



**Escola Superior  
de Tecnologia  
e Gestão**

Politécnico de Coimbra

CTeSP em Tecnologias e Programação de Sistemas de Informação

## **Relatório de Estágio**



**Digital**Progression

**Luís Francisco Costa Campos**

Oliveira do Hospital, 06 de 2024



Cofinanciado pela  
União Europeia





**Escola Superior  
de Tecnologia  
e Gestão**

Politécnico de Coimbra

## **Relatório de Estágio na DigitalProgression**

**De 14/02/2024 a 14/06/2024**

Luís Francisco Costa Campos

***Orientador:***

*Francisco Carlos Afonso*

*Professor Adjunto, ESTGOH*

***Supervisor na Entidade de  
Acolhimento:***

*José António dos Santos Fernandes  
Monteiro*



Cofinanciado pela  
União Europeia

## **Declaração de integridade e de honestidade intelectual**

*Eu, Luís Francisco Costa Campos, estudante n.º 2022117261 do Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação, declaro que o presente relatório de estágio é original e que, ao longo da sua elaboração, não pratiquei plágio ou qualquer forma de falsificação de conteúdo. Este relatório de estágio resulta do meu próprio trabalho, sendo reconhecidas todas as fontes utilizadas por se encontrarem devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Assumo ter plena consciência de que a prática de plágio - utilização como sendo criação ou prestação sua de obras, ideias, afirmações, dados, imagens ou ilustrações de outra autoria, no todo em parte, sem o adequado reconhecimento explícito - constitui, no âmbito académico, grave falta ética e desonestidade intelectual, tendo como consequência a anulação do trabalho apresentado, para além de poder constituir crime de violação dos direitos de autor e infração disciplinar.*

*Mais declaro que tomei conhecimento integral do Código de Ética e Conduta do Instituto Politécnico de Coimbra e demais regulamentos aplicáveis e que foram respeitadas as orientações recebidas quanto à pseudonimização ou anonimização de dados pessoais ou organizacionais.*

*Oliveira do Hospital, xx de xxxxxx de 20xx*

*Nome completo: ....*

*Assinatura: (digital)*

## Resumo

Este relatório final de estágio resulta, fundamentalmente, numa descrição e reflexão exaustiva e cuidada, acompanhando as várias etapas percorridas durante este estágio. O objetivo deste trabalho foi desenvolver um sistema de monitorização ambiental para eventos, focado na recolha e análise contínua de dados sobre temperatura, humidade, qualidade do ar e ruído. Esta solução IoT foi projetada para fornecer dados em tempo real, auxiliando tanto os participantes quanto os organizadores de eventos a gerir e melhorar as condições ambientais.

Ao longo do desenvolvimento deste relatório, procurou-se refletir de forma crítica sobre os diversos pontos do projeto, tais como as expectativas iniciais, as atividades de ensino-aprendizagem envolvidas, e as dificuldades e necessidades de formação encontradas. A capacidade de iniciativa, a responsabilidade e a importância do trabalho individual e de grupo foram também questões centrais abordadas.

As principais contribuições deste trabalho incluem a implementação bem-sucedida da infraestrutura IoT, a integração eficiente dos sensores e a criação de uma interface web funcional. As funcionalidades de alerta e sugestões automatizadas foram incorporadas para melhorar as condições de conforto durante os eventos monitorizados.

## Palavras-chave

Sistema, Monitorização, PHP, Python, Programação, Estágio, Base de Dados



# Índice

<b>1.</b>	<b>Introdução.....</b>	<b>1</b>
<b>2.</b>	<b>Estado da Arte .....</b>	<b>3</b>
2.1.	Nexxto .....	3
2.2.	Vaisala .....	4
2.3.	Jri MySirius .....	5
2.4.	EXTECH CO220.....	6
2.5.	Sismetrio .....	7
2.6.	NatéoSanté - Air Coach Pro.....	7
2.7.	Tabela de Comparação dos Sistemas .....	9
<b>3.</b>	<b>Objetivos e Metodologias .....</b>	<b>11</b>
3.1.	Ferramentas e Tecnologias .....	12
3.2.	Planeamento .....	15
<b>4.</b>	<b>Trabalho Desenvolvido .....</b>	<b>19</b>
4.1.	Fase 2 - Análise de Requisitos .....	19
4.2.	Fase 3 - Design e Arquitetura do Sistema .....	21
4.3.	Fase 4 - Desenvolvimento .....	24
4.4.	Fase 5 - Testes .....	34
4.5.	Fase 6 - Documentação .....	34
<b>5.</b>	<b>Conclusões .....</b>	<b>35</b>
5.1.	Forças .....	35
5.2.	Limitações .....	36
5.3.	Trabalho Futuro.....	36
<b>6.</b>	<b>Referências.....</b>	<b>39</b>
<b>7.</b>	<b>Anexos .....</b>	<b>42</b>
7.1.	Anexo 1 - Design Inicial da Dashboard .....	42
7.2.	Anexo 2 - Design do site .....	43
7.3.	Anexo 3 – Script da Base de Dados SQL .....	44
7.4.	Anexo 4 – Documento de Especificacao de Requisitos.....	45
7.5.	Anexo 5 – Documento de Testes.....	67
7.6.	Anexo 6 – Manual do Utilizador.....	81
7.7.	Anexo 7 – Documento do Diagrama de Deployment .....	89
7.8.	Anexo 8 – Documento do Diagrama ER .....	95





# Lista de Figuras

Figura 1 - Sistema Vaisala .....	4
Figura 2 - Sistema Jri MySirius .....	5
Figura 3 - Sistema EXTECH .....	6
Figura 4 - Sistema Sismetron .....	7
Figura 5 - Sistema NatéoSanté - Air Coach Pro .....	8
Figura 6 – Diagrama de Gantt do Planeamento .....	17
Figura 7 - Diagrama de Gantt do Executado .....	18
Figura 8 - Diagrama de Casos de Uso .....	21
Figura 9 - Diagrama ER.....	22
Figura 10 - Diagrama de Deployment.....	23
Figura 11 - Desenvolvimento da Infraestrutura .....	24
Figura 12 - Página de Dashboard.....	27
Figura 13 - Página de Histórico de dados .....	27
Figura 14 - Página de Configuração de Alertas.....	28
Figura 15 - Página de Sensores.....	28
Figura 16 - Página de Espaços e Associação de Polígonos .....	30
Figura 17 - Página de registo .....	31
Figura 18 - Página de Gestão de Contas.....	32
Figura 19 - Exibição de Alerta .....	32
Figura 22 - Caixa Final.....	33
Figura 21 - Planificação da caixa.....	33
Figura 20 - Caixa Inicial .....	33
Figura 23 – Design da dashboard inicial .....	42
Figura 24 - Design de algumas páginas do site.....	43



# Lista de Tabelas

Tabela 1 - Comparação Sistemas/Funcionalidades .....	9
Tabela 2 - Ferramentas e Tecnologias .....	14



# Lista de Acrónimos

<b>ADC</b>	Analog-Digital Converter (Conversor Analógico-Digital)
<b>DAC</b>	Digital-Analog Converter (Conversor Digital-Analógico)
<b>ER</b>	Modelo Entidade-Relacionamento
<b>I2C</b>	Inter-Integrated Circuit
<b>IDE</b>	Integrated Development Environment
<b>IoT</b>	Internet of Things
<b>IP</b>	Internet Protocol
<b>SoA</b>	State Of the Art
<b>SQL</b>	Structed Query Language
<b>SSH</b>	Secure Shell



# 1. Introdução

A crescente demanda por eventos seguros e confortáveis impulsionou a necessidade de sistemas avançados de monitorização ambiental. Este projeto visa desenvolver um sistema que mede e analisa, em tempo real, parâmetros críticos como temperatura, umidade, qualidade do ar e ruído em todo o tipo de eventos. Monitorar essas condições é essencial para garantir o bem-estar dos participantes e para cumprir com regulamentações de saúde e segurança.

Em eventos de grande escala, as condições ambientais podem variar significativamente, afetando a saúde e o conforto dos participantes. A falta de um sistema eficaz de monitorização pode levar a ambientes insalubres e experiências negativas. Além disso, a capacidade de reagir rapidamente a mudanças nas condições ambientais é limitada sem um sistema de monitorização em tempo real.

Para resolver este problema, foi desenvolvida uma solução baseada em tecnologia IoT dando o nome ao projeto de Sistema IoT de Monitorização. A solução envolve a integração de sensores específicos (temperatura, umidade, qualidade do ar e ruído) a um microcomputador Raspberry Pi. Os dados coletados pelos sensores são processados pelo Raspberry Pi e enviados para um servidor web, onde são armazenados e podem ser visualizados através de uma interface web.

O sistema proposto consiste em um conjunto de sensores conectados a um Raspberry Pi, que atua como unidade central de processamento. Através de um servidor Apache armazena os dados em um base de dados SQL e disponibiliza-os para visualização em tempo real por meio de uma interface Web.

Este documento está organizado nas seguintes seções: o estado da arte, que apresenta uma revisão das soluções existentes para o problema de monitorização ambiental em eventos; os objetivos e metodologias, que descrevem os objetivos globais do trabalho, a metodologia adotada e as ferramentas e tecnologias utilizadas; o trabalho desenvolvido, que detalha o que foi desenvolvido, os requisitos implementados e os principais resultados alcançados; e as conclusões, que fornecem um resumo do trabalho desenvolvido, destacando as principais contribuições e limitações, e apresentando sugestões para melhorias futuras. As referências e anexos incluem todas as fontes de informação utilizadas e documentos adicionais necessários para uma melhor compreensão do projeto.

