

# Pits A



Recentemente, diversas empresas do ramo alimentício têm se desvinculado dos grandes aplicativos de *delivery*. As causas dessa tendência são diversas e vão desde a transformação no modo de operação de cada estabelecimento, até às taxas abusivas das grandes plataformas.

Porém, em 2023, simplesmente não é viável voltar ao modo de trabalho “pré-Ifood”... Foi por isso que a pizzeria Pits A decidiu desenvolver seu próprio aplicativo de *delivery*. E adivinha só... vocês foram escolhidos para ajudar!

## User Stories

- US1** Eu, enquanto administrador do sistema, quero utilizar o sistema para **criar, editar e remover um estabelecimento**.
- Um estabelecimento deverá possuir um **código de acesso** ao sistema (com 6 dígitos).
  - O **código de acesso** deve ser informado sempre que se faz alguma operação enquanto estabelecimento. Se o código de acesso não for informado ou estiver incorreto, a operação irá obrigatoriamente falhar. Não há limite para o número de operações com inserção de código incorreto.
- US2** Eu, enquanto cliente, quero utilizar o sistema para me **cadastrar como cliente** do sistema. Mais detalhadamente, deve ser possível criar, ler, editar e remover clientes.
- Um(a) cliente é criado(a) a partir de seu **nome completo**, seu **endereço principal** e de seu **código de acesso** ao sistema (com 6 dígitos).
  - Nas operações de leitura de clientes, os **códigos de acesso não devem ser exibidos**.
  - Para realizar alguma operação sobre um(a) cliente, é necessário informar seu **código de acesso** (exceto leituras). Se o código de acesso não for informado ou estiver incorreto, a operação irá obrigatoriamente falhar.
  - **Apenas o cliente poderá editar ou excluir seu próprio cadastro**. Contudo, os

estabelecimentos poderão realizar operações de leitura sobre os clientes.

### US3

Eu, enquanto funcionário(a) terceirizado(a), quero utilizar o sistema para me **cadastrar como entregador(a)** do sistema. Mais detalhadamente, deve ser possível criar, ler, editar e remover entregadores.

- Um(a) entregador(a) é criado(a) a partir de seu **nome completo, placa do veículo, tipo do veículo (moto ou carro), cor do veículo** e de seu **código de acesso** (com 6 dígitos).
- Nas operações de leitura de entregadores, os **códigos de acesso não devem ser exibidos**.
- Para realizar alguma operação sobre um(a) entregador(a), é necessário informar seu **código de acesso** (exceto leituras). Contudo, o estabelecimento pode remover entregadores ou aprová-los/recusá-los e, nesses casos, é necessário informar o código de acesso do próprio estabelecimento. Se o código de acesso não for informado ou estiver incorreto, a operação irá obrigatoriamente falhar.

### US4

Eu, enquanto funcionário(a) terceirizado(a), quero utilizar o sistema para me **associar como entregador(a)** de um estabelecimento.

- Um(a) entregador(a) recém-associado(a) estará **sob análise** e só poderá fazer entregas após ser aprovado(a) pelo estabelecimento. Note que **os entregadores não podem mudar esse status por si mesmos**.
- Para se associar como entregador(a), o funcionário(a) terceirizado(a) precisará apresentar seu **próprio código de acesso**.

### US5

Eu, enquanto estabelecimento, quero utilizar o sistema para **aprovar ou rejeitar entregadores** do estabelecimento.

- Apenas entregadores que tenham sido **aprovados previamente** poderão fazer entregas para clientes.
- Para aprovar ou rejeitar o(a) entregador(a), o estabelecimento precisará apresentar seu **próprio código de acesso**.

### US6

Eu, enquanto estabelecimento, quero utilizar o sistema para **gerir os sabores de pizza** vendidos pelo estabelecimento. Mais detalhadamente, deve ser possível criar, ler, editar e remover sabores.

