



UNIVERSIDADE  
LUSÓFONA

# Sistema de Monitorização de Edifícios

## Trabalho Final de curso

Relatório Final 2º Semestre

Nome do Aluno: Felipe Genovese Bandeira – a22102727

Nome do Orientador: João Pedro Calado Barradas Branco Pavia

Nome do Co-Orientador: João Pedro Leal Abalada De Matos Carvalho

Trabalho Final de Curso | LIG | 27/11/2022

[www.ulusofona.pt](http://www.ulusofona.pt)

## **Direitos de cópia**

*(Sistema de Monitorização de Edifícios)*, Copyright de *(Felipe Genovese Bandeira)*, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

---

## Resumo

O mercado de automação predial e edifícios inteligentes está em ascensão. Cada vez mais empresas e pessoas têm interesse em Internet of Things (IoT) e em sistemas que facilitem o controle e monitorização de suas casas ou edifícios comerciais. Atualmente, existem diversos sistemas voltados para monitorização de edifícios que ajudam a coletar e analisar dados do edifício para auxiliar na tomada de decisões. Porém, a maior parte dos sistemas já estabelecidos no mercado, possuem alto custo de instalação e manutenção. Levando isso em consideração, este documento analisa soluções já estabelecidas no mercado e propõe a criação de um sistema que seja de fácil utilização para qualquer pessoa, com baixos requisitos de instalação e manutenção.

O presente documento tem como objetivo documentar o funcionamento de um sistema de monitorização de edifícios desenvolvido como trabalho final de curso. As funcionalidades do sistema têm o intuito de auxiliar na monitorização de um edifício, seja este comercial ou residencial. Através de um microcontrolador Arduino e múltiplos sensores, o sistema será capaz de medir níveis de temperatura, humidade, luminosidade, além de detectar quando uma porta é aberta ou fechada. Todos os dados adquiridos por um microcontrolador são enviados e armazenados numa base de dados. Estes dados podem ser acedidos e visualizados através de uma plataforma de visualização. Nesta plataforma, o utilizador poderá visualizar os dados armazenados e controlar algumas funcionalidades do sistema, como a ativação de alarmes, e será possível realizar a coleta de dados de forma manual ou através do estabelecimento de uma rotina periódica pré-definida pelo utilizador.

**Palavras-chave:** Edifícios Inteligentes, Microcontrolador, Monitorização de Edifícios, Dados, Arduino

## **Abstract**

The building automation and smart buildings market is growing fast. People and companies are increasingly more interested in Internet of Things (IoT) products and systems that help control and monitor their houses or commercial buildings. Currently, there are several systems aimed at monitoring buildings that help to collect and analyze data to assist in decision making. However, most of the systems that are already established in the market have high installation and maintenance costs. With this in mind, this document analyzes solutions that are already established in the market and proposes the creation of a system that is easy to use for any person, with low installation and maintenance requirements.

This document aims at detailing the operation of a Building Monitoring System developed as a Final Course Work. This system and its functionalities will help monitoring a build, it being a commercial or residential one. Through an Arduino microcontroller and multiple sensors, the system will be able to measure temperature, humidity, light and whether a door is opened or closed. All the data collected by the microcontroller is uploaded to a database. This data can be accessed and viewed through a viewing platform. In this platform, the user will be able to view all the stored data and control some of the main functionalities of the system, such as setting alarms that are triggered when a door opens. It will also be possible to collect new data from the microcontrollers manually or automatically, through a pre-determined period.

**Keywords:** Smart Buildings, Building Monitoring, Microcontroller, Data, Arduino

---

# Índice

Resumo.....	iii
Abstract .....	iv
Lista de Figuras .....	vii
Lista de Tabelas .....	ix
1 Identificação do Problema .....	1
1.1 Digitalização e Automação .....	1
1.2 Automação de edifícios .....	1
2 Viabilidade e Pertinência.....	2
3 Benchmarking.....	3
3.1 Visão Geral do Mercado .....	3
3.2 Schneider Electric .....	3
3.3 Siemens .....	4
3.4 Phoenix Contact .....	5
3.5 Conclusões.....	6
4 Engenharia.....	7
4.1 Levantamento e Análise de Requisitos .....	7
4.1.1 Requisitos Não-Funcionais .....	7
4.1.2 Requisitos Funcionais .....	8
4.1.3 Requisitos Técnicos .....	9
4.2 Casos de Uso .....	9
4.3 Mockups .....	10
4.3.1 Página Principal .....	10
4.3.2 Página de Alarmes .....	11
4.3.3 Página de Dados Armazenados .....	11
4.4 Base de Dados .....	12
4.5 Diagrama de Atividades .....	13
5 Solução Desenvolvida.....	16
5.1 Visão Geral .....	16
5.2 Arduino e Sensores .....	16
5.3 Plataforma de Visualização .....	19

6	Método e Planeamento .....	20
7	Resultados .....	22
7.1	Teste Cases .....	22
7.2	Testes com Utilizadores .....	24
8	Conclusão e Trabalhos Futuros .....	26
	Bibliografia .....	27
	Anexos .....	28
	Manual da Aplicação .....	28
	Progresso de Trabalho – Janeiro/Fevereiro .....	33
	Progresso de Trabalho – Março/Abril .....	40
	Progresso de Trabalho – Maio/Junho .....	43
	Teste de Usabilidade .....	47
	Questionário de Satisfação .....	50
	Glossário .....	52

---

# Lista de Figuras

Figura 2 - Principais objetivos do EcoStruxure .....	4
Figura 1 - Interface do EcoStruxure Building Operation.....	4
Figura 3 - Sistema Connect Box .....	5
Figura 4 - Sistema Emalytics .....	6
Figura 5 - Use cases .....	9
Figura 6 - Mockup Página Principal .....	10
Figura 7 - Mockup Página de Alarmes .....	11
Figura 8 - Mockup da Página de Dados Armazenados .....	11
Figura 9 - Diagrama Entidade-Relação da Base de Dados .....	12
Figura 10 - Diagrama de Atividades 1.....	13
Figura 11 - Diagrama de Atividades 2.....	14
Figura 12 - Diagrama de Atividades 3.....	15
Figura 13 - Microcontrolador Arduino.....	17
Figura 14 - Sensor de temperatura.....	17
Figura 15 - Sensor LDR.....	17
Figura 16 - Sensor DHT22 .....	17
Figura 17 - Sensor Ultrasónico.....	17
Figura 18 - Buzzer piezoelétrico .....	18
Figura 19 - Exemplo Simulação do Tinkercad .....	18
Figura 20 - Calendário em Formato Gantt.....	20
Figura 21 - Resultados do questionário.....	25
Figura 22 - Manual: Home Screen .....	28
Figura 23 - Manual: botões de controle de estores .....	28
Figura 24 - Manual: dispositivo conectado .....	29
Figura 25 - Manual: dispositivo desconectado .....	29
Figura 26 - Manual: bottom bar .....	29
Figura 27 - Manual: alarmes.....	29
Figura 28 - Manual: alarme de temperatura .....	30
Figura 29 - Manual: mensagem popup.....	30
Figura 30 - Manual: página de Dados Armazenados .....	31
Figura 31 - Manual: gráficos .....	32
Figura 32 - Manual: gráficos com filtros aplicados .....	32
Figura 33 - Simulador do Tinkercad.....	33
Figura 34 - Foto circuito.....	34
Figura 35 - Foto circuito (2) .....	35
Figura 36 - Exemplo de output dos sensores .....	35
Figura 37 - Ferramenta SceneBuilder para construção das interfaces com JavaFX .....	36
Figura 38 - Estado atual da página principal.....	37
Figura 39 - Estado atual da página de alarmes.....	38
Figura 40 - Estado atual da página de Dados .....	38
Figura 41 - Exemplo de dados enviados à plataforma.....	40
Figura 42 - Página Inicial (Update de Março/Abril) .....	41
Figura 43 - Página de Dados (Update de Março/Abril).....	42
Figura 44 - Página de Alarmes (Update de Março/Abril) .....	42
Figura 45 - Exemplo Código de Filtros .....	43
Figura 46 - Exemplo Contrução da Query.....	44
Figura 47 - Exemplo código de envio de queries à base de dados .....	44

Figura 48 - Página inicial (versão final) .....	45
Figura 49 - Página de Dados (versão final) .....	45
Figura 50 - Página de Alarmes (versão final) .....	46
Figura 51 - Página de Gráficos (versão final) .....	46



---

## Lista de Tabelas

Tabela 1 - Requisitos Não Funcionais .....	8
Tabela 2 - Requisitos Funcionais .....	8
Tabela 3 - Requisitos Técnicos.....	9
Tabela 4 - Resultados dos testes .....	24
Tabela 5 - Comandos do Arduino .....	40



# **1 Identificação do Problema**

## **1.1 Digitalização e Automação**

O processo de digitalização permite-nos viver num mundo cada vez mais conectado e onde tudo se encontra à distância de um clique. Através de softwares, websites e sistemas, processos que antes eram feitos manualmente passaram a ser realizados de maneira digital e automatizada (Tutida, 2021). As empresas que se adaptaram aos novos avanços tecnológicos veem grandes benefícios, como, aumento da produtividade e redução de desperdícios, que levam a aumento dos lucros da empresa (Tutida, 2021).

Para além da digitalização, as empresas também buscam aumentar a eficiência através da automação de múltiplos processos. O objetivo da automação é de substituir métodos antigos, ultrapassados e de baixa eficiência por métodos mais ágeis, rápidos e inovadores (DocuSign, 2021). De maneira geral, é possível automatizar todo tipo de tarefa, desde as mais simples às mais complexas, e assim torná-las mais rápidas e eficientes (DocuSign, 2021).

## **1.2 Automação de edifícios**

Com o surgimento de sistemas digitais e automáticos, surge também a ideia de implementar este tipo de sistemas em edifícios, com o intuito de torná-los mais eficientes, integrados, seguros e inteligentes. Apesar de o conceito de edifícios inteligentes já existir há algum tempo, atualmente o foco é tentar tornar as edificações integradas tanto no âmbito de sistemas, como também no âmbito da sua arquitetura. Isso possibilita a integração de diversas funcionalidades do edifício, tornando-o mais eficiente e sustentável (A (Re) Evolução Da Automação Predial, 2015).

No contexto de edifícios empresariais, algumas funcionalidades que já são implementadas são: sistema que regula persianas de acordo com a incidência de luz solar nas janelas, monitorização do ar condicionado, controle de acesso de pessoas no edifício, dentre outros (A (Re) Evolução Da Automação Predial, 2015). Muitas empresas veem este tipo de integração de sistemas como uma maneira de economizar os recursos que são consumidos no edifício, como água e energia. Edifícios que possuem garagem, por exemplo, podem utilizar muita energia de maneira desnecessária ao deixar as luzes da garagem ligadas por muito tempo, mas em um edifício inteligente as luzes são ligadas somente conforme necessário, o que reduz muito o consumo de energia (A (Re) Evolução Da Automação Predial, 2015).