O sistema proposto neste projeto visa combinar a robustez e a precisão dos sistemas mencionados, com um enfoque particular na acessibilidade e na customização para eventos. Utilizando um Raspberry Pi como unidade central de processamento e uma gama de sensores (temperatura, umidade, qualidade do ar e ruído), o sistema oferece monitorização em tempo real com a possibilidade de integração a um servidor Apache para armazenamento e visualização dos dados.



## **2. Estado da Arte**

No campo da monitorização, diversas soluções têm sido desenvolvidas para atender às necessidades de monitorização e gestão das condições de conforto em diferentes contextos. Estas soluções variam em termos de funcionalidades, conectividade e capacidade de armazenamento de dados.

Algumas das soluções mais notáveis incluem sistemas como Nexxto (Nexxto, n.d.), que oferece monitorização em tempo real, alertas e gestão de incidentes, com uma variedade de opções de conectividade e armazenamento. Vaisala (Vaisala, 2023) é outra opção que se destaca pelo seu monitorização portátil e alarmes remotos, cobrindo uma ampla gama de variáveis ambientais.

### **2.1. Nexxto**

O Nexxto oferece soluções completas de monitorização de ambiente, utilizando sensores IoT para medir temperatura, umidade e outros parâmetros ambientais. Este sistema se destaca pela sua capacidade de integração com diversas plataformas e pela facilidade de uso, proporcionando monitorização em tempo real e alertas automáticos via SMS ou e-mail. No entanto, seu custo elevado pode ser um fator limitante para eventos menores ou com orçamento restrito.

## 2.2. Vaisala

A Vaisala mostrado na figura 1 é conhecida pela alta precisão e confiabilidade dos seus sensores. Suas soluções incluem monitorização contínuo de temperatura, umidade e qualidade do ar, sendo amplamente utilizadas em ambientes críticos como laboratórios e indústrias farmacêuticas. Embora ofereça uma gama ampla de funcionalidades, a configuração e o custo de implementação são consideravelmente altos, tornando-o menos acessível para aplicações em eventos de curta duração.



Figura 1 - Sistema Vaisala

## 2.3. Jri MySirius

O Jri MySirius (Jri MySirius, n.d.) mostrado na figura 2 é um sistema de monitorização baseado em uma plataforma web que oferece medição de temperatura e umidade com dados em tempo real. É um sistema robusto de alertas. No entanto, como outros sistemas comerciais, pode representar um investimento significativo.



Figura 2 - Sistema Jri MySirius

## 2.4. EXTECH CO220

O EXTECH CO220 (Extech Instruments, n.d.) apresentado na figura 3 é um dispositivo portátil específico para monitorização de qualidade do ar, focado principalmente em medir níveis de dióxido de carbono (CO<sub>2</sub>). Embora seja eficaz para ambientes internos e para avaliar a ventilação, suas funcionalidades são limitadas quando comparadas a sistemas que oferecem monitorização múltiplo de parâmetros como temperatura e umidade.



Figura 3 - Sistema EXTECH

## 2.5. Sismetro

O Sismetro (SISMETRO, n.d.) mostrado na figura 4 fornece soluções completas para monitorização de temperatura e humidade, com uma plataforma que permite a visualização e análise dos dados coletados. É conhecido pela durabilidade e precisão dos seus sensores. No entanto, a necessidade de uma infraestrutura mais complexa e os custos associados podem ser um empecilho para eventos temporários.



Figura 4 - Sistema Sismetro

## 2.6. NatéoSanté - Air Coach Pro

O Air Coach Pro (NatéoSanté, n.d.) mostrado na figura 5 é um sistema portátil focado na monitorização de qualidade do ar, medindo partículas, CO<sub>2</sub>, temperatura e humidade. É ideal para uso em ambientes internos e oferece relatórios detalhados sobre as condições do ar. A principal limitação é a sua portabilidade, que, embora seja uma vantagem, pode não oferecer a mesma robustez que sistemas fixos mais complexos.



**Figura 5 - Sistema NatéoSanté - Air Coach Pro**

## 2.7. Tabela de Comparação dos Sistemas

Tabela 1 - Comparação Sistemas/Funcionalidades

Sistema Funcionalidades	Nexxto	Vaisala	Jri MySirius	EXTECH CO220	Sismetron	Air Coach Pro	Testo 175 H1	Sistema Desenvolvido
<b>Monitorização</b>								
Temperatura	✓	✓	✓	✓	✓	✓	✓	✓
Umidade	✓	✓	✓	✓	✓	✓	✓	✓
Ruído	✗	✗	✗	✗	✓	✗	✗	✓
CO2	✗	✓	✓	✓	✗	✓	✗	✓
Partículas	✗	✓	✗	✗	✗	✓	✗	✗
Pressão Barométrica	✗	✓	✓	✗	✗	✗	✗	✗
Ponto de Orvalho	✗	✓	✗	✓	✗	✗	✓	✗
<b>Comunicação</b>								
Wifi	✓	✓	✓	✓	✓	✓	✗	✓
Ethernet	✓	✓	✗	✗	✓	✗	✗	✓
Bluetooth	✓	✗	✓	✓	✗	✓	✗	✓

<b>USB</b>	✗	✓	✗	✓	✗	✗	✓	✓
<b>Modbus</b>	✗	✓	✗	✗	✗	✗	✗	✗
<b>LoRa</b>	✗	✗	✗	✗	✗	✗	✗	✗
<b>Armazenamento</b>								
<b>Interno</b>	✓	✓	✓	✓	✓	✗	✓	✓
<b>Cloud</b>	✓	✓	✓	✗	✓	✓	✗	✓
<b>Cartão SD</b>	✗	✓	✗	✓	✗	✗	✓	✓
<b>Exportação de dados</b>	✗	✗	✗	✗	✗	✓	✗	✗
<b>Híbrido</b>	✓	✗	✗	✗	✗	✗	✗	✗

A tabela acima resume as funcionalidades de monitorização, comunicação e armazenamento dos dados de cada sistema mencionado, nomeadamente Nexxto, Vaisala, Jri MySirius, EXTECH CO220, Sismetro e Air Coach Pro, em comparação com as funcionalidades previstas para o sistema do presente projeto.

Concluindo, a análise comparativa dos principais sistemas de monitorização ambiental em relação ao sistema desenvolvido neste projeto, revelou uma série de insights valiosos. Enquanto os sistemas existentes no mercado são eficazes e especializados, o sistema desenvolvido neste projeto destaca-se por sua abrangência, e alta customização para o organizador do evento no que toca a configuração e gestão do sistema o que torna o sistema competente e pronto para atender às demandas contemporâneas de monitorização ambiental.



### 3. Objetivos e Metodologias

O principal objetivo deste trabalho é desenvolver um sistema IoT para monitorizar e melhorar as condições de conforto em eventos como feiras e exposições. Para alcançar este objetivo geral, foram definidos objetivos específicos que guiarão o desenvolvimento e implementação do sistema. Inicialmente, foi realizada uma análise de requisitos para identificar as necessidades específicas dos utilizadores e organizadores de eventos, definindo os parâmetros de conforto a serem monitorizados. Em seguida, foi elaborado um cronograma detalhado, dividindo o projeto em fases claras de design, desenvolvimento, testes e deployment, estabelecendo metas realistas e prazos para cada etapa.

No sistema desenvolvido, executei integralmente as tarefas relacionadas ao desenvolvimento do site, exceto pela página de dashboard, que já estava desenvolvida e necessitava apenas de ajustes no código para poder funcionar corretamente com o código alterado. As principais atividades incluíram a criação das páginas e funcionalidades para a configuração de sensores, espaços e alertas, bem como a gestão de utilizadores Organizadores e SuperAdmin. Adicionalmente, foi desenvolvida a base de dados SQL para armazenar e organizar todas as informações coletadas e geridas pelo sistema.

O desenvolvimento de parte do script Python para o Raspberry Pi visto que este script já continha uma parte do código referente ao controlo do sensor da temperatura e umidade, mas tive que alterar o código para poder implementar os restantes sensores. Este script foi responsável pelo controlo dos sensores conectados ao Raspberry Pi e pela transmissão dos dados coletados para o servidor web. Esta transmissão permitiu a integração eficiente dos dados de monitorização ambiental na plataforma web, garantindo que os dados fossem acessíveis em tempo real para os administradores e participantes do evento.

Apesar de ter utilizado o servidor Apache para hospedar o sistema, o restante do desenvolvimento, desde a infraestrutura da aplicação web referente a área do

organizador até a lógica de comunicação entre o servidor e o sistema bem como armazenamento e solicitação de dados, foi criado de raiz. Isto inclui a interface de configuração do sistema, que permite adicionar, remover e gerir sensores e espaços, definir alertas baseados em condições específicas, e uma área administrativa para os SuperAdmins.

No que diz respeito ao design e arquitetura do sistema, foram usados sensores e dispositivos IoT apropriados para a recolha de dados, projetando uma arquitetura que integre hardware, software e redes de comunicação de forma eficiente. O desenvolvimento do sistema envolveu a implementação da infraestrutura IoT para a recolha contínua dos dados, o desenvolvimento de uma interface gráfica para a integração de funcionalidades de alerta de forma a notificar o organizador do evento e possibilidade de upload de uma planta do espaço do evento que permite desenhar o polígono do espaço.

Para validar a eficácia do sistema, foram conduzidos testes de funcionalidade e fiabilidade dos componentes IoT. A documentação técnica abrangente foi preparada, fornecendo o manual de utilizador e documentação relativa aos requisitos, diagramas e de testes. Em termos de metodologia, foi adotada uma abordagem que combina elementos tradicionais e ágeis, permitindo flexibilidade no processo de desenvolvimento e adaptação às necessidades emergentes. Foram recolhidos os requisitos, usadas metodologias ágeis para o planeamento e execução do projeto, práticas de design através de protótipos de baixa fidelidade e desenhos de esboço para facilitar a visualização da interface do projeto.