- Um sabor é criado a partir de seu **nome**, de seu **tipo** (salgado ou doce) e de seu **valor** (em reais) para os dois tamanhos de pizza disponíveis: média e grande.
- Ao ser criado, um sabor estará automaticamente **disponível** para os pedidos.

### US7

Eu, enquanto cliente, quero **visualizar o cardápio** de um estabelecimento.

- Nesse cardápio, devem estar listados **TODOS os sabores de pizza** do estabelecimento. Para cada sabor, devem estar apresentados o **nome** e o **valor** de cada tamanho de pizza (média e grande).
- Também deve ser possível ver o cardápio **por tipo de sabor**. Isto é, acessar um

cardápio somente dos sabores doces ou somente dos sabores salgados.

## US8

Eu, enquanto cliente, quero utilizar o sistema para **fazer pedidos de pizza** a um estabelecimento. Mais detalhadamente, deve ser possível criar, ler, editar e remover pedidos.

- Para fazer um pedido, o(a) cliente deverá **listar as pizzas que deseja comprar** e informar a **pizza escolhida**, o **endereço de entrega** (opcional) e o **código de acesso** do cliente.
- Se o **endereço de entrega** não for informado, o pedido deverá ser entregue no endereço principal do(a) cliente que fez o pedido.
- **Todas as pizzas possuem, pelo menos, um sabor.** Porém, as pizzas grandes podem ter até dois sabores. Note que, se alguma das pizzas de um pedido possuir um sabor não cadastrado, o pedido não será criado.
- O **valor de uma pizza** é calculado automaticamente pelo sistema com base no valor de seus sabores. Assim, se uma pizza possui dois sabores (com valores  $X$  e  $Y$ ), essa pizza custará  $(X+Y) / 2$ .
- O **valor total do pedido** é calculado automaticamente pelo sistema como o somatório do valor das pizzas do pedido. Esse custo total deve ser registrado no pedido.
- Após criar o pedido, o(a) cliente precisará **confirmar o pagamento** para que o estabelecimento comece a prepará-lo. Para isso, o(a) cliente deverá informar qual o pedido, qual o método de pagamento e qual o seu código de acesso.
- Para maior segurança dos entregadores, só são permitidos **pagamentos** via cartão de crédito, cartão de débito ou Pix.
- As operações sobre um pedido específico só poderão ser feitas pelo estabelecimento ou pelo(a) cliente que o criou. Em ambos os casos, é necessário o uso do **código de acesso** que, quando não for informado ou estiver incorreto, fará a operação obrigatoriamente falhar.

## US9

Eu, enquanto estabelecimento, quero **modificar a disponibilidade dos sabores** do cardápio. Mais detalhadamente, deve ser possível visualizar e editar a disponibilidade dos sabores de pizza — dado que, nem sempre, todos os produtos estão disponíveis.

- Se um sabor de pizza estiver indisponível, deve haver uma **indicação de indisponibilidade no cardápio**.
- No cardápio, os sabores indisponíveis devem ser **exibidos no final**.
- Quando um sabor voltar a estar disponível, os clientes que tiverem interesse nesse sabor **devem ser notificados**. A notificação deve ser representada como uma mensagem no terminal da aplicação (print), indicando o motivo e quem está recebendo a notificação.
- Sempre que o sabor se torna disponível, as notificações são disparadas apenas uma **vez**.

## US10

Eu, enquanto cliente, quero **demonstrar interesse em sabores** de pizza que não estão disponíveis no momento.

- Os clientes devem ser capazes de demonstrar interesse apenas por sabores que se

encontram **indisponíveis**.

**US11** Eu, enquanto estabelecimento, quero **disponibilizar diferentes meios de pagamento** para os pedidos, tal que cada meio de pagamento também gere descontos distintos.

- Os **pagamentos por cartão de crédito** não recebem nenhum desconto.
- Os **pagamentos por cartão de débito** recebem 2,5% de desconto sobre o valor total do pedido.
- Os **pagamentos por Pix** recebem 5% de desconto sobre o valor total do pedido.