A principal dificuldade que muitas empresas encontram está no alto custo que alguns destes sistemas podem ter. Para além disso existem também o problema com a instalação dos sistemas, algo que pode implicar muitas mudanças no funcionamento do edifício como um todo. Portanto, surge assim a necessidade de se encontrar um sistema de Building Intelligence que seja de baixo custo de instalação e de manutenção e que consiga dar resposta às necessidades das empresas e consumidores.

## **2 Viabilidade e Pertinência**

A procura por soluções no âmbito da Internet das Coisas (IoT) e sistemas integrados tem aumentado significativamente nos últimos anos. O objetivo da IoT é conectar dispositivos diversos pela Internet, permitindo sua gestão remota. Essa interconexão inteligente dos dispositivos traz consigo uma enorme quantidade de dados, que desempenham um papel crucial na tomada de decisões, deteção de problemas e na eficiência operacional. A grande quantidade de dados que é obtida pelos dispositivos inteligentes é de extrema importância, uma vez que a sua aquisição e análise auxilia, principalmente, na tomada de decisões e na resolução e deteção de problemas que antes não eram detetados e que poderiam gerar grandes prejuízos. Atualmente as empresas procuram por sistemas que sejam de baixo custo de instalação e de manutenção e que auxiliem o máximo na redução de custos e redução da utilização de recursos como energia e água.

Este projeto visa criar um sistema que facilite a monitorização de edifícios por meio da aquisição e análise de dados utilizando um único dispositivo. A principal ênfase é fornecer uma plataforma simples e de fácil utilização, permitindo o controle de aspetos do edifício e o armazenamento de dados para análise futura. A solução proposta concentra-se na simplicidade e usabilidade, oferecendo aos usuários uma plataforma intuitiva para controlar diferentes aspetos do edifício. Com esse sistema, espera-se alcançar uma gestão mais eficiente, otimizando o consumo de recursos, como energia e água. Embora a quantificação específica do custo baixo não seja mencionada, o foco está em fornecer uma solução acessível e eficaz para monitorar e controlar o edifício.

Inicialmente, o sistema será desenvolvido para edifícios residenciais e de pequeno porte. No entanto, sua flexibilidade permitirá que seja adaptado e expandido para atender às necessidades de edifícios empresariais. Embora não haja mais a comunicação sem fio entre dispositivos, a solução oferece a capacidade de controle centralizado e aquisição de dados para auxiliar na gestão do edifício. A abordagem centrada na simplicidade e nos benefícios práticos da solução torna-a relevante para atender às demandas do mercado atual por sistemas de monitorização eficientes e fáceis de usar.

## 3 Benchmarking

### 3.1 Visão Geral do Mercado

O mercado de automação predial e de prédios inteligentes está em ascensão. Estima-se que, em 2021, o tamanho do mercado global de sistemas de automação predial era de aproximadamente US\$77,1 mil milhões de dólares (*Building Automation System Market*, 2022). As projeções de crescimento do mercado indicam que, até 2030, o valor deve subir para mais de US\$200 mil milhões (*Building Automation System Market*, 2022). O aumento da procura por prédios mais sustentáveis, económicos, seguros e com baixos custos de manutenção é o que atualmente impulsiona este mercado (*Building Automation System Market*, 2022).

A maior parte deste mercado é ocupado por sistemas direcionados para edifícios comerciais, representando cerca de 40% do mercado em 2021. Atualmente existe uma busca muito grande por sistemas que utilizam tecnologia sem fios, uma vez que estes sistemas são mais fáceis de se instalar e apresentam uma maior flexibilidade e capacidade de controlo comparação com sistemas que utilizam fios (*Building Automation System Market*, 2022). A categoria de sistemas residenciais também deve obter um aumento considerável nos próximos anos, visto que existe uma procura cada vez maior por dispositivos IoT e por soluções que utilizam aplicações de *smartphone* para controlar casas inteligentes.

As principais empresas que atuam neste mercado são:

- Siemens
- Schneider Electric
- Honeywell International
- Bosch

### 3.2 Schneider Electric

Uma das maiores empresas que atua no setor de sistemas de gestão predial é a Schneider Electric, que é especializada em automação digital e gestão de energia. Através da criação de sistemas que combinam tecnologias sustentáveis, automação em tempo real e serviços integrados para casas e edifícios, a Schneider Electric tornou-se uma das principais referências no ramo de tecnologia e sustentabilidade (Schneider Electric Brasil, n.d.). Atualmente, o principal foco da empresa está em criar soluções para gestão predial que sejam sustentáveis e inovadoras (Schneider Electric Brasil, n.d.).

O EcoStruxure Building Operation é um sistema de gestão predial de última geração oferecido pela empresa. O foco do sistema está em providenciar funcionalidades que vão além dos sistemas tradicionais e cria edifícios que são inteligentes (Schneider Electric Brasil, n.d.). A solução é aberta, flexível, centrada nos dados e fornece um único centro de controlo capaz de monitorizar e gerir todos os tipos de edifício (Schneider Electric Brasil, n.d.).



Figura 1 - Principais objetivos do EcoStruxure

Fonte: <https://www.se.com/br/pt/work/products/product-launch/building-management-system/>



Figura 2 - Interface do EcoStruxure Building Operation

Fonte: <https://www.se.com/br/pt/work/products/product-launch/building-management-system/>

As principais funcionalidades do sistema são:

- Acesso a todos os dados sem um sistema *third-party*;
- Acesso aos dados através de uma aplicação móvel;
- Acesso em tempo real às operações do edifício com uma *dashboard* personalizada;
- Criação de relatórios;
- Ativação de alarmes;
- Controlos de iluminação e estores.

### 3.3 Siemens

Siemens é uma das maiores empresas de tecnologia do mundo com foco em indústrias, infraestrutura, transporte e assistência médica. A empresa possui diversos produtos que auxiliam na criação de edifícios inteligentes, permitindo que os recursos do edifício sejam utilizados de maneira mais inteligente, eficiente e sustentável.

Dentre as soluções de automação predial, a Siemens oferece a Connect Box, uma solução simples de IoT utilizada para controlar e monitorizar edifícios (*Connect Box - Your Simple IoT Solution (...)*, n.d.). Com foco em edifícios pequenos e médios, a Connect Box conecta todos os dispositivos de IoT existentes no edifício e permite integração com sistemas de automação predial ou sistemas de monitorização de edifício já existentes (*Connect Box - Your Simple IoT Solution (...)*, n.d.).

As principais funcionalidades e vantagens do sistema são:

- Compatibilidade com diversos dispositivos de IoT;
- Uma única interface para monitorização das operações do edifício;

- Interface pode ser acedida através do serviço de *cloud*;
- Integração com aplicações *third-party*;
- Notificações de alarmes;
- Criação de relatórios;
- Visualização em tempo real de dados recolhidos pelos dispositivos.



**Figura 3 - Sistema Connect Box**

Fonte: <https://new.siemens.com/global/en/products/buildings/automation/cloud-solutions/hardware/connect-box.html>

### 3.4 Phoenix Contact

A Phoenix Contact é uma empresa alemã líder de mercado na criação de sistemas e soluções voltados para as áreas de engenharia elétrica, eletrônica e de automação. A empresa oferece diversas soluções para gestão predial, dentre elas a Emalytics, que tem o intuito de ser um sistema de monitorização de edifícios com base em dispositivos IoT (*Automação Predial*, n.d.). O sistema funciona com uma plataforma única e central, onde todos os elementos do edifício podem ser monitorizados e controlados (*Automação Predial*, n.d.).

As principais funcionalidades e vantagens do sistema são:

- Plataforma única e centralizada para monitorização, automação e recolha de dados;
- Garantia de segurança das conexões entre dispositivos;
- Pode ser facilmente expandido e melhorado;
- Análise de dados a partir da criação de relatórios.



**Figura 4 - Sistema Emlytics**

Fonte: [www.phoenixcontact.com](http://www.phoenixcontact.com)

### **3.5 Conclusões**

A partir das pesquisas realizadas para este relatório, foi possível constatar que a maioria dos sistemas de monitorização de edifícios disponíveis no mercado seguem um padrão semelhante, com poucas diferenças entre eles. Todos esses sistemas têm como principal objetivo fornecer uma plataforma de visualização de dados única e centralizada, facilitando a monitorização dos dispositivos conectados à rede. Essa abordagem também é um dos principais objetivos do projeto proposto neste Trabalho Final de Curso (TFC). Outras características compartilhadas pelos sistemas analisados e pelo projeto incluem a ativação de alarmes, visualização de dados em tempo real e integração de dispositivos por meio de uma rede sem fio. É importante ressaltar que, embora os sistemas analisados tenham a capacidade de gerar relatórios a partir dos dados coletados, essa funcionalidade não estará presente inicialmente no projeto, mas poderá ser adicionada futuramente.

Portanto, a principal diferença entre os sistemas consolidados no mercado e o sistema a ser desenvolvido neste TFC reside no enfoque na usabilidade e facilidade de adoção. Ao contrário dos sistemas analisados, que requerem a aquisição de complementos ou pacotes extras para aproveitar todas as capacidades e funcionalidades, o sistema proposto tem como objetivo ser de fácil instalação e manutenção, além de ser facilmente expansível e adaptável às necessidades de cada usuário. A ênfase está em fornecer uma solução intuitiva, de utilização simplificada e acessível, que permita aos usuários controlar os aspetos do edifício e armazenar dados para análise futura. Essa abordagem busca proporcionar aos usuários uma experiência agradável e livre de obstáculos, eliminando a necessidade de investimentos significativos e complexos para obter um sistema eficiente de monitorização de edifícios.



## 4 Engenharia

### 4.1 Levantamento e Análise de Requisitos

Um importante passo para a criação de um software é o levantamento de seus principais requisitos. De maneira geral, um requisito é uma função ou propriedade que o sistema deve possuir para atingir um objetivo. Os requisitos são divididos em três categorias: funcionais, não funcionais e técnicos. Requisitos funcionais se referem ao que o sistema deve fazer para atender uma necessidade ou resolver um problema, enquanto os requisitos não-funcionais se referem à forma como o sistema irá realizar as ações. Por fim, os requisitos técnicos descrevem especificações do hardware e software que serão necessários para implementar o sistema.

