarLivroBD**, **removerLivroBD** e **getAllLivrosBD**;
  - 1.1.1.8.1. Será também necessário criar, na classe **Livro**, o método **setId(int)** para atualizar o id devolvido pela base de dados;
- 1.2. Na classe **SingletonGestorLivros**, para manipular a base de dados:
  - 1.2.1. Devem alterar o construtor e o método **getInstance()** para passarem a receber um parâmetro do tipo **Context**, necessário para instanciar a classe da base de dados;
    - 1.2.1.1. Neste caso será necessário corrigir todas as chamadas ao método **getInstance()** para que passe a receber o contexto adequado;
  - 1.2.2. Devem criar um atributo **livrosBD** do tipo **LivroBDHelper**;
  - 1.2.3. No construtor, devem iniciar a instância do **LivroBDHelper**, deixando de usar o método **gerarDadosDinamico()**;
  - 1.2.4. Após isso, devem alterar os métodos **adicionarLivro**, **removerLivro**, **editarLivro** e **getLivros**, renomeando-os para **adicionarLivroBD**, **removerLivroBD**, **editarLivroBD** e **getLivrosBD**, de modo a utilizar os métodos da base de dados;