



Teste Técnico - Histórico de Pedidos

Um pedido, após ter seu pagamento aprovado, gera novos status que podem ser acompanhados pelo app.

Por exemplo, é comum um pedido pago com boleto ter os seguintes status: aguardando pagamento, pagamento aprovado, preparando pedido, pedido preparado, enviado à transportadora, saiu para a entrega, entregue (pedido concluído).

Exemplo de pedidos com apenas um status:

| id | createdAt | name | orderId | value |
|----|---------------------|----------------------|---------|--------|
| 1 | 2020-08-02 01:00:00 | Aguardando pagamento | 123abc | 100.00 |
| 2 | 2020-08-03 17:48:16 | Aguardando pagamento | 456def | 100.00 |
| 5 | 2020-08-04 08:55:47 | Aguardando pagamento | 789eee | 100.00 |

Exemplo de pedidos com vários status:

| id | createdAt | name | orderId | value |
|----|---------------------|----------------------|---------|--------|
| 1 | 2020-08-02 01:00:00 | Aguardando pagamento | 123-abc | 100.00 |
| 2 | 2020-08-03 17:48:16 | Aguardando pagamento | 456-def | 100.00 |
| 3 | 2020-08-05 00:11:53 | Pagamento aprovado | 456-def | 100.00 |
| 4 | 2020-08-05 10:03:00 | Preparando pedido | 456-def | 100.00 |
| 5 | 2020-08-04 08:55:47 | Aguardando pagamento | 789-eee | 100.00 |

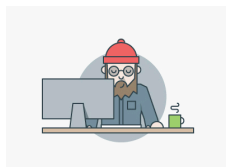
Desafios

- Agrupar status anteriores de um pedido dentro do mais recente
- Representar visualmente os status mais recentes e o histórico do pedido
- Mudar a cor de acordo com status
- Use a criatividade para criar pelo menos **uma funcionalidade de gestão de pedidos**. Exemplo: Cadastrar Pedido, Filtrar pedidos por valor/orderId/data criação...

Requisitos Técnicos

- Utilize umas destas linguagens: Nodejs, Python ou .NET Core;
- Utilize um destes framework: React, Angular ou Vue.js
- Você pode utilizar um framework de UI. Exemplo: bootstrap, material-ui, bulma, etc
- Banco de dados relacional ou não relacional;
- Testes unitários;
- Práticas do Clean Code.

Vamos lá?



Dúvidas e perguntas frequentes por e-mail