

UNIVERSIDADE FEDERAL DE SÃO CARLOS - UFSCar  
Centro de Ciências Exatas e de Tecnologia - CCET  
Departamento de Computação - DC  
**Curso de Bacharelado em Engenharia de Computação - EnC**

**Jhon Wislin Ribeiro Citron**

**Desenvolvimento de Aplicações para Cidades Inteligentes  
baseado na Plataforma InterSCity**

**São Carlos  
2025**

**Jhon Wislin Ribeiro Citron**

**Desenvolvimento de Aplicações para Cidades Inteligentes Baseadas na Plataforma  
InterSCity**

Trabalho de conclusão de curso apresentado ao curso de Engenharia de Computação, como requisito para obtenção do título de Bacharel, pela Universidade Federal de São Carlos.

Área de concentração: Levantamento, estruturação e desenvolvimento de software

Orientador: Fabiano Cutigi Ferrari

**São Carlos**

**2025**

---

## **Agradecimentos**

---

Agradeço aos meus pais Dejair e Sueli, e a minha avó Aparecida, pelo apoio e por me ajudarem e me incentivarem ao longo de toda minha trajetória. Considero que boa parte das minhas escolhas foram por causa dessas pessoas, e sem o apoio delas não estaria onde estou.

Agradeço aos meus amigos, Vinícius, Lucas, Vitor, Kleber, Caique, Juan e Luiz. Por estarem comigo nestes últimos 6 anos. Por toda a diversão que passamos juntos, pelos conselhos, e pela paciência que tiveram comigo.

Devo agradecer também ao professor Fabiano, por me acompanhar neste trabalho e ser meu orientador.

Tenho muitas outras pessoas que me ajudaram nestes 6 anos de curso, infelizmente não posso citar todas, mas agradeço a elas por todo apoio e ajuda que tive. Se estou aqui finalmente apresentando minha monografia, é graças a todos que me apoiaram e me ajudaram. Meus sinceros agradecimentos a todos.

*“Você não é derrotado quando perde, mas sim quando você desiste.”*  
Vegeta, Dragon Ball Z

---

## Resumo

---

Cada vez mais, as tecnologias estão integradas aos meios urbanos, como em equipamentos hospitalares, meios de transporte, câmeras, radares e outros dispositivos. Essas tecnologias geram dados continuamente e se comunicam, o que caracteriza os cenários das Cidades Inteligentes.

O desenvolvimento de aplicações que auxiliam e melhoram o cotidiano das pessoas é essencial, especialmente considerando o aumento populacional nos centros urbanos. Assim, a plataforma InterSCity oferece um ambiente de desenvolvimento no qual dados gerados por diferentes equipamentos, como sensores e atuadores, podem ser simulados e utilizados em testes e simulações de funcionamento dessas aplicações.

Neste trabalho, foi realizado um estudo de caso com o uso da plataforma InterSCity, demonstrando como ela pode ser integrada ao desenvolvimento de aplicações para Cidades Inteligentes. Como resultado, foi desenvolvida uma aplicação voltada ao monitoramento de temperaturas em estações de metrô.

**Palavras-chave:** Cidades Inteligentes, InterSCity, monitoramento de temperatura, sensores, simulação.

---

## Abstract

---

Increasingly, technologies are being integrated into urban environments, such as hospital equipment, transportation systems, cameras, radars, and other devices. These technologies continuously generate data and communicate with each other, characterizing the scenarios of Smart Cities.

The development of applications that assist and improve people's daily lives is essential, especially considering the population growth in urban centers. In this context, the InterSCity platform provides a development environment where data generated by different devices, such as sensors and actuators, can be simulated and used for testing and functional simulations of these applications.

In this study, a case study was conducted using the InterSCity platform, demonstrating how it can be integrated into the development of applications for Smart Cities. As a result, an application focused on temperature monitoring in subway stations was developed.

**Keywords:** Smart Cities, InterSCity, temperature monitoring, sensors, simulation.

---

## Lista de Ilustrações

---

Figura 1 - Arquitetura da plataforma InterSCity (extraída do trabalho de DEL ESPOSTE et al. (2018)).....	16
Figura 2 - Diagrama de casos de uso.....	25
Figura 3 - Diagrama de sequência 1.....	26
Figura 4 - Diagrama de sequência 2.....	27
Figura 5 - Diagrama de atividade.....	28
Figura 6 - Editar limiares.....	29
Figura 7 - Card da estação.....	30
Figura 8 - Card em alerta de temperatura muito acima do limite.....	31
Figura 9 - Card em alerta de temperatura acima do limite.....	31
Figura 10 - Card em alerta de temperatura parcialmente acima do limite.....	31
Figura 11 - Card em alerta de temperatura muito abaixo do limite.....	32
Figura 12 - Card em alerta de temperatura abaixo do limite.....	32
Figura 13 - Card em alerta de temperatura parcialmente abaixo do limite.....	32
Figura 14 - Card em alerta de princípio de incêndio.....	33
Figura 15 - Monitor de temperatura sem dados.....	34
Figura 16 - Monitor de temperatura.....	34
Figura 17 - Monitor de temperatura ajustado.....	35
Figura 18 - Status de comunicação.....	36
Figura 19 - Requisição de dados do recurso de station.....	37
Figura 20 - Informação de retorno da requisição.....	37

---

## SUMÁRIO

---

<b>1 INTRODUÇÃO.....</b>	<b>9</b>
1.1 Contextualização e Problemática.....	9
1.2 Objetivos.....	10
1.3 Justificativa.....	10
1.4 Estrutura do Trabalho.....	11
<b>2 FUNDAMENTOS TEÓRICOS.....</b>	<b>12</b>
2.1 Monitoramento de Temperatura.....	12
2.2 Internet das Coisas (IoT).....	12
2.3 Plataforma InterSCity.....	13
2.4 React e Node JS.....	16
<b>3 METODOLOGIA E DESENVOLVIMENTO.....</b>	<b>18</b>
3.1 Metodologia.....	18
3.2 Levantamento de Requisitos.....	18
3.3 Desenvolvimento do Sistema.....	19
3.3.1 Criação de Sensores.....	19
3.3.2 Geração de Dados.....	22
3.3.3 Desenvolvimento de Interface.....	23
3.4 Categorias de Alertas.....	24
3.5 Diagramas.....	25
<b>4 RESULTADOS E DISCUSSÕES.....</b>	<b>29</b>
4.2 Resultados Obtidos.....	33
4.3 Análise dos Resultados e Lições Aprendidas.....	38
<b>5 CONCLUSÃO.....</b>	<b>39</b>
5.1 Resultados e Contribuições.....	39
5.2 Limitações Encontradas.....	40
5.3 Trabalhos Futuros.....	41
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>42</b>



## 1 INTRODUÇÃO

Neste capítulo, apresentam-se o contexto e a problemática para o trabalho realizado (Seção 1.1), os objetivos gerais e específicos para este trabalho (Seção 1.2, subseções 1.2.1 e 1.2.2), a justificativa para o desenvolvimento deste trabalho (Seção 1.3), e uma descrição breve da organização dos capítulos subsequentes (Seção 1.4).

### 1.1 Contextualização e Problemática

O conceito de Cidades Inteligentes vem se consolidando ao longo dos anos como o principal caminho de desenvolvimento urbano, sendo este um conceito que se baseia em agregar soluções tecnológicas aos meios urbanos, de forma a oferecer um cenário mais econômico e sustentável. Assim, segundo a Secretaria de Ciência, Tecnologia e Inovação (SECTI), as Cidades Inteligentes visam melhorar a qualidade de vida das pessoas, integrando as tecnologias de comunicação e informação (TIC) aos seus serviços públicos, como meios de transporte, saneamento básico, iluminação, e entre outros serviços.

O mercado global de soluções/aplicações para Cidades Inteligentes apresentou um crescimento significativo nos últimos anos, com previsão de expansão ainda maior nas próximas décadas. Conforme estudo da empresa Grand View Research (GRAND VIEW RESEARCH, 2024), este mercado foi avaliado em US\$ 1 trilhão em 2022 e deve crescer à taxa de 25% ao ano até 2030. Esse cenário evidencia a necessidade de tecnologias que integrem eficientemente os dados gerados por dispositivos e sensores em ambientes urbanos. As aplicações para Cidades Inteligentes se caracterizam por criar soluções que otimizam e melhoram os serviços públicos com base na análise de dados de sensores e dispositivos IoT. Elas gerenciam recursos, como a distribuição de água e energia, controlam o tráfego e desenvolvem soluções inovadoras com o objetivo de melhorar a qualidade de vida dos cidadãos.

A plataforma InterSCity é resultado de um projeto de pesquisa coordenado pelo professor Fabio Kon, do Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP), e hospedado pelo Instituto Nacional de Ciência e Tecnologia (INCT), (INTERSCITY, 2025). A plataforma é um projeto de código aberto para o desenvolvimento de projetos voltados para o contexto de Cidades Inteligentes.

A InterSCity oferece recursos como descobertas de dispositivos, obtenção de dados de sensores, histórico de dados, comunicação com atuadores e filtragem de dados (SILVA et al.,

2018). No entanto, compreender as potencialidades e desafios da implementação dessa tecnologia em aplicações reais constitui uma lacuna no campo acadêmico e profissional. Este trabalho busca abordar essa lacuna, explorando o uso da InterSCity para o desenvolvimento de uma aplicação.

## 1.2 Objetivos

De forma geral, este trabalho consiste em explorar o funcionamento da plataforma InterSCity, buscando compreender seus principais serviços e como a plataforma pode ser integrada para o desenvolvimento de aplicações no contexto de Cidades Inteligentes. Assim, como um estudo de caso da plataforma, neste trabalho foi desenvolvido uma aplicação para monitorar a temperatura em estações de metrô.

O desenvolvimento desta aplicação visa demonstrar como a plataforma InterSCity pode ser integrada ao desenvolvimento de aplicações para Cidades Inteligentes, servindo como um exemplo de desenvolvimento para possíveis outras aplicações.

