

Seleção para Desenvolvedor Junior



www.facilconsulta.com.br

Teste

O Teste tem como objetivo validar as habilidades que o candidato(a) deve ter para concorrer a vaga.

Fácil Consulta

Entrega

A entrega será feita pelo o envio do link do repositório do teste do candidato para o **Whatsapp (53) 98464-8629**. Que deverá conter todo o código desenvolvido, incluindo o do Banco de Dados (estrutura + dados).

Todos os **requisitos e funcionalidades** do Sistema, descritos neste documento, serão avaliados, além da **qualidade do código, lógica, organização e usabilidade das páginas**.

1. Requisitos

- 1 - Utilizar apenas **PHP Orientado a Objetos** como linguagem Back-end;
- 2 - Utilizar a **Arquitetura MVC**;
- 3 - Utilizar **MySQL** no desenvolvimento do Banco de Dados;
- 4 - Desenvolver **2 tabelas** no Banco de Dados;
- 5 - Desenvolver um **Cadastro de Médico** e **Agendamento de Horário**;
- 6 - Utilizar **HTML, CSS** e **JavaScript** como linguagem Front-end;

Observações:

***Não é permitido utilizar framework** e/ou bibliotecas e/ou módulos no Back-end, exceto as nativas do PHP, como PDO e Date;

*É permitido o uso de **bibliotecas no Front-end**, como jQuery, Bootstrap etc;

*O Design deve ser **Responsivo**;

***Não é necessário desenvolver** um sistema de Login, ou seja, é possível editar um cadastro, fazer um agendamento, adicionar um horário e remover um horário sem estar logado.

2. Descrição

Deve ser desenvolvido um **Cadastro de Médico** e **Agendamento de Horário**, para isso deve ser utilizado PHP Orientado a Objetos, MySQL, HTML, CSS, JavaScript e desenvolvido 4 páginas: **Cadastro de Médico**, **Listagem dos Médico e seus Horários (Listagem)**, **Edição do Cadastro de Médico** e **Configurar Horários**.

As páginas estão descritas no tópico **4. Páginas e Funcionalidades**.

Fácil Consulta

Deve ser usado controle de versões no desenvolvimento, utilizando o GitHub como repositório. Para facilitar o desenvolvimento foi criado um modelo de projeto, disponível em: <https://github.com/KrishnaXavier/modelo-fc> (todas as recomendações do modelo devem ser seguidas).

3. Tabelas

Para o **Cadastro de Médico** deve ser criada a tabela **medico**, com os atributos: **id**, **email**, **nome**, **senha**, **data_criacao** e **data_alteracao**.

Para o **Agendamento de Horário** deve ser criada a tabela **horario**, com os atributos: **id**, **id_medico**, **data_horario**, **horario_agendado**, **data_criacao** e **data_alteracao**.

O atributo **data_horario** da tabela **horario** deve ser do tipo **datetime** e deve corresponder ao horário configurado para o agendamento.

O atributo **horario_agendado** da tabela **horario** deve corresponder a disponibilidade do horário:

- Se for 1 (um) o horário já foi agendado e está ocupado;
- Se for 0 (zero) o horário está liberado para o agendamento;

Os atributos **data_criacao** e **data_alteracao** das tabelas **medico** e **horario** devem ser fornecidos pelo sistema. O tipo desses atributos no Banco de Dados devem ser **timestamp**.

4. Páginas e Funcionalidades

As páginas devem seguir o Projeto Visual: [Cadastro de Médico](#), [Listagem](#), [Edição do Cadastro de Médico](#), [Configurar Horários](#) e [Cores e Tipografia \(fonte\)](#).

1 - Na página de [Cadastro de Médico](#) serão solicitados os dados de **email**, **nome** e **senha**.

1.1 - Os campos de **email**, **nome** e **senha** devem ter o tamanho mínimo de 6 caracteres e o campo **email** deve ser validado se é do tipo Email. Os campos devem ser validados no Front-end e Back-end;

1.2 - O campo **senha** deve ser salvo de forma criptografada;

2 - Na [Listagem dos Médico e seus Horários \(Listagem\)](#) deve ser mostrado todos os médicos e seus respectivos horários.

2.1 - Quando o usuário clicar em um **Horário** deve ser feito o **Agendamento do Horário**, ou seja, o **Horário** vai se tornar ocupado/agendado (horario_agendado = 1), ao fazer isso a página deve ser atualizada (recarregada);

2.2 - A página de **Listagem** deve ser a inicial/index;

2.3 - No topo da **Listagem** deve ter um botão para o **Cadastro de Médico**;

2.4 - Na **Listagem**, ao lado do nome do Médico deve ter um botão com “Editar cadastro” o qual leva para a página de **Edição do Cadastro de Médico** e outro botão com “Configurar horários” o qual leva para a página de **Configurar Horários**;

3 - Na página de [Edição do Cadastro de Médico](#) os atributos que podem ser atualizados são **nome** e **senha**.

3.1 - Para atualizar a **senha** deve ser informada a **senha antiga** e a **senha nova**. Se a **senha antiga** informada pelo usuário não corresponder a **senha antiga** a **senha** não deve ser atualizada e o usuário informado do problema;

3.2 - Os campos de **nome** e **senha** devem ter o tamanho mínimo de 6 caracteres. Os campos devem ser validados no Front-end e Back-end. E a **senha nova** deve ser salva de forma criptografada;

3.3 - Após atualizar os dados deve retornar para a página de **Listagem**;

4 - Na página de [Configurar Horários](#) deve ser possível: adicionar um novo horário (data, hora e minuto) e remover um horário.

4.1 - Ao adicionar um **Horário** ele deve ficar disponível (horario_agendado = 0);

4.2 - Os **Horários** agendados (horario_agendado = 1) não podem ser removidos;

5 - A ordenação dos Médicos na página de **Listagem** deve seguir as seguintes regras:

5.1 - A ordenação dos Médicos deve ser pelo horário disponível mais próximo, ou seja, o Médico na 1ª posição deve ser com o horário disponível mais próximo e assim por diante;

5.2 - Os **Horários** agendados (horario_agendado = 1) e que já passaram da data e hora atual não devem ser considerados na ordenação dos Médicos;

Para facilitar o entendimento da ordenação foi criado um exemplo, ele está disponível [neste link](#);

5. Considerações Finais

Para não ficar dúvidas sobre o que vai ser avaliado no Teste, segue os critérios que vão ser avaliados:

1. Se os requisitos foram atendidos;
2. Se as funcionalidades foram atendidas;
3. Se a lógica não apresenta nem um problema grave;
4. Organização e qualidade do código;
5. Usabilidade das páginas;
6. Utilização ou não de controle de versões;
7. Se foi utilizado o modelo de projeto fornecido;
8. Não será avaliado a rapidez de entrega;
9. A utilização correta do PDO no PHP;

Qualquer dúvida pode chamar pelo **Whatsapp (53) 98464-8629**