THIAGO ADRIANO

PÓS TECH

SOFTWARE ARCHITECTURE

TECH CHALLENGE

FASE 01

Tech Challenge

Tech Challenge é o projeto da fase que englobará os conhecimentos obtidos em todas as disciplinas da fase. Esta é uma atividade que, em princípio, deve ser desenvolvida em grupo. Importante atentar-se ao prazo de entrega, pois trata-se de uma atividade obrigatória, uma vez que vale 90% da nota de todas as disciplinas da fase.

O problema

Há uma lanchonete de bairro que está se 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 autoatendimento 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 cadastrar com nome e e-mail ou não se identificar, podendo montar o combo na seguinte sequência, sendo todas elas opcionais:

1. Lanche
2. Acompanhamento
3. Bebida
4. Sobremesa

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:

- Lanche

- Acompanhamento
- Bebida
- 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.

Entregáveis FASE 1:

1. Documentação do sistema (DDD) com Event Storming, incluindo todos os passos/tipos de diagrama mostrados na aula 6 do módulo de DDD, e utilizando a linguagem ubíqua, dos seguintes fluxos:
    a. Realização do pedido e pagamento;
    b. Preparação e entrega do pedido.

    É importante que os desenhos sigam os padrões utilizados na explicação.
2. Uma aplicação para todo o sistema de backend (monolito) que deverá ser desenvolvido seguindo os padrões apresentados nas aulas:
    a. Utilizando arquitetura hexagonal
    b. APIs:
        i. Cadastro do Cliente;
        ii. Identificação do Cliente via CPF;
        iii. Criar, editar e remover produtos;
        iv. Buscar produtos por categoria;
        v. Fake checkout, apenas enviar os produtos escolhidos para a fila. O checkout é a finalização do pedido;
        vi. Listar os pedidos.

        Disponibilizar também o Swagger para consumo dessas APIs
    c. Banco de dados à sua escolha

    i. 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, e um docker-compose.yml para subir o ambiente completo. É muito importante seguir boas práticas de segurança e melhorias para que a inicialização seja feita de forma rápida.

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

- 1 instância para banco de dados;
- 1 instâncias para executar aplicação;
- Obrigatório utilizar no mínimo um Dockerfile.

4. Link para vídeo demonstrando a arquitetura desenvolvida localmente.
    a. O vídeo deve ser postado no Youtube, Vimeo ou compartilhado via Google Drive/One Drive.
    b. Não esqueça de deixá-lo público ou não listado.
    c. O vídeo de demonstração sobre a execução da aplicação e banco através do Docker Compose, deve conter informações relevantes sobre o projeto e quais passos são necessários para que a aplicação funcione.

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

Para validação do código-fonte da aplicação, precisamos que vocês deixem o projeto como privado e adicionem o usuário **soat-architecture**.

Referente aos arquivos Dockerfile e docker-compose.yml, eles devem estar em um repositório no github, gitlab ou serviço semelhante, e com um arquivo README.md na raiz do projeto, propriamente formatado, indicando o que é o projeto, quais os objetivos e passos necessários para iniciar o projeto localmente.

Para a documentação pode ser usado o Miro ou qualquer outra ferramenta que conheçam. Se usar o Miro ou outra ferramenta disponível via web, deixar aberto ou compartilhar com os(as) professores(as) (mais sobre isso abaixo). Se decidirem enviar os diagramas, coloquem todos no mesmo arquivo compactado e suba o arquivo compactado.

A entrega deve ser feita pelo Portal do Aluno, em um arquivo com o nome do grupo, os nomes de todos(as) os(as) participantes do grupo, seus nomes de usuário no Discord e os links de acesso ao repositório e à documentação. Caso disponibilizem os diagramas em um arquivo, subir separadamente a este arquivo. O formato do arquivo deve ser, preferencialmente, PDF ou texto simples.

O nome de usuário dos participantes do grupo é importante para que a pessoa que irá corrigir consiga entrar em contato em caso de qualquer problema.


PDS TECH