

JOSÉ RUBENS RODRIGUES E DOUGLAS LIMA

POSTECH

SOFTWARE ARCHITECTURE

TECH CHALLENGE

# FASE 03

---

## **Tech Challenge**

O 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 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 identificarem 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:

- 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 3:**

Dando continuidade ao desenvolvimento do software para a lanchonete, teremos as seguintes melhorias e alterações:

1. Implementar um API Gateway para receber a solicitação externa e uma function serverless para autenticar/consultar o cliente com base no CPF.
  - a. Integrar/Consultar o sistema de autenticação (AD, Cognito) para identificar o cliente.
  - b. O cliente não precisa inserir qualquer tipo de senha para se identificar, somente o CPF.
  - c. O fluxo de integração/consulta pode utilizar JWT ou equivalente.
2. Implementar as melhores práticas de CI/CD para a aplicação, segregando os códigos em repositórios, por exemplo:
  - a. 1 repositório para o Lambda.
  - b. 1 repositório para sua infra Kubernetes com Terraform.
  - c. 1 repositório para sua infra banco de dados gerenciáveis com Terraform.
  - d. 1 repositório para sua aplicação que é executada no Kubernetes.
3. Os repositórios devem fazer deploy automatizado na conta da nuvem utilizando actions.

- a. As branches main/master devem ser protegidas, não permitindo commits direto, utilizando sempre o Pull Request para realizar o Merge, e posteriormente o Deploy.
  - b. Deve seguir boas práticas de segurança utilizando Secrets para valores sensíveis.
  - c. Toda ação que realizar criações/edições na cloud, devem ser feitas com Terraform e automatizada com CI/CD.
4. Melhorar a estrutura do banco de dados escolhido, documentar seguindo os padrões de modelagem de dados e justificar a escolha do banco de dados.
5. Você tem a liberdade para escolher qual Cloud desejar, mas terá de utilizar os serviços serverless: functions (AWS Lambda, Azure functions ou Google Functions, por exemplo), banco de dados gerenciáveis (AWS RDS, Banco de Dados do Azure, Cloud SQL no GCP etc), sistema de autenticação (AWS Cognito, Microsoft AD ou Google Identity platform no GCP, por exemplo).
6. Para validação do código-fonte da aplicação, precisamos que vocês deixem o projeto como privado e adicione o usuário **soat-architecture** para que possamos ver o projeto.
7. Link para vídeo demonstrando a arquitetura desenvolvida em cloud e as execuções de Pipelines CI/CD.
- 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. Não é necessário mostrar as pipelines em execução, somente os checks (verdes) indicando que todos os passos foram executados corretamente.
  - d. O vídeo deve exibir os serviços criados na Cloud escolhida e em pleno funcionamento, além de explicar qual a função e escolha de tal serviço/arquitetura na Cloud.

- e. Depois de gravar o vídeo, a infraestrutura criada poderá ser excluída, evitando gastos desnecessários. Nenhum professor ou professora vai testar o que foi criado, somente o vídeo evidenciando.

**Os artefatos de entrega são:**

**PDF** ou **TXT** com o link do repositório público ou com acesso dos docentes. Esse repositório deve conter todos os códigos (inclusive o Terraform e CI/CD actions), além de todos requisitos descritos anteriormente, como branch protegida e Pull Request fechados. Também deverá conter a URL do vídeo descrito no item 7.



POSTECH