

# **Sistema de Supervisão Predial**

## Sumário

Introdução .....	3
Objetivo.....	3
Levantamento de Requisitos.....	3
Arquitetura do Sistema.....	4
Estações Remotas.....	5
Configuração das Placas .....	6
O servidor.....	6
Gerenciamento de dados.....	6
Diagrama de Casos de uso .....	7
Descrição dos Casos de uso.....	7

## **Introdução**

O presente documento apresenta uma análise da necessidade do cliente referente a coleta de dados das instalações de um imóvel, a arquitetura do sistema responsável para resolução do sistema . Descrevendo os elementos que fazem parte do sistema, a comunicação entre as partes, a persistência dos dados e a interface do usuário.

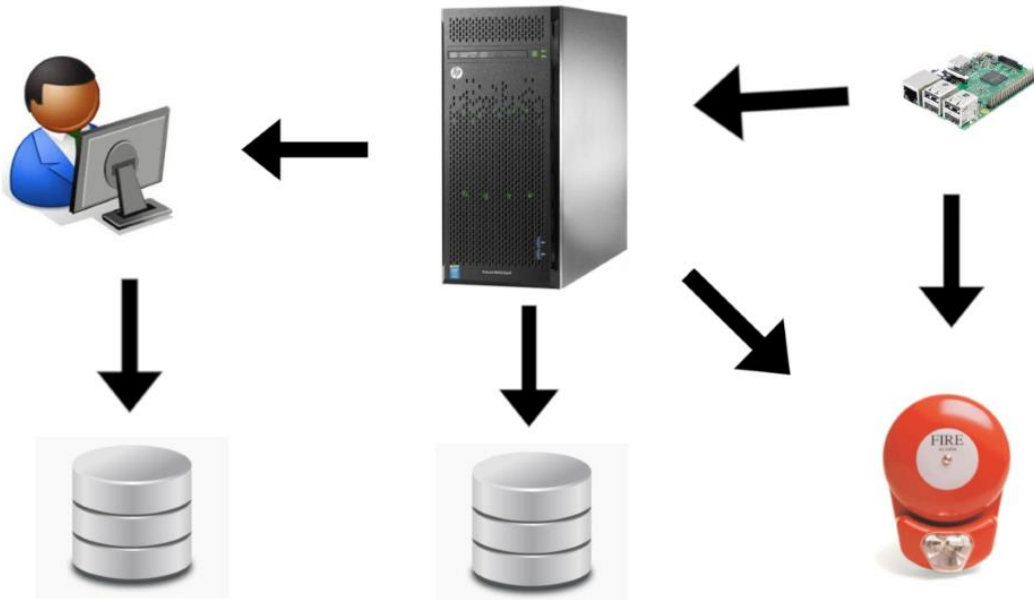
## **Objetivo**

O objetivo deste documento é análise de requisitos e a descrição das funcionalidades do sistema de acordo com as necessidades do usuário e a descrição de como o sistema será implantado.

## **Levantamento de Requisitos**

O cliente possui um imóvel com várias instalações, diante da preocupação de segurança, surgiu a necessidade de monitoramento de vários pontos deste imóvel, especificamente, 5000 pontos de monitoramento. Essas instalações podem possuir características e configurações diferentes umas das outras, como por exemplo saunas, frigoríficos, ambientes refrigerados entre outros.

## Arquitetura do Sistema



- Para cada ponto de coleta será instalado uma estação remota capaz de medir a temperatura, a localização entre outros dados
- A API da estação remota é capaz de enviar os dados via Rest em formato Json.
- As estações remotas serão programadas para guardar o nome da instalação onde serão colocadas.
- As instalações terão a temperatura máxima variável, cada estação remota estará programada para responder de acordo com a instalação posicionada.
- As estações remotas serão capazes de acionar o alarme de acordo com a temperatura limite na qual foi programado.
- Para evitar problemas de latência, cada estação remota é programada para consultar dados de coleta, como temperatura, e etc... E guardar estes dados em cache.
- As estações remotas se comunicam via rede local do prédio, pode ser via cabo ou wireless.
- Serão instalados alarmes de incêndio por todo o prédio, estes alarmes poderão ser acionados pela estação remota, pelo servidor ou manualmente.
- O alarme pode ser desligado manualmente (no próprio alarme).

