Sistema de coleta de dados de Estações Meteorológicas

tecsus

2025.1

Sumário

1. Introdução

Este documento descreve o desenvolvimento do sistema de monitoramento ambiental da Tecsus, voltado para coleta, processamento, armazenamento e visualização de dados meteorológicos em tempo real. O sistema será utilizado por administradores e usuários interessados em dados climáticos, promovendo a conscientização ambiental, apoio à prevenção de desastres naturais e incentivo ao aprendizado escolar.

2. Finalidade

A finalidade deste projeto é fornecer uma solução tecnológica escalável, acessível e educativa, que integre sensores ambientais com uma plataforma digital robusta, permitindo:

- Acompanhamento em tempo real de condições meteorológicas.
- Tomada de decisão rápida diante de alertas climáticos.
- Uso pedagógico por escolas do ensino médio.

3. Objetvo

O objetivo principal deste projeto é desenvolver uma plataforma moderna e eficiente para o monitoramento ambiental por meio de sensores IoT, possibilitando a coleta, o processamento e a visualização de dados meteorológicos em tempo real. Além disso, o sistema busca promover a educação ambiental, oferecendo materiais didáticos sobre os parâmetros climáticos monitorados, e apoiar ações preventivas relacionadas a desastres naturais por meio de alertas automáticos. Dessa forma, alia-se inovação tecnológica à responsabilidade social e educacional, atendendo tanto às necessidades de gestão ambiental quanto ao incentivo ao aprendizado escolar.

4. Escopo

O sistema será desenvolvido para abranger as seguintes funcionalidades:

- Cadastro e gerenciamento de **estações meteorológicas**, **sensores**, **parâmetros climáticos**, **usuários** e **alertas**.
- **Recepção de dados** em tempo real das estações.
- **Dashboards interativos** com gráficos e relatórios.
- **Geração automática de alertas** com base em parâmetros definidos.
- **Registro de dados históricos** com datalogger.
- **Tutoriais explicativos** sobre conceitos meteorológicos.
- **Montagem física da estação meteorológica** com sensores.

5. Visão Geral

O sistema de monitoramento ambiental da Tecsus será composto por uma plataforma web acessível, dividida em dois ambientes principais: o painel administrativo e o portal público. O painel administrativo será destinado aos responsáveis pela gestão das estações e sensores, permitindo o cadastro, edição e monitoramento em tempo real dos dados coletados. Já o portal público será voltado à visualização de informações meteorológicas por qualquer usuário, com foco em acessibilidade e uso educacional. Os dados serão exibidos por meio de dashboards interativos e gráficos atualizados, possibilitando análises detalhadas. A arquitetura adotada será escalável e modular, garantindo flexibilidade para futuras expansões e integrações com outras ferramentas ou sensores ambientais.

6. Visão de Casos de Uso

Este sistema será utilizado por dois tipos principais de usuários: administradores e usuários públicos.

Administrador:

- Gerenciar estações, parâmetros, usuários e alertas.
- Configurar sensores e montar estações.
- Monitorar dados em tempo real e gerar relatórios.
- Gerar alertas e visualizar logs.

Usuário:

- Acessar tutoriais meteorológicos.
- Visualizar dados ambientais em tempo real.

7. Documentação API

7.1 Objetivo

O objetivo deste documento tem como informar ao leitor sobre asmecânicas que foram desenvolvidas em nossa aplicação.

7.2 Tags

Abaixo estão listadas quais foram as seguintes tags utilizadas:

- **auth**
- **user**
- **station**
- **typeAlert**
- **typeParameter**
- **alert**
- **receiveJson**

7.3 Endpoints

- /auth/login

  Método: **POST**

  Objetivo: Autenticação de usuário

  Parâmetros:

  ```
  {
    "email": "usuario@exemplo.com",
    "password": "senha123"
  }
  ```

  Response: (200, OK)

- /auth/register

  Método: **POST**

  Objetivo: Registro de um novo usuário

  Parâmetros:

  {
  "name": "João Silva",
  "email": "joao@exemplo.com",
  "password": "senha123",
  "role": "user"
  }

  Response: (200, OK)

- /auth/createpassword

  Método: **POST**

  Objetivo: Registro de um novo usuário

  Parâmetros:

  {
  "email": "usuario@exemplo.com",
  "password": "novaSenha123"
  }

  Response: (200, OK)

- /user/update

  Método: **PUT**

  Objetivo: Atualiza um usuário

  Parâmetros:

  {

  "id": "123e4567-e89b-12d3-a456-426614174000",

  "name": "João Silva",

  "email": "joao@exemplo.com",

  "cpf": "123.456.789-00",

  "role": "user"

  }

  Response: (200, OK)

- /user/delete/{id}

  Método: **DELETE**

  Objetivo: Deleta um usuário

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /user/list

  Método: **GET**

  Objetivo: Lista todos os usuários

  Parâmetros: Nenhum

  Response: (200, OK)

- /user/read/{id}

  Método: **GET**

  Objetivo: Lista todos os usuários

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /user/change-password

  Método: **PUT**

  Objetivo: Altera a senha do usuário

  Parâmetros:

  ```
  {
    "currentPassword": "senhaAtual123",
    "newPassword": "novaSenha123"
  }
  ```

  Response: (200, OK)

- /station/create

  Método: **POST**

  Objetivo: Cria uma nova estação

  Parâmetros:

  ```
  {
    "uuid": "123e4567-e89b-12d3-a456-426614174000",
    "name": "Estação Central",
    "latitude": -23.55052,
    "longitude": -46.633308
  }
  ```

