**Documento de Diagrama de Classes**

**Autor:**

**Luís Campos**

**Março, 2023**

**Índice**

[**1.** **Introdução** 4](#_Toc166763499)

[**1.1.** **Descrição do Sistema** 4](#_Toc166763500)

[**1.2.** **Objetivo do Diagrama de Classes** 4](#_Toc166763501)

[**2.** **Diagrama de Classes** 5](#_Toc166763502)

[**3.** **Descrição das Classes** 6](#_Toc166763503)

[**3.1.** **Classe Organizador** 6](#_Toc166763504)

[**3.2.** **Classe Alerta** 6](#_Toc166763505)

[**3.3.** **Classe Sensor** 7](#_Toc166763506)

[**3.4.** **Classe Espaco** 7](#_Toc166763507)

[**4.** **Relações** 8](#_Toc166763508)

**Índice de Figuras**

[Figura 1 - Diagrama de Classes 5](file:///C:\Users\campo\OneDrive\Ambiente%20de%20Trabalho\Programmings%20and%20Shit\Curso%20TPSI\2%20Ano\2º%20Semestre\Estagio\docs\Documentação%20Diagramas\Documento%20do%20Diagrama%20de%20Classes.docx#_Toc166763441)

# **Introdução**

## **Descrição do Sistema**

O Sistema de Monitoramento de Eventos é um sistema web que permite monitorar eventos em tempo real. O sistema coleta dados de sensores de temperatura, humidade, qualidade do ar e ruído, e os apresenta em um dashboard para os organizadores do evento. Os organizadores também podem configurar alertas para serem notificados quando os valores dos sensores ultrapassam limites predefinidos.

## **Objetivo do Diagrama de Classes**

O diagrama de classes documenta as classes principais do sistema e seus relacionamentos. O objetivo do diagrama é descrever a estrutura do sistema, facilitar a comunicação entre os membros da equipe de desenvolvimento e documentar as regras de negócio do sistema.

# **Diagrama de Classes**

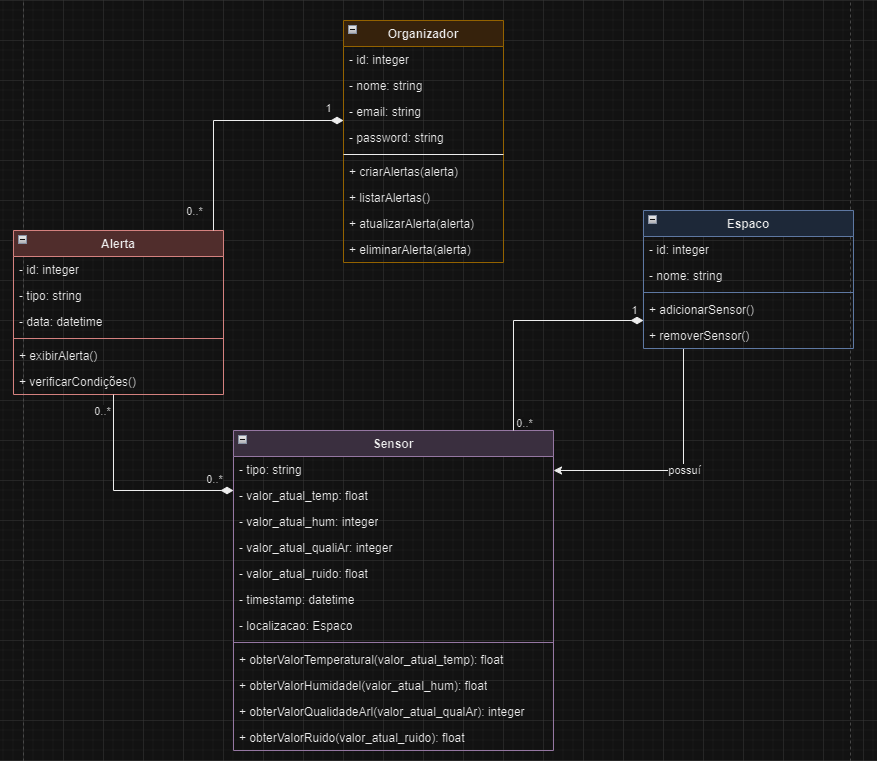


Figura - Diagrama de Classes

# **Descrição das Classes**

## **Classe Organizador**

* **Descrição:**

Representa a entidade que vai gerir os alertas e dados para assim agir conforme o necessário para manter as condições de conforto.

