

Teste Técnico — BK Company

Desenvolvedor(a) Fullstack

■ Atenção

Leia este documento **com atenção**. Ele contém informações importantes sobre os requisitos, critérios de avaliação e instruções de entrega. Ignorar qualquer seção pode **prejudicar sua avaliação**.

Contexto do Desafio

O objetivo deste teste é avaliar sua capacidade de **traduzir requisitos de um cliente real em requisitos técnicos**, implementando uma solução fullstack funcional. Abaixo você encontrará o pedido do cliente, seguido das especificações técnicas e critérios de avaliação.

■ O Pedido do Cliente

"Olá, tudo bem?

Nós trabalhamos com vendas online e hoje não temos visibilidade nenhuma do que realmente está acontecendo no nosso negócio. A gente vende bastante, mas no fim do mês nunca sabemos exatamente quanto entrou e quanto saiu.

*Eu gostaria de ter uma tela onde eu consiga ver o faturamento de um determinado período. Tipo, eu quero selecionar uma **data inicial** e uma **data final** e ver tudo que aconteceu ali dentro.*

*Preciso visualizar a **quantidade de pedidos** que entraram, o **faturamento total**, quanto eu gastei de **custo de produto** e, principalmente, qual foi o meu **lucro** naquele período.*

*Ah, e outra coisa: eu preciso de um **botão para cadastrar o custo** de cada **produto**. Hoje a gente não tem isso em lugar nenhum e fica tudo na planilha. Quero conseguir registrar quanto cada produto me custa pra poder calcular a margem.*

*Sobre os pedidos, a gente usa uma plataforma de e-commerce que **dispara um aviso** toda vez que um **novo pedido é criado**. Esse aviso chega num formato específico da plataforma, então vocês precisariam **receber esse aviso** e transformar ele no formato de vocês.*

*Também queria poder **cadastrar os produtos** no sistema, pra ter um catálogo organizado.*

Conseguem me ajudar com isso?"

Especificação Técnica

A partir do relato do cliente, você deverá construir uma aplicação fullstack que atenda aos requisitos extraídos abaixo.

Entidades do Sistema

O sistema deve contemplar as seguintes entidades:

- **Pedido** — Representa um pedido realizado na plataforma de e-commerce.
- **Produto** — Representa um item do catálogo.
- **Custo de Produto** — Representa o valor de custo associado a um produto específico.

Requisitos do Backend

- 1. Cadastro de Produtos:** Endpoint para **cadastrar produtos** no sistema.
- 2. Cadastro de Custo de Produto:** Endpoint para **cadastrar o valor de custo** associado a um produto.
- 3. Cadastro de Pedidos via Webhook:** O sistema deve expor um endpoint para **receber webhooks** simulando a notificação de criação de um pedido. O corpo do webhook terá um **formato diferente** do modelo de domínio da aplicação, sendo necessário implementar um **mapeador (mapper)** que converta o payload externo para a entidade de Pedido do domínio.
- 4. Persistência em Memória:** Todos os dados devem ser persistidos utilizando **repositórios em memória** (in-memory repositories). Não é necessário banco de dados.
- 5. Listagem de Dados da Dashboard:** Endpoint para retornar os dados consolidados da dashboard, incluindo **quantidade de pedidos, faturamento, custo total e lucro**, com suporte a **filtragem por período**.

Requisitos do Frontend

A tela inicial deve conter **4 seções** bem definidas:

Seção 1 — Visão Geral (Dashboard)

Painel com os indicadores consolidados do período selecionado:

- **Quantidade de pedidos**
- **Faturamento total**
- **Custo total de produtos vendidos**
- **Lucro** (faturamento - custo)

Deve possuir filtros de **data inicial** e **data final** para consulta dos indicadores.

Seção 2 — Listagem de Pedidos

Tabela ou listagem exibindo os **pedidos recebidos**, permitindo visualizar os detalhes de cada pedido (itens, valores, data, etc.).

Seção 3 — Listagem de Custos de Produto

Listagem que exiba os **produtos cadastrados** junto com seus respectivos **custos**. Deve ser possível visualizar rapidamente qual custo está associado a cada produto. O custo de cada produto deve ser **editável diretamente** na listagem.

Seção 4 — Listagem de Produtos

Listagem de todos os **produtos** cadastrados no sistema, com um **botão para cadastrar** um novo produto.