  Response: (200, OK)

- /station/update

  Método: **PUT**

  Objetivo: Atualiza uma estação

  Parâmetros:

  {

  "uuid": "123e4567-e89b-12d3-a456-426614174000",

  "name": "Estação Central Atualizada",

  "latitude": -23.55052,

  "longitude": -46.633308

  }

  Response: (200, OK)

- /station/delete/{id}

  Método: **DELETE**

  Objetivo: Deleta uma estação

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /station/list

  Método: **GET**

  Objetivo: Lista todas as estações

  Parâmetros: Nenhum

  Response: (200, OK)

- /station/read/{id}

  Método: **GET**

  Objetivo: Obtém uma estação específica

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /typeAlert

  Método: **GET**

  Objetivo: Lista todos os tipos de alerta

  Parâmetros: Nenhum

  Response: (200, OK)

- /typeAlert

  Método: **POST**

  Objetivo: Cria um novo tipo de alerta

  Parâmetros:

  ```
  {
    "name": "Temperatura Alta",
    "comparisonOperator": ">",
    "value": 30,
    "parameterId": "123e4567-e89b-12d3-a456-426614174000"
  }
  ```

  Response: (200, OK)

- /typeAlert

  Método: **PUT**

  Objetivo: Atualiza um tipo de alerta

  Parâmetros:

  {

  "id": "123e4567-e89b-12d3-a456-426614174000",

  "name": "Temperatura Muito Alta",

  "comparisonOperator": ">",

  "value": 35,

  "parameterId": "123e4567-e89b-12d3-a456-426614174000"

  }

  Response: (200, OK)

- /typeAlert/{id}

  Método: **GET**

  Objetivo: Obtém um tipo de alerta específico

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /typeAlert/{id}

  Método: **DELETE**

  Objetivo: Remove um tipo de alerta

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /typeParameter/create

  Método: **POST**

  Objetivo: Cria um novo tipo de parâmetro

  Parâmetros:

  {
  "name": "Temperatura",
  "unit": "°C",
  "numberOfDecimalsCases": 2,
  "factor": 1,
  "offset": 0,
  "typeJson": "temperature"
  }

  Response: (200, OK)

- /typeParameter/update

  Método: **PUT**

  Objetivo: Atualiza um tipo de parâmetro

  Parâmetros:

  {
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "name": "Temperatura",
  "unit": "°C",
  "numberOfDecimalsCases": 2,
  "factor": 1,
  "offset": 0,
  "typeJson": "temperature"
  }

  Response: (200, OK)

- /typeParameter/delete/{id}

  Método: **DELETE**

  Objetivo: Remove um tipo de parâmetro

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /typeParameter/read/{id}

  Método: **GET**

  Objetivo: Obtém um tipo de parâmetro específico

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /typeParameter/list

  Método: **GET**

  Objetivo: Lista de todos os parâmetros

  Parâmetros: Nenhum

  Response: (200, OK)

- /measure/create

  Método: **POST**

  Objetivo: Cria uma nova medição

  Parâmetros:

  {

  "unixTime": 1679876543,

  "value": 25.5,

  "parameterId": "123e4567-e89b-12d3-a456-426614174000"

  }

  Response: (200, OK)

- /measure/update

  Método: **PUT**

  Objetivo: Atualiza uma medição

  Parâmetros:

  {

  "id": "123e4567-e89b-12d3-a456-426614174000",

  "unixTime": 1679876543,

  "value": 26

  }

  Response: (200, OK)

- /measure/delete/{id}

  Método: **DELETE**

  Objetivo: Deleta uma medição

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /measure/list

  Método: **GET**

  Objetivo: Lista todas as medições

  Parâmetros: Nenhum

  Response: (200, OK)

- /measure/read/{id}

  Método: **GET**

  Objetivo: Obtém uma medição específica

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /alert/create

  Método: **POST**

  Objetivo: Cria um novo alerta

  Parâmetros:

  ```
  {
    "date": "2023-05-10T15:30:00Z",
    "typeAlerdId": "123e4567-e89b-12d3-a456-426614174000",
    "measureId": "123e4567-e89b-12d3-a456-426614174000"
  }
  ```

  Response: (200, OK)

- /alert/update

  Método: **PUT**

  Objetivo: Atualiza um alerta

  Parâmetros:

  {

  "id": "123e4567-e89b-12d3-a456-426614174000",

  "date": "2023-05-10T15:30:00Z",

  "typeId": "123e4567-e89b-12d3-a456-426614174000",

  "measureId": "123e4567-e89b-12d3-a456-426614174000"

  }

  Response: (200, OK)

- /alert/delete/{id}

  Método: **DELETE**

  Objetivo: Deleta um alerta

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /alert/list

  Método: **GET**

  Objetivo: Lista todos os alertas

  Parâmetros: Nenhum

  Response: (200, OK)

- /alert/read/{id}

  Método: **GET**

  Objetivo: Obtém um alerta específico

  Parâmetros:

  Example : 123e4567-e89b-12d3-a456-426614174000

  Response: (200, OK)

- /receiverJson

  Método: **POST**

  Objetivo: Recebe dados JSON de medições

  Parâmetros:

  ```
  {
    "stationId": "string",
    "typeParameter": 0,
    "unixTime": 0
  }
  ```

  Response: (200, OK)