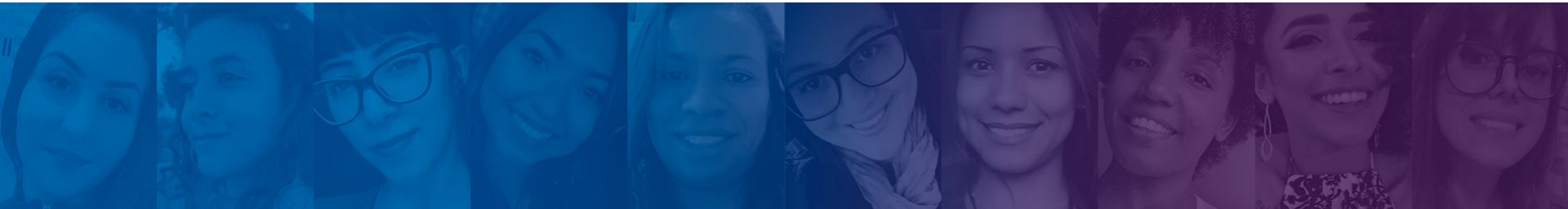


Luiza <Code> 3a Edição

DESAFIO FINAL

Como vai funcionar?

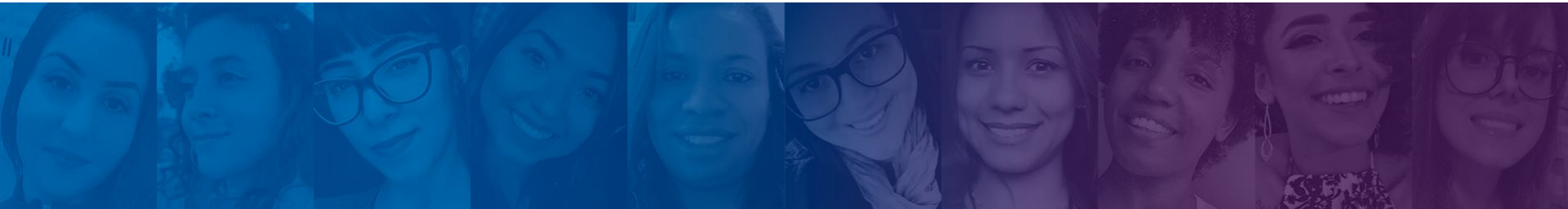
- Data da Banca Final: 13/09 2a feira (19h às 22h) e 14/09 3a feira (19h às 22h)
- Cada turma será dividida em grupos de 5 alunas
- Tempo: 10 min para a apresentação, seguidos por 5 min de perguntas da Banca avaliadora
- Apenas uma pessoa do grupo, a porta-voz, irá compartilhar a tela para apresentar o trabalho. As demais também devem contribuir ao longo da apresentação
- Mais orientações neste [Guia](#)



Desafio Final - Omni Channel

Uma das funcionalidades mais interessantes em um e-commerce é a *Omni Channel*. Ou seja, não há mais diferenças entre lojas físicas ou na web. O consumidor pode procurar o produto em uma loja física e comprar pela internet ou vice-versa.

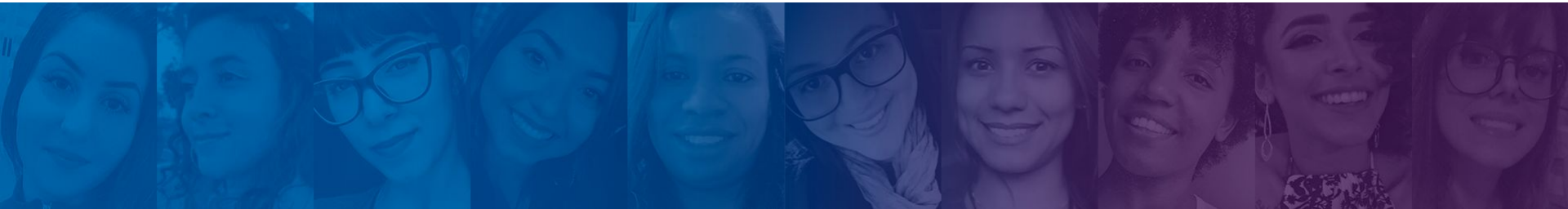
Basicamente, o cliente Omni Channel é aquele que utiliza de todos os canais simultaneamente, cruzando compras em ambientes físicos e online.



O que deve ser feito?