Com base no objetivo geral, definem-se para este trabalho os seguintes objetivos específicos:

- Compreender o funcionamento da plataforma InterSCity;
- Demonstrar de forma prática seu funcionamento, e possibilidades de uso;
- Apresentar como uma aplicação pode ser integrada à plataforma, e como é feita essa comunicação;
- Criar recursos que permitam simular os dados dentro da plataforma;
- Testar a aplicação conforme os dados simulados dos sensores.

## 1.3 Justificativa

Com o desenvolvimento populacional acelerado nos últimos anos, se faz necessário criar soluções tecnológicas para solucionar as altas demandas de recursos públicos. Estudos feitos pela ONU estimam que até 2050, ocorrerá um aumento de 55% para 68% no número de pessoas vivendo em áreas urbanas (NEC, 2024). Assim, reforça-se a necessidade de se criar soluções tecnológicas que permitam lidar com os impactos desse grande aumento populacional. Nesse contexto, plataformas como a InterSCity se tornam ferramentas que permitem desenvolver aplicações e tecnologias para cenários de Cidades Inteligentes. A

InterSCity é capaz de integrar diferentes fontes de dados e oferecer uma base sólida para o desenvolvimento de aplicações.

Explorar a plataforma InterSCity por meio do desenvolvimento de uma aplicação que simula um ambiente real consiste não apenas em validar a tecnologia, mas também identificar suas limitações e potencialidades, fornecendo subsídios para futuros desenvolvimentos na área. Assim, este trabalho contribui para a compreensão e utilização de plataformas para desenvolvimento de aplicações voltadas para o domínio de Cidades Inteligentes.

## 1.4 Estrutura do Trabalho

O Restante do trabalho se encontra estruturado da seguinte forma:

*O Capítulo 2: Fundamentação teórica* consiste em apresentar os principais conceitos abordados neste trabalho, como, por exemplo, a Internet das Coisas (IoT), a plataforma InterSCity, a importância do monitoramento, e o React e Node.js.

*O Capítulo 3: Metodologia e desenvolvimento* descreve o processo utilizado para planejar, desenvolver e testar o sistema. Assim, apresenta o levantamento de requisitos para a aplicação, o desenvolvimento do sistema (criação dos recursos, geração de dados e desenvolvimento do sistema), categorias de alertas e os diagramas.

*O Capítulo 4: Resultados e Discussões* visa demonstrar o sistema em funcionamento, destacando os benefícios, desafios e lições aprendidas.

*O Capítulo 5: Conclusão* apresenta de forma resumida os resultados obtidos neste trabalho, quais foram suas contribuições e possíveis melhorias, limitações encontradas e aprimoramentos para trabalhos futuros.

## 2 FUNDAMENTOS TEÓRICOS

Neste capítulo, é apresentada a base teórica para o desenvolvimento do trabalho. São apresentados neste capítulo a importância para o monitoramento de temperatura (Seção 2.1), o conceito de Internet das Coisas (IoT) (Seção 2.2), a apresentação da plataforma InterSCity, bem como sua arquitetura (Seção 2.3), e, por fim, uma visão geral das tecnologias React e Node (Seção 2.4).

### 2.1 Monitoramento de Temperatura

Monitorar a temperatura consiste em observar possíveis variações de temperatura que possam ocorrer, seja para monitorar alimentos que estão sendo conservados, ou manter fixamente a temperatura em um local controlado. Existem situações onde a temperatura dentro de um ambiente pode variar muito.

Um desses casos são as estações e vagões de metrô, que podem na maioria das vezes serem ambientes fechados com um acesso numeroso de pessoas diariamente. Existem relatos sobre condições extremas enfrentadas pelos passageiros em metrô devido ao calor. Por exemplo, no Rio de Janeiro, passageiros frequentemente reclamam do calor e do mau funcionamento do ar-condicionado nos vagões (G1 Rio de Janeiro, 2020).

Além disso, existem casos mais extremos, como o da linha 8-Diamante, onde passageiros foram obrigados a descer de trens devido ao calor extremo e à ausência de ar-condicionado (Barueri na Rede, 2024).

Dadas essas situações, o monitoramento de temperatura dentro desses locais se faz necessário, sendo uma forma de se monitorar e controlar a temperatura em ambientes fechados como as estações de metrô. Assim, a aplicação que foi desenvolvida neste trabalho, além de uma demonstração de aplicação da plataforma InterSCity, também representa uma ferramenta que segue uma necessidade de um contexto real.

### 2.2 Internet das Coisas (IoT)

A Internet das Coisas (IoT) ou “objetos inteligentes” é uma rede interconectada de dispositivos físicos que se comunicam entre si e com outros sistemas conectados à Internet. Esses dispositivos podem incluir veículos, eletrodomésticos e sensores equipados com

software e conectividade de rede, permitindo a coleta e a troca de dados entre eles. Esses dispositivos variam desde simples termostatos em residências até sistemas industriais complexos (IBM, 2025).

De acordo com a IBM (2025), as principais aplicações da IoT incluem:

- **Monitoramento do ambiente:** Sensores para medir temperatura e umidade, com aplicações na agricultura e em ambientes urbanos.
- **Gerenciamento do tráfego urbano:** Sistemas que controlam o fluxo de veículos para reduzir congestionamentos.
- **Otimização industrial:** Controle e automação de processos produtivos para melhorar a eficiência e reduzir custos.

As principais tecnologias que apoiam a IoT incluem sensores, redes de conectividade, plataformas de gerenciamento de dispositivos e ferramentas de análise de dados. Dessa forma, a IoT proporciona benefícios significativos, como eficiência operacional, redução de custos e otimização da tomada de decisões com base em análises realizadas por ferramentas de inteligência artificial. Todas essas soluções são aplicadas de forma integrada para criar ambientes conectados e inteligentes, que estão transformando diversos setores, como transporte, saúde e logística.

### 2.3 Plataforma InterSCity

A plataforma InterSCity é uma infraestrutura de código aberto com o objetivo de oferecer suporte à criação de soluções para Cidades Inteligentes. A plataforma se baseia em uma arquitetura composta por microsserviços que garantem escalabilidade, flexibilidade e facilidade de integração. Assim, oferecendo um ambiente de simulação que visa apoiar o desenvolvimento de aplicações para Cidades Inteligentes.

A plataforma foi detalhada no trabalho de Del Esposte et al. (2017), que destacou suas capacidades de processamento e análise de dados em grande escala, bem como a gestão eficiente de recursos urbanos.

## Recursos e Capacidades

A plataforma InterSCity define dois conceitos importantes relacionados ao seu funcionamento, sendo eles o conceito de recursos e capacidades. O trabalho de Del Esposte et al. (2017) define da seguinte forma esses conceitos:

- **Recursos:** Recursos são conceitos físicos encontrados em Cidades Inteligentes, podendo se referir a ambientes ou a dispositivos como câmeras, termômetros, veículos e dispositivos tecnológicos de IoT;
- **Capacidades:** Se refere aos principais atributos ou características de um recurso, bem como temperatura, umidade, decibels.

O conceito de recursos e capacidades está diretamente relacionado ao seu contexto de aplicação. Um recurso pode representar, por exemplo, um termômetro, cuja capacidade é medir e fornecer a temperatura. Da mesma forma, um recurso pode ser um ambiente, como um laboratório, onde suas capacidades incluem o monitoramento de temperatura, umidade e iluminação por meio de sensores.

Esse conceito se aplica a diversas áreas, como:

- **Transporte Público:** Um ônibus (recurso) pode ter a capacidade de monitorar sua localização, medir a lotação de passageiros e informar o tempo estimado de chegada;
- **Gestão de Energia:** Um poste de iluminação (recurso) pode ter como sua capacidade ajustar a intensidade da luz e monitorar o consumo de energia;
- **Medição Ambiental:** Uma estação meteorológica (recurso) pode possuir como suas capacidades medir a temperatura, umidade e qualidade do ar.

Dessa forma, a definição de recursos e capacidades depende do contexto em que são utilizados, permitindo a modelagem de sistemas inteligentes e eficientes para diferentes aplicações.

## Arquitetura e Funcionalidades

A figura 1, que foi extraída do trabalho de Del Esposte et al. (2017), apresenta a arquitetura da plataforma InterSCity. Essa arquitetura se resume em 6 micro serviços

diferentes que conectam as aplicações e funcionalidades com os dispositivos de IoT. Cada uma dessas funcionalidades é destacada a seguir:

- **Resource Discovery:** Permite a busca por recursos dentro da plataforma, podendo ser aplicados diferentes filtros como buscar por recursos de uma determinada capacidade ou pela localização;
- **Resource Viewer:** Visualização de dados dos recursos;
- **Data Collector:** Controla o acesso aos dados atuais e históricos referentes a cada recurso cadastrado na plataforma;
- **Resource Catalog:** Cria e armazena os UUIDs (Identificador do recurso) para cada recurso criado, e gerencia os dados de informação de cada recurso;
- **Actuator Controller:** Faz o controle de dados e recursos por meio de atuadores;
- **Resource Adaptor:** Permite integrar recursos na plataforma, atualizar esses recursos e fazer o envio de dados.

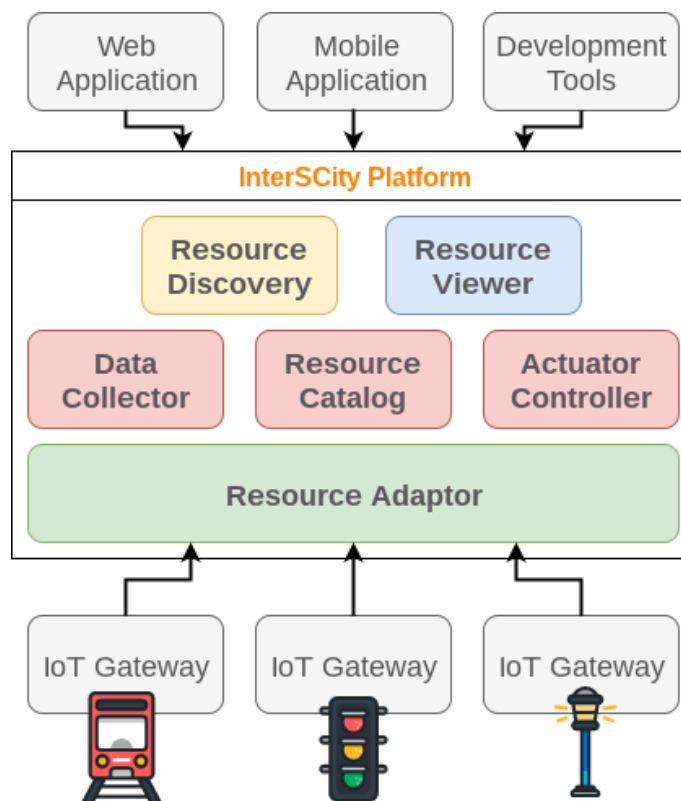


Figura 1 - Arquitetura da plataforma InterSCity (extraída do trabalho de DEL ESPOSTE et al. (2018)).

