

Desafio técnico: Ecommerce - Checkout transparente

A popularização das vendas pela internet tem ganhado gradativamente mais relevância. E com isso, otimizações em usabilidade e facilidade no checkout têm ganhado bastante atenção, pois uma alta porcentagem de clientes desistem da compra na fase de checkout, seja por má estruturação ou por muita burocracia. Uma abordagem de checkout é o transparente, um recurso o qual permite que o consumidor conclua a jornada no próprio ambiente da loja virtual (diferentemente do checkout padrão e do lightbox, que transferem o usuário para o site do intermediador de pagamento através de uma nova aba ou um pop-up).

Desafio Proposto

Implementar um checkout transparente de uma loja virtual, que por meio de um carrinho de compras, seleciona-se o endereço de entrega e a transportadora (frete) e, após isso, o pagamento. Considerando que o gerenciamento de pagamentos deve ser administrado pela própria loja virtual.

Os passos para utilizar o sistema, não necessariamente separados, devem ser:

1. O cliente cria um carrinho de compras;
2. O cliente coloca produtos nesse carrinho de compras;
3. O cliente decide fazer o checkout do carrinho de compras;
4. O cliente seleciona o endereço de entrega e o frete (Obs: considerar frete fixo por cada transportadora);
5. O cliente seleciona a forma de pagamento e confirma;
6. Caso a venda seja concluída retornar mensagem de sucesso com o número do pedido e código de rastreio.

Classes básicas para operar o sistema:

1. Produto;
2. Cliente;
3. Endereco;
4. Transportadora;
5. Pagamento;
6. TipoPagamento.

Essas entidades são o mínimo esperado. Podem ser adicionados mais entidades dependendo da necessidade. Além disso, é necessário que o aplicante faça o **design** das classes internas para executar o procedimento.

Entrega Principal:

- Entregar o código que valide as entradas/saídas e regras de negócio do contexto apresentado;
- Disponibilizar alguma forma do usuário testar entradas na aplicação, não é obrigatório que seja por telas (ex: argumentos, terminal, request HTTP);
- Documentação demonstrando como utilizar as chamadas aos webs-service.

Entrega Bônus:

- Diagrama Entidade Relacionamento;
- Código limpo (bons nomes a variáveis e métodos, evitar duplicação de código, criar métodos com apenas uma responsabilidade);
- Escrita de testes unitários;
- Desenvolvimento de tela que faz entrada das informações solicitadas no contexto do negócio;
- Expor endpoints via protocolo HTTP (REST) com as funcionalidades requeridas no contexto com representação em JSON;
- Disponibilizar a aplicação containerizada utilizando o Docker;
- Funcionalidades adicionais, como cupons de desconto e vale compras.

O que esperamos que você entregue ao final do teste?

- Todo seu código versionado no repositório dentro do diretório “implementacao”;
- Toda a parte de documentação do projeto incluindo banco de dados disponibilizada em um PDF que deve estar na pasta “documentacao”. Neste arquivo deverá conter as consultas e diagramas de resposta às “competências banco de dados” e demais competências caso exista.