Encontram-se, abaixo, os requisitos identificados para a realização deste projeto. Os requisitos estão classificados por grau de importância utilizando a escala “Have”, onde existe quatro graus de importância “Must Have”, “Should Have”, “Could Have” e “Will Not Have”. Os requisitos classificados como “Must Have” são obrigatórios para o funcionamento do sistema, “Should Have” são requisitos importantes, mas não obrigatórios, “Could Have” são requisitos que têm pouco impacto no funcionamento do sistema e “Will not Have” são requisitos de pouca importância e que podem não ser implementados.

#### 4.1.1 Requisitos Não-Funcionais

Requisito Não-Funcional	Descrição	Importância
RNF1	É necessário haver conexão com a internet.	Must Have
RNF2	O sistema deve ser capaz de armazenar dados do edifício	Must Have
RNF3	O sistema deve permitir o acionamento de alarmes para os diversos sensores	Must Have
RNF4	O sistema deve permitir que seja estabelecido um período para a coleta de novos dados do edifício	Must Have
RNF5	O sistema deve permitir a visualização em tempo real dos dados do edifício	Must Have

<b>RNF6</b>	O sistema deve permitir a visualização dos dados armazenados através de gráficos	Must Have
-------------	--	-----------

Tabela 1 - Requisitos Não Funcionais

#### 4.1.2 Requisitos Funcionais

Requisito Funcional	Descrição	Importância
<b>RF1</b>	Os dados coletados pelos sensores devem ser armazenados em uma base de dados	Must Have
<b>RF2</b>	O microcontrolador deve realizar a recolha de dados dos sensores de forma manual ou automática	Must Have
<b>RF3</b>	O sistema deve permitir que o utilizador veja os dados armazenados na base de dados através de uma tabela	Must Have
<b>RF4</b>	O Sistema deve permitir que o utilizador acione alarmes para temperatura, luminosidade e abertura de portas	Must Have
<b>RF5</b>	O sistema deve ativar o servo contínuo (simulação de estores) quando a intensidade de luz for muito elevada ou pode ser ativado manualmente pelo utilizador	Must Have
<b>RF6</b>	Os dados da base de dados devem ser visualizados através da tabela e dos gráficos	Must Have
<b>RF7</b>	Os gráficos podem ser filtrados por data	Must Have
<b>RF8</b>	A tabela de dados pode ser filtrada por data, temperatura e humidade	Must Have

Tabela 2 - Requisitos Funcionais

### 4.1.3 Requisitos Técnicos

Requisito Técnico	Descrição	Importância
<b>RT1</b>	Os sensores serão conectados a um microcontrolador Arduino UNO	Must Have
<b>RT2</b>	O microcontrolador deve realizar a recolha de dados dos sensores	Must Have
<b>RT3</b>	A plataforma de visualização irá correr no desktop	Must Have
<b>RT4</b>	O sistema deve aceder e recolher dados de uma base de dados MySQL	Must Have

Tabela 3 - Requisitos Técnicos

## 4.2 Casos de Uso

Os casos de uso descrevem visualmente alguns dos principais requisitos funcionais descritos acima.

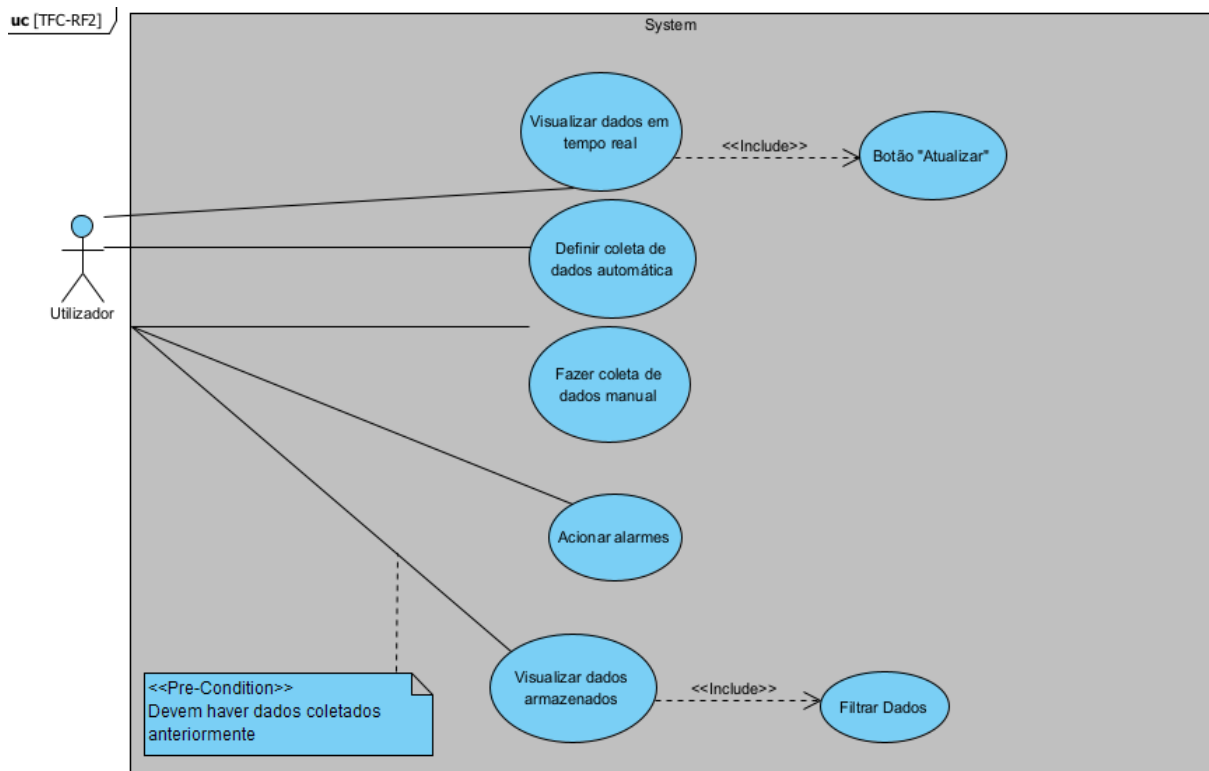


Figura 5 - Use cases

## 4.3 Mockups

Nesta secção, foram criados mockups para algumas páginas da plataforma de visualização. Nestes mockups, estão representadas as principais funcionalidades da plataforma e os principais ecrãs.

### 4.3.1 Página Principal

Nesta página, devem ser disponibilizados os seguintes dados do edifício:

- Temperatura atual;
- Humidade relativa;
- Luminosidade;
- Estado da conexão com microcontrolador.

Além dos dados, este ecrã também deve possuir um botão para acionar os estores, caso a luminosidade esteja muito alta, um botão para atualizar os dados que estão sendo exibidos e haverá uma barra de navegação na parte inferior do ecrã que dará acesso às outras páginas da plataforma de visualização.

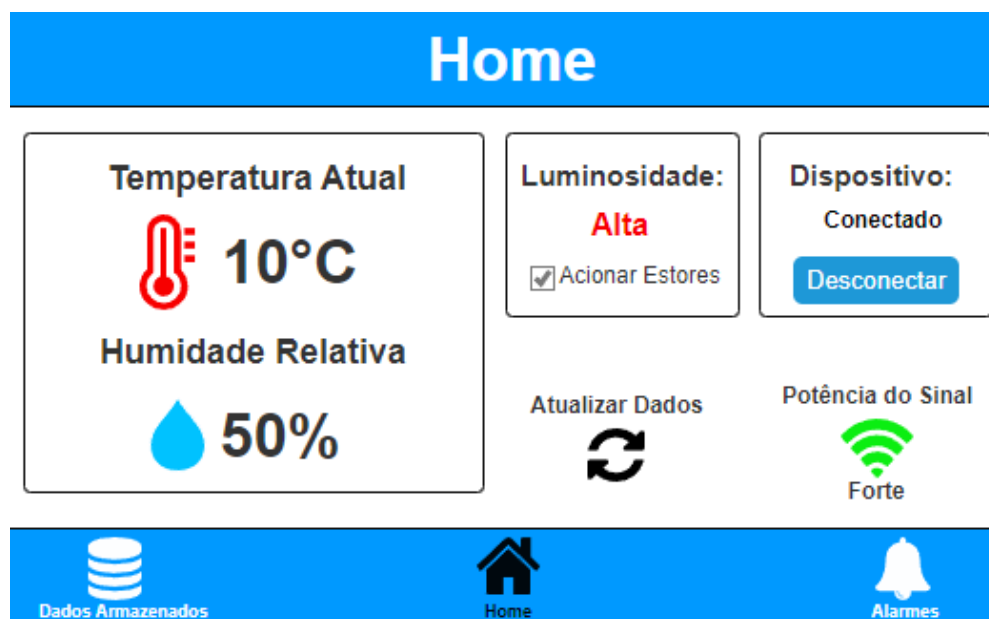



Figura 6 - Mockup Página Principal

### 4.3.2 Página de Alarmes

A página de alarmes deve fornecer ao utilizador uma forma de controlar quais os sensores estão com alarmes ativados ou desativados. Os alarmes disponíveis são:

- Ativação do Buzzer quando for detetado temperatura abaixo ou acima do limite;
- Ativação do Buzzer, piscar da luz no LED RGB e aviso na plataforma de visualização quando for detetada a abertura de uma porta;
- Ativação do Buzzer, piscar da luz vermelha no LED RGB e aviso na plataforma quando for detetada luminosidade acima do limite.



Alarmes

Temperatura acima do limite ☒ Desligado ☐ Ligado

Abertura de Porta ☒ Desligado ☐ Ligado

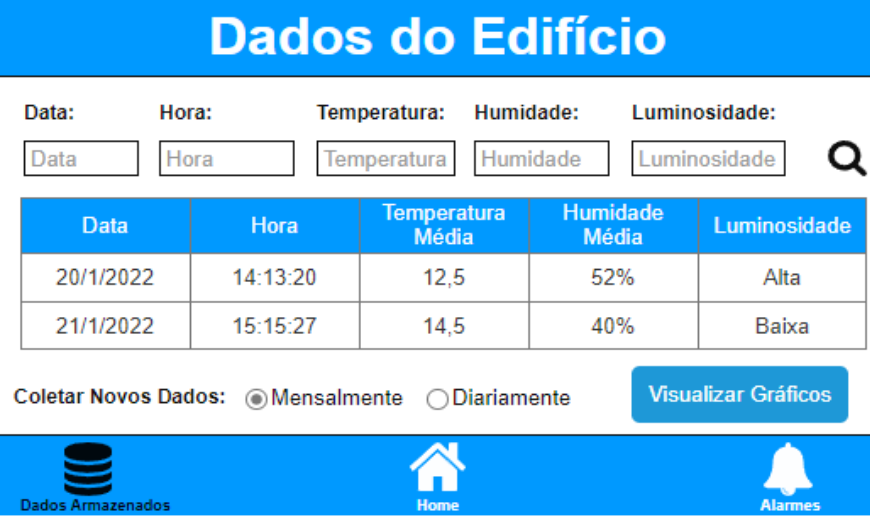
Luminosidade acima do limite ☒ Desligado ☐ Ligado

Dados Armazenados Home Alarmes

Figura 7 - Mockup Página de Alarmes


### 4.3.3 Página de Dados Armazenados

Nesta página, o utilizador poderá visualizar os dados do edifício que estão armazenados na base de dados. Os dados poderão ser filtrados por data, temperatura média e humidade relativa. Haverá também uma maneira de programar uma coleta automática dos dados periodicamente ou manualmente. Os dados são coletados, armazenados no banco de dados e exibidos nesta página. Os dados também poderão ser visualizados na forma de gráficos, para melhor visualização e interpretação dos mesmos.



