

Teste Prático para Desenvolvimento Web

Objetivo

O objetivo do teste é avaliar a forma como o candidato evolui a solução para o problema proposto. O candidato deve entregar o teste contemplando os itens obrigatórios.

Requisitos para o Teste

Conhecimento básico, intermediário ou avançado em:

CSharp (C#)

SQL

HTML/CSS

O que é obrigatório no teste?

Entregar todos os itens do tópico Descrição do Teste mais abaixo

Utilizar o GIT para controle de versão e ter ao menos dois commits

Uso de um banco de dados relacional, podendo ser o SQLite ou SQL Server

Disponibilizar um markdown (ex: README.md) na raiz da solução com as instruções para compilar, configurar banco e executar o projeto

O que não é obrigatório, mas contará pontos?

Uso de conceitos de Programação Orientada a Objeto (POO) e boas práticas (DDD, CleanCode, etc);

Separação de responsabilidade do código (código de aplicação, regra de negócio e acesso a dados); Utilizar alguma biblioteca de frontend como React (preferencial) ou jQuery.

Para React, demonstrar conhecimento não só em FunctionComponent, Hooks, etc, mas também com ClassComponent e seu ciclo de vida.

Descrição do Teste

Deve ser criada uma aplicação para o cadastro e listagem de pontos turísticos do país, cada ponto turístico deve ter o nome, descrição até 100 caracteres e localização (endereço ou referência de localização), cidade e estado).

Deve ser possível na página inicial listar de forma paginada os cadastros ordenados de forma decrescente pela data de inclusão e permitir buscar/filtrar digitando um termo de busca analisando tanto nome e descritivo quanto localização dos pontos turísticos cadastrados. A página deve listar os pontos com o nome e localização.

Ao ser selecionado o ponto turístico deve ser demonstrada o nome, descrição e localização. No cadastro, os estados devem ser listados como combo/dropdown e a cidade ou pode ser texto livre, ou buscar de algum webservice online disponível na internet as cidades de acordo com o estado selecionado. Criar um menu de navegação para alternar entre o formulário de cadastro e a listagem.

Modelos da Solução

Abaixo estão dispostos dois modelos solução que o candidato poderá escolher, escolha o que estiver mais confortável e compatível com seus conhecimentos:

Monolítica (perfil júnior)

Criar uma aplicação web comum, contendo as funcionalidades solicitadas, onde a mesma serve as páginas com as informações desejadas e processa as requisições se conectando ao banco para gravar ou consultar dados.

Em Camadas (perfil pleno/sênior)

Dividir a aplicação em duas camadas físicas rodando independentemente, sendo: O projeto com a API responsável pelo processamento de requisições e se conectar ao banco de dados para gravar ou consultar informações. O projeto com a aplicação front-end podendo ser um SPA React onde se comunicará com a API via Ajax via fetch/axios etc, ou aplicação comum onde o backend se comunicará com a API através de HTTP requests.

Telas que devem ser usadas como referência para as funcionalidades

Teste Prático de Conhecimento em SQL

Objetivo

Através da estrutura das tabelas criadas para a execução do aplicativo executável, gere um arquivo texto para cada item abaixo contendo os comandos SQL (SELECT, INSERT, UPDATE e DELETE)

1. Criar um comando SELECT que retorne APENAS o nome dos clientes da cidade "TUPA". Utilizar o EXISTS para realizar a condição.
2. Criar um comando SELECT que retorne o nome do cliente e o nome da cidade de todos os registros.
3. Criar um comando SELECT que retorne TODOS os nomes e os códigos dos clientes ordenados por nome do cliente.
4. Criar um comando DELETE que exclua clientes com código do cliente entre a numeração 100 e 200.
5. Criar um comando UPDATE que altere o estado de todas as cidades para o estado "SP" quando estiverem com estado igual a "PR".
6. Criar um comando INSERT na tabela de clientes de um registro qualquer com todos os campos da tabela. (valores livres)

Requisitos extras:

7. Criar uma view para consultar os registros contidos na tabela CIDADES. (Criar um arquivo separado);
8. Criar um procedimento armazenado (procedure) para realizar as operações de INSERT, UPDATE E DELETE na tabela de CIDADES. Criar uma procedure única que faça todas as operações. Ela pode ser usada no aplicativo criado, se desejar. (Criar um arquivo separado)

Teste Prático de Lógica

Objetivo

Monte um arquivo texto com um algoritmo, passo a passo, para troca de um pneu de carro que furou. Exemplo de um algoritmo bem básico para a realização de um churrasco.

1. Acender a churrasqueira;
2. Lavar os espetos;
3. Espetar a carne nos espetos
4. Colocar os espetos para assarem;
5. A cada 3 minutos trocar o lado do espeto;
6. Após 15 minutos retirar a carne;
7. Servir a carne aos convidados;
8. Apagar o fogo da churrasqueira;
9. Lavar os espetos;
10. Guardar os espetos.