

Teste prático de conhecimentos aplicados ao desenvolvimento web com PHP e frameworks juntamente com banco de dados para nível Júnior.

Esse teste consiste em executar ações tal qual um engenheiro o faz em seu dia a dia de trabalho.

Este teste tem por objetivo medir sua capacidade de aprender tecnologias novas, adaptação, lógica, código enxuto, documentação, pesquisas, o seu eu autodidata.

Este teste é uma fase essencial na etapa em que você demonstra aptidão para uma vaga caso ainda esteja disponível.

Tecnologias utilizadas

Linguagem	PHP 8.0 ou superior/Laravel
CSS	3
HTML	5
Javascript	6
Banco de dados	PostgreSQL 15 ou superior
Visual	VUE 2.5 ou superior
Ferramenta de desenvolvimento	De sua preferência

Como demonstrar seu projeto - Docker e container

1. Entenda e configure seu próprio ambiente
2. Para apresentar utilize o Ngrok ou se desejar subir a aplicação em um servidor, fique a vontade e disponibilize no momento da apresentação.

Dica: **laravel sail...**

Teste conforme nível de habilidade

Júnior	Realizar as rotinas em forma de funções integradas com formulários.
--------	---

	Realizar as rotinas construindo e consumindo pelo menos 1 api para as funções com documentação.
--	---

O que será medido

1. Capacidade de aprendizado (autodidata);
2. Rapidez;
3. Código limpo;
4. Código inteligente (reaproveitamento e pouco código);
5. Documentação e comentários de código (fluxo de dados e documentação);
6. Experiência do usuário (que seja fácil de operar);
7. Beleza (que seja agradável e bonito e prático);
8. Erros/bugs (exceções e logs) e testes do usuário.

Tarefa Geral

Criar um software de controle de revisões de veículos.

Tarefa 1.1 : Banco de Dados

1. Criar um esquema com seu nome e as tabelas dentro desse esquema;
2. Criar uma tabela com os campos para cadastro do proprietário do carro;
3. Criar uma tabela com os campos para os veículos dos proprietários;
4. Criar uma tabela com os campos para os dados das revisões;
5. Criar outras tabelas e adicionar mais recursos na aplicação poderá ser feito, como por exemplo login e outras funções que acredita serem relevantes.

Como gerar as consultas / relatórios / SQLs:

1. Crie o sql dos relatórios e deixe salvo em um arquivo de texto;
2. Os relatórios deverão ser apresentados em formato visual e legível ao usuário (listas) por exemplo, e incluir gráficos.

3. Gere relatórios. ~~Utilize o laravel Eloquent ORM para realizar as operações com o banco de dados.~~ Os relatórios devem ser em sql.
4. Todos os relatórios devem estar disponíveis para uso no sistema e todos devem conter gráficos. Relatórios que possuem apenas uma informação, seja criativo e pense como usar com um gráfico. Abaixo alguns relatórios para serem criados:
 - a. **Veículos**
 - i. Todos os veículos;
 - ii. Todos os veículos por pessoa ordenado por nome de pessoa;
 - iii. Informação de quem tem mais veículos (homens ou mulheres);
 - iv. Todas as marcas ordenadas pelo número de veículos;
 - v. Totais de marcas ordenadas do maior para o menor, separados entre homens e mulheres.
 - b. **Pessoas**
 - i. Todas as pessoas;
 - ii. Todas as pessoas separadas por homens e mulheres (com idade média de homens e mulheres).
 - c. **Revisões**
 - i. Todas as revisões dentro de um período;
 - ii. Marcas com maior número de revisões;
 - iii. Pessoas com maior número de revisões;
 - iv. Média de tempo entre uma revisão e outra de uma mesma pessoa;
 - v. Próximas revisões baseado no tempo médio baseado na última revisão.

Tarefa 2: Cadastro, edição, remoção e exibição(listar)

CRUD

1. Criar um cadastro com os campos para as pessoas com as funções crud;
2. Criar um cadastro para os veículos com as funções crud sendo chamado a partir de um campo na lista das pessoas;
3. Criar um cadastro para lançamento das revisões com as funções crud;
4. Se desejar crie mais cadastros para agregar ao seu sistema.

Visual

1. Todo Software deve ser responsivo;
2. Deve ter alinhamento;
3. Deve ter sequência lógica de preenchimento;
4. Deve ser bonito, agradável;
5. Se inspire em visuais que existem na internet feitos em vue.

Tarefa 3: Estilos/css

CSS

1. Criar estilos para serem utilizados em todo o software;
2. Utilizar estilos comuns, para re-uso;
3. Utilizar animações em relatórios e em cadastros, desde que as animações façam sentido ao usuário. Use animações prontas.

Tarefa 4: Usabilidade

Lado do cliente / front

1. Validar campos;
2. Formatar;
3. Opcional (não obrigatório)
 - a. Ciclo de vida de formulário (manter a informação local para ser recuperada em caso de fechamento da tela sem salvar ou queda de energia, por exemplo).

Tarefa 5: PHP/back

PHP

1. Atender aos padrões básicos de desenvolvimento (PSR). É importante seguir uma linha organizada;
2. Opcional (não obrigatório)
 - a. Atentar para complexidade ciclomática (para um número mínimo de testes); Este item não é obrigatório, mas se tiver conhecimento é importante.
3. Opcional(não obrigatório)
 - a. Valide seu código, por exemplo, com o uso do PHPStan (testes).

Tarefa 6: Prazo e local de entrega/desenvolvimento

Apresentação

Local de desenvolvimento	Onde desejar / Remoto
Prazo	3 dias
Consultas	Internet

	Caso use código pronto do git ou do chatGPT, ou outros, lembre-se que pode ser perguntado pra explicar o código.
--	--

</fim>

Desejamos que aplique seu conhecimento, que independente do resultado, consiga aprender por si mesmo algo novo, que seja um teste desafiador e inspirador, que mostre a você mesmo sua capacidade de aprendizado e evolução.

Bons códigos...