

Bing

Claro! Organizar as pastas de um projeto em **PHP com o padrão MVC (Model-View-Controller)** é uma prática excelente para manter o código modular, escalável e fácil de manter. Vou te mostrar como estruturar seu projeto seguindo esse padrão:

1. Estrutura Básica: Comece criando uma estrutura de pastas que represente as três camadas do MVC:

```

meu_projeto/
├── app/
│   ├── controllers/
│   │   └── HomeController.php
│   ├── models/
│   │   └── UserModel.php
│   └── views/
│       ├── home/
│       │   └── index.php
│       └── layout/
│           ├── header.php
│           └── footer.php
├── public/
│   ├── css/
│   ├── js/
│   └── index.php
├── config/
│   └── config.php
└── .htaccess
  
```

- A pasta app contém as três camadas do MVC:
 - controllers: Lida com a lógica de negócios.
 - models: Gerencia a interação com o banco de dados.
 - views: Contém os arquivos de interface.
- A pasta public é acessível pelo navegador e contém arquivos estáticos como CSS, JavaScript e o arquivo index.php.
- A pasta config pode conter configurações globais do projeto.
- O arquivo .htaccess é usado para configurar URLs amigáveis.

2. Arquivos e Nomes Descritivos:

- Nomeie seus arquivos e pastas de forma descritiva. Por exemplo, HomeController.php em vez de Ctrl.php.
- Dentro da pasta views, organize os arquivos por funcionalidade (por exemplo, home, sobre, contato).

3. Exemplo de Arquivos:

- HomeController.php (Controller):

```

<?php
class HomeController {
    public function index() {
        // Lógica para carregar dados e renderizar a view
        include 'views/home/index.php';
    }
}
  
```

- UserModel.php (Model):

```

<?php
class UserModel {
    // Lógica para interagir com o banco de dados
}
  
```

- index.php (View):

```
<?php include 'layout/header.php'; ?>
<h1>Bem-vindo ao Meu Projeto</h1>
<!-- Conteúdo da página -->
<?php include 'layout/footer.php'; ?>
```

4. Rotas e Front Controller:

- Use um arquivo `index.php` na pasta `public` como ponto de entrada (Front Controller).
- Configure rotas para direcionar URLs para os controladores corretos.

5. **Frameworks**: Se estiver usando um framework PHP (como Laravel, Symfony ou CodeIgniter), ele já oferecerá uma estrutura MVC bem definida. Caso contrário, você pode criar sua própria estrutura seguindo essas diretrizes.

Lembre-se de que a organização do projeto é essencial para facilitar a colaboração e a manutenção contínua. 🚀