### **3.1. Ferramentas e Tecnologias**

Durante o desenvolvimento deste projeto sendo a equipa constituída por mim com a ajuda do supervisor que me ajudava quando tinha dúvidas, diversas ferramentas e tecnologias foram empregues para garantir uma implementação eficiente e precisa do sistema IoT proposto. Para a integração e controlo dos dispositivos IoT, a plataforma

Raspberry Pi foi utilizada como base principal devido à sua flexibilidade e capacidade de suportar uma variedade de sensores e interfaces de comunicação.

No aspeto dos sensores, foram selecionados dispositivos apropriados em que apesar de os sensores já terem sido escolhidos pelo supervisor foi ainda assim efetuado uma tabela de orçamento juntamente com o supervisor e pesquisa acerca de cada sensor com fim de obter mais conhecimento da minha parte acerca dos sensores usados para monitorar os parâmetros de conforto desejados, Sensor de SHT31-D para a temperatura e humidade visto que o sensor mede as duas coisas, Sensor de qualidade do ar MQ135 e Sensor de nível de ruído KY-038. Esses sensores desempenham um papel fundamental na coleta de dados em tempo real, fornecendo informações para a análise e otimização das condições ambientais em eventos.

**Tabela 2 - Ferramentas e Tecnologias**

<b>Componente</b>	<b>Descrição</b>	<b>Detalhes</b>
<b>Raspberry Pi 4 Model B</b>	Microcomputador	Microcomputador de placa única com processador Broadcom BCM2711 quad-core de 1,5 GHz, 4 GB de RAM LPDDR4, Wi-Fi, Bluetooth 5.0, Gigabit Ethernet, 2 portas USB 3.0, 2 portas USB 2.0, porta HDMI, porta CSI para câmara, porta DSI para display e slot para cartão MicroSD. Ideal para projetos de IoT, automação residencial, desenvolvimento de software, educação e muito mais.
<b>Cartão MicroSD 8GB</b>	Armazenamento para o sistema operativo e dados	Armazenamento para o sistema operativo Raspbian e dados do projeto.
<b>Caixa para Raspberry Pi</b>	Proteção para o Raspberry Pi	Caixa que protege o Raspberry Pi de diversos fatores
<b>Sensor de temperatura/Humidade SHT31-D</b>	Sensor digital de temperatura e humidade	Sensor digital de temperatura e humidade com interface de 1 fio. Faixa de medição de temperatura: 0°C a 50°C. Faixa de medição de humidade: 20% a 80%.
<b>Sensor de qualidade do ar MQ135</b>	Sensor analógico de qualidade do ar	Sensor analógico de qualidade do ar que detecta gases como CO, H2S, NH3 e SO2. Requer um circuito de conversão analógico-digital para ser utilizado com o Raspberry Pi.
<b>Sensor de nível de ruído KY-038</b>	Sensor analógico de nível de ruído	Sensor analógico de nível de ruído que converte o nível de som em um sinal elétrico. Requer um circuito de conversão analógico-digital para ser utilizado com o Raspberry Pi.

<b>Cabo Ethernet</b>	Permite conectar o Raspberry Pi à rede via cabo	
----------------------	---	--

Para o desenvolvimento do software referente ao Raspberry Pi que permite a funcionalidade dos sensores a coleta dos dados e envio para o servidor Web através de Python e para o site que recebe os dados para os exibir, utilizei uma combinação de tecnologias web, incluindo HTML, JavaScript e PHP. O Visual Studio (MICROSOFT, 2016) Code foi adotado como ambiente de desenvolvimento integrado (IDE) para a parte do site, oferecendo uma ampla gama de recursos e extensões que facilitaram a codificação, debug e teste do site já o Pycharm (JetBrains s.r.o., n.d.) foi o IDE utilizado para programação em Python.

Além disso, o Postman (Postman, n.d.) foi utilizado para testar e validar as APIs, garantindo a integração adequada entre os diferentes componentes do sistema e do site. Para a gestão e análise dos dados coletados, o DataGrip (JetBrains s.r.o., n.d.) foi a ferramenta selecionada, permitindo a manipulação eficiente da base de dados, consultas e visualizações que facilitaram a interpretação e utilização das informações recolhidas pelo sistema.

## 3.2. Planeamento

O planeamento do projeto foi estruturado para garantir uma organização e a conclusão de cada fase do projeto dentro dos prazos estabelecidos. As tarefas foram distribuídas ao longo de várias fases, cada uma com objetivos específicos e prazos definidos. A seguir, está definido a distribuição das tarefas e o tempo alocado para cada uma delas.

### **3.2.1. Análise de Requisitos**

Nesta fase inicial foram estipulados 11 dias e foram identificados os requisitos do sistema, tanto funcionais quanto não funcionais. Isso incluiu reuniões com o supervisor da empresa e a revisão de sistemas semelhantes existentes no mercado. A análise detalhada permitiu a definição clara dos objetivos do sistema.

### **3.2.2. Pesquisa e Seleção de Componentes**

Uma pesquisa abrangente foi conduzida para selecionar os sensores e componentes adequados para o sistema dando a esta fase um prazo de 8 dias. Foram considerados fatores como precisão, custo e compatibilidade com o Raspberry Pi. Os componentes selecionados incluíram sensores de temperatura, umidade, qualidade do ar e ruído.

### **3.2.3. Design e Arquitetura do Sistema**

Com os requisitos e componentes definidos, foi desenvolvida a arquitetura do sistema. Esta fase teve uma duração prevista para 19 dias e incluiu a criação de protótipos de baixa e de alta fidelidade, desenhos de design do site do sistema, o Diagrama de Deployment e o Diagrama ER. Foi atribuído uma maior duração a esta fase devido ao tempo de alterações constantes de forma a coincidir com expectativa do supervisor.

### **3.2.4. Desenvolvimento do Sistema**

O desenvolvimento do sistema foi dividido em subfases pelo que durou 19 dias subfases estas que foram:

- **Configuração do Raspberry Pi e Instalação de Software:** Incluiu a configuração do Raspberry Pi, instalação do sistema operativo e bibliotecas necessárias, e configuração das ferramentas de acesso remoto (PuTTY e FileZilla).
- **Integração dos Sensores:** Envolveu a conexão física dos sensores ao Raspberry Pi, a programação inicial para leitura dos dados dos sensores e a configuração do ADC (MCP3008) e DAC (MCP4725).

- **Desenvolvimento da Interface Web:** Criação do servidor Apache para armazenamento dos dados e desenvolvimento da interface web para visualização dos dados em tempo real e administração do sistema.

### 3.2.5. Testes e Validação

Os testes foram conduzidos para garantir a precisão e confiabilidade do sistema. Incluiu testes unitários para cada sensor, testes de integração para o sistema completo e testes de desempenho para verificar a capacidade do sistema de lidar com grandes volumes de dados em tempo real. Esta fase teve a duração de 7 dias.

### 3.2.6. Documentação e Relatório de Estágio

A documentação do sistema foi elaborada ao longo de todo o projeto daí a duração de 75 dias o que corresponde ao número total de dias do decorrer do estágio. Incluiu manual de utilizador, documento de especificação requisitos, documento dos diagramas, documento de testes, bem como a preparação do relatório final do projeto. A documentação foi essencial para assegurar que o sistema possa ser replicado e mantido no futuro.