- O servidor será instalado localmente no próprio prédio dentro de um CPD para maior estabilidade e independência de provedores de internet.
- O servidor será Robusto mais que o suficiente para comportar todas as tarefas necessárias.
- Docker será utilizado para separar os softwares necessários para o funcionamento do sistema.
- Um microservice será responsável por recuperar os dados das estações remotas e guardá-las no banco de dados.
- O microservice usará threads, onde cada thread será responsável por pegar uma série de pontos de coleta diferente. De forma que o tempo máximo da coleta seja a cada 5s.
- A falta de informação de qualquer estação remota resulta no acionamento do alarme.
- As informações obtidas pelo ponto de coleta serão persistidas em um banco de dados NoSql em um container diferente.
- Em outro container deste servidor estará uma aplicação MVC para ser acessada pelo usuário.
- Esta estação pode ser acessada via rede local ou internet.
- Ele terá um banco de dados próprio que constará os dados das estações remotas e dos alarmes.
- Uma UI contendo o mapeamento das estações remotas e alarmes.

## **Estações Remotas**

O hardware utilizado para a coleta dos dados são placas Raspberry Pi, acoplados com uma placa de extensão com sensores de temperatura, umidade e pressão atmosférica, bibliotecas de softwares fornecem o software para a placa de extensão. Cada placa possui duas chamadas de API já implementadas. Os sensores se comunicarão com a rede local do cliente podendo ser via Wi Fi ou por cabo RJ-45.

## **Configuração das placas**

- Raspberry Pi 3 Model B
  - CPU Quad-Core 64-bit 1.2 GHz
  - 1 GB RAM
  - Micro SD 15 GB
  - Wireless Lan / Ethernet 100 Mbps
  - GPIO – 40 pinos
- ANAVI Infrared pHAT
  - Sensor pressão barométrica
  - Sensor de temperatura e umidade

## **O Servidor**

O servidor é o responsável por fazer as requisições para os sensores. As requisições são feitas por um sistema interno que consome os dados dos pontos de coleta e persiste esses em um banco de dados, este mesmo sistema contém as regras definidas para alarme em situações anormais.

