Documentação Técnica: Sistema de Gerenciamento de Contatos Pessoais

1. Visão Geral

O Sistema de Gerenciamento de Contatos Pessoais é uma aplicação web desenvolvida em PHP para permitir aos usuários gerenciar contatos de maneira eficiente. O sistema apresenta uma interface responsiva, utiliza Bootstrap para estilização e organização dos elementos visuais, e inclui funcionalidades como cadastro, edição, exclusão e visualização de contatos.

2. Tecnologias Utilizadas

- Linguagens: PHP, HTML, CSS

- Frameworks:

- Bootstrap 4.1.3 (para estilização e responsividade)

- Bootstrap Icons (para ícones)

- Banco de Dados: MySQL (supondo pela utilização do arquivo `db/conexao.php`)

- Bibliotecas de Fontes: Google Fonts (Poppins e Roboto Slab)

- Outras Dependências utilizadas:

- jQuery

- Popper.js

---

3. Estrutura do Projeto

3.1. Diretórios Principais

- Arquivo principal: index.php é o ponto de entrada do sistema.

Banco de dados “/db”:

- Contém o arquivo “conexao.php”, responsável pela conexão com o banco de dados.

Pasta: /paginas

- Contém os módulos organizados por funcionalidade:

- home : Contém a página inicial (`home.php`).

- Contatos: Contém os arquivos relacionados ao gerenciamento de contatos:

- contato.php: Exibição de contatos.

- cad-contatos.php: Formulário de cadastro de novos contatos.

- inserir-contato.php: Lógica para inserir novos contatos.

- editar-contato.php: Formulário para edição de contatos.

- atualizar-contato.php: Lógica para atualizar contatos existentes.

- excluir-contato.php: Lógica para excluir contatos.

`/css`

- Contém arquivos de estilo personalizados como `estilo-padrao.css` e o logotipo (`agenda.png`).

---

4. Arquitetura e Funcionamento

4.1. Controle de Páginas com `$menuop`

O arquivo `index.php` utiliza a superglobal `$\_GET` para controlar qual página será exibida. O parâmetro `menuop` define a funcionalidade atual.

Exemplo de Uso:

1. URL: `index.php?menuop=home` -> Carrega `home.php`.

2. URL: `index.php?menuop=contatos` -> Carrega `contato.php`.

Se `menuop` não for especificado, o sistema assume o valor padrão `home` e carrega `home.php`.

4.2. Switch para Inclusão de Páginas

O controle é feito com um `switch` que seleciona os arquivos baseados no valor de `$menuop`:

```php

switch($menuop) {

case 'home':

include("./paginas/home/home.php");

break;

case 'contatos':

include("./paginas/contatos/contato.php");

break;

case 'cad-contato':

include("./paginas/contatos/cad-contatos.php");

break;

case 'inserir-contato':

include("./paginas/contatos/inserir-contato.php");

break;

case 'editar-contato':

include("./paginas/contatos/editar-contato.php");

break;

case 'atualizar-contato':

include("./paginas/contatos/atualizar-contato.php");

break;

case 'excluir-contato':

include("./paginas/contatos/excluir-contato.php");

break;

default:

include("paginas/home/home.php");

break;

}

```

4.3. Banco de Dados

- O arquivo `db/conexao.php` deve conter a lógica de conexão ao banco de dados (MySQL).

- As operações de CRUD (Create, Read, Update, Delete) são realizadas pelos arquivos em `/paginas/contatos/`.

4.4. Responsividade e Estilo

- A estrutura responsiva é gerenciada pelo Bootstrap 4.1.3.

- Estilos adicionais são definidos em `estilo-padrao.css`, garantindo uma identidade visual personalizada.

---

5. Funcionalidades

5.1. Exibição de Contatos

- A página `contato.php` lista os contatos cadastrados.

- Deve incluir botões para editar ou excluir cada contato.

5.2. Cadastro de Contatos

- O formulário em `cad-contatos.php` permite inserir novos contatos.

- Os dados são processados por `inserir-contato.php` e salvos no banco.

5.3. Edição de Contatos

- O formulário em `editar-contato.php` permite modificar informações de um contato existente.

- As alterações são enviadas para `atualizar-contato.php`.

5.4. Exclusão de Contatos

- A página `excluir-contato.php` realiza a remoção de contatos.

- Deve incluir uma verificação de confirmação antes de excluir definitivamente.

---

6. Possíveis Melhorias

1. Segurança:

- Validar e sanitizar todos os dados recebidos via `$\_GET` e `$\_POST`.

- Proteger contra SQL Injection utilizando prepared statements (PDO ou MySQLi).

2. Feedback para o Usuário:

- Adicionar mensagens de sucesso ou erro após cada operação (ex.: "Contato cadastrado com sucesso").

3. Estilo:

- Melhorar o design das tabelas e formulários para maior usabilidade.

4. Novas Funcionalidades:

- Implementar busca e filtros para os contatos.

- Adicionar opção para exportar contatos (CSV ou PDF).

5. Arquitetura:

- Migrar para um padrão MVC (Model-View-Controller) para maior organização do código.

---

7. Conclusão

O Sistema de Gerenciamento de Contatos é uma aplicação simples e funcional que utiliza boas práticas de desenvolvimento. Com melhorias pontuais, como reforço na segurança e aprimoramento na interface, o sistema pode se tornar ainda mais eficiente e profissional.