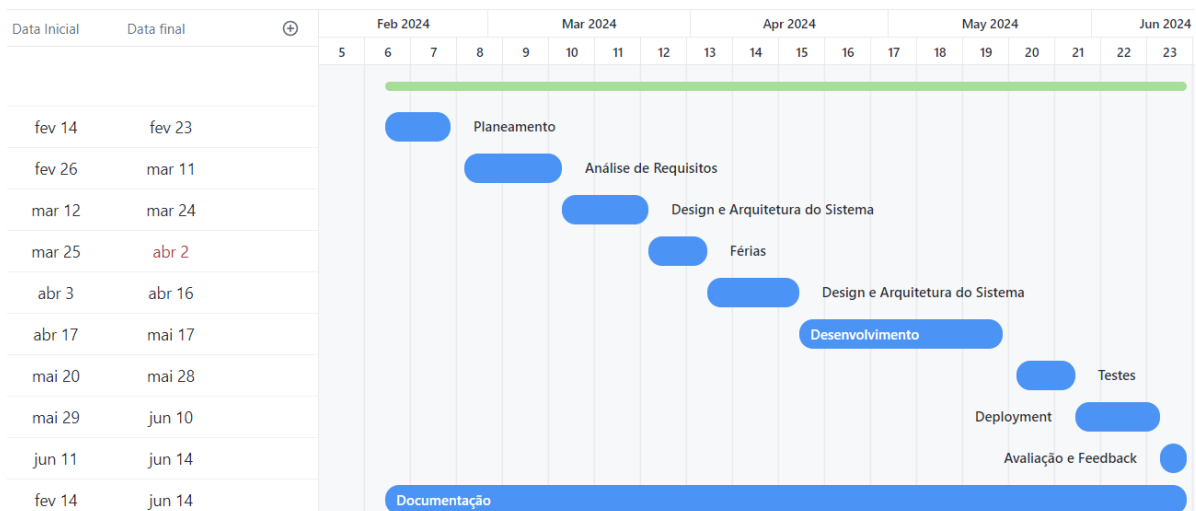


Figura 6 – Diagrama de Gantt do Planeamento

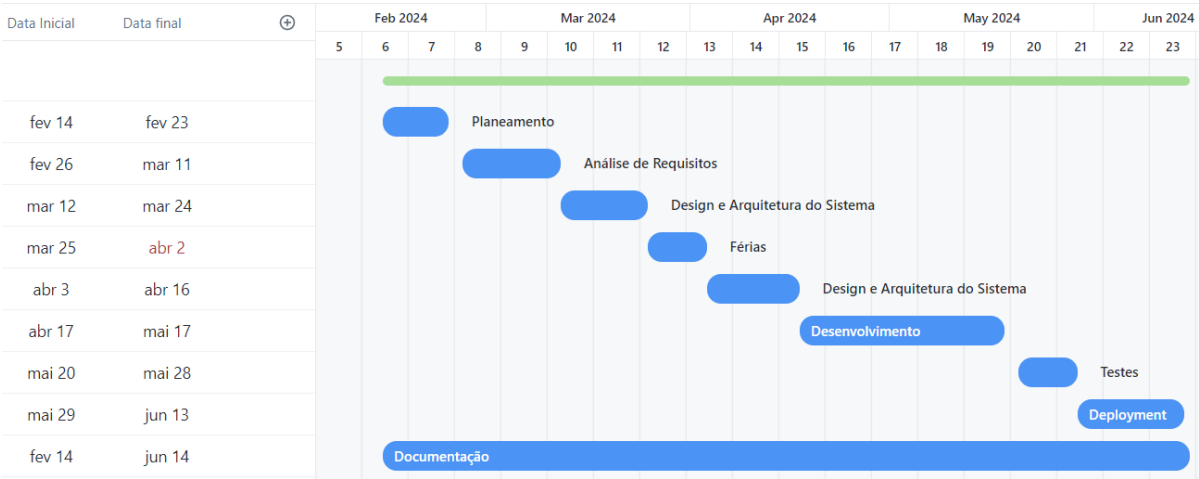


Figura 7 - Diagrama de Gantt do Executado

O diagrama de gantt do executado mostrado na Figura 7 em relação ao gantt do planeamento apresentado na Figura 6 é que foi tudo cumprido nos prazos definidos com exceção da Fase de Avaliação e Feedback que não foi executada. Para realização destes diagramas de gantt foi usado o ClickUp (*ClickUp<sup>TM</sup>, n.d.*).



## **4. Trabalho Desenvolvido**

### **4.1. Fase 2 - Análise de Requisitos**

Após a fase 1 que foi o Planeamento deu início à etapa de análise de requisitos onde foram identificadas as necessidades específicas dos utilizadores e organizadores de eventos, definindo os parâmetros de conforto a serem monitorizados. A análise de requisitos envolveu a coleta de informações detalhadas para garantir que o sistema desenvolvido atendesse às expectativas e necessidades. No relatório apenas estão alguns dos principais requisitos. As tabelas de requisitos e a informação mais completa esta na documentação da Análise de Requisitos e o respetivo diagrama estão disponíveis nos anexos deste relatório.

#### **4.1.1. Requisitos Funcionais:**

##### **Monitorização de Temperatura:**

- O sistema deve ser capaz de medir a temperatura ambiente em diferentes pontos do evento.
- As leituras de temperatura devem ser atualizadas em tempo real e armazenadas para análise posterior.

##### **Monitorização de Umidade:**

- Deve ser possível monitorar a umidade relativa do ar em várias áreas do evento.
- As leituras de umidade devem ser precisas e refletir as condições em tempo real.

##### **Monitorização da Qualidade do Ar:**

- O sistema deve medir a qualidade do ar, incluindo a concentração de partículas (PM2.5 e PM10) e níveis de CO2.
- Deve fornecer alertas quando a qualidade do ar cair abaixo de níveis definidos pelo organizador.

#### **Monitorização de Ruído:**

- O nível de ruído deve ser monitorado em tempo real para garantir um ambiente confortável e seguro.
- Deve ser possível definir limites de ruído e gerar alertas se esses limites forem excedidos.

#### **Interface Web para Visualização de Dados:**

- Os dados coletados devem ser acessíveis através de uma interface web intuitiva.
- A interface deve permitir a visualização de dados em tempo real e gestão e configuração de sensores, espaços e alertas.
- Associar polígonos aos espaços desenhando-os em uma planta adicionada previamente através do upload.

#### **Alertas e Notificações:**

- O sistema deve enviar alertas automáticos via SMS ou e-mail quando os parâmetros monitorados excederem os limites pré-definidos.

### **4.1.2. Requisitos Não Funcionais:**

#### **Precisão e Confiabilidade:**

- Os sensores utilizados devem ter alta precisão e confiabilidade para garantir a qualidade dos dados coletados.

#### **Escalabilidade:**

- O sistema deve ser escalável para permitir a adição de novos sensores ou a expansão para monitorar áreas maiores, se necessário.

#### **Acessibilidade e Usabilidade:**

- A interface web deve ser acessível e fácil de usar para todos os utilizadores, independentemente do seu nível de conhecimento técnico.

#### **Segurança dos Dados:**

- Os dados coletados devem ser armazenados de forma segura, garantindo a privacidade e integridade das informações.

### Manutenção e Suporte:

- O sistema deve ser de fácil manutenção, com suporte disponível para resolução de problemas e atualizações.

A análise de requisitos foi crucial para garantir que o sistema de monitoramento desenvolvido atendesse às expectativas e necessidades dos utilizadores e organizadores de eventos, proporcionando um ambiente mais seguro e confortável.

## 4.2. Fase 3 - Design e Arquitetura do Sistema

Com base nos requisitos identificados, foi projetado a arquitetura do sistema através dos diagramas mostrados nas figuras 8, na figura 9 e na figura 10, escolhendo os sensores e dispositivos IoT adequados para a recolha de dados. Foi dada atenção especial à integração de hardware, software e redes de comunicação para garantir a eficiência e a escalabilidade do sistema. Para o design foi usado o Figma (Figma Inc., 2019) que permitiu desenhar livremente o design do site.

### 4.2.1. Diagrama de Casos de Uso:

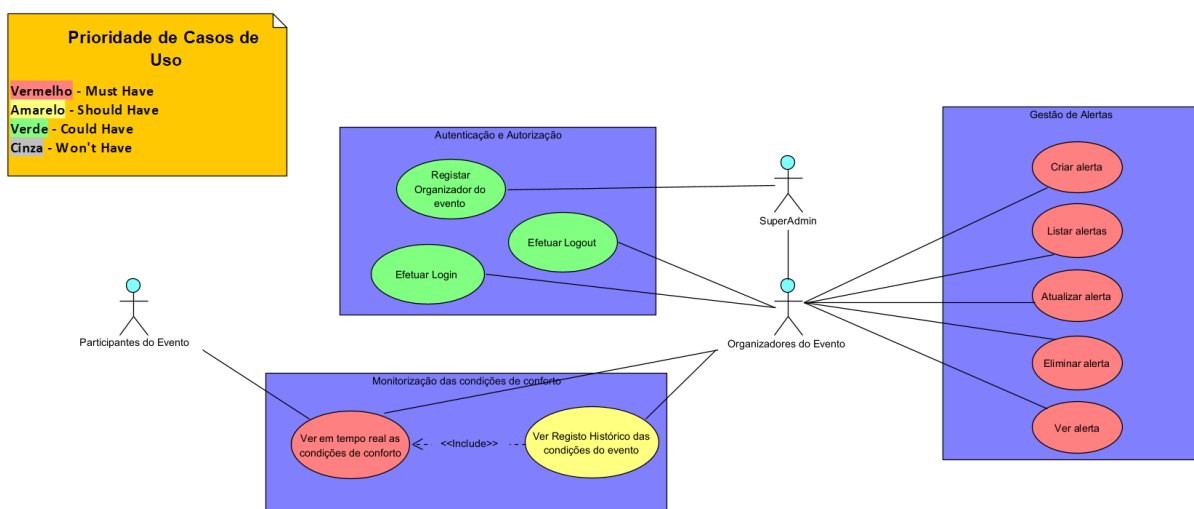


Figura 8 - Diagrama de Casos de Uso

A Figura 8 descreve o diagrama de casos de uso e as interações entre os utilizadores (administradores de eventos e visitantes) e o sistema, este diagrama foi desenvolvido no Visual Paradigm (Visual Paradigm, 2019). Ele define as funcionalidades que o sistema oferece, tais como monitorização em tempo real, alertas de condições anormais e acesso a relatórios históricos como se pode ver na nota do diagrama foi tudo implementado no sistema visto que consoante as cores não existe nenhum caso de uso com a cor cinza. Na especificação de requisitos foram utilizados casos de uso e esta disponível informação mais detalhada dos casos no documento de especificação de requisitos nos anexos.