Dados do Edifício

Data: Hora: Temperatura: Humidade: Luminosidade:



Data	Hora	Temperatura Média	Humidade Média	Luminosidade
20/1/2022	14:13:20	12,5	52%	Alta
21/1/2022	15:15:27	14,5	40%	Baixa

Coletar Novos Dados: ☒ Mensalmente ☐ Diariamente

Dados Armazenados Home Alarmes

Figura 8 - Mockup da Página de Dados Armazenados

## 4.4 Base de Dados

A base de dados será implementada utilizando MySQL. A princípio, haverá duas tabelas principais, sendo elas: Dados e Utilizadores. A tabela “Dados” será utilizada para armazenar os dados coletados pelo Arduino, como temperatura, humidade, luminosidade e a data e hora em que a coleta foi realizada. Os dados desta tabela serão exibidos para o utilizador na página de Dados Armazenados na Plataforma de Visualização. A tabela “Utilizadores” terá dados relacionados aos alarmes e à coleta automática de dados, caso o utilizador ative um dos alarmes ou a coleta automática isto ficará registado na tabela para que na próxima vez que entrar na plataforma o alarme continue acionado. Abaixo está o diagrama de entidade-relação da base de dados.

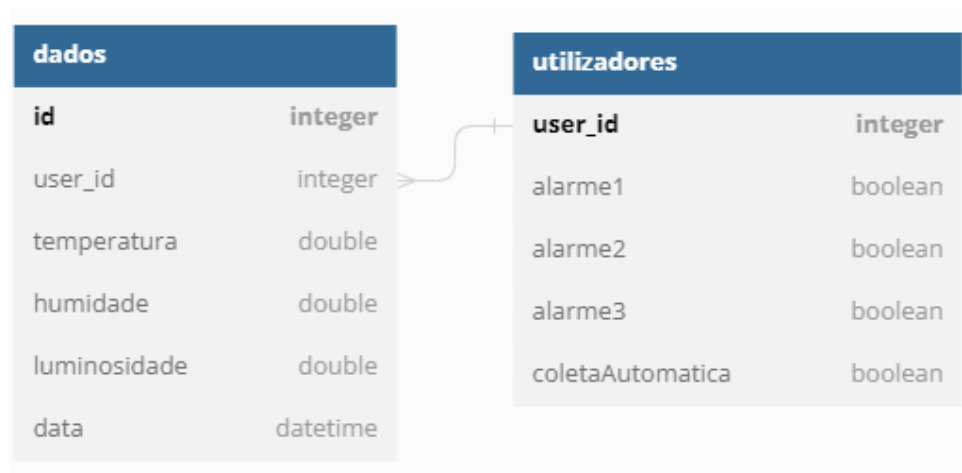


Figura 9 - Diagrama Entidade-Relação da Base de Dados

## 4.5 Diagrama de Atividades

Diagramas que descrevem o fluxo das principais atividades do sistema. Nestes diagramas temos o utilizador, o sistema e o microcontrolador Arduino como atores.

- Visualizar dados em tempo real:

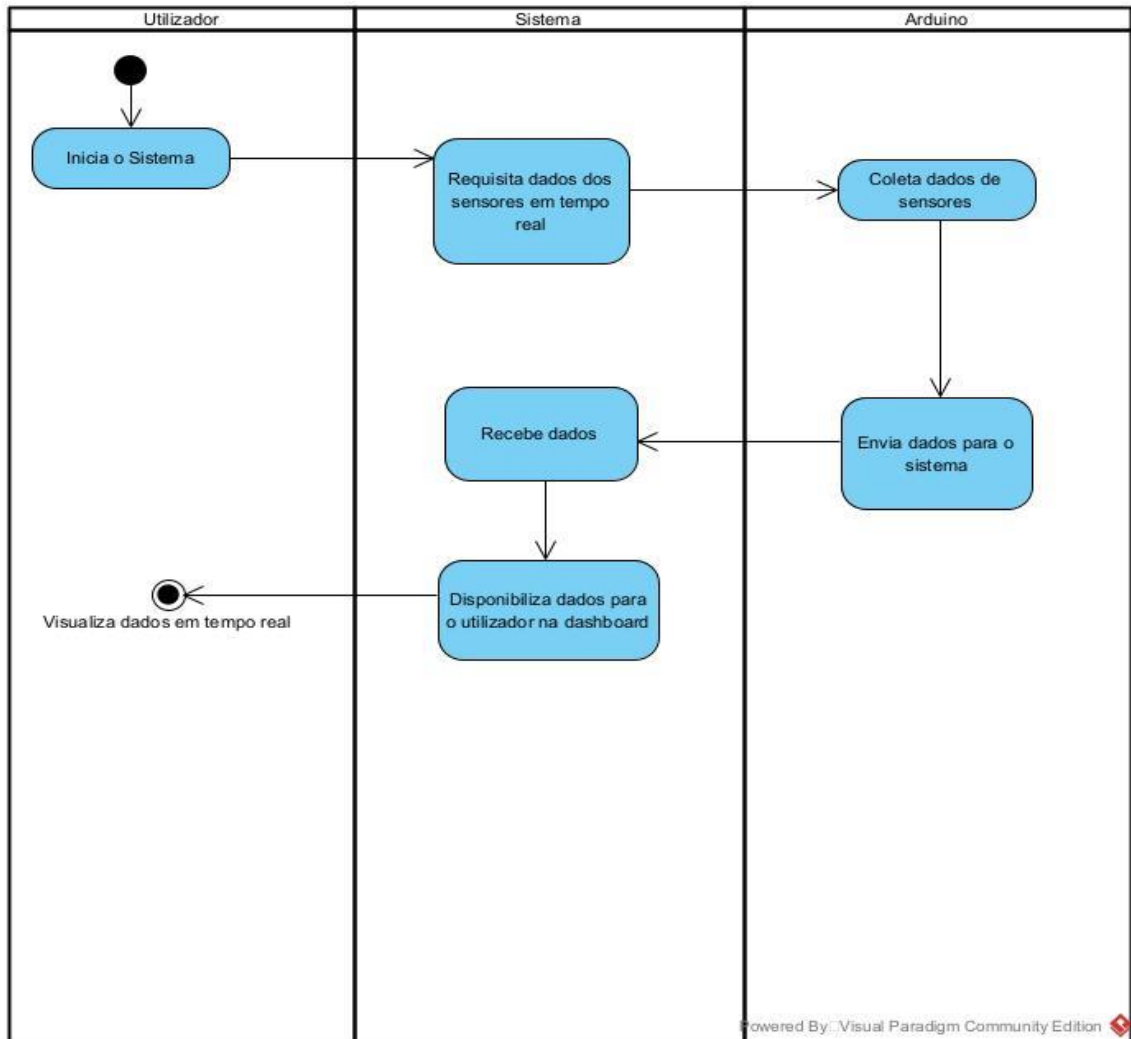


Figura 10 - Diagrama de Atividades 1

Esta primeira atividade descreve o funcionamento da visualização de dados em tempo real. O Arduino constantemente irá coletar os dados dos sensores e enviar estes dados ao sistema para que esse possa exibi-los corretamente na interface do utilizador.

