Sumário

[Objetivo do projeto 1](#_Toc74877691)

[Épicos identificados 1](#_Toc74877692)

[Tecnologias utilizadas 1](#_Toc74877693)

[Riscos identificados 1](#_Toc74877694)

[Estimativa inicial e cronograma 1](#_Toc74877695)

[Etapa 1: Implementação mínima e analise do deploy 2](#_Toc74877696)

[Etapa 2: Conclusão da implementação e definição do servidor cloud 2](#_Toc74877697)

[Documentação da API 2](#_Toc74877698)

[Melhorias e débito técnico 3](#_Toc74877699)

[Links do Projeto 3](#_Toc74877700)

# Objetivo do projeto

Criar uma REST API para gerenciamento de dados de clientes. A interface da aplicação deve permitir a criação, atualização e pesquisa de registros de clientes. A aplicação deve ser disponibilizada em um provedor cloud para acesso.

## Épicos identificados

1. Desenvolver API REST
2. Disponibilizar aplicação em um provedor cloud.

## Tecnologias utilizadas

As tecnologias foram selecionadas com base nos seguintes critérios: Conhecimento prévio da tecnologia; Interesse do desenvolvedor em conhecer a tecnologia necessária e adaptabilidade / qualidade da documentação disponibilizada pelo mantenedor da ferramenta.

1. Java 8
2. Spring Boot 2.5
3. FlywayDB
4. Docker
5. MySQL 8
6. Microsoft Azure
7. Spring Tool Suite 4

## Riscos identificados

Na análise inicial do projeto identificou-se a necessidade de realizar treinamento técnico da publicação do sistema em cloud usando Docker devido menor experiência do desenvolvedor nas tecnologias usadas para este fim.

## Estimativa inicial e cronograma

Visando reduzir riscos e ter software funcionando no menor tempo possível o desenvolvimento foi dividido em duas etapas

### Etapa 1: Implementação mínima e analise do deploy

1. Desenvolver API REST
   1. Implementar o cadastro de clientes.
   2. Implementar a listagem dos clientes (sem filtros por campos).
2. Disponibilizar aplicação em um provedor cloud.
   1. Criar POC publicando uma aplicação existente em um servidor cloud.

Conclusão estimada em 15/06/2021

### Etapa 2: Conclusão da implementação e definição do servidor cloud

1. Desenvolver API REST
   1. Concluir a implementação com a API de atualização
   2. Implementar a listagem dos clientes com filtros por campos.
   3. Implementar validação da API
2. Disponibilizar aplicação em um provedor cloud.
   1. Decisão de uso da Microsoft Azure para publicação das imagens
   2. Estudo e configuração do ambiente Cloud.

Conclusão estimada em 18/06/2021

# Documentação da API

A API REST criada é composta por 3 serviços: Criar Cliente, Atualizar Cliente e Pesquisar Cliente. A Tabela abaixo detalha os serviços e seus endpoints.

1. Cadastrar Cliente

Permite o cadastro de clientes

Apenas um cadastro é permitido por cpf.

Link: http://52.224.249.98:9000/GerenciamentoClientes/Cadastro/criar

Exemplo de corpo da requisição

{

"cpf" : "69068174022",

"nome" : "Rodrigo Freund de Moraes",

"dataNascimento" : "1983-10-30",

"estadoCivil": "SOLTEIRO"

}

Estado civil disponíveis: SOLTEIRO, CASADO, DIVORCIADO e VIUVO.

1. Atualizar Cliente

Permite atualização dos dados de um cliente (identificado por um cpf)

Link: http://52.224.249.98:9000/GerenciamentoClientes/Cadastro/atualizar

Exemplo de corpo da requisição

{

"cpf" : "69068174022",

"nome" : "Rodrigo Freund de Moraes",

"dataNascimento" : "1980-10-30",

"estadoCivil": "SOLTEIRO"

}

1. Pesquisar Cliente

Permite pesquisa de clientes utilizando filtros.

Sem filtros o sistema retorna todos os clientes cadastrados.

O filtro é passado para o sistema através de variáveis no link

Link exemplo: http://52.224.249.98:9000/GerenciamentoClientes/Pesquisa/pesquisar?page=0&size=5&cpf=69068174002

# Melhorias e débito técnico

Como débito técnico temos teste unitários na camada de serviço e os estes integrados podem ser incrementados.

Como melhorias pode ser adicionado o Swagger para documentação da API.

# Links do Projeto

Projeto postman de produção: <https://www.getpostman.com/collections/fed1ae6780dcbff13ea6>

Github do projeto: <https://github.com/rodrigofreund/GerenciamentoClientes>