

# Escopo para Desenvolvimento de um CRUD Simples

**Objetivo:** Desenvolver um aplicativo CRUD (Create, Read, Update, Delete) básico para gerenciamento de dados, utilizando um banco de dados simples e uma linguagem de programação à escolha do aluno.

**Público:** Estudantes iniciantes em desenvolvimento de software e banco de dados.

## Requisitos:

### 1. Modelo de Banco de Dados:

- Criar um banco de dados relacional com 3 entidades simples, por exemplo:
  - **Clientes:** (id, nome, email, telefone)
  - **Produtos:** (id, nome, descrição, preço)
  - **Pedidos:** (id, id\_cliente, id\_produto, data\_pedido)
- Definir chaves primárias e estrangeiras para garantir a integridade do banco de dados.
- Criar o script SQL para a criação do banco de dados e tabelas.

### 2. Linguagem de Programação:

- O aluno deve escolher uma linguagem de programação para desenvolver o aplicativo (ex: Python, Java, C#, JavaScript, etc.).
- Utilizar um framework ou biblioteca para facilitar o desenvolvimento da interface e acesso ao banco de dados (ex: Flask, Django, Spring Boot, Node.js, etc.).

### 3. Funcionalidades do CRUD:

- Implementar as operações básicas de CRUD para cada entidade:
  - **Criar (Create):** Inserir novos registros no banco de dados.
  - **Ler (Read):** Consultar registros existentes no banco de dados.
  - **Atualizar (Update):** Modificar registros existentes no banco de dados.
  - **Deletar (Delete):** Remover registros do banco de dados.

### 4. Interface do Usuário (UI):

- Criar uma interface de usuário simples para interagir com o aplicativo.
- A UI pode ser desenvolvida utilizando HTML, CSS e JavaScript para aplicações web, ou bibliotecas gráficas para aplicações desktop.
- A interface deve permitir:
  - Visualizar a lista de registros de cada entidade.
  - Adicionar novos registros.
  - Editar registros existentes.
  - Excluir registros.
  - Pesquisar registros por critérios específicos.

### 5. Conexão com o Banco de Dados:

- Utilizar um conector de banco de dados adequado para a linguagem escolhida (ex: psycopg2 para PostgreSQL, mysql-connector-python para MySQL, etc.).
- Implementar a lógica para conectar ao banco de dados, executar consultas SQL e manipular os resultados.

### 6. Documentação:

- Criar um documento com a descrição do projeto, incluindo:

- Modelo de banco de dados (diagrama ER ou script SQL).
- Descrição das funcionalidades do CRUD.
- Instruções para instalação e execução do aplicativo.
- Código fonte do aplicativo.

### **Etapas:**

1. **Planejamento:**
  - Definir o modelo de banco de dados e criar o script SQL.
  - Escolher a linguagem de programação e as ferramentas a serem utilizadas.
  - Definir a estrutura do projeto e a arquitetura do aplicativo.
2. **Desenvolvimento do Banco de Dados:**
  - Criar o banco de dados e as tabelas utilizando o script SQL.
3. **Desenvolvimento do CRUD:**
  - Implementar as operações CRUD para cada entidade, utilizando a linguagem de programação escolhida.
  - Testar cada funcionalidade individualmente.
4. **Desenvolvimento da Interface do Usuário:**
  - Criar a interface do usuário para interagir com o aplicativo.
  - Implementar a lógica para conectar a interface com as funcionalidades do CRUD.
5. **Testes:**
  - Realizar testes completos do aplicativo para garantir que todas as funcionalidades estejam funcionando corretamente.
  - Corrigir eventuais erros e bugs.
6. **Documentação:**
  - Criar a documentação do projeto, incluindo todos os requisitos e instruções.

### **Entrega:**

- Script SQL para criação do banco de dados e tabelas.
- Código fonte do aplicativo.
- Documentação do projeto.

### **Avaliação:**

- Funcionamento correto do CRUD.
- Qualidade do código fonte (organização, legibilidade, eficiência).
- Qualidade da interface do usuário (usabilidade, design).
- Qualidade da documentação.

### **Observações:**

- O aluno pode utilizar qualquer SGBD (Sistema de Gerenciamento de Banco de Dados) de sua preferência (ex: MySQL, PostgreSQL, SQLite, etc.).