

...SOFTWARE ARCHITECTURE

FASE 01

**TECH
CHALLENGE**

O PROBLEMA

VALE NOTA

Há uma lanchonete de bairro que está expandindo devido seu grande sucesso. Porém, com a expansão e sem um sistema de controle de pedidos, o atendimento aos clientes pode ser caótico e confuso. Por exemplo, imagine que um cliente faça um pedido complexo, como um hambúrguer personalizado com ingredientes específicos, acompanhado de batatas fritas e uma bebida. O atendente pode anotar o pedido em um papel e entregá-lo à cozinha, mas não há garantia de que o pedido será preparado corretamente.

Sem um sistema de controle de pedidos, pode haver confusão entre os atendentes e a cozinha, resultando em atrasos na preparação e entrega dos pedidos. Os pedidos podem ser perdidos, mal interpretados ou esquecidos, levando à insatisfação dos clientes e a perda de negócios.

Em resumo, um sistema de controle de pedidos é essencial para garantir que a lanchonete possa atender os clientes de maneira eficiente, gerenciando seus pedidos e estoques de forma adequada. Sem ele, expandir a lanchonete pode acabar não dando certo, resultando em clientes insatisfeitos e impactando os negócios de forma negativa.

Para solucionar o problema, a lanchonete irá investir em um sistema de autoaten-

dimento de

fast food, que é composto por uma série de dispositivos e interfaces que permitem aos clientes

selecionar e fazer pedidos sem precisar interagir com um atendente, com as seguintes

funcionalidades:

PEDIDO

- Os clientes são apresentados a uma interface de seleção na qual podem optar por se identificar via CPF, se cadastrarem com nome, e-mail ou não se identificar, podendo montar o combo na seguinte sequência, sendo todas elas opcionais:

1. Lanche
2. Acompanhamento
3. Bebida

- Em cada etapa é exibido o nome, descrição e preço de cada produto.

PAGAMENTO

O sistema deverá possuir uma opção de pagamento integrada para MVP. A forma de

pagamento oferecida será via QRCode do Mercado Pago.

ACOMPANHAMENTO

Uma vez que o pedido é confirmado e pago, ele é enviado para a cozinha para ser preparado.

Simultaneamente deve aparecer em um monitor para o cliente acompanhar o progresso do seu

pedido com as seguintes etapas:

- Recebido
- Em preparação
- Pronto
- Finalizado

ENTREGA

Quando o pedido estiver pronto, o sistema deverá notificar o cliente que ele está pronto para

retirada. Ao ser retirado, o pedido deve ser atualizado para o status finalizado.

Além das etapas do cliente, o estabelecimento precisa de um acesso administrativo:

- Gerenciar clientes: Com a identificação dos clientes o estabelecimento pode trabalhar em campanhas promocionais.
- Gerenciar produtos e categorias: Os produtos dispostos para escolha do cliente serão gerenciados pelo estabelecimento, definindo nome, categoria, preço, descrição e imagens.

Para esse sistema teremos categorias fixas:

- a. Lanche
- b. Acompanhamento
- c. Bebida
- d. Sobremesa

- Acompanhamento de pedidos: Deve ser possível acompanhar os pedidos em andamento e tempo de espera de cada pedido.

As informações dispostas no sistema de pedidos precisarão ser gerenciadas pelo estabelecimento através de um painel administrativo.

ENTREGA FASE 1

1. Documentação do sistema (DDD) utilizando a linguagem ubíqua, dos seguintes fluxos:
 - a. Realização do pedido e pagamento
 - b. Preparação e entrega do pedido

2. Uma aplicação para todo sistema de backend (monolito) que deverá ser desenvolvido

seguindo os padrões apresentados nas aulas:

a. Utilizando arquitetura hexagonal

b. APIs:

- Cadastro do Cliente
- Identificação do Cliente via CPF
- Criar, editar e remover de produto
- Buscar produtos por categoria
- Fake checkout, apenas enviar os produtos escolhidos para a fila
- Listar os pedidos

c. Aplicação deverá ser escalável para atender grandes volumes nos horários de pico

d. Banco de dados a sua escolha

- Inicialmente deveremos trabalhar e organizar a fila dos pedidos apenas em banco de dados

3. A aplicação deve ser entregue com um Dockerfile configurado para executá-la corretamente.

Para validação da POC, temos a seguinte limitação de infraestrutura:

a. 1 instância para banco de dados

b. 1 instâncias para executar aplicação

Não será necessário o desenvolvimento de interfaces para o frontend, o foco deve ser total no backend.