#### 4.2.2. Diagrama ER (Entidade-Relacionamento):

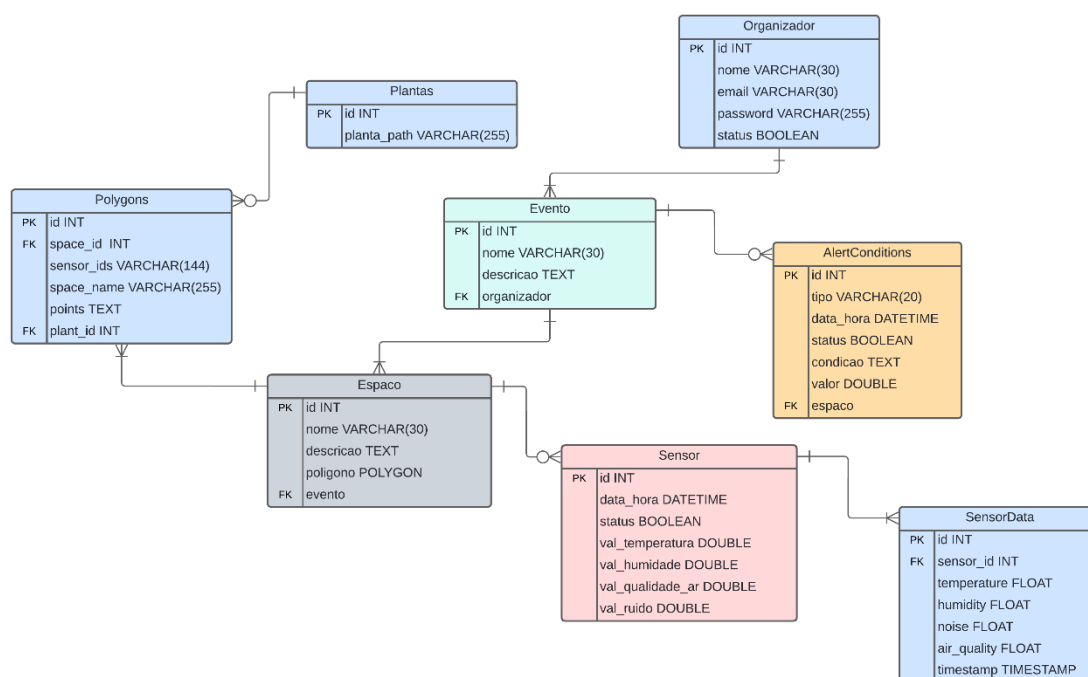


Figura 9 - Diagrama ER

O diagrama ER mostrado na Figura 9 ilustra a estrutura da base de dados referente ao site do sistema onde esta a dashboard e as páginas de administração do sistema como adicionar espaço, sensor alertas etc., destacando as tabelas principais, os campos em cada tabela e os relacionamentos entre elas. Este diagrama é essencial para entender

como os dados são armazenados e organizados no sistema e foi desenvolvido no LucidChart (Lucid Software Inc., 2019).

### 4.2.3. Diagrama de Deployment:

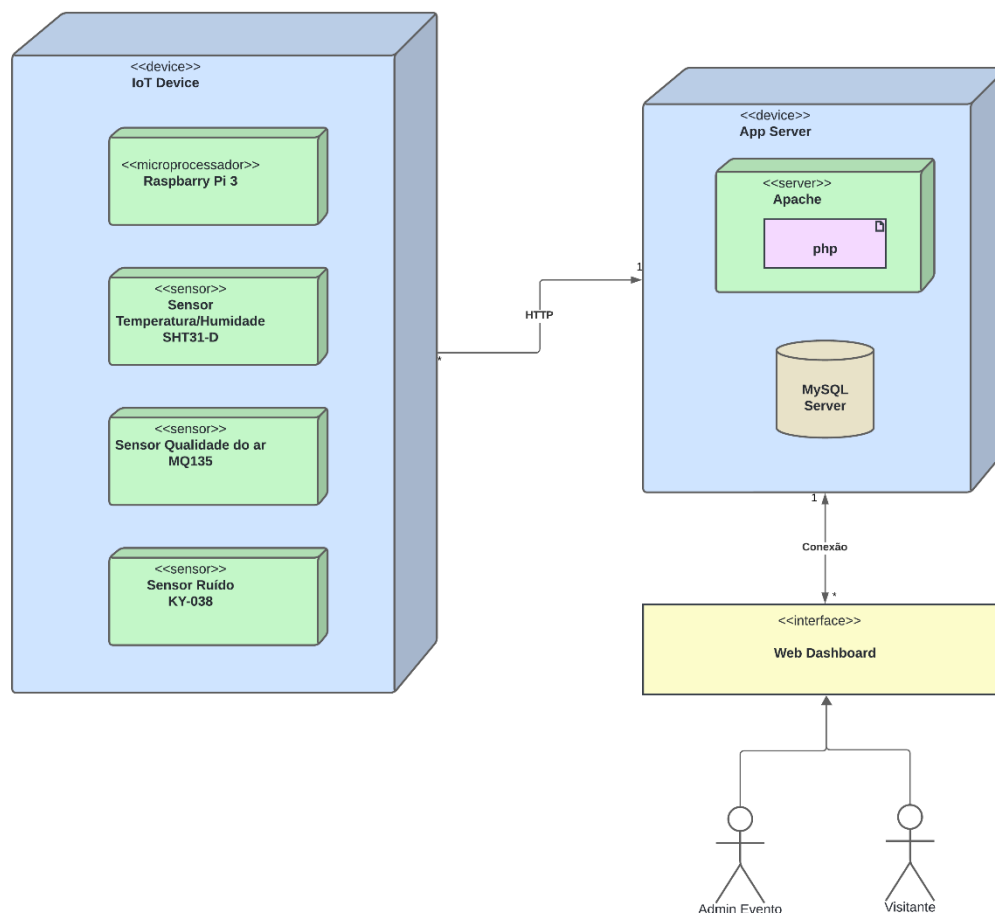


Figura 10 - Diagrama de Deployment

A Figura 10 representa o diagrama de deployment que também foi desenvolvido no LucidChart mostra a disposição física dos componentes de hardware e software no sistema. Ele descreve como os dispositivos (como o Raspberry Pi e o servidor Apache) estão distribuídos na rede e como eles se comunicam entre si. Também mostra que ambos os utilizadores têm acesso à página de dashboard mas no caso do Admin Evento tem acesso as páginas de configuração e gestão protegidas por autenticação dos componentes do sistema como sensores, espaços e alertas.

## 4.3. Fase 4 - Desenvolvimento

A fase de desenvolvimento foi fundamental para a implementação de um sistema de monitorização ambiental completo e funcional. O processo começou com a criação da infraestrutura IoT necessária para a recolha contínua dos dados ambientais. Esta infraestrutura incluiu a instalação e configuração de diversos sensores para monitorizar temperatura, humidade, qualidade do ar e ruído. A escolha e integração dos sensores adequados foram críticas para garantir a precisão e confiabilidade dos dados coletados.

### 4.3.1. Implementação da Infraestrutura IoT

O desenvolvimento da infraestrutura IoT envolveu a utilização do Raspberry Pi, como o principal componente de processamento do sistema como demonstra a Figura 11. O Raspberry Pi 4 Model B foi escolhido devido à sua capacidade de processamento robusta e compatibilidade com uma ampla gama de sensores. Para conectar os sensores analógicos ao Raspberry Pi, que possui apenas entradas digitais, foi utilizado um shield ADC (Analog-to-Digital Converter). Este shield permitiu a conversão dos sinais analógicos dos sensores para sinais digitais, possibilitando a integração dos sensores analógicos ao sistema.

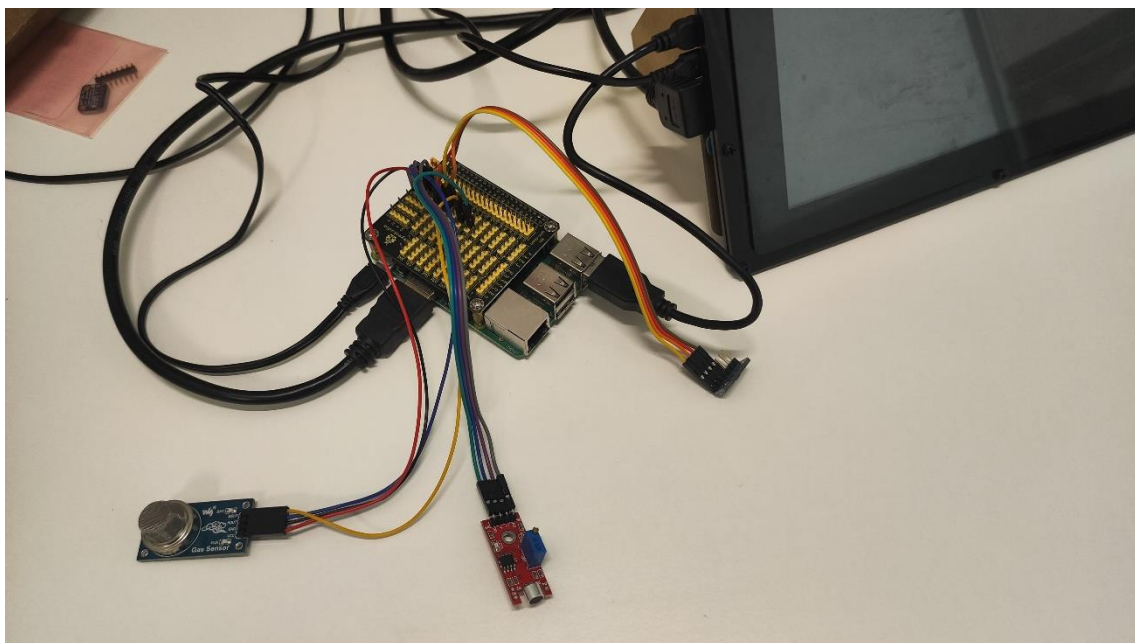


Figura 11 - Desenvolvimento da Infraestrutura

Além disso, foi necessário utilizar um DAC (Digital-to-Analog Converter) para converter sinais digitais de volta para analógicos, se necessário. Isso garantiu que todos os tipos de sensores pudessem ser integrados ao sistema, aumentando a sua versatilidade e capacidade de monitorização.

#### **4.3.2. Configuração Inicial do Raspberry Pi**

A configuração inicial do Raspberry Pi começou com a instalação do Raspberry Pi OS (Raspberry Pi Foundation, n.d.) em um cartão SD. Após a instalação do sistema Operativo, foram realizadas configurações básicas para preparar o Raspberry Pi para o desenvolvimento e operação contínua. As bibliotecas necessárias para o desenvolvimento do sistema foram instaladas, incluindo bibliotecas específicas para o controle do ADC e do DAC. Este processo envolveu a utilização de ferramentas como o IDE PyCharm para o desenvolvimento de scripts Python, e o FileZilla (FileZilla, n.d.) para transferir esses scripts para o Raspberry Pi.

