



Teste técnico

Desenvolvedor(a)

FullStack





Por que estamos fazendo este teste?

O objetivo desse teste é conhecer como você programa, o conhecimento que possui, o profissionalismo que emprega no código e também a capacidade que tem de aprender rapidamente.

Valorizamos mais a qualidade do que a quantidade.

Por isso não se apresse em entregar o teste completo exatamente no prazo, mas sim preocupe-se em demonstrar o seu melhor trabalho no que conseguir fazer no tempo disponível.

Se precisar de mais tempo para concluir o teste, entre em contato.

O que deve ser feito

Uma aplicação web simulando de forma simplificada a emissão de pedidos.

Você tem liberdade para escolher a linguagem de programação, a forma de persistência dos dados (SQL, NoSQL, arquivos, etc.) e os frameworks que irá utilizar no projeto. A única restrição é que a aplicação seja web. Sugerimos que escolha tecnologias com as quais tenha familiaridade.

A aplicação deve estar hospedada e acessível pela internet, pois consideramos importante que um programador saiba, pelo menos o mínimo necessário, como configurar um servidor e colocar sua aplicação no ar. Você pode utilizar qualquer serviço de hospedagem gratuito disponível na web (AWS, Heroku, Azure, PythonAnywhere, entre outros).

O usuário desta aplicação poderá criar novos pedidos e alterar os pedidos existentes. Portanto, é indispensável que estas informações sejam armazenadas de forma persistente.



Um pedido é composto pelas seguintes informações:

Cliente: O usuário deve escolher uma opção entre os clientes pré-cadastrados no sistema (tabela 1). Não é necessário criar o CRUD de clientes.

Itens: Cada item do pedido é composto pelas seguintes informações:

- Produto: o usuário deve escolher uma opção entre os produtos pré-cadastrados no sistema (tabela 2). Não é necessário criar o CRUD de produtos.
- Quantidade: a quantidade do produto deve ser um número inteiro maior que zero.
- Preço unitário: o sistema deve sugerir o preço unitário do produto, mas deve permitir que o usuário o altere (tanto para mais quanto para menos). O preço deve ter no máximo 2 casas decimais e precisa ser maior que zero.

Informações pré-cadastradas

A tabelas a seguir listam as informações utilizadas no pedido que devem ser pré-cadastradas no sistema.

Clientes

| ID | Nome |
|----|---------------------|
| 1 | Darth Vader |
| 2 | Obi-Wan Kenobi |
| 3 | Luke Skywalker |
| 4 | Imperador Palpatine |
| 5 | Han Solo |



Produtos

| ID | Nome | Preço Unitário (R\$) | Múltiplo |
|----|----------------------------|----------------------|----------|
| 1 | Millenium Falcon | 550.000,00 | |
| 2 | X-Wing | 60.000,00 | 2 |
| 3 | Super Star Destroyer | 4.570.000,00 | |
| 4 | TIE Fighter | 75.000,00 | 2 |
| 5 | Lightsaber | 6.000,00 | 5 |
| 6 | DLT-19 Heavy Blaster Rifle | 5.800,00 | |
| 7 | DL-44 Heavy Blaster Pistol | 1.500,00 | 10 |

Regras de negócio

Este sistema possui duas regras de negócio, além das validações básicas já mencionadas:

Rentabilidade

Os itens do pedido devem ser classificados em três níveis de rentabilidade, de acordo com a diferença entre o preço do item (que é informado pelo usuário) e o preço do produto (que é fixo):

Rentabilidade ótima: quando o preço usado no pedido é maior que o preço do produto. Ex: se o preço do produto é de R\$ 100, a rentabilidade será ótima se o item for vendido por R\$ 100,01 (inclusive) ou mais.



Rentabilidade boa: quando o preço do item é no máximo 10% menor que o preço do produto. Ex: se o preço do produto é de R\$ 100, a rentabilidade será boa se o item for vendido por qualquer preço entre R\$ 90 (inclusive) e R\$ 100 (inclusive).

Rentabilidade ruim: quando o preço do item é inferior ao preço do produto menos 10%. Ex: se o preço do produto é de R\$ 100, a rentabilidade será ruim se o preço for menor ou igual a R\$ 89,99.

Quando o usuário escolher o produto para inserir no pedido, o sistema deve calcular e exibir a rentabilidade na tela. Sempre que o preço for modificado, a rentabilidade deve ser recalculada e reexibida. Itens que ficarem com rentabilidade ruim não podem ser inseridos no pedido.

Múltiplo de venda

Alguns produtos só podem ser vendidos em quantidades múltiplas de um determinado número. Por exemplo, o produto X-Wing só pode ser vendido em múltiplos de 2, por exemplo, 2, 4, 6, 8, etc. Já o produto Lightsaber só pode ser vendido em múltiplos de 5, ou seja, 5, 10, 15, 20 e assim por diante. Produtos que não possuem múltiplos podem ser vendidos em qualquer quantidade.

Pontos que são importantes para nós

- **Clareza de código:** As variáveis estão com bons nomes? Quão grandes estão os métodos e classes? Quantos comentários existem no código?
- **Arquitetura:** Como ficou a estrutura do código? As responsabilidades estão bem divididas? Isolou as regras de negócio?
- **Testes automatizados:** Quais tipos de teste escreveu? Quão rápidos são os testes? Onde e porque usou mocks?
- **Workflow:** Qual o tamanho das alterações que fez no Git? Começa um projeto por qual parte? Quão fácil é instalar e rodar o projeto?



- **UI:** Seguiu o protótipo? Usou uma tecnologia moderna no Front? Sabe lidar com eventos e manipulação do estado na tela?
- **UX:** Qual a noção de UX na hora de desenvolver? As telas tem navegação clara? Como funciona a validação dos formulários?
- **Surpreenda-nos!**

Entrega

Você deve retornar por e-mail:

- O endereço do repositório no GitHub contendo o código-fonte da aplicação;
- O link da solução hospedada na internet para acessarmos;

Agora é só pegar o café ☕ e programar!

Obrigado por se candidatar!