## Casos de Uso

O trabalho de Silva et al. (2018) demonstra alguns cenários de aplicação da plataforma interSCity como, por exemplo, gestão de trânsito por meio de uma aplicação de estacionamento inteligente, onde com base na plataforma é possível simular o fluxo de carros e a posição geográfica deles. A plataforma InterSCity (INTERSCITY, 2025), apresenta alguns outros cenários, como uma aplicação para controle de semáforos com o objetivo de controlar o fluxo de veículos em uma cidade, e uma aplicação que determina as áreas mais propícias para esportes conforme uma análise climática.

## 2.4 React e Node JS

React é uma ferramenta que integra o uso do Javascript, com HTML e CSS. Ela permite mesclar os conceitos de uma forma fácil e intuitiva, permitindo a criação de interfaces Web. O React se baseia no conceito de estados de uma variável, onde esse estado é variado conforme a renderização do código fonte, e a interação do usuário com a interface.

No contexto de desenvolvimento deste trabalho, o React permitiu integrar a comunicação com as informações da plataforma InterSCity com a interface sendo apresentada ao usuário, facilitando a manipulação dos dados vindos da plataforma e criando uma interface acessível ao usuário. Sendo o React uma biblioteca do JavaScript, APIs como fetch são possíveis de serem utilizadas dentro de um projeto, permitindo o uso de requisições HTTP de dados. Outro conceito interessante sobre o React, é a possibilidade de renderizar uma interface web em um browser, mesclando conceitos de HTML como criação de tabelas e botões que podem ser personalizados com o uso de CSS, dessa forma facilitando o desenvolvimento de interfaces (React, 2024).

O Node JS é uma ótima ferramenta para comunicação com servidores e leituras de rede, o que facilita o processo de fazer requisições de mudanças e atualizações em um banco de dados. Para este trabalho, ele permitiu criar um processo cíclico de atualização de dados dentro da plataforma InterSCity por meio de requisições.



O Node JS integra o uso do JavaScript, facilitando manipular as requisições e ter um melhor controle do banco (Node.JS, 2024).

### 3 METODOLOGIA E DESENVOLVIMENTO

Neste capítulo, apresentam-se a metodologia de desenvolvimento do trabalho (Seção 3.1), os requisitos levantados para a aplicação (Seção 3.2), o desenvolvimento prático da aplicação (Seção 3.3 e subseções 3.3.1, 3.3.2 e 3.3.3), as categorias de alertas gerados pela aplicação (Seção 3.4) e, por fim, os diagramas que ilustram o funcionamento da aplicação conforme o levantamento de requisitos (Seção 3.5).

#### 3.1 Metodologia

A metodologia adotada seguiu o modelo incremental, com fases de levantamento de requisitos, desenvolvimento iterativo e validação contínua. As tecnologias utilizadas incluíram Javascript, Node JS, a plataforma InterSCity e React para a interface.

#### 3.2 Levantamento de Requisitos

Para o desenvolvimento da aplicação, foi feito um levantamento de requisitos, explorando todas as funcionalidades esperadas. Dessa forma, foram definidos os requisitos funcionais e não funcionais.

##### **Requisitos funcionais:**

Todas as funcionalidades que a aplicação deve possuir são apresentadas abaixo:

- **[RF001]** O sistema deve monitorar a temperatura em tempo real em cada plataforma do metrô;
- **[RF002]** O sistema deve alertar ao usuário caso a temperatura em uma plataforma esteja acima de um valor pré-especificado pela companhia gestora da estação (por exemplo, 25 graus Celsius);
- **[RF003]** O sistema deve alertar ao usuário caso a temperatura em uma plataforma esteja abaixo de um valor pré-especificado pela companhia gestora da estação (por exemplo, 15 °C);

- [RF004] Caso o sistema detecte uma alta variação na temperatura ao longo do tempo e essa temperatura ultrapasse um limite pré-especificado (por exemplo, 60 °C), o sistema deve emitir um alerta para possível princípio de incêndio;
- [RF005] O sistema deve remover o alerta caso a temperatura retorne a uma faixa de valores entre os valores pré-especificados;
- [RF006] O sistema deve apresentar a temperatura atual em cada plataforma.

#### **Requisitos não funcionais:**

Representam as características de uso da aplicação.

- [NF001] O sistema deve ser acessível via web em qualquer navegador;
- [NF002] O sistema deve se comunicar com a plataforma InterSCity para obter as informações dos sensores;

### **3.3 Desenvolvimento do Sistema**

As subseções apresentadas a seguir (subseção 3.3.1 *Criação de Sensores*, 3.3.2 *Geração de Dados* e 3.3.3 *Desenvolvimento da Interface*) apresentam o desenvolvimento do software de monitoramento de temperaturas das estações de metrô. Cada subseção apresenta um componente desenvolvido que compõe o funcionamento da aplicação como um todo.

#### **3.3.1 Criação de Sensores**

Como apresentado na seção 2.3 *Plataforma InterSCity* do capítulo 2, o InterSCity é uma plataforma que possibilita criar aplicações para Cidades Inteligentes, fornecendo recursos para a simulação de dispositivos de IoT como sensores e atuadores. Dessa forma, para simular o funcionamento dos sensores de temperatura em cada estação de metrô, foi desenvolvido um script em Node.js, onde é feita a criação dos sensores de cada estação dentro da plataforma. O código do script e outros artefatos desenvolvidos neste trabalho estão disponíveis no repositório online criado para o projeto (CITRON, 2025).

Para criar cada um dos recursos, deve ser criada uma capacidade, que é basicamente a definição de um tipo de dado ou um atributo, que para essa aplicação foi a capacidade “*sensor-temperature*”. O módulo de código apresentado na Listagem 1 representa a criação

das capacidades dentro da plataforma, no qual é feita uma requisição do tipo POST, mandando como estrutura o nome da capacidade a ser criada (isto é, “*sensor-temperature*”), uma descrição para a capacidade, e o tipo de capacidade a ser criada (sensor).

```
const capability = await
fetch("http://10.10.104:8000/catalog/capabilities", { // Request to
create capability
  method: "POST", // POST method
  headers: { // Request headers
    "Content-Type": "application/json", // Request content type
  },
  body: JSON.stringify({
    name: "sensor-temperature", // Name of the capability being
created
    description: `Capability for temperature sensor`, // Defines
capability description
    capability_type: "sensor", // Capability type (sensor)
  }),
});
```

Listagem 1 - Criação das capacidades

Com a criação da capacidade “*sensor-temperature*”, o script fez a criação dos recursos, que neste caso foram as estações (*station*). Neste caso foram criados 9 recursos de “*station*” com a capacidade de “*sensor-temperature*”, onde a estação de metro representa o recurso que vai fornecer como informação a capacidade de temperatura na estação. Vale-se ressaltar, que a escolha por se criar 9 recursos ou estações, reflete a simular algumas das linhas de metrô de São Paulo que originalmente possuíam esta quantidade de estações.

A Listagem 2 apresenta o trecho de código referente à criação dos recursos dentro da InterSCity por meio de uma requisição fetch do tipo POST para a plataforma, com as informações bases de cada recurso sendo passadas por um JSON. O script completo se encontra no repositório do GitHub (CITRON, 2025).

```
for (const resource of stations) { // For each station resource
  const sendData = resource; // Resource data
  const result = await
fetch("http://10.10.104:8000/adaptor/resources", { // Request to create
resource
  method: "POST", // POST method
  headers: { // Request headers
```

```

        "Content-Type": "application/json",
    },
    body: JSON.stringify({ data: sendData }), // Request body with
resource data
});

```

Listagem 2 - Criação dos recursos

Assim, com os recursos criados dentro da plataforma InterSCity, a plataforma gera uma estrutura de informações referentes a cada recurso criado, mas a mais significativa delas é o UUID, que é um identificador único gerado aleatoriamente para cada recurso. Com base nestes identificadores, é possível acessar o recurso desejado dentro da plataforma.

A Listagem 3 apresenta os dados de um dos recursos criados dentro do InterSCity em JSON por meio do script de criação, como referenciado anteriormente pela Listagem 2, onde podem-se destacar a capacidade desse recurso (*sensor-temperature*) e seu UUID único. O JSON com todos os recursos criados se encontra disponível no GitHub (CITRON, 2025).

```

"resources" : [
  {
    "capabilities" : [
      "sensor-temperature"
    ],
    "city" : null,
    "collect_interval" : null,
    "country" : null,
    "created_at" : "2024-11-12T11:20:04.193Z",
    "description" : "station-9",
    "id" : 21,
    "lat" : -23.5158948777673,
    "lon" : -47.4656380735414,
    "neighborhood" : null,
    "postal_code" : null,
    "state" : null,
    "status" : "active",
    "updated_at" : "2024-11-12T11:20:04.193Z",
    "uri" : null,
    "uuid" : "dcd35c35-3688-45da-8c75-ffcdfe61692c"
  }
]

```

Listagem 3 - Recurso da estação 9

### 3.3.2 Geração de Dados

Para testar o funcionamento da aplicação, os recursos (*station*) criados precisavam ser alimentados com dados de temperatura. Para isso, foi desenvolvido um segundo script em Node.js para enviar temperaturas aleatórias variando de -5 a 45 °C para cada recurso de estação criado. Esse script se encontra disponível no GitHub (CITRON, 2025), deste documento.

O script se baseou essencialmente em buscar os recursos cuja capacidade e tipo correspondiam aos de “*sensor-temperature*”, assim obtendo os dados referentes a identificação (UUIDs) de cada recurso. Com base nos UUIDs obtidos, foi possível se comunicar com cada um dos recursos por meio de mensagens HTTP, e fazer requisições do tipo GET ou POST como explicado anteriormente na subseção 2.3.

