# Desafio Técnico Backend

Oi! Tudo certo?

Estamos felizes com o seu interesse em querer fazer parte do time MaxMilhas. Queremos te convidar para a etapa do Desafio Técnico. É o momento de conhecer os seus skills técnicos, sua lógica e solução de problema.

Estamos a disposição para qualquer dúvida. Não deixe de entrar em contato ;) Boa sorte!

## Contextualização do Problema

Atualmente o time de análise antifraude do ecommerce realiza um controle de CPFs em uma planilha eletrônica. Nesta planilha são adicionados CPFs com risco de fraude. Com o aumento da nossa base de clientes têm ficado cada vez mais difícil manter o controle manual.

Com isso o Product Owner do time levantou os principais requisitos funcionais para desenvolvimento de um sistema que controle os CPFs adicionado-os em uma lista restrita.

## Requisitos Funcionais

Os requisitos fucionais foram levantados utilizando o modelo de estória de usuário (User Stories).

| # | Funcionalidade | Descrição | Critérios de Aceite |
|---|---|---|---|
| US1 | Adicionar CPF na lista restrita | Eu, enquanto Product Owner, quero adicionar um CPF na lista restrita para consulta futura. | 1. Deve ser adicionado um CPF válido na lista, sem dígitos repetidos e formatação.<br>2. Se o CPF for inválido deve retornar a exceção do tipo "InvalidCpfException".<br>3. Se um CPF existir na lista deve retornar a exceção do tipo |

| | | | |
|---|---|---|---|
| | | | "ExistsCpfException".<br>4. O CPF adicionado na lista deve conter a data de inclusão (createdAt) no formatado ISO 8601 - UTC. |
| US2 | Verificar se um determinado CPF está na lista restrita | Eu, enquanto Product Owner, quero verificar se um CPF esta na lista restrita. | 1. Se um CPF existir deve retornar o CPF e a data de criação (createdAt) no formato ISO 8601 - UTC.<br>2. Se o CPF não existir deve retornar uma exceção do tipo "NotFoundCpfException".<br>3. Se o CPF for inválido deve retornar a exceção do tipo "InvalidCpfException". |
| US3 | Remover um CPF da lista restrita | Eu, enquanto Product Owner, quero remover um CPF da lista restrita. | 1. Se o CPF não existir deve retornar uma exceção do tipo "NotFoundCpfException".<br>2. Se o CPF for inválido deve retornar a exceção do tipo "InvalidCpfException". |
| US4 | Visualizar todos os CPFs que estão na lista restrita | Eu, enquanto Product Owner, quero visualizar todos os CPFs que estão na lista restrita para gerar um relatório de controle de CPFs. | 1. Se nenhum CPF existir na lista deve retornar um array vazio. |

# Descrição da API

Uma descrição básica da API foi criada para orientar a integração por outros times.

É esperado aplicar o padrão `{ "type": "InvalidCpfException", "message": "CPF is not valid."}` no retorno das mensagens de erro.

## 1. Add CPF

Adiciona um CPF na lista restrita (US01)

| Item | Descrição |
| --- | --- |
| URL | /cpf |
| Data Params | `{ "cpf": "64852893055" }` |

## 2. Check CPF

Verifica se um CPF está adicionado na lista restrita (US02)

| Item | Descrição |
| --- | --- |
| URL | /cpf/{cpf} |
| Success Result | Content: `{ "cpf": "64852893055", createdAt: "2019-12-17T22:22:08.547Z"}` |

## 3. Remove CPF

Remove um CPF adicionado na lista restrita (US03)

| Item | Descrição |
| --- | --- |
| URL | /cpf/{cpf} |

## 4. Find All CPFs

Retorna a lista de CPFs da lista restrita (US04)

| Item | Descrição |
| --- | --- |
| URL | /cpf |
| Success Result | Content: `[ { "cpf": "64852893055", createdAt: "2019-12-17T22:22:08.547Z"} ]` |

# Requisitos não Funcionais

- A aplicação deve expor suas funcionalidades em um serviço Restful utlizando o formato JSON;
- Temos preferência nas seguintes linguagens para realização do desafio técnico: Java, C#, Node.js (Javascript/Typescript), PHP ou Python. Se a linguagem de sua preferêcia for outra, não têm problema, o time de engenharia pode demorar um pouco para corrigir seu desafio;
- Fica a seu critério o uso de frameworks (web, persistência, etc). Pense na sua produtividade na hora de desenvolver o desafio técnico;
- A utilização de banco de dados de dados relacional (MySQL, PostgreSQL, HSQLDB, H2, etc) ou não relacional (MongoDB) é livre. O uso de um banco de dados embutido também é válido.
- Testes unitários não são obrigatórios mas é uma segurança que tudo está funcionando como esperado. Utilize os cenários dos critérios de aceite das User Stories como base dos seus testes;
- A utilização de Container (Docker) para executar sua aplicação é uma boa prática e garante que tudo vai funcionar igual ao seu ambiente.
- É imporntate a criação de um Readme.md com orientações de build e execução da aplicação. Aproveite para expor suas motivações na escolha do framework, banco de dados, estilo arquitetural, organização dos pacotes, etc. Esta pode ser uma oportunidade para conhecermos melhor sua experiência :).

## Lembra-se, você será avaliado por:

- Utilização de melhores práticas de desenvolvimento (Design Pattern, SOLID, Clean Code, etc);
- Utilização dos recursos mais recentes da linguagem escolhida;
- Boa organização lógica e documental (readme, comentários, logs, diagramas, etc);
- Quantidade de requisitos funcionais e não implementados.

# Como entregar o Desafio Técnico?

Você deverá entregar um arquivo compactado (zip/tar.gz) contendo o projeto e todos os artefatos necessários para execução do mesmo em outro ambiente.

# Quando entregar este Desafio Técnico?

O envio do arquivo deverá ser realizado dentro do prazo alinhado e descrito no e-mail. Se ocorrer algum imprevisto e precisar de mais tempo entre em contato com a gente ;).

Boa sorte!

Time MaxMilhas