Utilize qualquer **biblioteca de Design System** de sua preferência para implementar o frontend (ex: Chakra UI, Shadcn/UI, Material UI, Ant Design, etc.).

Referência de Layout

A imagem abaixo serve como **inspiração e recomendação de layout** para a tela inicial. Não é obrigatório seguir fielmente o protótipo, mas ele ilustra a disposição esperada das seções e a organização visual dos dados.

The dashboard mockup is divided into several sections:

- Dashboard:** Visão geral do seu negócio.
- Resumo Financeiro:** Mostra Lucro (R\$ 147.550,00), Faturamento (R\$ 245.890,00), Custo Total (R\$ 98.340,00) e Total de Pedidos (1.248).
- Pedidos Recentes:** Últimos pedidos recebidos via webhook, com uma lista de 6 itens.
- Produtos:** Lista de produtos com opção de adicionar novo. Itens incluem Camiseta Básica, Calça Jeans, Tênis Esportivo, Jaqueta Corta-Vento, entre outros.
- Custos de Produto:** Tabela de custos para os mesmos produtos, com colunas para Produto, Custo (ex: R\$ 18,00, R\$ 52,00, etc.) e botões de edição.

Formato do Webhook

O webhook recebido da plataforma de e-commerce terá um corpo igual ao corpo abaixo. Observe que o formato **não corresponde diretamente** às entidades do seu domínio — o mapeamento é parte do desafio.

```
{
  "id": "ORD-98432",
  "buyer": {
    "buyerName": "Maria Souza",
    "buyerEmail": "maria@email.com"
  },
  "lineItems": [
    {
      "itemId": "P-001",
      "itemName": "Camiseta Básica",
      "qty": 2,
      "unitPrice": 49.90
    },
    {
      "itemId": "P-002",
      "itemName": "Calça Jeans",
      "qty": 1,
      "unitPrice": 129.90
    }
  ],
  "totalAmount": 229.70,
  "createdAt": "2025-02-10T14:32:00Z"
}
```

Stack Recomendada

Camada	Tecnologias
Frontend	React
Backend	Express + Zod ou NestJS + Class Validator
Runtime	Node.js ou Bun

Utilize as **bibliotecas adicionais** que bem entender para chegar a um resultado de melhor qualidade. O conhecimento de ferramentas externas demonstra **conhecimento prático**.

Princípios e Boas Práticas

Utilize princípios do **SOLID** sempre que necessário, em especial:

- **Open/Closed Principle** — O sistema deve ser aberto para extensão e fechado para modificação.
- **Dependency Inversion Principle** — Módulos de alto nível não devem depender de módulos de baixo nível; ambos devem depender de abstrações.

A implementação dos **repositórios em memória** deve seguir uma abstração (interface/contrato) que permita futura substituição por um repositório real sem alterar a lógica de negócios.

O **mapeador** do webhook deve estar isolado da lógica de domínio, permitindo que novos formatos de plataforma sejam adicionados sem impactar o restante do sistema.

■■ Aviso Importante

Com o avanço da inteligência artificial, reconhecemos que seu uso como ferramenta de auxílio na construção de software é uma realidade. No entanto, os principais critérios de avaliação serão a **organização, extração, modularização, componentização, encapsulamento e abstração** do código. A capacidade de produzir uma implementação bem estruturada e aderente a **boas práticas** será o fator decisivo na avaliação.

Entregáveis

1. **Repositório público** no GitHub (ou equivalente) com o código-fonte da aplicação.
2. **README** com instruções claras para execução do projeto.
3. **Vídeo demonstrativo** (máximo 3 minutos, sem necessidade de falar) apresentando:
 - Overview da arquitetura do projeto, estrutura de pastas, arquivos, componentes.
 - Demonstração do sistema funcionando.

Instruções de Entrega

- O link para o **repositório público** deve ser enviado como resposta deste e-mail até o **horário de finalização**.
- O projeto deve ser capaz de ser **executado na máquina de teste** sem necessidade de configurações adicionais complexas.
- O vídeo pode ser disponibilizado via link (YouTube não listado, Loom, Google Drive, etc.) incluído no README ou no corpo do e-mail.

Boa sorte! Estamos ansiosos para ver sua solução. ■