Dessa forma, para cada estação (Recurso), o script envia uma temperatura randômica dentro do intervalo definido, preenchendo cada recurso de estação com novos dados a cada 15 segundos. A Listagem 4 apresenta o bloco de código referente ao envio de dados aos recursos de estação.

```
const updateTemperature = async () => {
  try {
    for (const station of stations) {

      const newTemperature = geraTemperatura(TEMP_MIN, TEMP_MAX); // Gera
      temperatura

      // Envia a nova temperatura para o servidor
      const response = await
      fetch(`http://10.10.10.104:8000/adaptor/resources/${station.uuid}/data`, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          data: {
            'sensor-temperature': [
              {
                timestamp: new Date().toISOString(),
                value: newTemperature,
              },
            ],
          },
        }),
      },
```

```

    }},
  });
}
} catch (error) {
  console.error('Erro ao gerar e enviar temperaturas:', error);
}
};

```

Listagem 4 - Enviando dados de temperatura

### 3.3.3 Desenvolvimento de Interface

Com base na criação dos recursos de estação e na geração de dados apresentada anteriormente, foi possível desenvolver o *dashboard* para a apresentação dos dados de monitoramento. Utilizou-se o React para o desenvolvimento da interface de monitoramento. O código para a interface foi colocado no GitHub do projeto (CITRON, 2025).

A estrutura principal do projeto em React foi definida da seguinte forma: a cada 15 segundos, o estado atual dos recursos (*station*) eram atualizados pela função ***updateStations*** apresentada no bloco de código da Listagem 5.

A função ***updateStations***, tinha o papel de buscar os recursos com a capacidade “*sensor-temperature*” existentes conforme o recurso de “*station*”, buscar os dados de temperatura referentes a esses recursos, e verificar a existência de alertas de temperatura, assim atualizando o estado em que se encontra o sensor (Capacidade) dentro de cada estação (Recurso) e seus respectivos alertas. Na Subseção 3.4 apresentam-se as categorias de alertas possíveis criados na aplicação.

```

const updateStations = useCallback(async () => {
  try {
    // Busca sensores reais
    const response = await
fetch(`${IP_ADDRESS}/discovery/resources?capability=sensor-temperature`);
    const data = await response.json();

    // Atualiza sensores com os reais encontrados, se houver
    const realSensors = data.resources?.map(resource => ({
      uuid: resource.uuid,
      description: resource.description,
      lastTemperature: '-',
      lastDate: '-',

```

```

    alert: false,
    alertAscAltLimit: false,
    alertAscLimit: false,
    alertAscParcLimit: false,
    alertDecAltLimit: false,
    alertDecLimit: false,
    alertDecParcLimit: false,
    alertInc: false,
  ))) || defaultStations;
  // Busca as temperaturas para os sensores
  const updatedResources = await searchTemperatures(realSensors);
  // Verifica alertas
  const checkAlerts = checkTemperatures(updatedResources);
  // Atualiza o estado com as estações verificados
  setStations(checkAlerts);

} catch (error) {
  console.error('Erro ao atualizar sensores:', error);
}
}, [checkTemperatures]);

```

Listagem 5 - Função updateStations

### 3.4 Categorias de Alertas

As categorias de alertas se baseiam em categorizar as temperaturas em 7 tipos diferentes, conforme a temperatura seja muito alta ou muito baixa. A Tabela 1 apresenta os alertas para a aplicação.

Condição	Categoria	Mensagem de Alerta
Temperatura até 5 °C acima do limite máximo definido	Temperatura parcialmente alta	Temperatura parcialmente acima do limite permitido!
Temperatura entre 5 °C até 10 °C acima do limite máximo definido	Temperatura alta	Temperatura acima do limite permitido!
Temperatura entre 10 °C até 15 °C acima do limite máximo definido	Temperatura muito alta	Temperatura muito acima do limite permitido!
Temperatura até 5 °C abaixo do limite mínimo definido	Temperatura parcialmente baixa	Temperatura parcialmente abaixo do limite permitido!
Temperatura entre 5 °C até 10 °C abaixo do limite mínimo definido	Temperatura baixa	Temperatura abaixo do limite permitido!



Temperatura com variação superior a 10 °C abaixo do limite mínimo definido	Temperatura muito baixa	Temperatura muito abaixo do limite permitido!
Temperatura com uma variação superior a 15 °C acima do limite máximo definido	Possível princípio de incêndio	Possível princípio de incêndio!

Tabela 1 - Alertas

### 3.5 Diagramas

Com base no levantamento de requisitos feito, e apresentado na Seção 3.1, foram criados quatro diagramas, com o propósito de elaborar o funcionamento da aplicação e como o usuário interage com ela.

Na Figura 2, tem-se o diagrama de casos de uso, o qual expressa as ações de uso do usuário com o sistema. Neste diagrama há dois atores: Funcionário do Metrô e Sistema. O funcionário pode ajustar os limites de temperatura (temperatura máxima e mínima), e monitorar a temperatura.

O Caso de uso *Monitorar a temperatura* inclui obter a temperatura atual nas estações, pois é a informação retornada ao usuário (funcionário do metrô) por meio do caso de uso. O ator Sistema possui como casos de uso emitir alertas de temperatura fora do limiar, e emitir alertas de princípio de incêndio, onde ambos incluem obter a temperatura atual na estação.

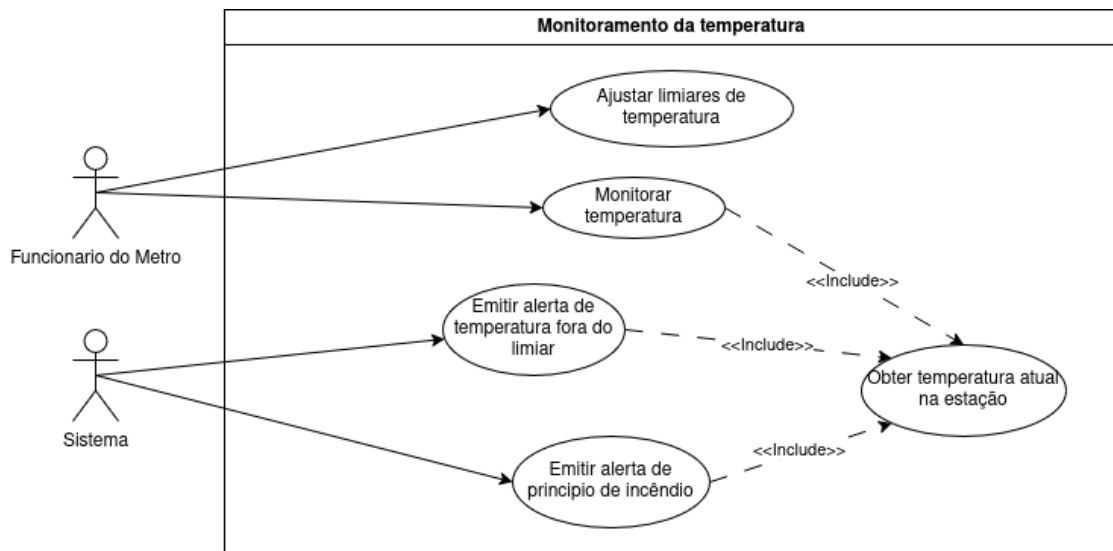


Figura 2 - Diagrama de casos de uso

Na Figura 3, o diagrama de sequência reflete a sequência de eventos relacionados ao caso de uso *Monitorar temperatura*. Dessa forma, o diagrama segue a seguinte sequência:

1. O Ator funcionário do metrô acessa o monitor de temperatura;
2. O objeto monitor de temperatura busca novos sensores;
3. O objeto sensores retorna as informações dos sensores encontrados;
4. O objeto monitor de temperatura busca os dados dos sensores encontrados;
5. O objeto sensores retorna os dados de Temp (Temperatura) e Date (Data);
6. O objeto monitor de temperatura verifica o tipo de alerta com base no atributo Temp (Temperatura);
7. O objeto monitor de temperatura retorna ao ator funcionário do metrô o status dos sensores.

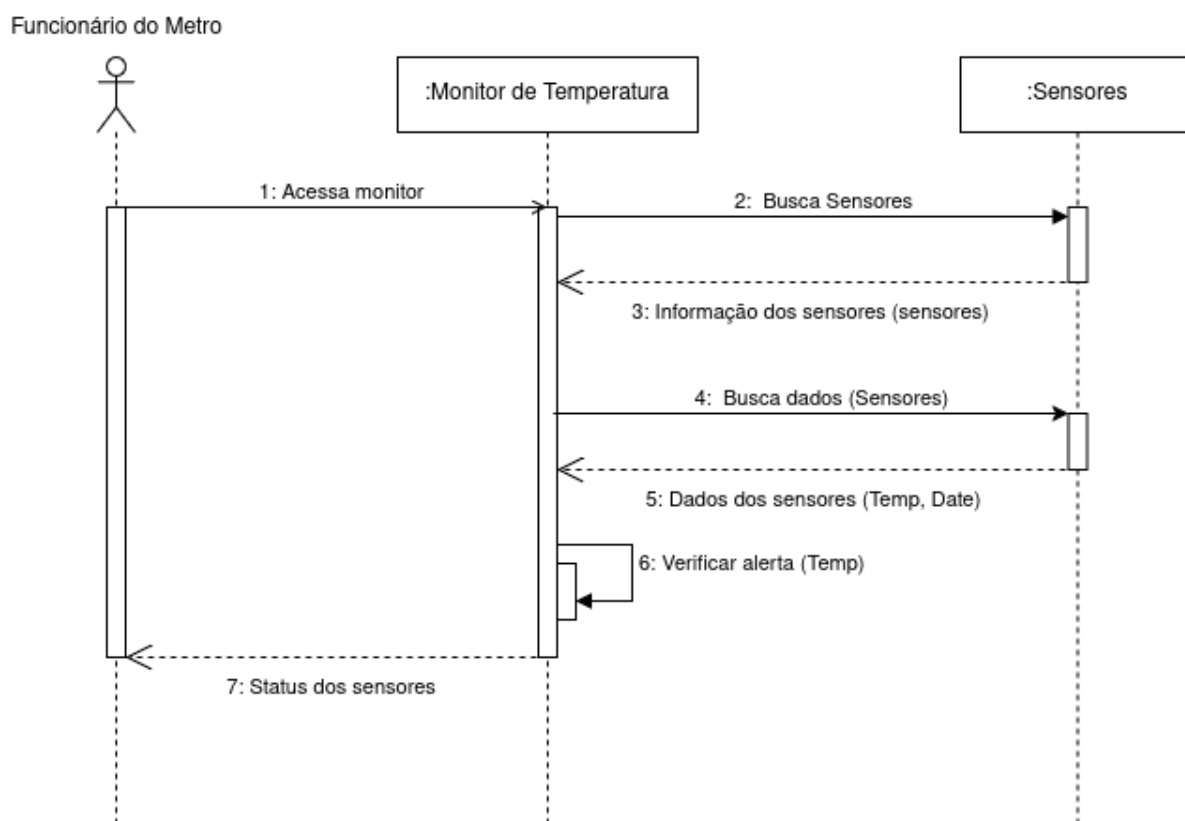


Figura 3 - Diagrama de sequência 1

O Diagrama de Sequência 2 (Figura 4) apresenta a sequência de eventos para o caso de uso *Ajustar limiares de temperatura*. Dessa forma, o diagrama segue a seguinte sequência:

