

## CRUD Mundo – Programação Web

Você foi contratado para desenvolver uma aplicação web completa que permita o gerenciamento de dados geográficos, com foco em países e cidades do mundo. O sistema deverá contemplar tanto a camada de Front End (HTML, CSS e JavaScript), quanto a camada de Back End (PHP), com armazenamento e manipulação de dados em um banco de dados MySQL.

### Objetivo Geral:

Implementar um CRUD (Create, Read, Update e Delete) que possibilite ao usuário:

- Cadastrar novos países e associar cidades a eles.
- Consultar a lista de países e cidades existentes.
- Editar informações de países e cidades.
- Excluir registros, mantendo a integridade referencial no banco de dados.

### Requisitos Funcionais:

#### 1. Gerenciamento de Países:

- Inserir, listar, editar e excluir países.
- Cada país deve conter ao menos: ID, nome oficial, continente, população e idioma principal.

#### 2. Gerenciamento de Cidades:

- Inserir, listar, editar e excluir cidades associadas a um país existente.
- Cada cidade deve conter ao menos: ID, nome da cidade, população, país (chave estrangeira).

#### 3. Interface Web (Front End):

- Criar páginas em HTML5 estruturadas semanticamente.
- Aplicar CSS3 para estilização, garantindo boa usabilidade e design responsivo.
- Utilizar JavaScript para validação de formulários e pequenas interações dinâmicas (ex.: alertas, confirmação de exclusão).

#### 4. Camada Back End (PHP + MySQL):

- Implementar scripts em PHP que realizem a comunicação com o banco de dados.
- As operações de CRUD deverão ser realizadas através de queries SQL (INSERT, SELECT, UPDATE, DELETE).
- Garantir a integridade dos dados ao excluir países que tenham cidades associadas (ex.: só permitir exclusão se não houver cidades vinculadas, ou realizar exclusão em cascata, se configurado no banco).

## 5. Banco de Dados (MySQL):

- Criar um banco de dados chamado *bd\_mundo*.
- Criar duas tabelas principais:
  - *paises (id\_pais, nome, continente, populacao, idioma)*.
  - *cidades (id\_cidade, nome, populacao, id\_pais)*.
- Estabelecer relacionamento 1:N (um país pode ter várias cidades).

### Critérios de Avaliação:

- Organização da estrutura de pastas (separação de front end e back end).
- Uso correto de HTML, CSS e JavaScript no front end.
- Implementação funcional das operações CRUD em PHP.
- Correção e eficiência das queries SQL.
- Qualidade visual da interface (layout, responsividade, clareza na navegação).
- Tratamento de erros e validações adequadas (ex.: campos obrigatórios, confirmação de exclusão).

### Desafio Extra (opcional):

- Criar uma funcionalidade de pesquisa dinâmica (com JavaScript) para buscar países ou cidades pelo nome.
- Exibir estatísticas simples, como a cidade mais populosa de um país ou o total de cidades cadastradas por continente.

### Integração com APIs Externas:

- O sistema deverá consumir pelo menos duas APIs externas para enriquecer os dados cadastrados:
  - **API 1 (Países):** REST Countries → fornecer informações complementares sobre países (ex.: bandeira, moeda, capital).
  - **API 2 (Clima):** OpenWeatherMap → exibir informações climáticas em tempo real de uma cidade cadastrada.

### Controle de Versionamento (Git/GitHub):

- Utilizar Git para versionar o projeto.
- Criar um repositório no GitHub com:
  - Histórico de commits descritivos.
  - Organização em branches.
  - README.md com descrição do projeto, instruções de instalação e uso.

#### Entrega:

- Individual.
- Arquivo ReadMe contendo:
  - Nome do(a) aluno(a), nome do projeto, tecnologias utilizadas, descrição detalhada do projeto.
- Projeto disponibilizado no GitHub.