

Olá, candidato(a)!

Primeiro gostaríamos de agradecer por seu interesse em fazer parte do time de desenvolvimento da Shopper.com.br.

Abaixo vamos descrever um teste técnico onde queremos avaliar o que você conhece sobre desenvolvimento de software.

## CENÁRIO

Em qualquer empresa de e-commerce é essencial que os usuários possam atualizar os preços de suas lojas para se manterem competitivos e manterem seus preços alinhados com os custos de operação. Essa tarefa parece simples, porém quando falamos de lojas com milhares de produtos, se torna essencial a existência de uma ferramenta que permita atualizar os produtos de forma massiva e com recursos adicionais para evitar erros que possam prejudicar o negócio.

Você foi encarregado de desenvolver essa ferramenta e após uma série de reuniões com as áreas envolvidas, os seguintes requisitos foram levantados:

- 1- O time Compras, responsável por definir os preços, se comprometeu em gerar um arquivo CSV (exemplo em anexo) contendo código do produto e o novo preço que será carregado.
- 2- O time Financeiro, preocupado com o faturamento, solicitou que o sistema impeça que o preço de venda dos produtos fique abaixo do custo deles;
- 3- O time de Marketing, preocupado com o impacto de reajustes nos clientes, solicitou que o sistema impeça qualquer reajuste maior ou menor do que 10% do preço atual do produto
- 4- Alguns produtos são vendidos em pacotes, ou seja, um produto que composto por um ou mais produtos em quantidades diferentes.  
Estabeleceu-se a regra que, ao reajustar o preço de um pacote, o mesmo arquivo deve conter os reajustes dos preços dos componentes do pacote de modo que o preço final da soma dos componentes seja igual ao preço do pacote.

Exemplos 1 -

Imagine o produto **PACK GUARANA 1L – 6 Unidades**

Ele é composto por 6 unidades do produto **GUARANA 1L**

O preço do pack é de R\$ 24,00. O preço do componente é de R\$ 4,00.

Se o arquivo do time de precificação pedir um reajuste do preço do pacote para R\$ 30,00, o mesmo arquivo deve conter o reajuste do preço do componente, no caso mudando o preço para R\$ 5,00 ( $6 \times 5 = 30$ )

Exemplos 2 -

Imaginando o produto KIT ESCOVA DE DENTE + PASTA DE DENTE, vendido a R\$ 25,00

O produto é composto por 1 unidade do produto ESCOVA DE DENTES (R\$ 10,00) e 1 unidade do produto PASTA DE DENTE (R\$ 15,00). Se o preço da ESCOVA DE DENTES for reajustado para R\$ 20,00, o arquivo também deve conter um reajuste do preço do pacote para R\$ 35,00 (R\$ 20,00 + R\$ 15,00)

A ferramenta deve impedir atualizações de preço que quebrem essa regra.

## REQUISITOS

Diante desse cenário, você deve construir um sistema com os seguintes requisitos:

- O sistema deve ter um back end (node.js), contendo as todas as regras definidas e um front-end (React.js) que será utilizado pelo usuário da ferramenta
- Você deve escrever seu código em Javascript ou TypeScript (preferencialmente)
- O banco de dados deve ser MySQL (versão 5 ou 8)
- O sistema deve permitir que o usuário carregue o arquivo de precificação
- O sistema deve ter um botão chamado VALIDAR.
- Ao clicar em VALIDAR, o sistema deve ler todo o arquivo e fazer as seguintes verificações:
  - Todos os campos necessários existem?
  - Os códigos de produtos informados existem?
  - Os preços estão preenchidos e são valores numéricos validos.?
  - O arquivo respeita as regras levantadas na seção CENARIO?
- Ao final da validação o sistema deve exibir as seguintes informações dos produtos que foram enviados
  - Código, Nome, Preço Atual, Novo Preço
- Caso uma ou mais regras de validação tenham sido quebradas, o sistema também deve exibir ao lado de cada produto qual regra foi quebrada.
- O sistema também deve ter um botão ATUALIZAR. Que só ficará habilitado se todos os produtos dos arquivos estiverem validados e sem regras quebradas
- Ao clica em ATUALIZAR, o sistema deve salvar o novo preço no banco de dados e já deixar a tela pronta para o envio de um novo arquivo.
- O preço de custo dos pacotes também deve ser atualizado como a soma dos custos dos seus componentes. Os preços de custo dos produtos que não são pacotes não deve ser atualizado

## COMO ENTREGAR SEU PROJETO

- Junto com esse teste você deve ter recebido o arquivo database.sql – Esse arquivo contém o script que define e preenche a tabela de Produtos, que contém todos os dados de produtos e a tabela que relaciona os pacotes.
- Seu projeto deve estar versionado no Github em um repositório público você deve encaminhar o link do seu projeto para o e-mail [talentos@shopper.com.br](mailto:talentos@shopper.com.br) para oficializar sua entrega. **Serão aceitas entregas até as 23h59 do dia 22/04/2024**
- Seu projeto deve ter um arquivo README.md contendo todas as instruções para instalação e execução local (passos do setup, variáveis de ambiente, etc...)
- Não há necessidade de publicar seu projeto em plataformas de host (ex. Heroku)

## ARMADILHAS QUE VOCE DEVE EVITAR

- Atenha-se ao que foi pedido nos requisitos. Não crie complicações não solicitadas (ex. uma tela de login).
- Caso queira implementar mais funcionalidades para mostrar seus conhecimentos, deixe-as para o final e só as inicie assim que todos os requisitos solicitados tenham sido 100% atendidos. Os requisitos serão contados na avaliação, as funcionalidades extras serão consideradas apenas diferenciais

Te desejamos um ótimo trabalho e, caso tenha qualquer dúvida, fique à vontade para entrar em contato conosco.