1. O ator funcionário do metrô ajusta o limiar do objeto Monitor de Temperatura, passando, como parâmetros, TempMax (Temperatura Máxima) e TempMin (Temperatura Mínima);
2. O objeto Monitor de Temperatura redefine os limiares com base nos parâmetros TempMax (Temperatura Máxima) e TempMin (Temperatura Mínima);
3. O objeto Monitor de temperatura verifica os alertas com base no limiar redefinido.

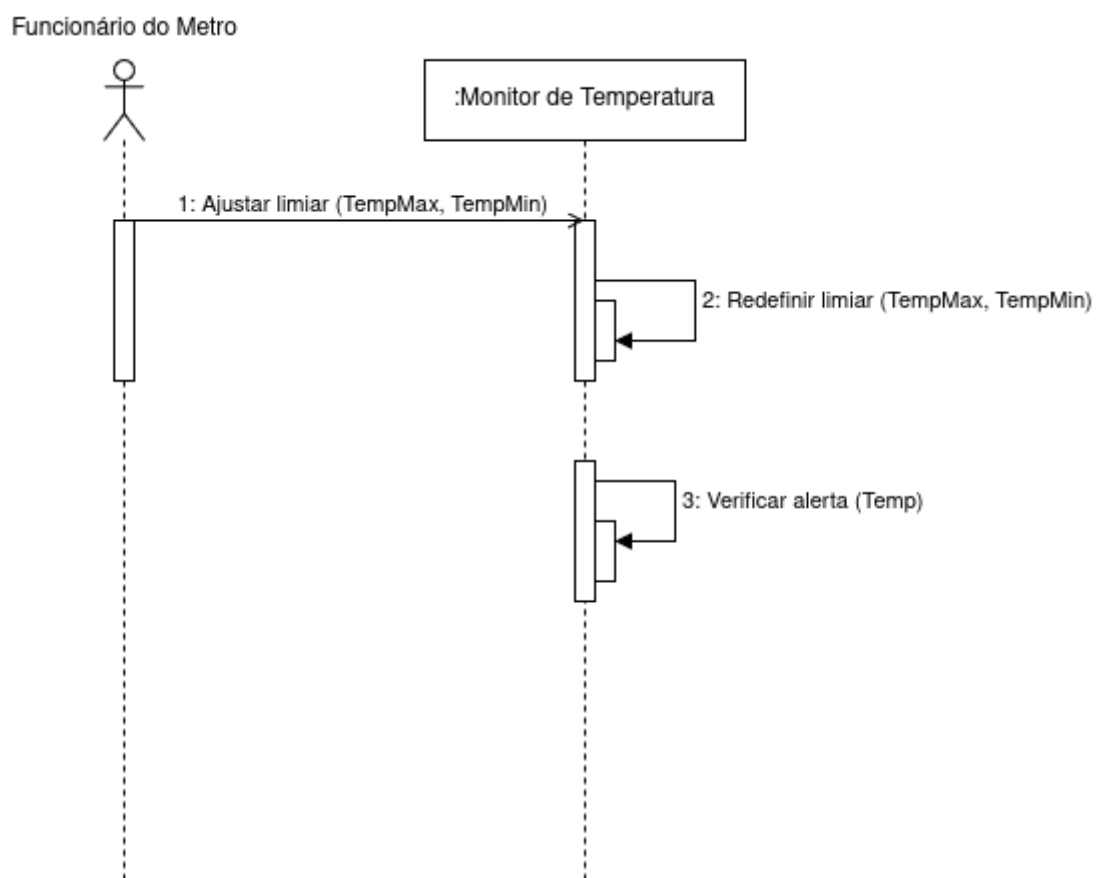


Figura 4 - Diagrama de sequência 2

Por fim, o diagrama de atividades apresentado na figura 5 reflete o fluxo de funcionamento da aplicação. A seguir, é apresentado o caminho do ponto de partida ao ponto final em relação às sequências de ações do usuário:

1. O usuário acessa o monitor de temperatura;
2. O usuário pode escolher entre modificar os limiares de temperatura;

- 2.1. Caso tenha modificado o limiar, os limiares são reajustados, e após isso, é feita a busca de temperaturas das estações
- 2.2. Caso não tenha reajustado, a busca de temperaturas das estações é feita com o limiar atual;
3. É feita uma verificação na temperatura da estação;
  - 3.1. Caso a temperatura esteja dentro do limiar, exibe a temperatura atual e finaliza o processo;
  - 3.2. Caso esteja fora do limiar, verifica tipo de alerta;
    - 3.2.1. Caso a temperatura seja de princípio de incêndio, exibe alerta de princípio de incêndio e finaliza o processo;
    - 3.2.2. Caso não seja uma temperatura de princípio de incêndio, exibe alerta de temperatura fora do limite e finaliza o processo.

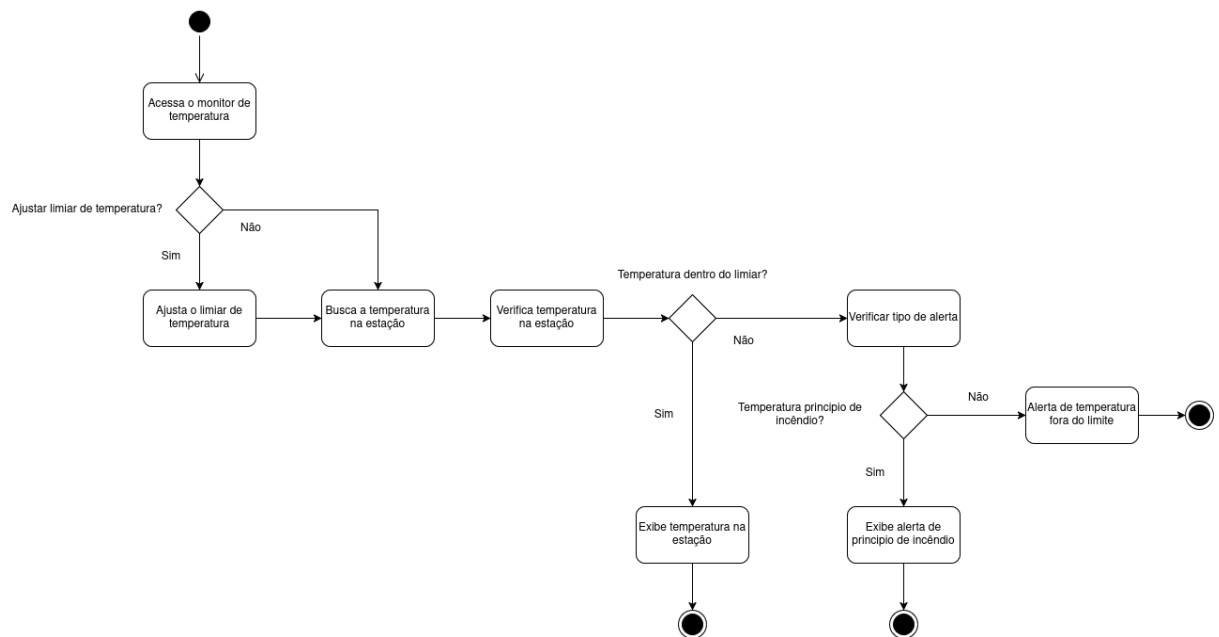


Figura 5 - Diagrama de atividade

## 4 RESULTADOS E DISCUSSÕES

Este capítulo consiste em apresentar as funcionalidades da aplicação desenvolvida (Seção 4.1), bem como os resultados obtidos (Seção 4.2) e uma análise dos resultados (Seção 4.3).

### 4.1 Funcionalidade da Aplicação

A aplicação possui funcionalidades relacionadas ao ajuste de temperaturas (limiares mínimo e máximo), e à apresentação dos dados de sensores de cada estação do metrô. Descrições para cada uma delas são apresentadas a seguir.

**Ajustar a temperatura máxima e temperatura mínima:** Como é possível observar na Figura 6, é possível ajustar conforme necessário os limites de temperatura máxima e mínima a serem usados como métricas para avaliar cada estação e gerar os alertas.

Figura 6 - Editar limiares

**Apresentar os dados dos recursos de cada estação (UUID, nome, data, temperatura):** Para cada recurso encontrado referente a uma estação do metrô com a capacidade de sensor de temperatura, é adicionado um cartão (do inglês, *card*), no qual são apresentados as informações referentes ao recurso daquela estação, como o nome, UUID (Identificador único), data e hora de coleta do dado, e caso seja pertinente, uma mensagem de alerta de temperatura. Um exemplo de card é apresentado na Figura 7.



Figura 7 - Card da estação

**Apresentar alertas conforme a temperatura da estação:** Conforme os limites de temperatura máxima e mínima forem definidos, mensagens serão emitidas dentro do card específico informando que a temperatura está fora dos limites definidos. Mensagens de alerta irão depender do quão superior a temperatura for da temperatura máxima, sendo que a tonalidade avermelhada vai ganhando um aspecto mais escuro.

