**Documento de requisitos – SMMC – Sistema Monitor de MicroServiços**

**Autores:**

João Vitor Santos Dias

**Sumário**

[**Introdução**](#_lz0cbbf8tdj6) **2**

[**Escopo**](#_6wuqt6f669xx) **2**

[**Técnicas de Obtenção de Requisitos Utilizadas**](#_bdofjaxqbb6j)2

[**Dicionário de Termos e Siglas**](#_euw51znqpa84)2

[**Atores do sistema**](#_7nbrladz4lv9)2

[**Requisitos Funcionais**](#_lr7hjlji33df)2

[Informações Gerais](#_ovf7b85hjeas) 2

[Funcionalidades](#_e30bo914kqoq) 2

[Restrições de Acesso](#_x0jy2tjy7ueb) 3

[**Requisitos Não-Funcionais**](#_ylgrkqq70txs)3

[Características Técnicas](#_prgh8jb5uxcl) 3

[Layout](#_8cr3pxcbhthx) 3

[**Regras de negócio**](#_6tysarx4a3ba)4

# 

# Introdução

Com o aumento da digitalização pelo globo diversas aplicações que antes pouco tinham de se preocupar em com a escalabilidade, resiliência e maleabilidade notaram que a rigidez do padrão de arquitetura monolítica não mais atenderia as demandas exigidas pelo mercado.

Neste cenário surgiu os padrões de arquitetura voltados aos serviços, também intitulado de arquitetura de microserviços. Esse padrão orienta que a aplicação possua componentes/serviços fracamente acoplados e fortemente interdependentes, cujas interações deverão ser realizadas através de APIs (Application Programming Interface).

Contudo, esse novo modelo trouxe outros desafios como monitoramento, orquestração, configuração, sincronismo e comunicação entre todos esses componentes/serviços da aplicação.

Pensando em sistemas que possuem elevada sensibilidade a oscilações/interferências na rede, o objetivo deste trabalho será propor soluções quando ao desafio de monitoramento. Para que possamos alcançar tal objetivo este projeto busca criar uma ferramenta que registre em tempo real a latência de comunicação entre os serviços com o menor impacto possível na rede de computadores na qual está inserida.

# Escopo do projeto

O trabalho tem como objetivo montar uma aplicação que monitore continuamente a performance do tráfego de dados de aplicações REST distribuídas em serviços. Serão apresentadas as latências entre cada rota da arquitetura e calculada as estimativas de tempo de resposta para as próximas transmissões.

# Atores do sistema

Usuário administrador (Administrador);

Usuário visualizador (Visualizador);

Serviço Monitorador (Serviço);

# Requisitos Funcionais

## Informações Gerais

1. RF(Essencial): O sistema deve permitir o cadastro de serviços externos para serem monitorados.
2. RF(Essencial): O sistema deve permitir que o usuário registre as **rotas de requisição** de cada serviço mencionando o serviço destino da requisição;

## Funcionalidades

1. RF(Essencial): O sistema deve monitorar continuamente a performance do tráfego entre os serviços apresentando em tela o tempo de latência de cada trecho;
2. RF(Essencial): O sistema deve contabilizar a quantidade total de respostas enviadas e requisições recebidas em um determinado período;
3. RF(Desejável): O sistema deve identificar a contabilizar a quantidade de falhas recebidas a partir de requisições **realizadas** (ausência de resposta, perda na comunicação ou aplicação destino fora de operação);
4. RF(Desejável): O sistema deve identificar a contabilizar a quantidade de falhas enviadas a partir de requisições **recebidas** (exceção na aplicação, falha interna);
5. RF(Desejável): O sistema deve contar com uma tela que relatará métricas de cada serviço. (Exemplo: Período mínimo/médio/máximo para responder requisições do dia, quantidade de respostas enviadas, timeout de requisições, quantidade de falha recebidas, quantidade de falha enviadas, outras.);
6. RF: Ao ser registrado um novo serviço para o monitoramento o sistema deverá consultar imediatamente se o serviço é acessível e colorir o item que representa o serviço recentemente adicionado com a cor verde, caso tenha-o encontrado ou cor vermelha, caso não tenha;

## Restrições de Acesso

1. RF (Essencial): A interface deve possuir um sistema de autenticação, contemplando 2 níveis de acesso: (2) Administrador, (1) Visualizador.
2. RF (Essencial): Apenas usuários administradores podem realizar o gerenciamento de usuários do sistema.
3. RF (Essencial): Qualquer tipo de usuário deve ter permissão de deletar própria conta.
4. RF (Importante): Apenas usuários administradores podem conceder aos usuários visualizadores a elevação do nível de acesso para administrador.

# Requisitos Não-Funcionais

## Características Técnicas

1. RNF(Essencial): A interface do sistema deverá ser apresentada em um website.
2. RNF: O sistema fará o monitoramento apenas de pacotes http e https;
3. RNF: As informações sobre tempos de tráfego em cada trecho deverão ser atualizadas a cada 5s.

## Layout

1. RNF: O layout do website deve ser simples, apresentar um menu com as seguintes opções:
   * acessar as configurações de conta do usuário;
   * acessar o ambiente de registro de serviços;
   * acessar o ambiente de registro de rotas;
   * lista das telas de métricas;

Ao centro apresentar o conteúdo da tela selecionada.

1. RNF: As métricas calculadas deverão ser apresentadas no formato de relatórios em texto.
2. RNF: O layout deverá permitir que cada serviço possa ser selecionado para a apresentação de suas métricas individualmente.
3. RNF: A Tela do monitoramento principal deverá ser apresentada no formato de lista.
4. RNF: O sistema deverá persistir os dados no banco Postgres.
5. RNF: O sistema deverá ser programado na linguagem Java utilizando o Framework Spring Boot;
6. RNF: O conteúdo das requisições interceptadas não deverá ser exposto ou armazenado.

# Regras de Negócio

Não há conteúdo aqui.