

**Sistema de Biblioteca com SQLite**

**Objetivo final**

Desenvolver um sistema de Biblioteca em Python com persistência em banco de dados SQLite.

O projeto será construído por etapas, versionado no Git, e finalizado com uma interface no Streamlit.

**Preparação (antes de começar)**

1. Criar uma pasta chamada biblioteca.

2. Dentro da pasta, iniciar o repositório ( git init )

3. Criar um arquivo README.md com a descrição do projeto.

4. Criar um commit inicial:

git add .

git commit -m "Estrutura do projeto"

**Etapas do Desenvolvimento**

**Etapa 1 – Criação do Banco e Tabela**

• Criar o arquivo main.py.

• Conectar ao banco biblioteca.db.

• Criar a tabela livros com os campos:

o id → chave primária (INTEGER, autoincremento). o título → texto obrigatório.

o autor → texto obrigatório.

o ano → número inteiro.

o disponível → texto (valores "Sim" ou "Não"). Commit esperado: "Etapa 1 - Criação do banco e tabela"

**Etapa 2 – Função de Cadastro**

• Implementar no main.py uma função que permita inserir livros no banco.

• função cadastrar\_livro(titulo, autor, ano) que insere um livro na tabela.

• Todo livro novo deve ser cadastrado com

disponivel = "Sim".

Commit esperado: "Etapa 2 - Função de cadastro de livros"

**Etapa 3 – Listagem de Livros**

• Criar uma função listar\_livros() que mostre todos os livros cadastrados.

• Exibir colunas: ID, Título, Autor, Ano, Disponibilidade. Commit esperado: "Etapa 3 - Função Listagem de livros"

**Etapa 4 – Atualização de Disponibilidade**

• Criar função que altere o campo disponivel: o Se estava "Sim", vira "Não".

o Se estava "Não", vira "Sim".

Commit esperado: "Etapa 4 - Atualização de disponibilidade"

**Etapa 5 – Remoção de Livros**

• Criar função remover\_livro(id) que remova um livro pelo ID.

Commit esperado: "Etapa 5 - Função de remoção de livros implementada"

**Etapa 6 – Menu Interativo no Console**

• Implementar um menu de opções no main.py: 1. Cadastrar livro

2. Listar livros

3. Atualizar disponibilidade

4. Remover livro

5. Sair

Cada opção deve chamar a função correspondente. Commit esperado: "Etapa 6 - Menu interativo no console"

**Etapa 7 – Integração com Streamlit**

• Criar o arquivo app.py.

• Implementar a interface gráfica com:

o Cadastrar Livro (formulário).

o Listar Livros (tabela).

o Atualizar Disponibilidade (botão).

o Remover Livro (bônus).

o Relatório por Autor (bônus).

• A interface deve usar as mesmas funções de banco já criadas.

Commit esperado: "Interface gráfica no Streamlit"

**Critérios de Avaliação**

• Código funcional.

• Estrutura organizada em dois arquivos (main.py e app.py).

• Uso correto de SQLite.

• Uso de Git com commits bem descritos.

• Funcionalidades bônus valem pontos extras.

**Bônus (Extras):**

**Pesquisa de livros (bônus)**

• Criar funções de pesquisa:

o pesquisar\_livros\_por\_titulo(titulo)

o pesquisar\_livros\_por\_autor(autor)

• Utilizar LIKE para permitir buscas parciais.

Commit: git commit -m "Funções de pesquisa por título e autor"

**Relatório por autor (bônus)**

• Criar função relatorio\_por\_autor() que mostra quantos livros cada autor possui.

• Utilizar GROUP BY no SQL.

Commit: git commit -m "Relatório de livros por autor “