O mesmo ocorre para temperaturas abaixo do mínimo. Nesse caso, a coloração azul ganha tons mais escuros conforme mais baixa for a temperatura. As Figuras 8 a 14 ilustram os diferentes alertas gerados pela aplicação para uma temperatura máxima de 12 °C e mínima de 10 °C. Por exemplo, o card apresentado na Figura 8, apresenta as informações referentes a estação 6, ou mais especificamente, as informações do recurso station-6, com os dados de identificação desse recurso (UUID), a última temperatura registrada neste recurso, data de atribuição da informação ao recurso e por fim o tipo de alerta, que neste caso refere-se a uma temperatura muito acima do limite permitido, o que corresponde com a tonalidade de vermelho mais escurecido.

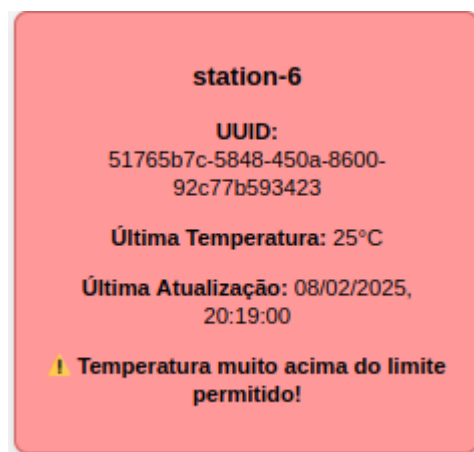


Figura 8 - Card em alerta de temperatura muito acima do limite

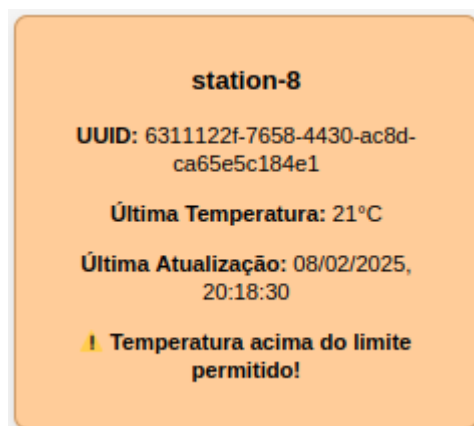


Figura 9 - Card em alerta de temperatura acima do limite

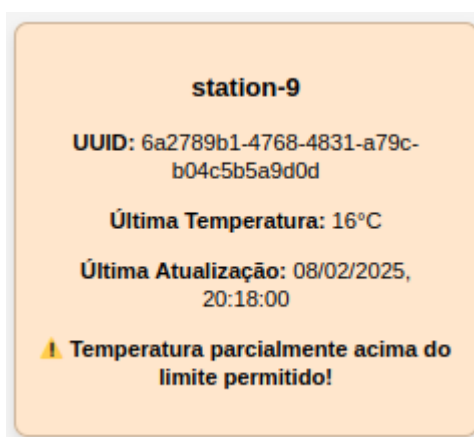


Figura 10 - Card em alerta de temperatura parcialmente acima do limite



Figura 11 - Card em alerta de temperatura muito abaixo do limite



Figura 12 - Card em alerta de temperatura abaixo do limite



Figura 13 - Card em alerta de temperatura parcialmente abaixo do limite



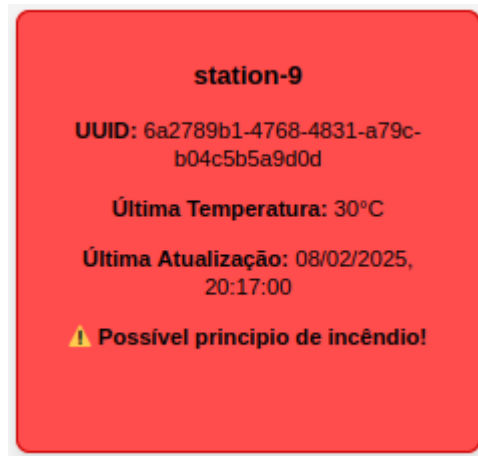


Figura 14 - Card em alerta de princípio de incêndio

## 4.2 Resultados Obtidos

A aplicação inicialmente possui um estado de default, onde são criados 9 cards iniciais com informações padrão. Esse estado inicial ocorre inicialmente quando nenhum recurso de estação dentro da plataforma InterSCity foi localizado, assim criando uma estrutura padrão enquanto aguarda as informações dos recursos. A Figura 15 representa este estado inicial da aplicação.

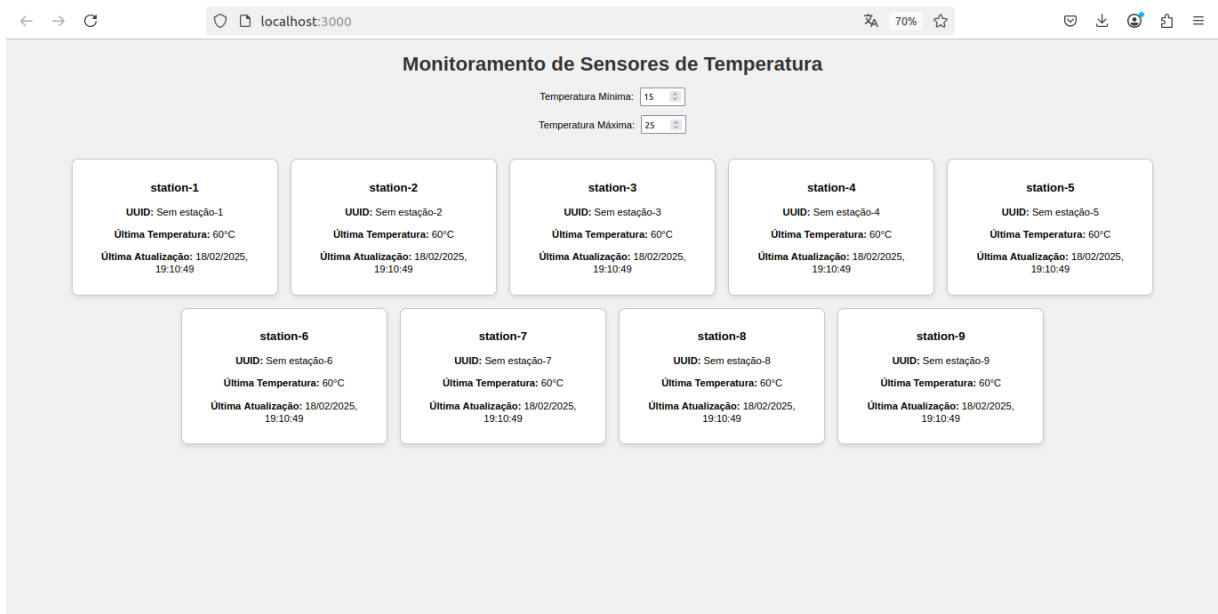


Figura 15 - Monitor de temperatura sem dados

Na Figura 16 é apresentada a aplicação em funcionamento. No dashboard apresentado, cada card representa uma estação de metrô; mais especificamente, cada card representa um sensor de temperatura, caracterizando com isso as principais informações dessa estação (UUID, nome, temperatura, data). É possível notar os diferentes tipos de alertas gerados para cada estação, conforme a variação das cores.

A temperatura máxima, configurada por padrão, se encontra em 25 °C. Da mesma forma, a temperatura mínima configurada por padrão se encontra em 15 °C. Observando de forma mais detalhada os cards das estações (Figura 16), é possível notar que as estações 2 e 9 se encontram em alerta de temperatura muita acima do limite permitido.

Considerando as condições de ocorrência dos alertas apresentados no Capítulo 3 (Seção 3.4), as estações 2 e 9 registraram uma temperatura de 40 °C, o que corresponde ao alerta de temperatura muita acima do limite (temperatura entre 10 a 15 °C acima do limite máximo), dado que a temperatura sensorizada se encontra 15 °C acima do limite máximo.

Em contrapartida, a estação 1 registrou uma temperatura de -5 °C, estando dentro do intervalo de 10 a 15 °C abaixo do limite mínimo. Assim, gerando um alerta de temperatura muito abaixo do limite permitido.

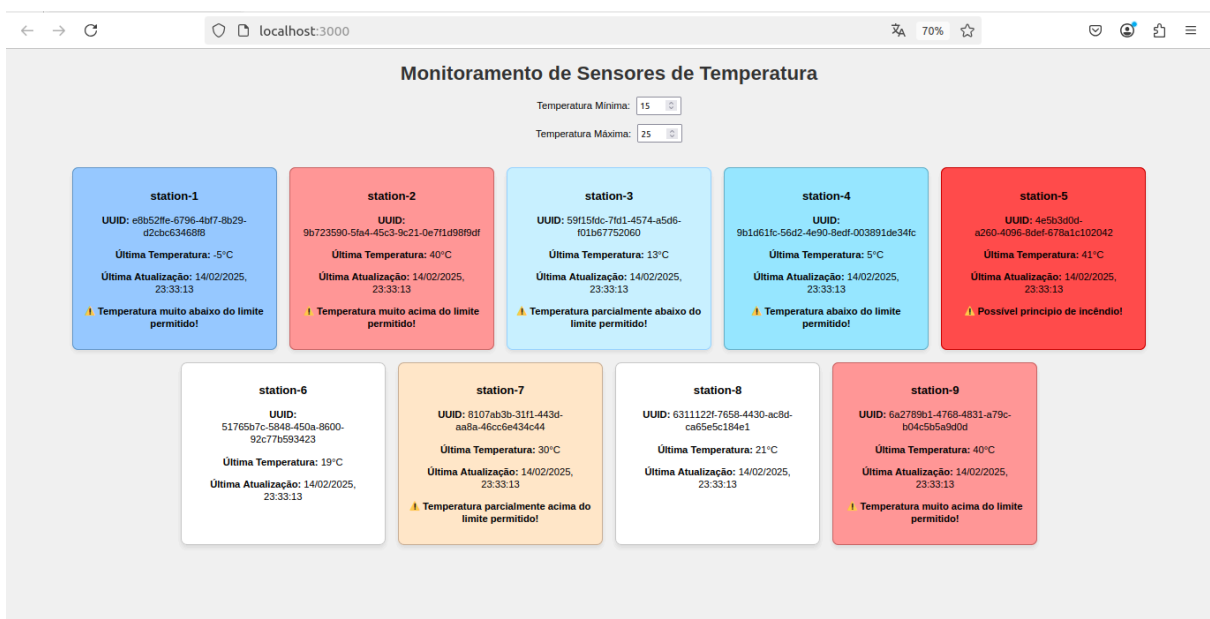


Figura 16 - Monitor de temperatura

Ajustando-se os limites de temperatura máxima e mínima do monitor para 10 °C (Temperatura mínima) e 12 °C (Temperatura máxima), são obtidos os resultados apresentados na Figura 17, onde, para a estação 6, a temperatura se encontra acima do limite, com alerta de temperatura acima do limite permitido.