#### **4.3.3. Desenvolvimento do Script Python**

O desenvolvimento do script Python foi uma parte crucial desta fase também é importante ressaltar que parte do script já estava funcional que era a parte de controlo do sensor de temperatura e umidade e o resto do script desenvolvi para os outros sensores ficarem funcionais. O script foi responsável por ler os dados dos sensores conectados ao Raspberry Pi, processá-los e enviá-los para um servidor remoto. Este script foi escrito localmente no PyCharm e, posteriormente, transferido para o Raspberry Pi utilizando o FileZilla para ser testado isto devido a biblioteca board não funcionar localmente no computador, mas sim na placa Raspberry Pi. Para executar o script remotamente, foi utilizado o PuTTY (*Simon Tatham, n.d.*), que permitiu uma conexão segura e eficiente ao Raspberry Pi.

O próximo passo no desenvolvimento foi a integração do Raspberry Pi com um servidor Apache remoto. O Raspberry Pi foi configurado para enviar os dados coletados dos sensores para este servidor através de uma API REST. O servidor Apache, por sua vez, foi responsável por armazenar os dados em uma base de dados MySQL e exibi-los em

uma interface web. Esta interface foi desenvolvida para ser acessível tanto para os participantes do evento quanto para os organizadores.

#### **4.3.4. Desenvolvimento da Interface Gráfica**

A interface gráfica foi projetada para proporcionar uma visualização e análise em tempo real dos dados coletados tanto para organizadores como para participantes. Os participantes do evento têm acesso a uma interface onde podem visualizar as condições ambientais em tempo real no caso é a dashboard mostrada na figura 12, enquanto os organizadores tinham acesso a uma área administrativa protegida por autenticação. Nesta área, os organizadores podem configurar diversos aspetos do sistema, como:

- **Configuração de Condições de Alerta:** Permite definir limiares para os parâmetros monitorizados (temperatura, humidade, qualidade do ar e ruído) e configurar alertas automáticos quando esses limiares fossem ultrapassados.
- **Gestão de Sensores:** Facilitava a adição e remoção de sensores, permitindo uma adaptação rápida às necessidades do evento
- **Gestão de Espaços:** Permitida o upload de plantas dos locais do evento e o desenho de polígonos para uma melhor visualização e gestão dos espaços monitorizados.
- **Consulta de Histórico de Dados:** Oferecia a possibilidade de consultar os dados históricos coletados pelos sensores, que podiam ser ordenados de diferentes formas para facilitar a análise.





Figura 12 - Página de Dashboard

A Figura 12 representa a dashboard e página inicial do site onde todos os utilizadores têm acesso sejam eles organizadores do evento ou visitantes do evento e nesta página podem ver os dados coletados pelos sensores e caso o utilizador esteja autenticado também pode aceder as páginas de configuração dos espaços, sensores e alertas. Esta página já me foi entregue desenvolvida apenas fiz as alterações necessárias para o funcionamento, mas o resto das páginas foram desenvolvidas de raiz.



Figura 13 - Página de Histórico de dados

A Figura 13 representa a página de Histórico do sistema todos os dados coletados dos sensores são exibidos nesta página e permite também ordenar a informação por ordem

crescente e decrescente por cada tipo de valor por ID, Data, Temperatura, Humidade, Ruído e Qualidade do Ar.

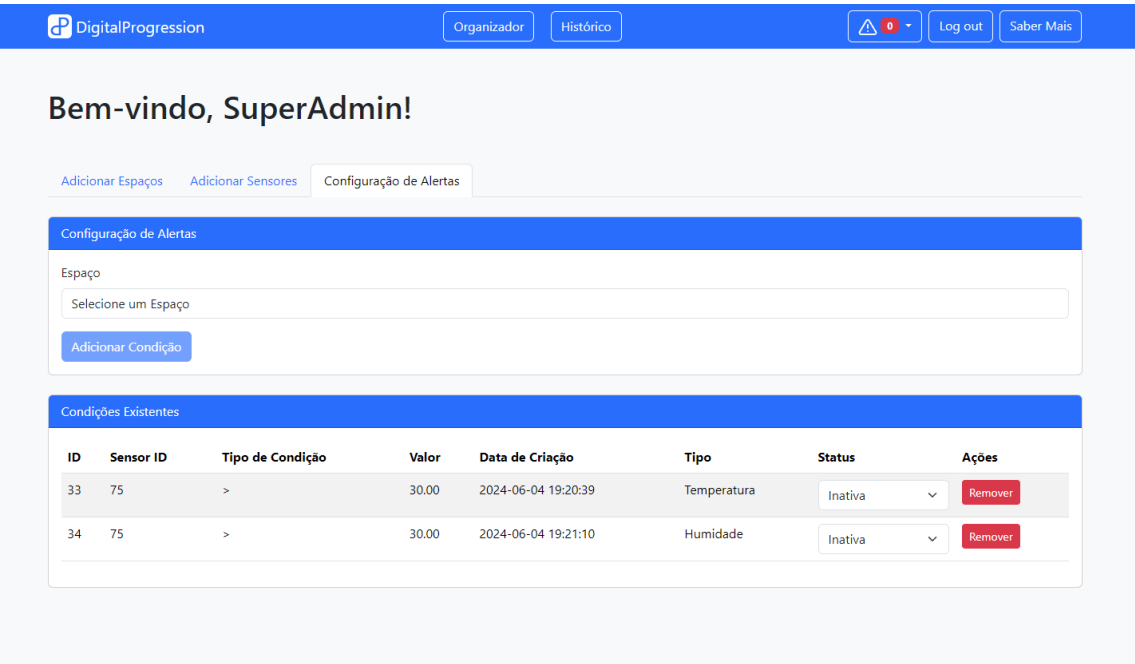


Figura 14 - Página de Configuração de Alertas

A Figura 14 representa a página de configuração de alertas onde o organizador pode criar, remover condições e editar o status das condições e também permite visualizar a informação das condições existentes. As condições definem os limiares de condição para exibir os alertas, ou seja, quando o valor de um sensor esta a cima ou abaixo do valor definido pelo organizador exibe um alerta para esse sensor.

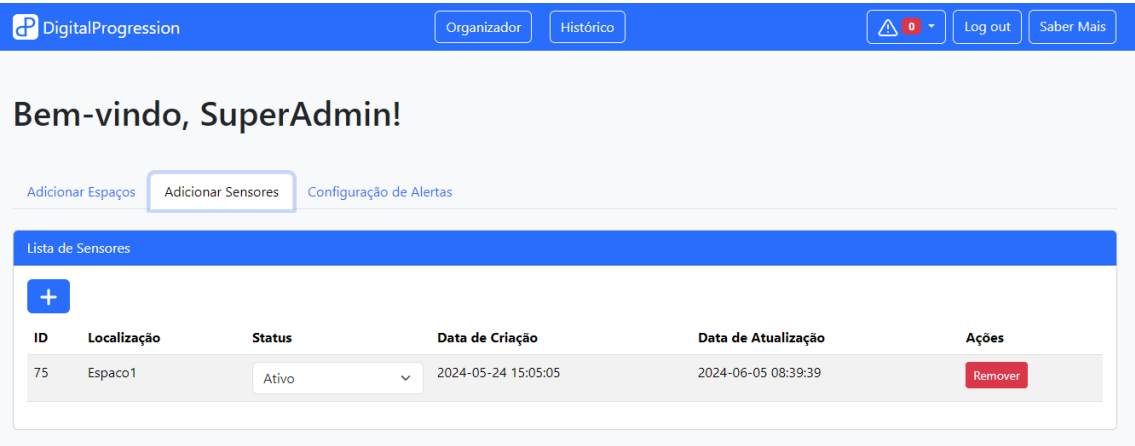


Figura 15 - Página de Sensores

A Figura 15 é a página de configuração de sensores onde permite ao organizador adicionar, remover sensores e editar o status do sensor bem como visualizar a informação dos sensores existentes

DigitalProgression

Organizador

Histórico

Log out

Saber Mais

Bem-vindo, SuperAdmin!

Adicionar Espaços

Adicionar Sensores

Configuração de Alertas

Lista de Espaços

ID	Nome do Espaço	Descrição	Ações
16	Espaco1	Descrição do Espaco1	<div>Remover</div>
17	Espaco2	Descrição do Espaco2	<div>Remover</div>

Upload de Plantas e Adição de Polígonos


Adicionar Planta

Remover Planta

Planta

5863620be6f596e41b737feb96409ecb.jpg

ÁREA DA CASA = 106.41 m<sup>2</sup> -  
ÁREA EXTERNA = 43.50 m<sup>2</sup> -  
ÁREA TOTAL = 149.91 m<sup>2</sup> ..



ID do Espaço	Nome do Espaço	IDs dos Sensores	Polígonos	Ações
--------------	----------------	------------------	-----------	-------

Adicionar Polígono

Travar Polígono

Salvar Polígono

Figura 16 - Página de Espaços e Associação de Polígonos

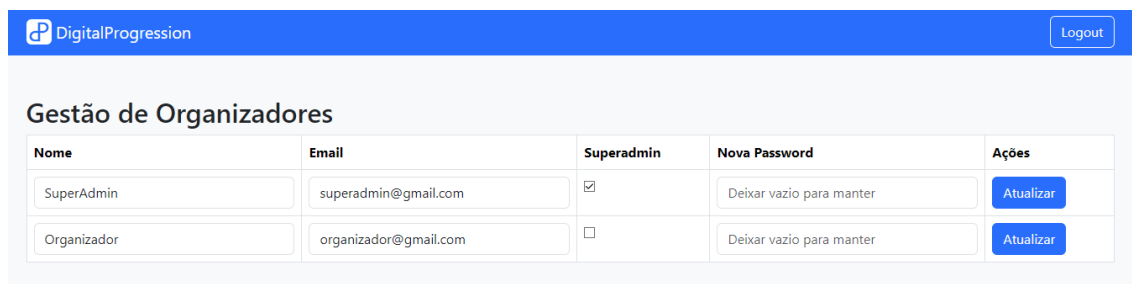
30

A Figura 16 é a página de espaços e nesta página foi implementado a possibilidade de adicionar, remover espaços e também permite ao organizador realizar o upload de plantas do evento bem como remove-las e nessas plantas permite desenhar, editar e remover os pontos do polígono correspondentes a um espaço tendo este de ser posteriormente associado ao polígono o polígono é guardado na BD bem como a informação referente a esse polígono. Os polígonos existentes também são exibidos na planta e a informação do espaço pode ser vista ao passar o cursor por cima de um polígono existente.

