

Sistema de Controle de Estoque de Materiais de Limpeza

Sobre o Projeto

Este é um projeto de controle de estoque de materiais de limpeza desenvolvido em PHP, HTML, CSS e MySQL. Ele segue a arquitetura MVC (Model-View-Controller) para organizar o código de forma clara, escalável e de fácil manutenção.

O objetivo do sistema é permitir o gerenciamento de entrada, saída e controle de validade dos materiais de limpeza de forma eficiente.

Estrutura de Pastas

A estrutura de pastas foi projetada para separar as responsabilidades da aplicação, facilitando o desenvolvimento e a manutenção do código. Abaixo está a descrição detalhada de cada diretório principal, partindo da raiz sistema/.

```
sistema/
├── app/
│   ├── Controlador/
│   │   ├── LoginController.php      (PHP)
│   │   ├── InicioController.php     (PHP)
│   │   └── MateriaisController.php  (PHP)
│   ├── Database/
│   │   └── Conexao.php              (PHP)
│   ├── Modelo/
│   │   ├── Entidade/
│   │   │   ├── Usuario.php          (PHP)
│   │   │   └── Material.php         (PHP)
│   │   └── Repositorio/
│   │       ├── UsuarioRepositorio.php (PHP)
│   │       └── MaterialRepositorio.php (PHP)
│   └── Utilidades/                  (PHP)
├── views/
│   ├── login/
│   │   └── index.php                (HTML e PHP)
│   ├── inicio/
│   │   └── index.php                (HTML e PHP)
│   └── materiais/
│       ├── adicionar.php            (HTML e PHP)
│       └── listar.php                (HTML e PHP)
└── public/
    ├── css/
    │   └── estilo.css               (CSS)
    └── img/
        └── logo.png                 (Imagens)
```

Por que esta Estrutura é a Ideal?

Para iniciantes, pode parecer que uma estrutura com muitas pastas deixa o projeto mais complexo, mas é o oposto. Essa organização resolve problemas futuros e permite que o trabalho em equipe seja muito mais eficiente.

Separação de Responsabilidades: Cada equipe sabe exatamente onde trabalhar. A equipe de Front-end se concentra nas pastas views/ e public/. A equipe de Back-end foca em app/. A equipe de Banco de Dados tem uma pasta dedicada (Database/).

Foco no Aprendizado: A arquitetura MVC é um padrão profissional. Ao segui-la,

você estão aprendendo e aplicando as melhores práticas do mercado desde o início, o que é um diferencial valioso.

Manutenção e Escalabilidade: Quando o projeto crescer, será muito mais fácil encontrar e corrigir erros ou adicionar novas funcionalidades. Por exemplo, para criar uma nova funcionalidade (ex: "Remover Material"), você sabe que precisará criar uma rota no controlador (MateriaisController.php) e talvez uma nova visualização na pasta views/materiais/.

📁 **Detalhamento de cada Pasta e Arquivo**

app/: O coração do sistema. Contém toda a lógica de negócio e os arquivos aqui não devem ser acessados diretamente pelo navegador por questões de segurança.

Controlador/: Responsável por receber requisições do usuário, processar dados e determinar qual view será exibida.

Modelo/: Comunica-se com o banco de dados. É dividido em Entidade/, que representa as tabelas (ex: Usuario.php), e Repositorio/, que contém as funções para interagir com o MySQL.

views/: A camada de apresentação. A equipe de Front-end trabalha aqui. Os arquivos HTML (que na verdade são .php) recebem os dados dos Controladores e os exibem.

A divisão por subpastas (login/, materiais/, inicio/) ajuda a evitar conflitos de nomes (ex: ter dois arquivos chamados index.php) e mantém o projeto organizado.

public/: A única pasta acessível diretamente pelo navegador. Contém todos os arquivos estáticos como CSS, JavaScript e imagens. Isso aumenta a segurança do projeto.