## - Armazenar dados na base de dados:

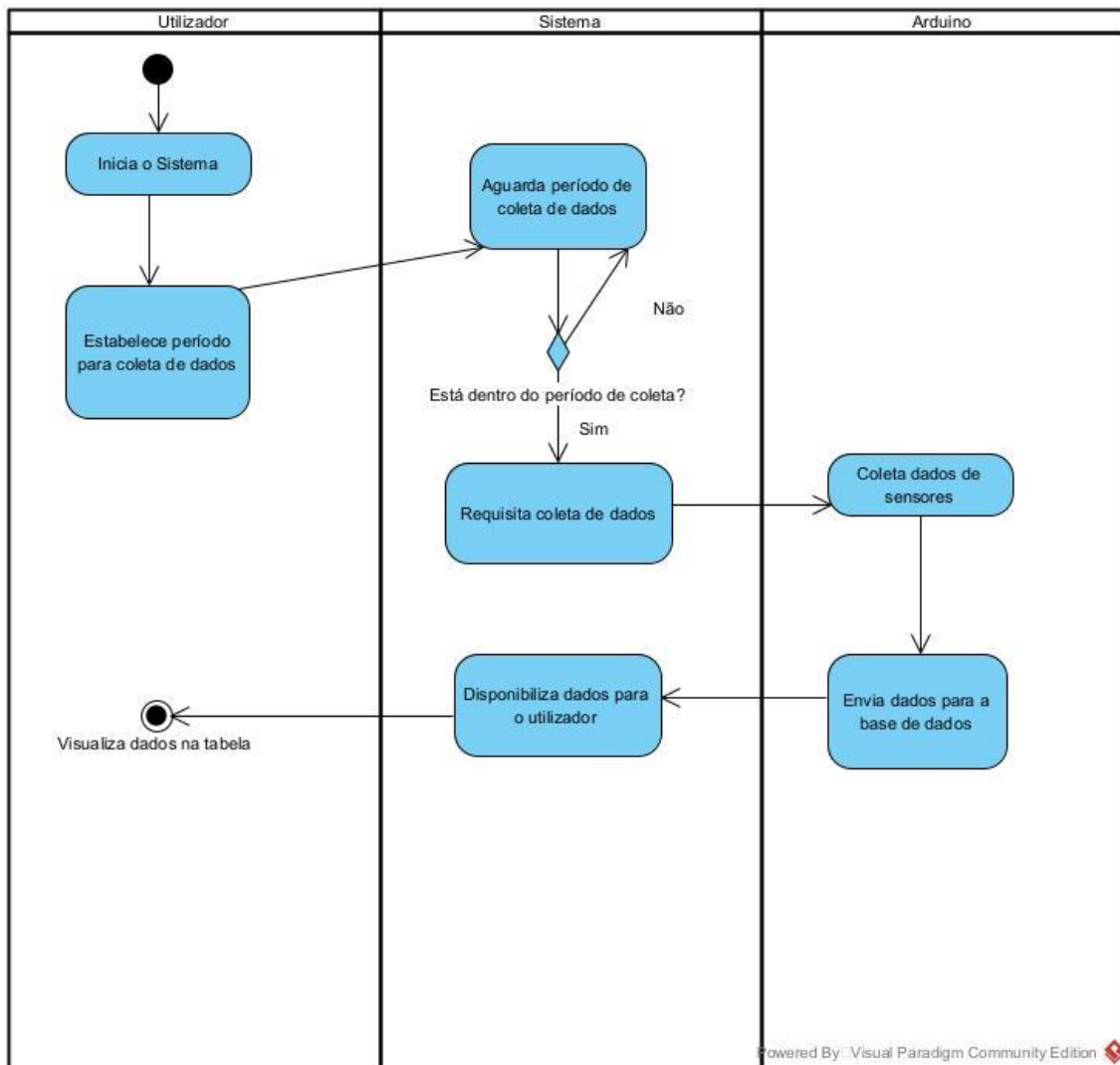
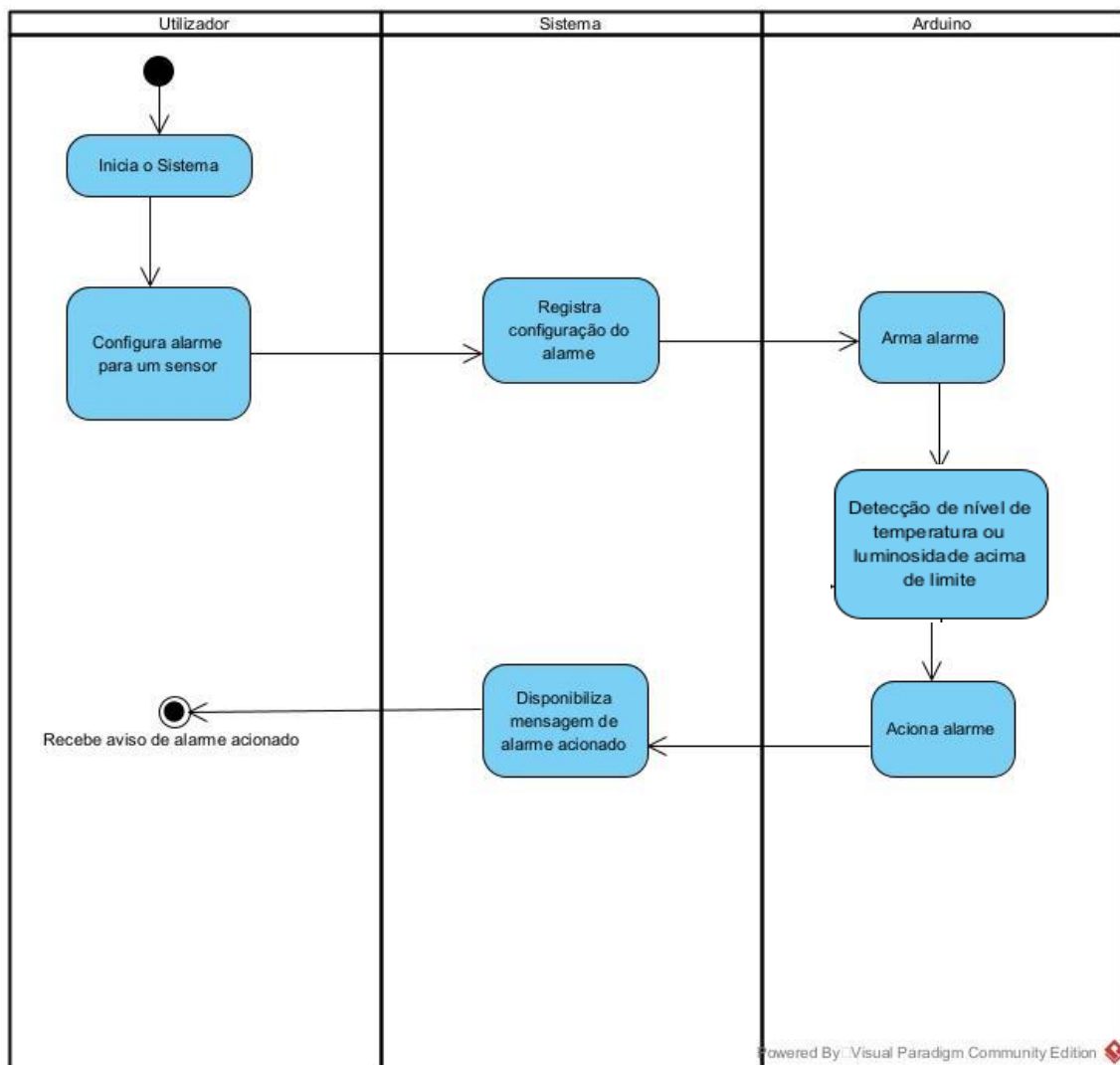


Figura 11 - Diagrama de Atividades 2

Este diagrama descreve como será o funcionamento da coleta e armazenamento de dados na base de dados. O utilizador pode estabelecer um período para a coleta automática dos dados. Após o estabelecimento deste, o sistema verifica se está dentro do período de coleta dos dados. Caso esteja dentro do período, o sistema recebe os dados mais recentes dos sensores e os envia para a base de dados. O utilizador também poderá armazenar os dados instantaneamente, sem o estabelecimento do período de coleta.



**- Configurar alarmes:****Figura 12 - Diagrama de Atividades 3**

O diagrama descreve o funcionamento do sistema de Alarmes do projeto. Através da página de alarmes, o utilizador pode ativar ou desativar os mesmos. Quando um alarme é ativado, é enviado um sinal ao Arduino, que agora verifica se, por exemplo, os níveis de temperatura estão acima do limite. Caso o limite seja ultrapassado, o Arduino aciona o buzzer e o LED e envia um sinal à plataforma para que seja disponibilizada uma mensagem ao utilizador na interface.

## 5 Solução Desenvolvida

### 5.1 Visão Geral

A solução desenvolvida para o problema foi o desenvolvimento de um sistema inteligente que realiza a monitorização de diversos dispositivos integrados em um edifício e exibe os dados e informações coletados em uma plataforma de visualização. Para a realização da monitorização, foi utilizado um microcontrolador com múltiplos sensores.

O sistema é capaz de realizar as seguintes ações:

- Medição de temperatura e umidade relativa através de um sensor;
- Medição de luminosidade através de um sensor LDR (Light Dependent Resistor);
- Modo de acionamento de diversos alarmes;
- Ativação de um servo contínuo (utilizado para simular estores) quando a intensidade de luz é muito alta.

Todos os dados e medições são armazenados em uma base de dados e podem ser visualizados através da plataforma de visualização. Essa plataforma serve tanto para a visualização dos dados adquiridos pelos sensores quanto para a ativação de alarmes, definição de períodos de coleta de dados e ativação manual de alguns sensores. As medições podem ser feitas de maneira automática ao estabelecer um período para a aquisição dos dados ou podem ser feitas manualmente através da plataforma de visualização.

#### **GitHub do Projeto:**

<https://github.com/felipegbandeira/TFC-DEISI318-SistemaDeMonitorizacaoDeEdificios>

#### **Protótipo Axure do Projeto:**

<https://3ubu2t.axshare.com>

#### **Vídeo de Demonstração do Projeto:**

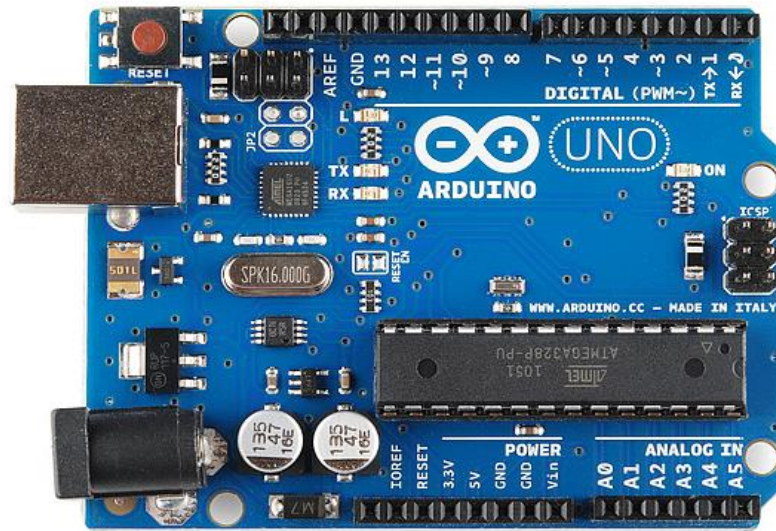
<https://youtu.be/AdfsF-fyks0>

### 5.2 Arduino e Sensores

O sistema funciona por meio de um microcontrolador Arduino em conjunto com o ambiente de desenvolvimento Arduino IDE. O Arduino é uma plataforma open source amplamente utilizada em projetos de eletrônica. Consiste em microcontroladores físicos, que são placas de circuito programáveis, e um software de desenvolvimento chamado Arduino IDE (Integrated Development Environment). O Arduino IDE permite escrever código que é enviado para os microcontroladores e utiliza uma versão simplificada da linguagem C (What Is an Arduino? - SparkFun Learn, n.d.).

O Arduino é uma plataforma altamente versátil, adequada para o desenvolvimento de uma ampla variedade de projetos de prototipagem eletrônica, desde os mais simples até os mais complexos, como robôs, máquinas e sistemas de automação. No caso deste

projeto, o Arduino é utilizado como base para a implementação do sistema de monitorização e controle dos dispositivos do edifício.



**Figura 13 - Microcontrolador Arduino**

Fonte: [www.techtudo.com.br](http://www.techtudo.com.br)

As funcionalidades do Arduino são ampliadas através do acréscimo de sensores e módulos ao microcontrolador. Para este projeto, os sensores utilizados são:

- Sensor de temperatura BMP180 (figura 14)
- Sensor de humidade relativa DHT22 (figura 16);
- Sensor LDR para detectar luminosidade (figura 15);
- Sensor ultrasónico (figura 17).