Para a estação 1, a temperatura se encontra abaixo do limite, gerando um alerta de temperatura muito abaixo do limite permitido. Para as estações 2, 5 e 9, a temperatura se encontra dentro dos limites máximos e mínimos, não gerando nenhum alerta. As estações 3 e 4, se encontram com uma temperatura acima do limite, com um alerta de possível princípio de incêndio.

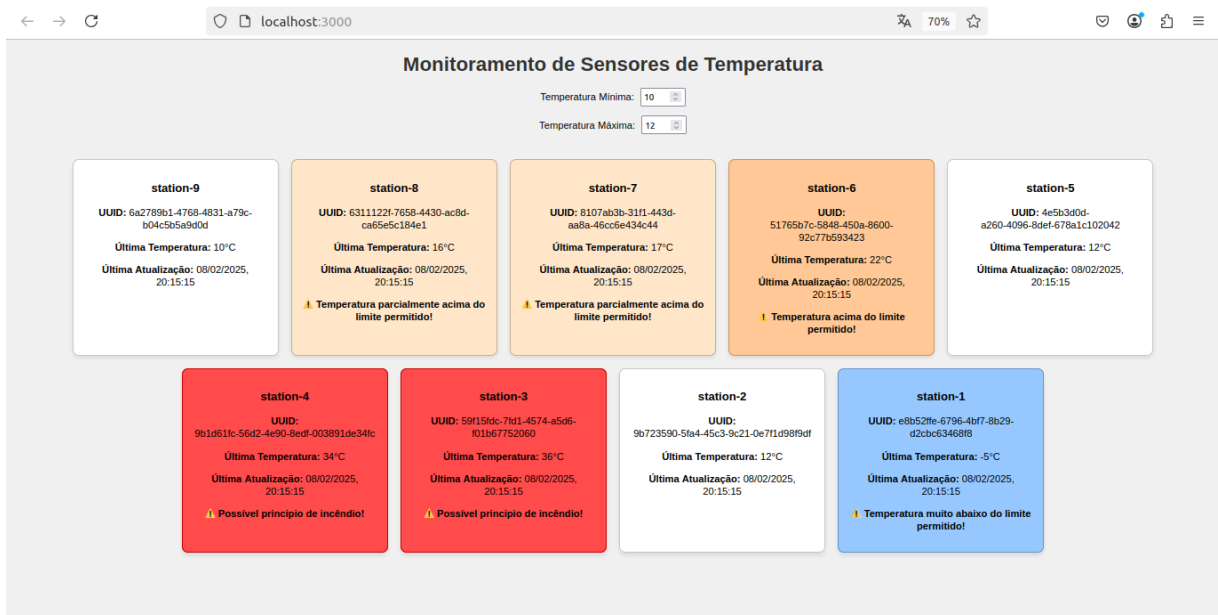


Figura 17 - Monitor de temperatura ajustado

A Figura 18 apresenta os status de comunicação da aplicação com a plataforma InterSCity. É possível observar que o status retornado para cada requisição feita foi de 200, o que corresponde a OK para o retorno da requisição.

Analisando melhor a comunicação, é possível notar que inicialmente é feita uma requisição para a obtenção dos dados das estações (Recursos), e após a obtenção desses dados, são feitas nove requisições de data, onde cada uma dessas requisições é feita especificamente para obter os dados de temperatura (Capacidade) de uma estação (Recurso) que foi cadastrada dentro do InterSCity.

A Figura 19 apresenta uma das requisições feitas para a obtenção de dados de temperatura. Os dados obtidos por meio dessa requisição são apresentados na Figura 20, com as informações de value (temperatura) e data (coleta da informação). Analisando melhor os dados da Figura 20, além de apresentar a informação atual do recurso, é obtido também dados anteriormente já registrados neste mesmo recurso.

Status	Método	Domínio	Arquivo	Iniciador	Tipo	Transferido	Tamanho	0 ms
304	GET	10.10.10.10...	resources?capability=sensor-temperature	bundle.js:177 (f...	json	em cache	3,43 kB	33 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,69 kB	41,93 kB	340 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,70 kB	41,95 kB	278 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,69 kB	41,93 kB	1048 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,68 kB	41,92 kB	1044 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,69 kB	41,93 kB	1044 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,70 kB	41,95 kB	278 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,68 kB	41,92 kB	341 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,68 kB	41,92 kB	855 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,71 kB	41,95 kB	388 ms
304	GET	10.10.10.10...	resources?capability=sensor-temperature	bundle.js:177 (f...	json	em cache	3,43 kB	58 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,74 kB	41,98 kB	261 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,75 kB	41,99 kB	454 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,74 kB	41,98 kB	395 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,72 kB	41,96 kB	754 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,74 kB	41,98 kB	852 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,75 kB	41,99 kB	905 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,73 kB	41,97 kB	904 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,73 kB	41,97 kB	563 ms
200	GET	10.10.10.10...	data	bundle.js:89 (fe...	json	42,75 kB	41,99 kB	900 ms

20 requisições | 762,05 kB / 768,87 kB transferidos | Tempo: 16,00 s

Figura 18 - Status de comunicação

Arquivo

Ini

Tr

Ta

C

data

bu

j: 42, 41,

C

data

bu

j: 42, 41,

C

resources

bu

j: em 3,4

C

data

bu

j: 42, 42,

C

data

bu

j: 42, 42,

C

data

bu

j: 42, 42,

C

data

bu

j: 42, 42,

C

data

bu

j: 42, 42,

C

data

bu

j: 42, 42,

Cabeçalhos

Cookies

Requisição

Resposta

Tempos

Stack Trace

Filtrar cabeçalhos

Bloquear

Reenviar

GET

Scheme: http

Host: 10.10.10.104:8000

Filename: /collector/resources/51765b7c-5848-450a-8600-92c77b593423/data

Endereço: 10.10.10.104:8000

Status

200 OK

Versão

HTTP/1.1

Transferido

42,77 kB (tamanho 42,01 kB)

Referrer Policy

strict-origin-when-cross-origin

Resolução de DNS

Sistema

Figura 19 - Requisição de dados do recurso de station

CabeçalhosCookiesRequisiçãoRespostaTemposStack Trace

Filtrar propriedades

JSONBruto

▼ resources: [ {... } ]

▼ 0: Object { uuid: "51765b7c-5848-450a-8600-92c77b593423", capabilities: { ... } }

uuid: "51765b7c-5848-450a-8600-92c77b593423"

▼ capabilities: Object { "sensor-temperature": [ ... ] }

▼ sensor-temperature: [ { ... }, { ... }, { ... }, { ... }, { ... }, { ... }, { ... }, { ... }, { ... }, { ... }, { ... }, ... ]

▼ 0:

value: 6

date: "2024-12-08T19:10:51.627Z"

▼ 1:

value: 44

date: "2024-12-08T19:11:06.650Z"

▶ 2:

▶ 3:

▶ 4:

▶ 5:

▶ 6:

▶ 7:

▶ 8:

▶ 9:

▶ 10:

▶ 11:

▶ 12:

▶ 13:

▶ 14:

Figura 20 - Informação de retorno da requisição

### 4.3 Análise dos Resultados e Lições Aprendidas

Considerando os resultados apresentados nas subseções anteriores, constata-se que a aplicação está se comunicando corretamente com a plataforma InterSCity, conseguindo acessar os dados dos sensores e obter as temperaturas, o que demonstra uma correta integração entre a aplicação e a plataforma.

Além disso, as funcionalidades da aplicação seguem o escopo inicial definido no levantamento de requisitos e nos diagramas apresentados no Capítulo 3. Assim, a aplicação se demonstra funcional, apresentando corretamente os alertas de temperatura para cada estação e suas respectivas informações.

Com base nos resultados e no processo de desenvolvimento, vale destacar alguns pontos centrais que foram absorvidos. Como apresentado previamente no trabalho, é possível buscar recursos e coletar dados da plataforma. Esse processo, conforme comprovado durante o desenvolvimento descrito nesta seção, demonstra também as principais funcionalidades oferecidas pela plataforma.

Outro ponto a ser destacado é a capacidade de manter os dados já atribuídos anteriormente a um recurso armazenados dentro da plataforma. Dessa forma, o InterSCity possibilita trabalhar não apenas com os últimos dados adicionados, mas todos que já foram registrados previamente como uma lista ou histórico de dados. Sendo assim, é possível extrair diferentes informações como oscilação ou variação das informações dos sensores. Porém, esse espaço de armazenamento de dados por recurso ainda é limitado, o que levanta um ponto negativo da plataforma, que não permite remover dados atribuídos a um recurso ou limpar essa lista de dados. Assim, caso o limite de armazenamento seja atingido, não é possível atribuir novos dados aos recursos.

Observa-se, também, que a plataforma InterSCity permite desenvolver diferentes tipos de aplicações para diferentes fins, como aplicações web (como apresentado previamente neste capítulo), mas também aplicações móveis e diferentes tipos de ferramentas de software.

Por fim, é possível registrar e utilizar diferentes tipos de recursos e capacidades dentro da plataforma, não se limitando apenas a sensores de temperatura, mas também de umidade ou câmeras, semáforos, iluminação, dentre outros tipos de recursos e capacidades que podem ser abordados utilizando a plataforma. Favorece-se, assim, o desenvolvimento de aplicações variadas no contexto de Cidades Inteligentes.

