

PROCESSO SELETIVO CERTI - FULL STACK JÚNIOR

O teste é composto de duas etapas, uma teórica e outra prática, explicadas a seguir.

Teste Teórico

O teste teórico é composto de algumas perguntas-problema, cuja resposta deve ser em código. Nela, avaliamos a capacidade de interpretação de texto e pensamento lógico e desenvolvimento de algoritmos.

Para realizá-lo, você deve acessar o [ambiente de testes do InterviewZen](#).

Preencha **seu** nome e e-mail, e clique em **"Start the Interview"**. Cada questão tem um timer que conta o tempo total que você levou no problema. Ao concluir cada problema, clique no botão **"Submit solution"**.

Quando acabar o teste, **envie o link do resultado como resposta ao email que enviou e coloquei como cópia o email:**

Nota: você só pode realizar o teste uma vez, e não pode editar seu código após enviar a solução

Teste Prático

Para o teste prático, você precisa elaborar uma solução que aborde o ambiente de frontend e uma API Rest.

1. Crie uma aplicação web em Angular 14+ para mostrar todos os pokemons já capturados pelo usuário. A aplicação deve consumir a API para cumprir os requerimentos mencionados abaixo.
 - Deve mostrar um formulário com um input para pegar um novo pokemon digitando o seu nome.
 - Deve ser mostrada uma lista com todos os pokemons capturados pelo usuário. A lista deve conter nome, id e a imagem do pokemon. Quando a lista está vazia, deve ser mostrada a seguinte mensagem: "Ainda não capturou nenhum pokemon".
 - Os pokemons capturados no passo 1 devem automaticamente aparecer na lista mencionada no passo 2.
2. Elabore uma API que dê suporte ao frontend criado no exercício 1.
 - Como tecnologia sugerimos: Node.js, NEST e Spring boot.
 - Não há necessidade de um banco de dados, você pode elaborar em memória

Bônus:

As tarefas bônus são opcionais, mas contam pontos no processo seletivo. Elas nos ajudam a identificar não sua pró-atividade e suas habilidades em organização de código, arquitetura, etc.

- Elaborar um banco de dados físico, como sugestão MongoDB e elaborar o passo 1 e 2 em Docker
- Entregar o endereço da imagem que está no Dockerhub

O que apreciamos ++

- Código estruturado, documentado e minimamente comentado, preferencialmente em SWAGGER
- Verificação de consistência dos dados de entrada;
- Retorno de erros para o front;
- Testes unitários;
- Código limpo e bem organizado; Finalizando

Como entregar o teste

Suba a sua proposta para o projeto que você criou no GitHub. Exemplo: <https://github.com/seuNome>

- Envie-nos o link do seu projeto no prazo estipulado por email.
 - Exemplo: <https://github.com/seuNome/test-backEnd.git>

Aguarde o RH entrar em contato