**Figura 14 - Sensor de temperatura**

Fonte: <https://www.makerhero.com/blog/temperatura-pressao-bmp180-arduino/>



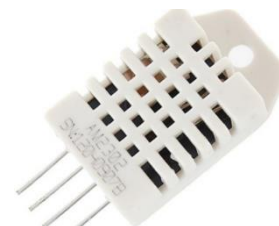
**Figura 15 - Sensor LDR**

Fonte: <https://www.ptrobotics.com>



**Figura 17 - Sensor Ultrasónico**

Fonte: <https://blogmasterwalkershop.com.br>



**Figura 16 - Sensor DHT22**

Fonte: <https://www.electrofun.pt/sensores-arduino/sensor-humidade->

Além dos sensores, serão utilizados também dispositivos de saída, para comunicar informações ao utilizador:

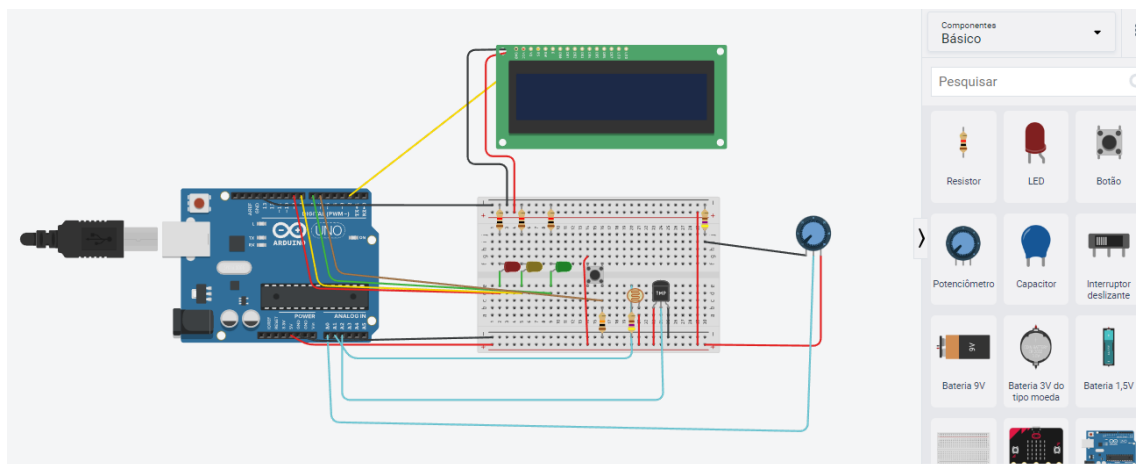
- Buzzer piezoelétrico para sinalização de alarme (*figura 18*)
- LED vermelho
- Servo contínuo para simular estores



**Figura 18 - Buzzer piezoelétrico**

Fonte: <https://www.casadarobotica.com/componentes-eletronicos/componentes/buzzer/2x-buzzer-passivo-piezoelétrico-de-alta-potencia-at3040>

Adicionalmente, foi utilizado o site Tinkercad para a criação e simulação de circuitos com um microcontrolador Arduino. Dessa forma, foi possível testar diversos circuitos diferentes, encontrando um que funcione corretamente e sem erros, antes de colocá-lo em prática.



**Figura 19 - Exemplo Simulação do Tinkercad**

Fonte: tinkercad.com

### **5.3 Plataforma de Visualização**

O sistema conta com uma plataforma de visualização, escrita em Java, onde podem ser realizadas diversas ações. Foi utilizada a biblioteca JavaFX para desenvolvimento das GUIs, por causa de sua fácil utilização. Nesta plataforma, o utilizador pode visualizar os microcontroladores que estão conectados e o seu atual estado, se estão ligados, se estão com algum problema, se há algum alarme acionado, entre outros detalhes do microcontrolador.

A principal função da plataforma de visualização é fornecer uma maneira para o utilizador visualizar os dados que são recolhidos pelos sensores. Será possível visualizar em tempo real dados como temperatura, humidade relativa e intensidade da luz. Os dados recolhidos pelos sensores são armazenados numa base de dados mySQL e podem ser visualizados pelo utilizador a qualquer momento. A recolha de dados poderá ser feita manualmente pelo utilizador, através de um botão, ou pode ser feita de maneira automática após o utilizador ativar a opção de coleta automática de dados. Os dados armazenados também podem ser visualizados em gráficos para melhor visualização da evolução dos mesmos.

As principais unidades curriculares que serviram de base para este projeto são:

- Linguagens de Programação I
- Linguagens de Programação II
- Bases de Dados
- Engenharia de Requisitos e Testes
- Interação Humano-Máquina

## 6 Método e Planeamento

O cronograma abaixo (figura 13) apresenta uma proposta de ordem de trabalho para este TFC em formato Gantt. Como descrito anteriormente, o projeto está dividido em duas partes principais, sendo elas o circuito feito com um microcontrolador Arduino e a plataforma de visualização. A primeira parte consiste em criar o circuito e o código do Arduino que servirão para coletar os dados dos sensores. A segunda parte será a criação da Plataforma de Visualização, que dará ao utilizador uma forma de visualizar os dados coletados. O cronograma contempla também as entregas a serem feitas ao longo do ano.

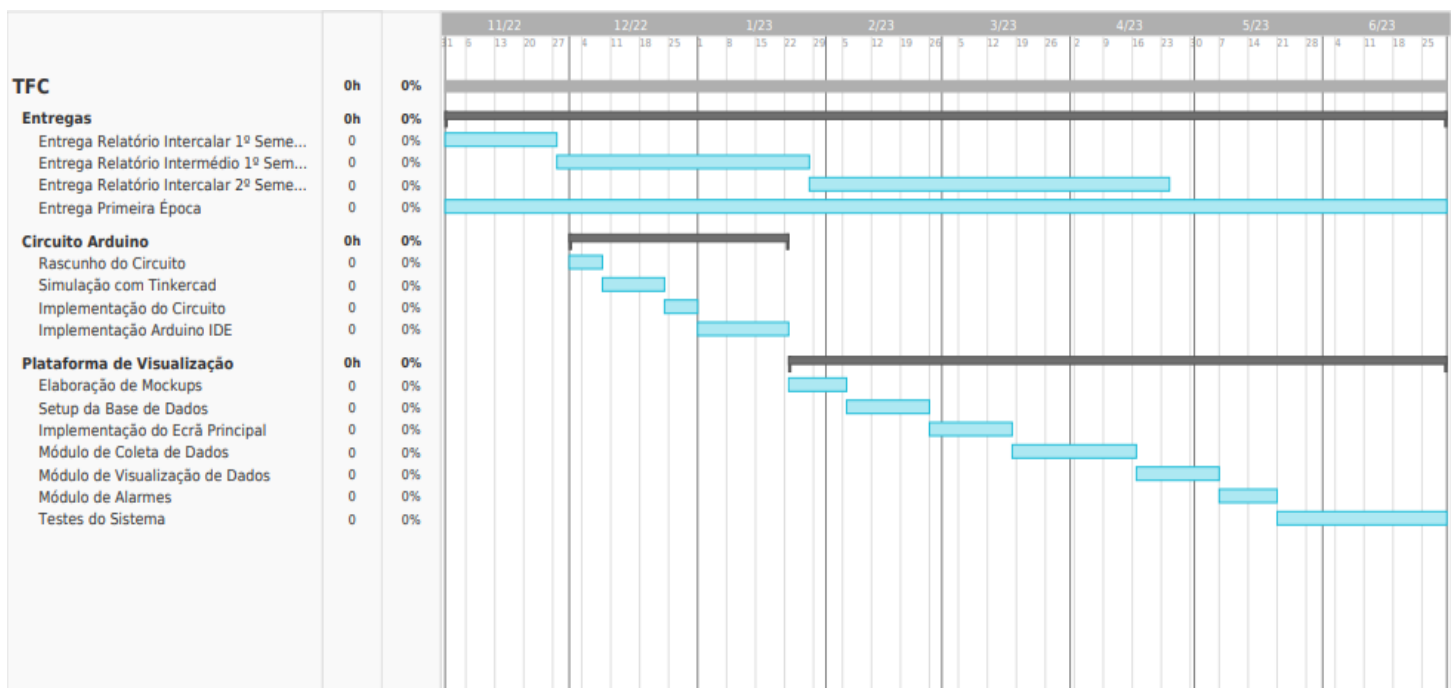


Figura 20 - Calendário em Formato Gantt

Para este Trabalho Final de Curso, foi escolhido utilizar o modelo de desenvolvimento incremental, dividindo o desenvolvimento do projeto em tarefas menores que compõem o trabalho final. As principais tarefas identificadas foram:

- Levantamento de Requisitos
- Estudo das ferramentas a serem utilizadas (Arduino IDE)
- Rascunho do circuito
- Simulação do circuito
- Implementação do código do Arduino IDE
- Setup da base de dados
- Implementação dos ecrãs principais da Plataforma de Visualização
- Implementação da funcionalidade de coleta de dados
- Implementação da funcionalidade de visualização em tempo real
- Implementação da funcionalidade de alarmes
- Realização de testes do sistema

- Features adicionais

Entregáveis:

- Relatório Intercalar (Primeiro Semestre)
- Relatório Intermédio (Primeiro Semestre)
- Relatório Intercalar (Segundo Semestre)
- Entrega Final (Segundo Semestre)

## 7 Resultados

### 7.1 Teste Cases

Os testes têm como objetivo garantir que tanto a plataforma de visualização quanto o Arduino estejam funcionando da maneira esperada. As principais funcionalidades do sistema serão testadas, como a visualização em tempo real, o armazenamento dos dados e o sistema de alarmes.