* **Atributos**
* id: integer : Identificador único do organizador.
* nome: string : Nome do organizador.
* email: string : E-mail do organizador.
* password: string : Password do organizador
* **Métodos**
* criarAlertas(alerta) : Alerta:
* listarAlertas() :
* atualizarAlerta(alerta) : Alerta :
* eliminarAlerta(alerta) : Alerta :

## **Classe Alerta**

* **Descrição:**

Representa um alerta e o seu tipo que é exibido ocorra alguma variação nas condições de conforto e tambem fornece a data e hora do ocorrido.

* **Atributos:**
* **id: int :** Identificador único do Alerta.
* **tipo: string :** Define o tipo de alerta.
* **data\_hora:** datetime : data e a hora que ocorreu o alerta
* **Métodos:**
  + - **exibirAlerta() :** Exibe um alerta caso algum dado dos sensores esteja fora do normal.
    - **verificarCondicoes() :** verifica se os dados estam fora do normal recebidos dos sensores estam fora do normal.

## **Classe Sensor**

* **Descrição:**

Representa um dispositivo que coleta dados do ambiente.

* **Atributos:**
  + - **tipo: string :** Tipo do sensor (ex: temperatura, ruído, etc..).
    - **valor\_atual\_temp: float :** valor medido pelo sensor exclusivamente de temperatura.
    - **valor\_atual\_hum: integer :** valor medido exclusivamente pelo sensor de humidade.
    - **valor\_atual\_qualiAr: integer :** valor medido exclusivamente pelo sensor de qualidade do ar.
    - **valor\_atual\_ruido: float :** valor medido exclusivamente pelo sensor de ruido.
    - **timestamp: datetime :** o timestamp das medições feitas pelo sensor.
    - **localizacao: Espaco :** localização do sensor em um espaco especifico do evento.
* **Métodos:**
* **obterValorTemperatural(valor\_atual\_temp): float :**
* **obterValorHumidadel(valor\_atual\_hum): float :**
* **obterValorQualidadeArl(valor\_atual\_qualAr): integer :**
* **obterValorRuido(valor\_atual\_ruido): float :**

## **Classe Espaco**

* **Descrição:**

Representa um espaço físico dentro do evento (ex: sala, corredor) onde estam instalados os sensores.

* **Atributos:**
  + - **id: integer :** Identificador único do evento.
    - **nome: string :** Nome do evento.
* **Métodos:**
  + - **adicionarSensor(sensor): Sensor : Adiciona um sensor ao espaço.**
    - **removerSensor(sensor): Sensor : Remove ao espaço.**

# **Relações**

* **Um organizador pode criar vários eventos (relacinamento one-to-many).**
* **Um evento pode ser criado por apenas um organizador (relacionamento one-to-many).**
* **Um evento pode ter vários sensores associados (relacionamento one-to-many).**
* **Um sensor pode estar presente em vários eventos (relacionamento many-to-many).**
* **Um evento pode gerar vários alertas durante o monitoramento (relacionamento one-to-many).**
* **Um participante pode visualizar os dados de vários eventos (relacionamento many-to-many).**
* **Somente organizadores cadastrados no sistema podem criar eventos.**
* **Um SuperAdmin tem permissão para registrar novos organizadores, listar todos os organizadores registrados, buscar um organizador específico, atualizar dados de um organizador e eliminá-lo do sistema.**
* **Os participantes não possuem conta no sistema e acessam os dados do evento por meio de um convite ou link compartilhado pelo organizador.**