

RELATÓRIO DO CRUD EM PHP

Candidato: Cauê Felix Urbini

Processo Seletivo: Estágio em Desenvolvimento

Empresa: DirectWeb

Data: 05 de setembro de 2025

1. Introdução

Este relatório mostra o desenvolvimento de um sistema CRUD feito em PHP como parte de um desafio técnico para estágio. O sistema serve para gerenciar notas (criar, listar, editar e apagar). Durante o desenvolvimento, pratiquei conceitos básicos de programação web, organização de código e medidas de segurança.

2. Tecnologias Utilizadas

- PHP puro (sem frameworks).
- Arquivo JSON como banco de dados.
- HTML5, CSS3 e JavaScript para a interface.
- Configuração de fuso horário para America/Sao_Paulo.
- Layout responsivo.

3. Estrutura do Sistema

- A aplicação é simples, mas organizada entre a lógica do PHP e a parte visual (HTML/CSS).
- Cada nota tem os campos: *id*, *content*, *created_at* e *updated_at*.
- Os dados ficam salvos em um arquivo chamado notes.json.

4. Funcionalidades

- Adicionar notas(*addNotes*): Faz validação para não permitir salvar notas vazias, usa *trim()* para remover espaços. Cria um *id* único com *uniqid()*, adiciona a data de criação e depois chama *saveNotes(\$notes)* para gravar no arquivo JSON.
- Listar notas(*getNotes*): Verifica se o arquivo existe, lê o conteúdo e mostra as notas, se não houver notas, exibe uma mensagem.
- Editar notas(*editNotes*): Ao clicar em editar, o formulário já vem preenchido com os dados atuais da nota, o usuário pode alterar o texto, e ao salvar, o sistema atualiza a data de modificação e chama *saveNotes(\$notes)* para salvar a alteração.
- Excluir notas(*deleteNotes*): Antes de excluir, o sistema mostra um modal de confirmação para evitar exclusões acidentais. Se confirmado, remove a nota com *array_filter()*, reorganiza os índices com *array_values()* e usa *saveNotes(\$notes)* para atualizar o arquivo JSON.

5. Segurança

- Usei *htmlspecialchars()* para evitar que alguém insira código malicioso no conteúdo da nota.
- Validação no frontend e backend para não salvar campos vazios.
- Tratamento seguro dos parâmetros de URL para evitar problemas.

6. Testes Realizados

- Funcionais: testei criar, editar, listar e excluir, incluindo casos de erro (ex.: nota vazia).
- Segurança: tentei injetar código no campo, mas o sistema mostrou só como texto.
- Interface: testei em celular, tablet e computador, e ficou responsivo.
- Usabilidade: testei cancelar edições/exclusões e o sistema manteve os dados corretos.

7. Código e Organização

- Código dividido em funções .
- Comentários explicando partes importantes e aprendizados que obtive pelo caminho.
- Uso de boas práticas como validação dupla e tratamento de erros.

8. Interface do Usuário

- Modal de confirmação para excluir notas.
- Mensagens de feedback rápidas para avisar o usuário.
- Design simples, mas responsivo e acessível.

9. Conclusão

O CRUD em PHP foi uma boa prática para aprender como funciona a criação de aplicações web básicas. Consegui aplicar validações, segurança simples, salvar dados em JSON e deixar a interface responsiva. Esse projeto me ajudou a entender melhor como organizar código e pensar em melhorias para o futuro, como: autenticação de usuários (login), usar banco de dados no lugar do Json e melhorar o design.