A imagem mostra a interface de usuário para o registro de organizadores. No topo, há uma barra azul com o logo 'DigitalProgression' à esquerda, um botão 'Contas' no centro e um botão 'Logout' à direita. Abaixo, centralizado, há um formulário branco com o título 'Registo de Organizador' em uma barra azul. O formulário contém campos de entrada para 'Nome', 'Email' e 'Password'. Abaixo desses campos, há uma opção de seleção 'Superadmin' com uma caixa de seleção vazia. No final do formulário, há um botão azul com o texto 'Registrar'.

**Figura 17 - Página de registo**

A Figura 17 é a página de registo de contas de SuperAdmin e de Organizadores e apenas SuperAdmins têm acesso esta página. Ao efetuar login com conta de SuperAdmin é redirecionado para esta página onde pode registar contas de SuperAdmin e de Organizador para posteriormente efetuar login.



Nome	Email	Superadmin	Nova Password	Ações
<input type="text" value="SuperAdmin"/>	<input type="text" value="superadmin@gmail.com"/>	<input checked="" type="checkbox"/>	<input type="text" value="Deixar vazio para manter"/>	<button>Atualizar</button>
<input type="text" value="Organizador"/>	<input type="text" value="organizador@gmail.com"/>	<input type="checkbox"/>	<input type="text" value="Deixar vazio para manter"/>	<button>Atualizar</button>

Figura 18 - Página de Gestão de Contas

Para além da página de registo o SuperAdmin também tem acesso a uma página de gestão de contas como se pode ver na Figura 18 onde pode alterar vários aspetos das contas como nome, email, se é SuperAdmin ou não e alterar a password.

#### 4.3.5. Funcionalidades de Alerta

O sistema também foi integrado com funcionalidades de alerta para a melhoria das condições de conforto. Estas funcionalidades foram desenvolvidas para fornecer notificações em tempo real sobre condições ambientais fora dos parâmetros desejados e para sugerir ações corretivas com base nos dados coletados. Isto funciona através da criação de condições de alerta por parte do organizador e o sistema esta constantemente de 5 em 5 segundos a verificar se essas condições se cumprem se sim aparece um alerta em forma de toast com a informação do alerta como o timestamp, o valor (temperatura, umidade, etc..) e o id do sensor como se pode ver na Figura 19. Os alertas também ficam guardados em um dropdown na navbar para consulta posterior.

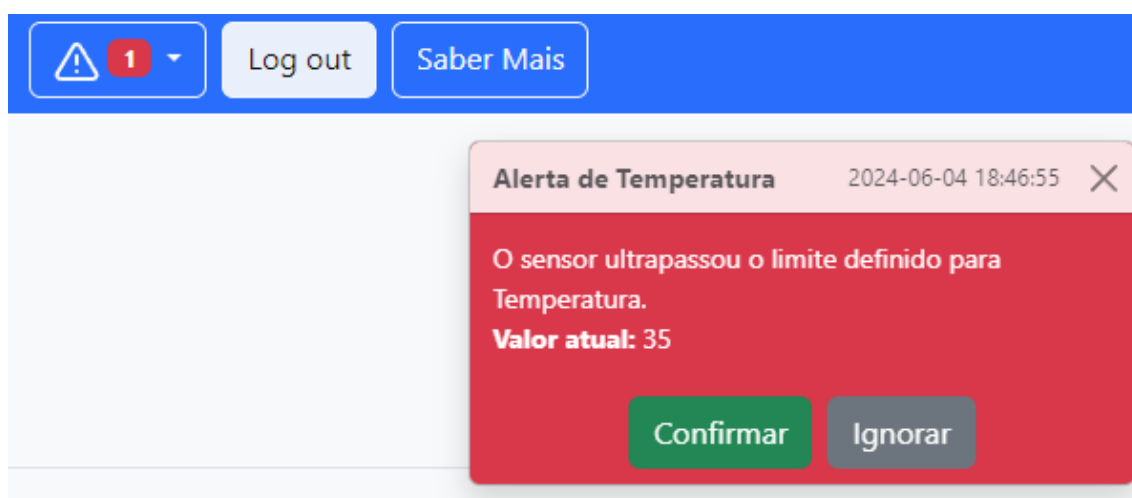


Figura 19 - Exibição de Alerta

#### 4.3.6. Montagem e Ligações dos Sensores

A montagem e ligações dos sensores ao Raspberry Pi foi realizada por mim no local da empresa com atenção especial à precisão das conexões. Foram usados um sensor por cada unidade de medida, ou seja, uma para temperatura e umidade, um para ruído, e outro para qualidade do ar totalizando 3 sensores como se pode ver na figura 22 abaixo e cada sensor foi conectado aos pinos apropriados do Raspberry Pi, garantindo que os dados coletados fossem precisos e confiáveis. A montagem física foi feita de forma a garantir que os sensores estivessem posicionados adequadamente.

Sendo apenas um projeto protótipo foi idealizado pelo supervisor o Raspberry Pi e os sensores ficarem em uma caixa fechada o que esteticamente ficaria mais agradável como se pode ver nas Figuras 20 e 21, mas faria com que os sensores não funcionassem corretamente por estarem dentro da caixa então no fim foi utilizada uma caixa diferente que permitia a melhor coleta dos dados onde os sensores ficam de fora da caixa e também permitiu uma maior proteção do Raspberry Pi.



Figura 22 - Caixa Inicial



Figura 21 - Planificação da caixa



Figura 20 - Caixa Final

## **4.4. Fase 5 - Testes**

Realizei testes de funcionalidade e fiabilidade dos componentes IoT, incluindo a calibração dos sensores de modo a fornecerem resultados precisos. Os resultados dos testes foram cuidadosamente analisados e documentados para garantir a qualidade e a confiabilidade do sistema. Para além disso foram também testadas as funcionalidades da interface web na área do Organizador para analisar a existência de possíveis bugs no sistema ou algumas propriedades do sistema que estejam em falta.

## **4.5. Fase 6 - Documentação**

Durante todo o processo, foi preparada documentação técnica abrangente, incluindo documento de requisitos, documento de testes e um documento para os diagramas no caso o diagrama de Casos de Uso, Diagrama ER e Diagrama de Deployment. Também foi elaborado um manual de utilizador e configuração para utilizadores finais documentação esta que é essencial para garantir uma utilização eficaz e segura do sistema. Nos anexos estão a documentação do diagrama e casos de uso que foi abordado na documentação de especificação de requisitos que também está nos anexos bem como o documento de testes e o manual do utilizador.



## 5. Conclusões

Neste relatório, foi descrito o desenvolvimento de um sistema de monitorização para eventos, abrangendo desde a escolha dos componentes de hardware até a implementação e teste do sistema completo. O objetivo principal do projeto foi criar uma solução eficiente e escalável para monitorar temperatura, umidade, qualidade do ar e ruído em eventos, utilizando sensores IoT integrados a um Raspberry Pi, armazenando e exibindo os dados por meio de um servidor Apache.

### 5.1. Forças

Este projeto apresentou várias forças significativas que contribuíram para o seu sucesso. Primeiramente, a integração bem-sucedida de hardware e software resultou em um sistema funcional e eficiente, capaz de monitorar múltiplos parâmetros ambientais em tempo real. A escolha do Raspberry Pi como unidade central de processamento, juntamente com os sensores de temperatura DS18B20 e DHT22, demonstrou ser eficaz na coleta de dados precisos.

Além disso, a utilização de um shield ADC (MCP3008) e de um DAC (MCP4725) permitiu a integração de sensores analógicos, ampliando as capacidades do sistema. A configuração de acesso remoto ao Raspberry Pi via SSH utilizando PuTTY e a transferência de arquivos com FileZilla facilitaram a administração e a manutenção do sistema.

A implementação de um servidor Apache para armazenamento e visualização dos dados coletados foi outra força do trabalho visto que já possuía algum conhecimento acerca de PHP e servidor Apache, garantindo que os dados pudessem ser acedidos de maneira eficiente e organizada. A documentação detalhada e os diagramas desenvolvidos ajudaram a esclarecer a arquitetura do sistema, facilitando sua compreensão e possível replicação.

## 5.2. Limitações

Apesar das forças significativas, o projeto também apresentou algumas limitações e fraquezas que devem ser reconhecidas. A complexidade da integração inicial dos sensores analógicos com o Raspberry Pi representou um desafio considerável, mas com o site StackOverflow (Stack Overflow, n.d.) ajudou me bastante no que toca a programação em Python. Esta tarefa exigiu ajustes e testes adicionais, o que aumentou o tempo necessário para a implementação.

Além disso, tive dificuldades específicas na implementação de plantas e polígonos, áreas nas quais eu não tinha experiência prévia. A falta de familiaridade com essas tarefas levou a um processo de aprendizagem mais longo e a necessidade de buscar recursos adicionais para entender e aplicar as técnicas necessárias.

A criação de diagramas e documentos técnicos foi outra área que apresentou desafios. Embora essa fosse uma tarefa nova para mim, exigiu um esforço significativo para aprender e desenvolver documentação clara e precisa. Este processo foi essencial para garantir a comunicação efetiva do projeto, mas também foi uma das fraquezas iniciais que precisei superar.