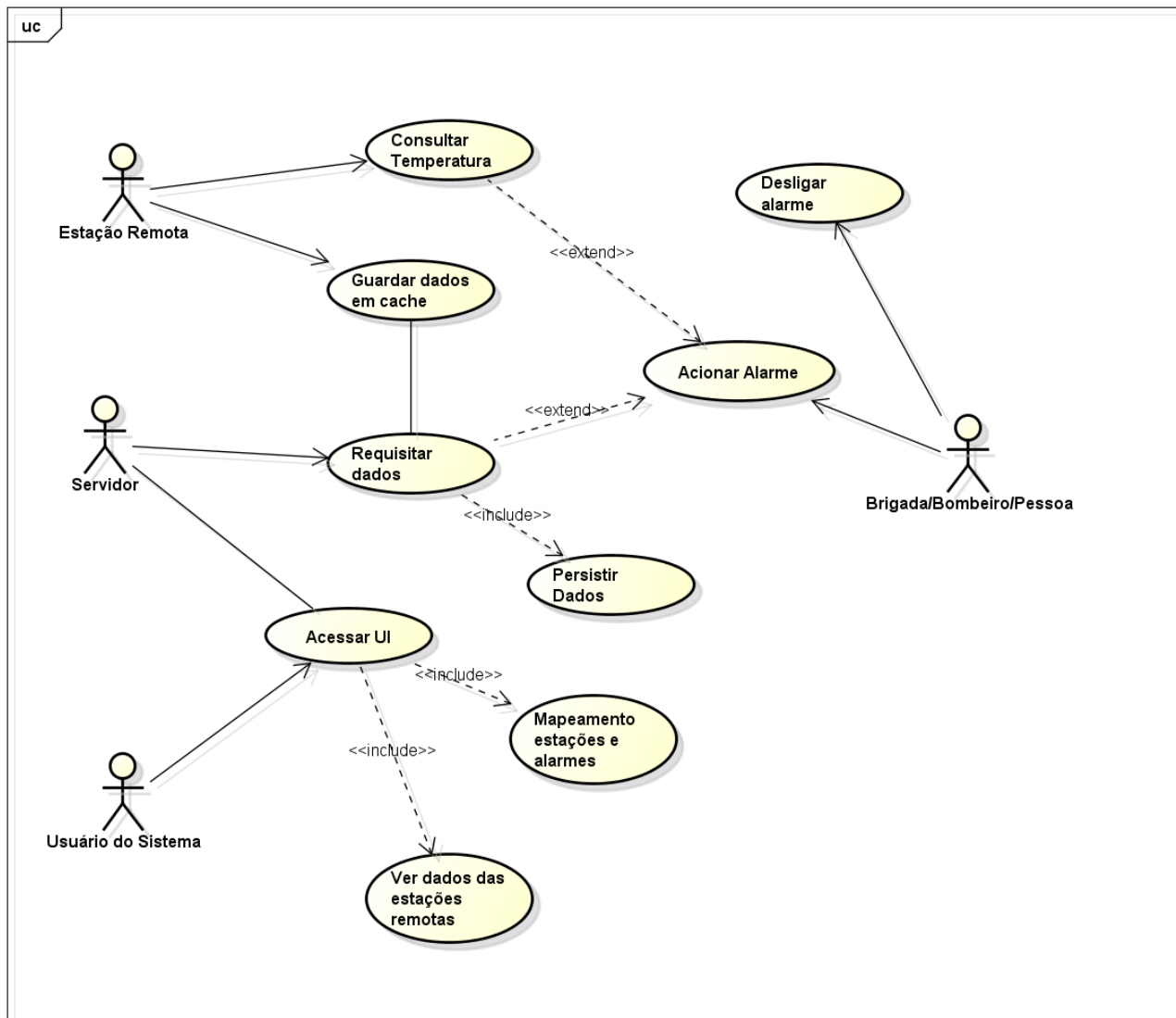
O servidor também contará com um sistema que atuará como uma interface de serviço e disponibilizará os dados coletados para usuário por esta interface.

## **Gerenciamento de Dados**

Será utilizado Docker, os sistemas, os bancos de dados e a UI para o usuário fazer as consultas.

O microservice requisita os dados, já armazenados em cache, das estações remotas, o sistema é multithreading, cada thread consulta um range de estações, afim de contemplar o tempo de 5 segundos de monitoramento, a cada consulta os dados são armazenados em banco de dados NoSql (mongoDB). Em outro container, fica a aplicação MVC que contém a UI do para a consulta dos pontos de coleta e também um banco de dados separado contendo as informações dos pontos de coleta.

## Diagrama dos Casos de Uso



powered by Astah

## Descrição dos Casos de Uso

### Consultar Temperatura

#### Cenário Principal

A API da estação remota mede a temperatura e outros dados do ponto de coleta ciclicamente, durante o menor intervalo de tempo possível.

#### Cenário Alternativo

A temperatura está fora do normal para o ponto de coleta na qual a estação foi instalada. A estação remota aciona o alarme.

## **Guardar dados em cache**

### **Cenário Principal**

A cada consulta de dados dos pontos de coleta, estes mesmos são guardados em cache. O cache guarda até 10 conjuntos de dados, após os 10, os dados vão sendo substituídos.

## **Requisitar Dados**

### **Cenário principal**

O microservice, separado em várias threads, cada thread coleta uma série de estações remotas e seus dados. A ação se repete a cada 5 segundos, em toda consulta os dados são armazenados em banco de dados não relacional.

### **Cenário Alternativo**

A falta de comunicação de qualquer uma das estações remotas resulta no acionamento do alarme.

## **Acessar UI**

### **Cenário Principal**

O usuário do sistema, acessa uma aplicação MVC que possui o mapeamento dos pontos de coleta e dos alarmes, entre outras informações pertinentes ao sistema como um todo.

## **Acionar o Alarme**

### **Cenário Principal**



O alarme pode ser acionado manualmente caso um incêndio seja presenciado e não detectado pela estação remota.

## **Desligar o alarme**

### **Cenário principal**

Embora seja recomendado um bombeiro, ou alguém da brigada de incêndio, o alarme possui a opção de ser desligado manualmente.