

# Desafio 1:

## User Story: Pré-cadastro de clientes

Como área de Comercialização da Cielo, desejo manter um pré-cadastro de clientes (prospect) para possibilitar uma futura oferta de produtos e serviços a esses clientes.

### Regras:

#### 1) Informações do cadastro:

##### a) Se Pessoa Jurídica:

- CNPJ
  - número de 14 dígitos formatado com zeros à esquerda
- Razão Social
  - máximo de 50 caracteres
- MCC - “*Merchant Category Code*”
  - número com no máximo 4 caracteres
- CPF do contato do estabelecimento
  - número de 11 dígitos formatado com zeros à esquerda
- Nome do contato do estabelecimento
  - máximo de 50 caracteres
- Email do contato do estabelecimento
  - expressão regular para validação:  
`"^([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})$"`

##### b) Se Pessoa Física:

- CPF da pessoa
  - número de 11 dígitos formatado com zeros à esquerda
- MCC – “*Merchant Category Code*”
  - número com no máximo 4 caracteres
- Nome da pessoa
  - máximo de 50 caracteres
- Email da pessoa
  - expressão regular para validação:  
`"^([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})$"`

\* Todas as informações são obrigatórias

## 2) Consistências:

a) A operação de **cadastrar** cliente deverá validar se o cadastro não existe.

Se o cadastro **já** existir, o sistema deverá retornar um status coerente informando que o cliente já está cadastrado e não realizar qualquer alteração nos dados existentes.

b) A operação de **alterar** cliente deverá validar se o cadastro já existe.

Se o cadastro **não** existir, o sistema deverá retornar um status coerente informando que o cliente ainda não está cadastrado e não deverá realizar a inclusão de um novo registro.

c) Ambas as operações de **cadastrar** ou **alterar** cliente deverão validar se todos os dados foram informados, se estão consistentes conforme tamanhos, tipos de dados e formatações disponibilizadas na regra “1”.

Em caso de qualquer inconsistência, o sistema deverá retornar um status coerente informando os detalhes do erro.

d) A operação de **consultar** um cliente deverá validar se o cadastro já existe.

Se o cadastro **não** existir, o sistema deverá retornar um status coerente informando que o cliente ainda não está cadastrado.

## Desafio:

a) modelar uma API REST com operações que possibilitem a **criação, alteração, exclusão e consulta** de pré-cadastros de clientes. O entregável deverá ser um documento **swagger**.

b) implementar na linguagem java utilizando o framework spring boot as APIs modeladas no item 1. Os dados podem ser armazenados em memória.

c) Desenvolver teste unitário para as regras descritas

## Desafio 2:

### User Story: Fila de atendimento

Como área de Comercialização da Cielo, desejo ter uma fila de atendimento aos prospect, para que cada cliente possa ser analisado de forma sequencial pelos gestores comerciais.

#### Regras:

- 1) Toda vez que um **novo cadastro** ou uma **alteração de cadastro** for realizada no sistema, o cliente deverá entrar na **última** posição da fila de atendimento.
- 2) Possibilitar a **retirada** do cliente na **primeira** posição da fila de atendimento, apresentando seus dados para o tratamento.
- 3) Caso o gestor comercial solicite um *prospect* da fila para atendimento e não houver nenhum cliente na fila, deverá retornar um status coerente informando que a fila de atendimento está vazia.

#### Desafio:

- a) incluir na API criada no desafio “1” uma nova operação que possibilite a **retirada do próximo cliente** da fila de atendimento e retorne os dados disponíveis
- b) implementar na linguagem java uma estrutura de dados para uma **fila**, onde seja possível acrescentar e retirar clientes na fila no modelo **FIFO** (*First In, First Out*).
- c) contemplar as **regras da história de usuário** através da implementação da operação modelada no item “a”, utilizando a estrutura de fila criada no item “b”

# Desafio 3:

## User Story: Web Front End

Como área de Comercialização da Cielo, desejo poder realizar a gestão de pré-cadastro de clientes e, no mesmo sistema, fornecer aos gestores comerciais a possibilidade de recuperar dados dos prospects da fila de atendimento

### Desafio:

- a) Desenvolver uma aplicação em React ou Angular que consuma a API descrita nos Desafios 2 e 3
- b) As telas devem contemplar as operações de **criação, alteração, exclusão e consulta** de pré-cadastro de clientes
- c) Ao recuperar um prospect da fila, deve ser exibido o mesmo resultado da consulta de pré-cadastro deste cliente
- d) Deve ser apresentada uma mensagem amigável quando for solicitado um prospect da fila quando ela estiver vazia
- e) Responsividade: o desafio deve ser desenvolvido para ser responsivo em diferentes dispositivos e tamanhos de tela.
- f) Acessibilidade: o desafio deve ser desenvolvido de forma acessível, seguindo as diretrizes do WCAG2.
- g) Design e UX: o desafio deve ter um bom design e uma boa experiência do usuário, seguindo as melhores práticas de design e UX.