Teste	Requisito	Descrição	Resultado Esperado	Resultado Obtido
Armazenar dados na base de dados	<b>RF1</b>	Quando requisitado pelo utilizador, os dados mais recentes coletados pelo microcontrolador devem ser armazenados na tabela "Dados"	Os dados coletados + data e hora da coleta são armazenados na tabela correta da base de dados.	Dados armazenados corretamente na tabela 'dados'
Visualizar os dados em uma tabela	<b>RF3</b>	Quando o utilizador vai à página "Dados Armazenados", uma tabela com todos os dados armazenados deve ser disponibilizada	O utilizador consegue ver a tabela na página	Visualização correta da tabela
Alarme de abertura de portas	<b>RF4</b>	Quando o alarme de abertura de portas estiver acionado, o sistema deve detetar quando a porta for aberta	Ao detetar a abertura de uma porta, o sistema deve disponibilizar uma mensagem pop-up para o utilizador, e também devem ser acionados o buzzer e o LED vermelho ligados ao Arduino.	Alarme funciona como o esperado
Alarme de temperatura	<b>RF4</b>	Quando o alarme de temperatura estiver acionado, o sistema deve detetar quando os limites mínimos ou máximos são ultrapassados	Ao detetar que a temperatura está acima do limite máximo, ou abaixo do limite mínimo, o sistema deve disponibilizar uma mensagem pop-up para o utilizador, e	Alarme funciona como o esperado



			também devem ser acionados o buzzer e o LED vermelho ligados ao Arduino.	
Alarme de luminosidade	<b>RF4</b>	Quando o alarme de luminosidade estiver acionado, o sistema deve detetar quando o limite máximo é ultrapassado	Ao detetar que a luminosidade está acima do limite máximo, o sistema deve disponibilizar uma mensagem pop-up para o utilizador, e também devem ser acionados o buzzer e o LED vermelho ligados ao Arduino.	Alarme funciona como o esperado
Geração automática de gráficos	<b>RF6</b>	Quando requisitado pelo utilizador, os dados armazenados na base de dados devem ser disponibilizados em forma de gráficos	O sistema deve obter os dados armazenados na tabela "Dados" e gerar gráficos que possam ser visualizados pelo utilizador na página de Dados Armazenados	Gráficos disponibilizados corretamente
Dados visualizados em tempo real	<b>RNF5</b>	Os dados coletados pelo microcontrolador devem ser exibidos em tempo real na página principal do sistema	Ao abrir a plataforma, o sistema deve automaticamente conectar-se ao Arduino e exibir os dados que são coletados pelos sensores	Visualização em tempo real funciona como o esperado
Ativação automática do servo motor	<b>RF5</b>	O servo motor (simulação de estores) deve ser automaticamente acionado quando a luminosidade estiver alta	Ao detetar níveis de luminosidade alta, o sistema deve enviar um sinal ao microcontrolador para que o servo motor seja acionado	Há um pequeno delay na ativação do servo motor, mas está funcionando
Filtrar gráficos por data	<b>RF7</b>	O utilizador escolhe uma data específica e os gráficos devem mostrar os dados armazenados daquela data	A plataforma envia um query à base de dados com a data escolhida e os dados recebidos são disponibilizados	Gráficos disponibilizados corretamente

Filtrar tabela	<b>RF8</b>	A tabela pode ser filtrada utilizando os filtros disponíveis e que podem ser combinados de diversas formas	A plataforma envia uma query à base de dados e os dados recebidos são disponibilizados na tabela	Foram testadas diversas combinações de filtros e todos passaram no teste
----------------	------------	--	--	--

**Tabela 4 - Resultados dos testes**

Com base nos testes realizados, conclui-se que a plataforma desenvolvida neste projeto apresentou um desempenho satisfatório, atendendo a todos os requisitos estabelecidos. Os testes abrangeram as principais funcionalidades e cenários de uso, e em todas as instâncias, a plataforma manteve-se consistente, fornecendo os resultados desejados de forma correta. Todos os requisitos funcionais foram alcançados, conforme planeado, e a integração com o banco de dados, bem como o armazenamento dos dados, também foram confirmados nos testes realizados. É importante destacar que, além dos testes mencionados acima, foram realizados testes adicionais para garantir a qualidade e confiabilidade da plataforma. Com base na validação desses testes abrangentes, podemos afirmar que a plataforma atinge os objetivos propostos, oferecendo uma solução funcional e confiável para a monitorização de edifícios.

## 7.2 Testes com Utilizadores

Foram também realizados teste com utilizadores para garantir que a plataforma seja o mais fácil de se utilizar quanto possível. O sistema, apesar de possuir um uso muito específico, deve poder ser utilizado por qualquer pessoa, mesmo que esta não tenha qualquer conhecimento técnico. Para os testes, os utilizadores seguiram um Guião de Tarefas ([ver anexo](#)) e, após concluírem o guião, preencheram um questionário de satisfação com suas impressões sobre a plataforma ([ver anexo](#)). O guião de tarefas é dividido em 3 partes, que testam cada uma das páginas principais da plataforma.

Foram realizados 5 testes, sendo que 3 utilizadores eram do sexo masculino e 2 do sexo feminino. Os testadores tinham idades variadas, entre 20 e 55 anos, sendo a média de idade de 40 anos. Todos os testes foram realizados presencialmente. Após a realização dos testes, os dados obtidos pelo questionário de satisfação e pela grelha de observação mostram que, no geral, a aplicação causou boa impressão nos utilizadores e cumpre o que foi proposto.

A primeira parte do teste focava-se no ecrã principal e esta foi a parte em que os utilizadores tiveram menor dificuldade para realizar as ações descritas no guião de tarefas. Todos os utilizadores conseguiram realizar as tarefas em pouco tempo. Alguns comentários acerca deste ecrã diziam que estava bem organizado e fácil de entender, porém alguns componentes estavam um pouco desalinhados. A segunda parte do teste é focada na página de alarmes. Aqui, os utilizadores testaram dois dos três alarmes disponíveis e também os popups. Por ser uma página bastantes simples, os utilizadores não encontraram dificuldades em realizar as tarefas propostas, mas novamente foram apontados pequenos defeitos visuais na página.

Por fim, a terceira parte do teste focava-se na página de dados e nos gráficos. Aqui, os utilizadores tiveram um pouco mais de dificuldade de realizar as ações, porém isto já era esperado, tendo em vista que a página é um pouco mais densa e complexa do que as outras. Ao realizar a ação de carregar no botão “Atualizar Dados”, um utilizador sugeriu que houvesse algum tipo de confirmação de que a ação foi concluída com sucesso, podendo ser algo semelhante ao popups de alarme. Os utilizadores conseguiram filtrar a tabela, porém indicaram que o texto desta estava demasiado pequeno e difícil de ler. Por fim, conseguiram aceder à página de gráficos e aplicar os filtros conforme indicado. Abaixo encontra-se a média de cada item do questionário sobre usabilidade preenchido pelos utilizadores.



**Figura 21 - Resultados do questionário**

No geral, a plataforma foi bem recebida e cumpre o objetivo de ser utilizável por pessoas que não têm qualquer conhecimento técnico. A maior parte das críticas foi para alguns elementos visuais e para a página de Dados Armazenados. Os comentários feitos pelos utilizadores auxiliaram a melhorar alguns aspetos da plataforma, principalmente os aspetos visuais, e contribuíram para a melhora do projeto.

## **8 Conclusão e Trabalhos Futuros**

Após o desenvolvimento e implementação do sistema proposto, podemos concluir que todas as metas e requisitos foram alcançados com sucesso. O sistema de monitorização de edifícios apresenta um desempenho consistente e confiável, cumprindo todas as funcionalidades propostas e fornecendo uma plataforma de utilização simples e intuitiva. Ao longo do projeto, foram realizados testes para garantir a precisão e a eficiência das medições realizadas pelos sensores, bem como a correta ativação dos alarmes e controle dos dispositivos. Essa abordagem minuciosa garantiu que o sistema funcionasse corretamente em diferentes cenários e condições. A facilidade de uso foi uma prioridade durante todo o desenvolvimento do sistema. A interface gráfica amigável permite que qualquer pessoa, mesmo sem conhecimentos técnicos avançados, possa operar e interagir com o sistema de forma simples e direta, o que foi comprovado pelos resultados dos testes com utilizadores.

No entanto, é importante destacar que este projeto não se limita apenas à implementação atual. Há um amplo espaço para futuros trabalhos e melhorias. Uma das possibilidades é a adaptação do sistema para funcionar com conexões sem fio, possibilitando uma maior flexibilidade na instalação dos dispositivos em diferentes locais do edifício. Além disso, o sistema pode ser expandido com a inclusão de novos sensores e microcontroladores, permitindo a monitorização de uma variedade ainda maior de variáveis e áreas do edifício. Essa escalabilidade é um fator importante, pois permite que o sistema seja adaptado de acordo com as necessidades específicas de cada edifício e cenário. Outra perspetiva de desenvolvimento futuro é a expansão para edifícios maiores e mais complexos. Com ajustes e personalizações adequadas, o sistema pode ser aplicado em ambientes de maior escala, como edifícios empresariais e industriais, contribuindo para a eficiência operacional, segurança e economia de recursos.

Em resumo, o sistema proposto demonstrou ser uma solução viável e eficiente para a monitorização de edifícios. Sua implementação bem-sucedida, juntamente com as possibilidades de expansão e aprimoramento, posicionam o sistema como uma ferramenta promissora no campo da automação e gestão de edifícios. O resultado alcançado neste projeto estabelece uma base sólida para o contínuo desenvolvimento e aperfeiçoamento do sistema, oferecendo benefícios reais em termos de facilidade de uso, controle e otimização de recursos.

## Bibliografia

- [BASM22] *Building Automation System Market*. (2022, June 15). P&S Intelligence. <https://www.psmarketresearch.com/market-analysis/building-automation-system-market>, acessado em Nov/2022
- [SEB22] Schneider Electric Brasil. (n.d.). *EcoStruxure Building Operation: Energy Conservation at University of Nottingham | Schneider Electric*. <https://www.se.com/br/pt/work/products/product-launch/building-management-system/>, acessado em Nov/2022
- [CB22] *Connect Box - Your simple IoT solution to connect and monitor your building*. (n.d.). siemens.com Global Website. <https://new.siemens.com/global/en/products/buildings/automation/cloud-solutions/hardware/connect-box.html>, acessado em Nov/2022
- [AP22] *Automação predial*. (n.d.). Phoenix Contact. Retrieved November 16, 2022, from <https://www.phoenixcontact.com/pt-br/industrias/automacao-predial>, acessado em Nov/2022
- [DC21] DocuSign, C. (2021, April 15). *Automação de processos: como a tecnologia pode ajudar sua empresa*. DocuSign. <https://www.docusign.com.br/blog/automacao-de-processos-como-tecnologia-pode-ajudar-sua-empresa>, acessado em Nov/2022
- [AEAP15] *A (re) evolução da automação predial*. (2015, October 19). SMART-Automação Residencial, Home-Theater & Som Ambiente. <http://www.smartautomacao.com.br/home/a-re-evolucao-da-automacao-predial/>, acessado em Nov/2022
- [WA22] *What is an Arduino? - SparkFun Learn*. (n.d.). <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>, acessado em Nov/2022
- [TD20] Tutida, D. (2021, September 20). *Digitalização de processos: o que é? Como implementar?* <https://encontreumnerd.com.br/blog/digitalizacao-processos-automatizacao>, acessado em Nov/2022
- [MH22] Medeiros, H. (n.d.). *Introdução a Requisitos de Software*. DevMedia. <https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>, acessado em Dez/2022

## Anexos

### Manual da Aplicação

- A plataforma só pode ser utilizada em conjunto com o microcontrolador Arduino com os seguintes sensores e dispositivos de saída:

- Sensor de temperatura BMP180;
- Sensor de humidade relativa DHT22;
- Sensor LDR para detectar luminosidade;
- Sensor ultrasónico;
- Buzzer piezoelétrico;
- LED vermelho;
- Servo contínuo para simular estores.

- Ao iniciar a aplicação o utilizador verá a dashboard principal.