**US12** Eu, enquanto estabelecimento, quero que o sistema garanta a **corretude nas mudanças de status dos pedidos**.

- Um pedido começa com o status "*Pedido recebido*" e, posteriormente, poderá ser modificado para "*Pedido em preparo*". **Essa mudança ocorre após a confirmação do pedido pelo cliente.**
- Um pedido com o status "*Pedido em preparo*" poderá ser modificado para "*Pedido pronto*". **Essa mudança ocorre quando o funcionário do estabelecimento indica o término do preparo.**
- Um pedido com o status "*Pedido pronto*" poderá ser modificado para "*Pedido em rota*". **Essa mudança ocorre quando a entrega do pedido é atribuída a um entregador.**
- Um pedido com o status "*Pedido em rota*" poderá ser modificado para "*Pedido entregue*". **Essa mudança ocorre após a confirmação de entrega do pedido pelo cliente.**

**US13** Eu, enquanto cliente, quero ser **notificado(a) quando meus pedidos estiverem em rota** e, por medidas de segurança, quero ser **informado(a) com o nome do(a) entregador(a)** responsável pela entrega e os **detalhes sobre seu veículo**. A notificação deve ser representada como uma mensagem no terminal da aplicação (print), indicando o motivo e quem está recebendo a notificação.

**US14** Eu, enquanto cliente, quero ser responsável por **confirmar a entrega** dos meus pedidos.

- O cliente será responsável por mudar o status de seus pedidos para "*Pedido entregue*".

**US15** Eu, enquanto estabelecimento, quero ser **notificado(a)** sempre que o status de um pedido for modificado para "*Pedido entregue*". A notificação deve ser representada como uma mensagem no terminal da aplicação (print), indicando o motivo e quem está recebendo a notificação.

**US16** Eu, enquanto cliente, quero ter a possibilidade de **cancelar um pedido** que fiz no estabelecimento.

- Um pedido só pode ser cancelado se **não tiver atingido** o status de "*Pedido pronto*".
- Um pedido só pode ser cancelado pelo **cliente que o fez** e, quando cancelado, deve ser completamente **excluído** do sistema.

**US17** Eu, enquanto cliente, quero poder **verificar os pedidos que já realizei** no estabelecimento.

- O(a) cliente poderá visualizar **um pedido específico**, desde que possua o identificador

único deste pedido e, sobretudo, que esse pedido seja seu.

- O(a) cliente poderá visualizar o seu próprio **histórico de pedidos**. Nesse caso, a listagem dos pedidos deve estar ordenada de modo que pedidos ainda não entregues e/ou mais recentes estejam no início.
- O(a) cliente também poderá aplicar **filtragem por status** sobre o seu próprio histórico de pedidos.

**US18** Eu, enquanto funcionário(a) terceirizado(a), desejo definir se estou **disponível (ou não) para realizar as entregas** do estabelecimento.

- Apenas o(a) entregador(a) pode definir sua **própria disponibilidade** (em atividade ou em decanso).
- Ao ser **aprovado(a) por um estabelecimento** como entregador(a), o(a) funcionário(a) terceirizado(a) fica inicialmente indisponível para realizar entregas (“Descanso”). Apenas após ficar em atividade (“Ativo”) o(a) entregador(a) poderá ser atribuído(a) a uma entrega.

**US19** Eu, enquanto funcionário(a), gostaria que o sistema **atribuísse automaticamente as entregas dos pedidos** com status “*Pedido Pronto*” a um(a) entregador(a) que esteja disponível para realizar entregas.

- Como é o estabelecimento que prepara o pedido, também será sua responsabilidade **informar quando já estiver pronto** e puder ser enviado para o(a) cliente.
- A atribuição do pedido a um(a) entregador(a) deve, também, **atualizar o status do pedido**.
- Buscando equilibrar os ganhos dos entregadores, a atribuição das entregas deve ser feita de modo a **priorizar que os entregadores que estejam aguardando por mais tempo**.
- Quando um pedido está pronto, mas não há entregadores disponíveis, esse pedido deve ser **alocado a um(a) entregador(a) tão logo haja algum(a) disponível**.