## 5 CONCLUSÃO

Este capítulo conclui o trabalho desenvolvido e descrito nos capítulos anteriores. Resumem-se, aqui, os resultados obtidos neste trabalho e quais foram suas contribuições (Seção 5.1), quais foram as limitações encontradas (Seção 5.2) e possíveis melhorias para trabalhos futuros (Seção 5.3).

### 5.1 Resultados e Contribuições

O objetivo central deste trabalho foi apresentar de forma aplicada o funcionamento da plataforma InterSCity dentro do escopo de desenvolvimento de aplicações. De forma geral, visou-se compreender como a plataforma pode ser utilizada no desenvolvimento de tecnologias para Cidades Inteligentes, demonstrando seus principais recursos e funcionalidades oferecidas. Assim, podem-se destacar os seguintes resultados:

- **Integração da plataforma InterSCity com a aplicação:** Foi possível integrar a aplicação com a plataforma, conseguindo adicionar os recursos (*Stations*), adicionar dados a cada um dos recursos, buscar os recursos criados na plataforma por meio da aplicação, e buscar os dados de cada uma das estações;
- **Aplicação para monitoramento de temperatura nas estações:** Foi possível fazer com que a aplicação apresentasse uma interface simples e intuitiva para monitorar as temperaturas obtidas de cada estação.
- **Compreensão da plataforma InterSCity:** Com base no desenvolvimento prático e utilização da plataforma, foi possível observar as principais funcionalidades oferecidas, como descoberta de recursos (*stations*), criação de novos recursos dentro da plataforma, adição de dados, e acesso ao histórico de dados de cada recurso.

Este trabalho contribui para o desenvolvimento de aplicações para Cidades Inteligentes, apresentando uma forma de integrar a plataforma InterSCity com o desenvolvimento de ferramentas tecnológicas voltadas para o contexto de Cidades Inteligentes. Ademais, este trabalho apresentou uma aplicação prática dessa plataforma integrada ao desenvolvimento de uma aplicação para o monitoramento de temperaturas em estações de metrô. Entre as principais contribuições, destacam-se:

- **Integração de tecnologias:** O trabalho demonstrou como utilizar uma plataforma para aplicações para Cidades Inteligentes, baseado em um cenário simulado que reflete um cenário real, facilitando criar e desenvolver situações de testes para as aplicações.
- **Aplicação para monitoramento:** A aplicação desenvolvida se adequa a situações reais em centros urbanos com grande movimentação e sendo uma solução para situações de alta variação de temperatura encontradas dentro de estações de metrô.
- **Modelo Reutilizável:** A arquitetura modular do sistema pode ser aplicada em outros cenários de monitoramento ambiental, como umidade ou qualidade do ar, além de ser um modelo para desenvolvimento de outras aplicações voltadas para o contexto de Cidades Inteligentes.

## 5.2 Limitações Encontradas

Durante a fase de desenvolvimento e teste da aplicação, a plataforma InterSCity mostrou algumas limitações. Dentre elas, destacam-se:

- **Conexão com a plataforma:** Os serviços do InterSCity, em muitas situações, ficavam indisponíveis, se mostrando um pouco instáveis.
- **Funcionamento da plataforma:** As configurações de instalação e funcionamento da plataforma são muito dependentes de diversos pacotes de instalação como Mongo, VirtualBox, Ruby, dentre outros pacotes que estão sujeitos a atualizações. Assim, pode-se gerar quebras de pacotes e falhas de funcionamento na plataforma.
- **Limitações dos Recursos:** A plataforma não oferece nenhum serviço de exclusão de dados de um recurso ou atualização de dados existentes. Caso um recurso tenha seu espaço de armazenamento de dados preenchido, a InterSCity não disponibiliza nenhum serviço de exclusão, restringindo a inclusão de novos dados referentes a um recurso. Dessa forma, as únicas opções são utilizar os dados já armazenados sem a possibilidade de adicionar novos dados, ou criar um novo recurso.



### 5.3 Trabalhos Futuros

Considerando todos os serviços oferecidos pela plataforma InterSCity, é possível ampliar e aprofundar mais o desenvolvimento de diversos tipos de ferramentas por meio da plataforma.

Existem alguns recursos oferecidos pelo InterSCity que não foram abordados ou foram pouco explorados durante o desenvolvimento que abrem margem para possíveis melhorias e aprimoramentos para a aplicação de monitoramento, mas também no escopo geral de outros projetos. Nesse sentido, destacam-se as seguintes atualizações e aprimoramentos possíveis:

- **Implementação e Simulação de Atuadores:** Como apresentado no Capítulo 2, a InterSCity oferece o serviço de Actuator Controller (ou, Controlador de Atuação), que consiste em controlar e manipular recursos por meio de atuações. Assim, um aprimoramento para a aplicação desenvolvida neste trabalho seria a adição de controladores ao projeto, o que permitiria a manipulação de temperaturas dentro de cada estação e, de forma conjunta, aprofundar as possibilidades de uso do InterSCity dentro do desenvolvimento de uma aplicação.
- **Análise de Histórico de Dados dos Recursos:** Outra possibilidade de aprimoramento consiste em se aprofundar no acesso aos dados históricos de cada recurso. Durante o desenvolvimento da aplicação, a ideia central era obter os dados mais recentes de cada recurso (estação ou *station*). Uma atualização possível seria criar um gráfico para cada estação (Recurso), onde seria possível observar a variação de temperatura ao longo do tempo. Com base no histórico de dados, também seria possível criar um histórico único de atualização de dados de cada recurso (*Station*) e obter dados como taxas de variação de temperatura ou padrões de temperatura conforme a mais recorrente.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARUERI NA REDE. Linha 8-Diamante: calor extremo e ar-condicionado desligado obrigam passageiros a descer de trens. 2024. Disponível em: <<https://baruerinarede.com.br/linha-8-diamante-calor-extremo-e-ar-condicionado-desligado-obrigam-passageiros-a-descer-d-e-trens/>>. Acesso em: 2 nov. 2024.

CCSL DO IME/USP. Usando a plataforma InterSCity. 2020. Disponível em: <<https://www.youtube.com/watch?v=tItCiCsA4Xw>>. Acesso em: 1 set. 2024.

CUDE. Thinking about becoming a smart city? 10 benefits of smart cities. 2018. Disponível em: <<https://www.plantemoran.com/explore-our-thinking/insight/2018/04/thinking-about-becoming-a-smart-city-10-benefits-of-smart-cities>>. Acesso em: 25 jan. 2025.

ESPOSTE, A. M. D.; KON, F.; COSTA, F. M.; LAGO, N. InterSCity: A scalable microservice-based open source platform for smart cities. In: Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems, 2017, p. 35–46.

G1 RIO DE JANEIRO. Passageiros reclamam de calor e ar-condicionado sem funcionar no metrô do Rio. 2020. Disponível em: <<https://g1.globo.com/rj/rio-de-janeiro/noticia/2020/01/09/passageiros-reclamam-de-calor-e-ar-condicionado-sem-funcionar-no-metro-do-rio.ghtml>>. Acesso em: 2 nov. 2024.

G1 SÃO PAULO. Novo trem do metrô de SP é 3 graus mais frio que o convencional. 2009. Disponível em: <[https://g1.globo.com/Noticias/SaoPaulo/0,,MUL1064714-5605\\_00-NOVO+TREM+DO+METRO+DE+SP+E+GRAUS+MAIS+FRIO+QUE+O+CONVENCIONAL.html](https://g1.globo.com/Noticias/SaoPaulo/0,,MUL1064714-5605_00-NOVO+TREM+DO+METRO+DE+SP+E+GRAUS+MAIS+FRIO+QUE+O+CONVENCIONAL.html)>. Acesso em: 2 nov. 2024.

GLOBOPLAY. Calor no metrô: com problema de ar-condicionado, vagões ficam com temperatura acima dos 30 graus. 2024. Disponível em: <<https://globoplay.globo.com/v/12454041/>>. Acesso em: 2 nov. 2024.

GRAND VIEW RESEARCH. Smart Cities Market Size & Trends Analysis Report By Application (Smart Transportation, Smart Utilities, Smart Governance), By Region, And Segment Forecasts, 2023 - 2030. Disponível em: <https://www.grandviewresearch.com/industry-analysis/smart-cities-market>. Acesso em: 25 jan. 2025.

IBM. Internet das coisas (IoT). Disponível em: <<https://www.ibm.com/br-pt/topics/internet-of-things>>. Acesso em: 20 jan. 2025.

INTERSCITY. Sobre o projeto InterSCity. Disponível em: <<https://interscity.org/about/>>. Acesso em: 25 jan. 2025.

CITRON, Jhon. Repositório do projeto TCC. Disponível em: <<https://github.com/JhonWislin/TCC>>. Acesso em: 25 jan. 2025.

MDPI. Sustainability, 12(19), 8038. 2020. Disponível em: <<https://www.mdpi.com/2071-1050/12/19/8038>>. Acesso em: 25 jan. 2025.

METRÓPOLES SÃO PAULO. Ar-condicionado é vilão das reclamações nos vagões do metrô de SP. 2024. Disponível em: <<https://www.metropoles.com/sao-paulo/ar-condicionado-e-vilao-das-reclamacoes-nos-vagoes-do-metro-de-sp>>. Acesso em: 2 nov. 2024.

NEC. Conceito de Smart City: Uma Nova Dinâmica para o Desenvolvimento Urbano. 2024. Disponível em: <<https://docs.nec.com.br/conceito-smart-city-nova-dinamica>>. Acesso em: 25 jan. 2025.

SECTI. O que são cidades inteligentes? 2024. Disponível em: <<https://www.secti.df.gov.br/o-que-sao-cidades-inteligentes/>>. Acesso em: 25 jan. 2025.

SILVA, G. R.; ESPOSTE, A. M. D.; LAGO, N.; KON, F.; COSTA, F. M. Design and evaluation of a scalable smart city software platform with large-scale simulations. Future Generation Computer Systems, 2018. Disponível em: <[https://InterSCity.org/assets/design\\_evaluation\\_scalable\\_platform\\_-\\_FGCS\\_2018.pdf](https://InterSCity.org/assets/design_evaluation_scalable_platform_-_FGCS_2018.pdf)>.