O objetivo é que você **desenvolva um serviço HTTP resolvendo a funcionalidade de Omni Channel do cliente**. Esse serviço deve atender os seguintes requisitos:

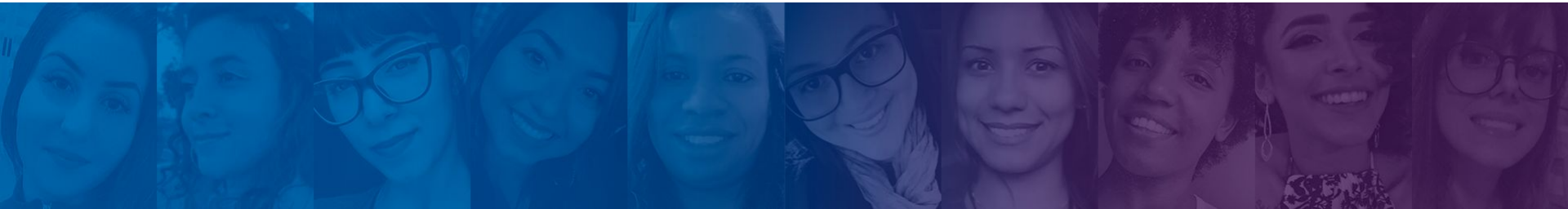
- Adicionar um produto na lista da compra da cliente;
- Remover um produto da lista da compra da cliente;
- Consultar todos os produtos disponíveis;
- Consultar a lista de compras da cliente;



Orientações

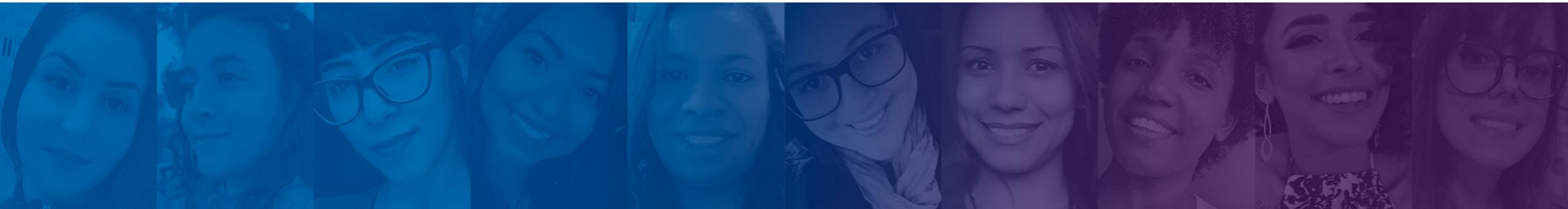
Imagine que esse serviço fará parte de uma plataforma construída em uma arquitetura baseada em micro-serviços.

É importante estabelecer alguns limites para proteger o bom funcionamento do ecossistema, dessa forma o cliente só pode comprar um produto de cada tipo. Após realizar a compra o status dessa compra é 'Realizada' e após a retirada do produto na loja física passa a ser 'Retirado'



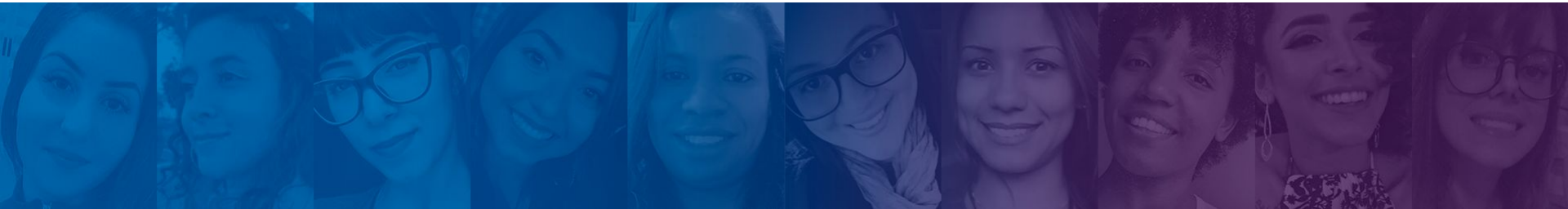
Entregas obrigatórias

- README do projeto explicando como configurar o ambiente e rodar o projeto.
- SWAGGER com acesso aos endpoints
- API contendo os seguintes endpoints:
 - Listar produtos
 - Listar lojas físicas
 - Cadastrar cliente
 - Adicionar um produto na lista de compra da cliente;
 - Remover um produto da lista de compra da cliente;
 - Finalizar compra
 - Consultar todas as compras realizadas da cliente;



Requisitos não-funcionais

- Performance (Volume de milhões de transações por dia)
- Testes automatizados, integração e unitários.
- Logs / Rastreabilidade
- Tratamento de Exceção
- Front-end da aplicação
- Controle de concorrência na persistência dos objetos
- Autenticação / Login da API



Algumas dicas

- Testes são excelentes para descrever como o código funciona!
- Organizar a estrutura do código, classes e pacotes, também é relevante, utilizamos clean architecture em nossos serviços.
- Fique a vontade para usar containers, caso faça sentido;