**US20** Eu, enquanto cliente, quero ser **notificado(a) quando meu pedido não puder ser atribuído para entrega** devido à indisponibilidade de entregadores. A notificação deve ser representada como uma mensagem no terminal da aplicação (print), indicando o motivo e quem está recebendo a notificação

## Grupos

- Os grupos devem ser formados por **até 5 pessoas**.
- É obrigatória a colaboração via **GitHub**, sendo apenas um repositório por grupo. A lista dos *commits* no repositório será utilizada durante a avaliação da participação de cada integrante.
- Todo o código-fonte e testes desenvolvidos devem estar disponíveis no repositório do GitHub, que deve ser **gerado a partir do [link](#) do github classroom**. Repositórios pessoais não serão

considerados para a avaliação.

## Formato das Entregas

- Link para o repositório no GitHub com todo o **código-fonte** e **testes automáticos** desenvolvidos.
- Modelagem em diagrama de classe com os relacionamentos entre as principais classes do sistema (não considerar classes de controller, service ou repository).
- Um documento descrevendo as **user stories** implementadas no projeto, a **participação de cada integrante** do grupo nessa implementação (tanto o percentual quanto a descrição da participação, ou seja, informar quem desenvolveu cada US) e, por fim, as **principais decisões de design** tomadas durante seu desenvolvimento (como estilos arquiteturais e/ou padrões de projeto adotados).
- **Apresentação do projeto desenvolvido.** As apresentações serão presenciais (em sala de aula), em grupo e ocorrerão nos dias 07/11/2023 e 10/11/2023 (a ordem de apresentação será sorteada).

## Avaliação

- Serão avaliadas a **quantidade**, a **complexidade** e a **corretude** das *user stories* implementadas.
- Serão avaliadas as **decisões de design** tomadas e implementadas durante o desenvolvimento. Por exemplo, o uso adequado de arquiteturas e padrões de projeto.
- Será avaliada a **qualidade do código** desenvolvido. Por exemplo, o uso de princípios de projeto, a atribuição de responsabilidades e a presença de *bad smells* no código.
- Serão avaliadas a **cobertura** e a **qualidade dos testes** automáticos desenvolvidos, abordagem funcional, com a correta utilização do MockMvc aplicados aos *endpoints* no formato REST para todos os membros da API disponibilizada.

## Estrutura do sistema

- O sistema web a ser desenvolvido deverá utilizar o **framework Spring Boot** e, assim como o sistema Mercado Fácil, deve seguir os **padrões controller, service e repository**.
- O código-fonte deve estar **organizado em pacotes**, de modo que se facilite a diferenciação

das classes e de suas respectivas responsabilidades.

- Apesar de **não possuir um *frontend web***, todas as funcionalidades do sistema devem ser acessíveis via API REST e estar **documentadas através do [Swagger](#)**.
- A **cobertura dos testes** para as classes de serviço e controladores deve ser de 100% (cem por cento), a fim de que fique evidenciada a prática de TDD.

## Calendário de Entregas

A realização do projeto será realizada em quatro entregas (milestones) seguindo o seguinte cronograma de entregas:

- 1ª Etapa: USs 01-07 (18/09/2023) - vale 10% da nota do projeto
- 2ª Etapa: USs 08-11 (09/10/2023) - vale 20% da nota do projeto
- 3ª Etapa: USs 12-17 (23/10/2023) - vale 30% da nota do projeto
- 4ª Etapa: USs 18-20 (06/11/2023) - vale 40% da nota do projeto

## Bônus (0,5 ponto-extra)

Pela substituição **de todas as notificações** na console padrão (print) por:

- Envio de notificação por E-mail; ou
- Envio de notificação por SMS; ou
- Envio de notificação por WhatsApp.

**Observação:** este bônus será aplicado **exclusivamente à nota do projeto**, a todos os membros da equipe, caso corretamente implementado.