## 5.3. Trabalho Futuro

O desenvolvimento deste sistema de monitorização para eventos abriu várias possibilidades para melhorias e expansões futuras. Com base nas experiências adquiridas e nas limitações encontradas, algumas áreas chave foram identificadas para trabalho futuro.

A interface de utilizador é uma das coisas que pode ser melhorada está funcional, mas podia estar mais bonita e agradável para além de proporcionar uma experiência mais intuitiva e informativa. Deixar o sistema mais elaborado visto que é apenas um protótipo e por isso esta bastante simples.

Embora o sistema atual monitore temperatura, umidade, qualidade do ar e ruído, há oportunidades para expandir as capacidades de monitoramento com a adição de novos tipos de sensores. Sensores de luminosidade, pressão atmosférica e detecção de gases específicos podem ser integrados para fornecer um quadro mais completo das condições ambientais. Essa expansão permitirá uma análise mais detalhada e uma resposta mais precisa às necessidades dos eventos.



## 6. Referências

Adafruit Industries. (n.d.). *Adafruit's Raspberry Pi Lesson 4 and GPIO Setup*.  
<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c>

Adafruit Industries. (n.d. -a). *CircuitPython on Linux and Raspberry Pi*.  
<https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/installing-circuitpython-on-raspberry-pi>

ClickUp™. (n.d.). *One app to replace them all*. <https://www.clickup.com>

Extech Instruments. (n.d.). *Environmental & Mechanical Test*.  
<https://pt.farnell.com/b/extech-instruments>

Figma, Inc. (2019). *Figma: the collaborative interface design tool*.  
<https://www.figma.com>

FileZilla. (n.d.). *The free FTP solution*. <https://filezilla-project.org>

JetBrains s.r.o. (n.d.). *DataGrip: The Cross-Platform IDE for Databases & SQL by JetBrains*.  
JetBrains. from <https://www.jetbrains.com/datagrip>

JetBrains s.r.o. (n.d. -a). *PyCharm: the Python IDE for Professional Developers by JetBrains*. <https://www.jetbrains.com/pycharm>

JGraph Ltd. (n.d.). *Draw.io - Diagram Software and Flowchart Maker*.  
<https://www.drawio.com>

Jri MySirius. (n.d.). *Cloud-based temperature monitoring solution*. <https://www.jri-mysirius.com>

Lucid Software Inc. (2019). *LucidChart*. <https://www.lucidchart.com>

MICROSOFT. (2016, April 14). *Visual Studio Code*. Visualstudio.com.  
<https://code.visualstudio.com>

NatéoSanté. (n.d). *Air Coach: CO2 sensor and Indoor Air Quality monitor*.  
<https://www.nateosante.com/fr/categorie-produit/mesure-qualite-air/>

Nexxto (n.d.). *Nexxto | Sistema monitor de temperatura e umidade no setor de saúde*.  
<https://nexxto.com>

Postman. (n.d.). *The Collaboration Platform for API Development*.  
<https://www.postman.com>

Raspberry Pi Foundation. (n.d.). *Raspberry Pi OS*.  
<https://www.raspberrypi.com/software>

Simon Tatham. (n.d.). *PuTTY - a free SSH and telnet client for Windows*.  
<https://www.putty.org>

SISMETRO. (n.d.). *Manutenção, telemetria e IoT*. <https://www.sismetron.com/ambientes>

Stack Overflow. (n.d.). *How do I setup a local HTTP server using Python*.  
<https://stackoverflow.com/questions/27977972/how-do-i-setup-a-local-http-server-using-python>

Vaisala. (2023, December 20). *Sistema de medição ambiental e industrial*.  
<https://www.vaisala.com/pt>

*Visual Paradigm*. (2019). A suite of design, analysis and management tools to drive your IT project development and digital transformation. <https://www.visual-paradigm.com>

## 7. Anexos

### 7.1. Anexo 1 - Design Inicial da Dashboard

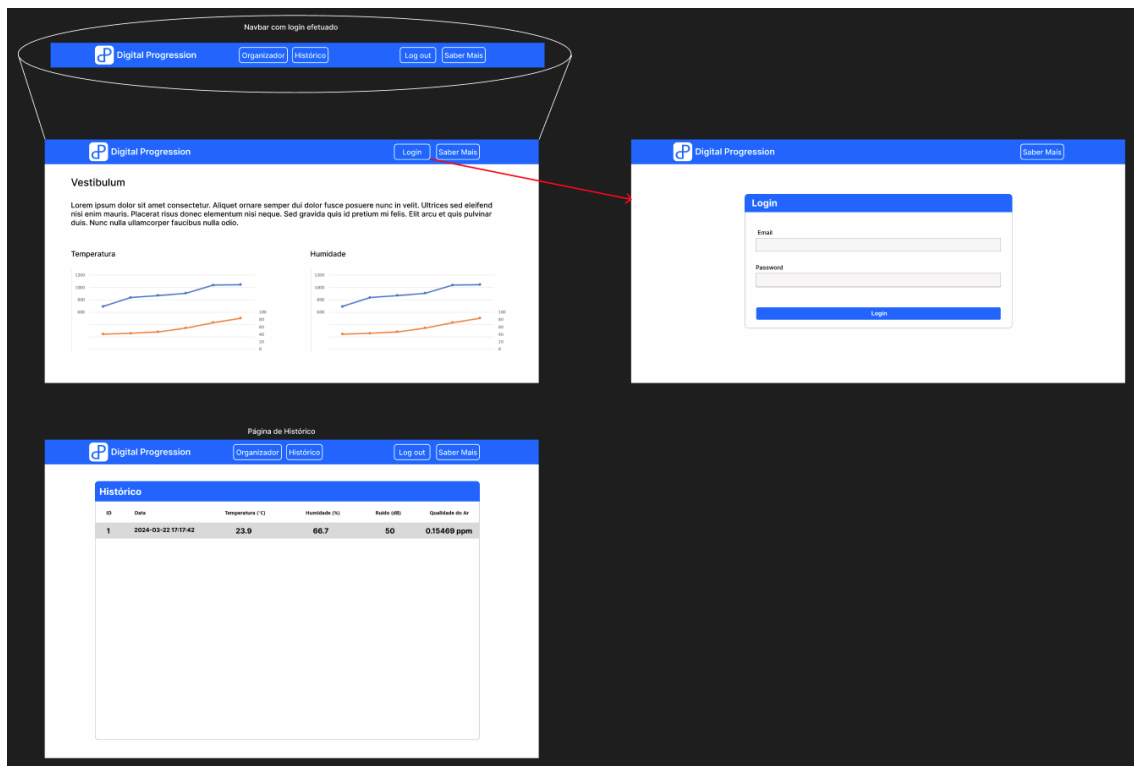


Figura 23 – Design da dashboard inicial

O design acima foi idealizado por mim sem qualquer tipo de opinião do supervisor.



## 7.2. Anexo 2 - Design do site



**Figura 24 - Design de algumas páginas do site**

Já este foi juntamente e com a opinião e ajuda do supervisor.

## 7.3. Anexo 3 – Script da Base de Dados SQL

```
create table alertconditions
(
    id            int auto_increment
                primary key,
    sensor_id     int
not null,
    condition_type varchar(50)
not null,
    value         decimal(10, 2)
not null,
    created_at    timestamp                        default
CURRENT_TIMESTAMP null,
    type          varchar(30)
null,
    status        enum ('active', 'inactive') default 'active'
null
);

create table espacos
(
    id            int auto_increment
                primary key,
    nome          varchar(255) null,
    descricao     text         null
);

create table organizadores
(
    id            int auto_increment
                primary key,
    email         varchar(255)      not null,
    password      varchar(255)      not null,
    name          varchar(255)      not null,
    is_superadmin tinyint(1) default 0 not null,
    constraint email
                unique (email)
);

create table plantas
(
    id            int auto_increment
                primary key,
    planta_path   varchar(255) not null
);

create table polygons
(
    id            int auto_increment
                primary key,
    space_id     int      not null,
    sensor_ids   varchar(144) null,
    space_name   varchar(255) null,
    points       text      not null,
    plant_id     int       not null,
    constraint polygons_pk
                unique (space_id),
    constraint fk_plant
                foreign key (plant_id) references plantas (id),
```

```

        constraint polygons_ibfk_1
            foreign key (space_id) references espacios (id)
    );

create index space_id_index
    on polygons (space_id);

create table sensors
(
    id            int auto_increment
        primary key,
    location      varchar(255)                not null,
    status        varchar(50) default 'ativo'  null,
    created_at    timestamp default CURRENT_TIMESTAMP null,
    updated_at    timestamp default CURRENT_TIMESTAMP null on update
CURRENT_TIMESTAMP,
    username      varchar(255)                not null,
    password      varchar(255)                not null
);

create table espacios_sensores
(
    id            int auto_increment
        primary key,
    espacio_id    int null,
    sensor_id     int null,
    constraint espacios_sensores_ibfk_1
        foreign key (espaco_id) references espacios (id),
    constraint espacios_sensores_ibfk_2
        foreign key (sensor_id) references sensors (id)
);

create index espaco_id
    on espacios_sensores (espaco_id);

create index sensor_id
    on espacios_sensores (sensor_id);

create table sensordata
(
    id            int auto_increment
        primary key,
    sensor_id     int                not null,
    temperature    float             not null,
    humidity       float             not null,
    noise          float             not null,
    air_quality    float             not null,
    timestamp      timestamp default CURRENT_TIMESTAMP null,
    constraint sensordata_ibfk_1
        foreign key (sensor_id) references sensors (id)
);

create index sensor_id
    on sensordata (sensor_id);

```