#### 1 - Página principal:



Figura 22 - Manual: Home Screen

- Ao aceder à página principal, se o Arduino estiver conectado, o utilizador verá os dados que são capturados pelos sensores em tempo real. Os dados são: temperatura atual, humidade relativa, luminosidade (pode ser “Baixa”, “Média” ou “Alta”) e se o Arduino está conectado ou desconectado.

- A secção de Luminosidade possui ainda duas botões: “Abrir Estores” e “Fechar Estores”.



- Ambos os botões realizam a mesma ação, que é ativar o servo motor que está conectado ao Arduino e é utilizado para simular o acionamento de estores

Figura 23 - Manual: botões de controle de estores

- A secção “Dispositivo” possui dois botões para controle da conexão com o microcontrolador. Caso esteja desconectado, o botão “Conectar” irá tentar estabelecer uma conexão com o microcontrolador. Se estiver conectado, o botão “Desconectar” irá encerrar a conexão, podendo ser reestabelecida novamente a qualquer momento.



Figura 24 - Manual: dispositivo conectado



Figura 25 - Manual: dispositivo desconectado

- **Navegação:** a navegação para outras páginas se dá através da bottom bar localizada na parte inferior do ecrã. O botão “Home” leva o utilizador à página inicial, o botão “Dados Armazenados” leva à página de “Dados Armazenados” e o botão “Alarmes” leva à página de alarmes.



Figura 26 - Manual: bottom bar

## 2 - Página de Alarmes:

- Ao aceder à página de alarmes, através da bottom bar, o utilizador verá três opções de alarmes para acionar ou desligar.



Figura 27 - Manual: alarmes

- **Alarme de temperatura:** Ao ativar o alarme de temperatura, o utilizador tem a opção de escolher um valor mínimo e máximo de temperatura. Se a temperatura atual estiver abaixo do valor mínimo escolhido, ou acima do valor máximo, o alarme será acionado. O utilizador deve inserir

os valores e carregar em “Confirmar”, podendo inserir ambos os valores ou apenas um dos dois. Se o utilizador não escolher os valores mínimo e máximo, serão utilizados os valores padrões, sendo temperatura máxima = 30 e temperatura mínima = 15.

Figura 28 - Manual: alarme de temperatura

- **Alarme de luminosidade:** ao ativar este alarme, o microcontrolador verifica se a luminosidade atual está acima da média. Em caso positivo, o alarme será acionado.
- **Alarme de abertura de portas:** ao ativar este alarme, o microcontrolador verifica se a porta está aberta ou fechada. Caso esteja aberta, será acionado o alarme. Esta verificação se dá através do sensor ultrassónico para calcular a distância entre o microcontrolador e a porta.
- Em todos os casos, o comportamento do alarme é o mesmo: ao ser acionado, será ativado o buzzer, o LED vermelho e haverá uma mensagem popup disponibilizada para o utilizador na plataforma.
- **Mensagem popup:** se um dos alarmes forem acionados, será disponibilizada uma mensagem popup ao utilizador, nesta mensagem há dois botões, um para apenas fechar o popup, e outro para desligar o alarme que foi acionado e fechar o popup.

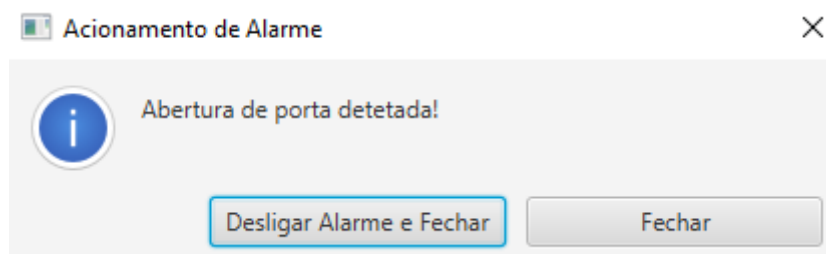


Figura 29 - Manual: mensagem popup

### 3 - Página de Dados Armazenados:

- Nesta página o utilizador pode visualizar uma tabela com os dados que foram coletados e enviados à base de dados.



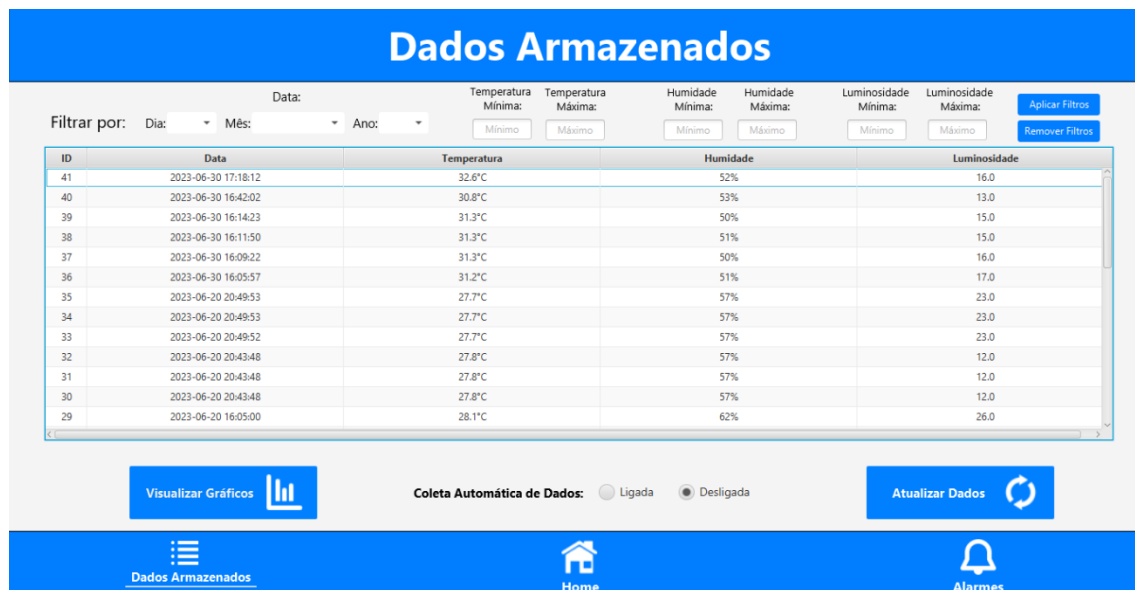


Figura 30 - Manual: página de Dados Armazenados

- **Filtros:** existem diversas opções de filtros que podem ser aplicados à tabela. Pode-se escolher uma data, selecionando o dia, mês e ano do respetivo *choicebox*. Existem também filtros para os valores de temperatura e humidade: pode-se escolher uma temperatura mínima ou máxima e humidade mínima ou máxima. Após inserir os valores nos respetivos locais de input, o botão “Aplicar Filtros” é utilizado para aplicar estes filtros à tabela e disponibilizar os dados corretamente de acordo com o que o utilizador requisitou. O botão “Remover Filtros” remove todos os filtros inseridos pelo utilizador e volta a mostrar todos os valores que estão armazenados na base de dados. Ao carregar nos títulos das colunas (Ex: “Data”, “Temperatura”) os dados serão ordenados de forma ascendente ou descendente.
- **Coleta automática de dados:** esta opção define se os dados coletados pelo microcontrolador serão enviados automaticamente à base de dados ou não. Caso esta opção esteja ligada, são definidos automaticamente 3 períodos de coleta dos dados: um às 10h, outro às 14h e o último às 18h. Após a última coleta, serão definidos novamente estes três períodos até que o utilizador desligue esta opção.
- **Botão “Atualizar Dados”:** realiza manualmente o armazenamento dos dados mais recentes coletados pelos sensores conectados ao microcontrolador. Os dados são coletados e enviados à base de dados.
- **Botão “Visualizar Gráficos”:** este botão leva o utilizador à página de gráficos, onde é possível visualizar os dados da tabela em gráficos de linha.

## 4 - Página de Gráficos:

- Nesta página é possível visualizar os dados de temperatura e humidade através de gráficos de linha.

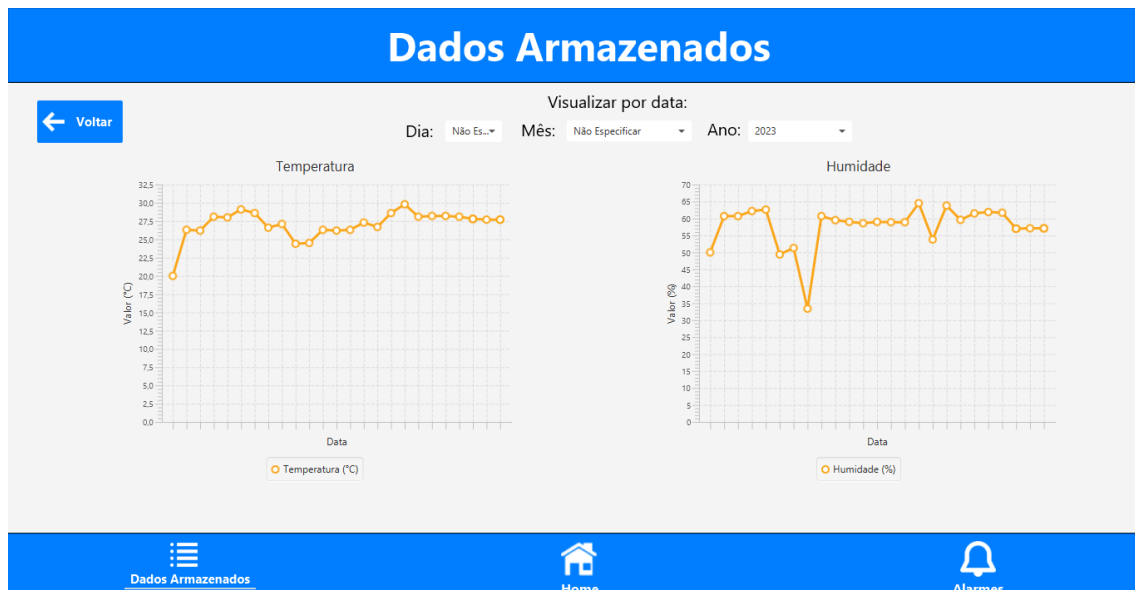


Figura 31 - Manual: gráficos

- **Filtros aplicados aos gráficos:** o utilizador pode aplicar filtros aos gráficos para visualizar os dados coletados de um determinado dia ou de um determinado mês. Ao escolher a opção “Não Especificar” na *choicebox* “Dia” e selecionar um mês específico serão mostrados todos os dados coletados daquele mês. Ao escolher a opção “Não Especificar” em ambos “Dia” e “Mês” serão mostrados todos os dados armazenados na base de dados sem filtros.

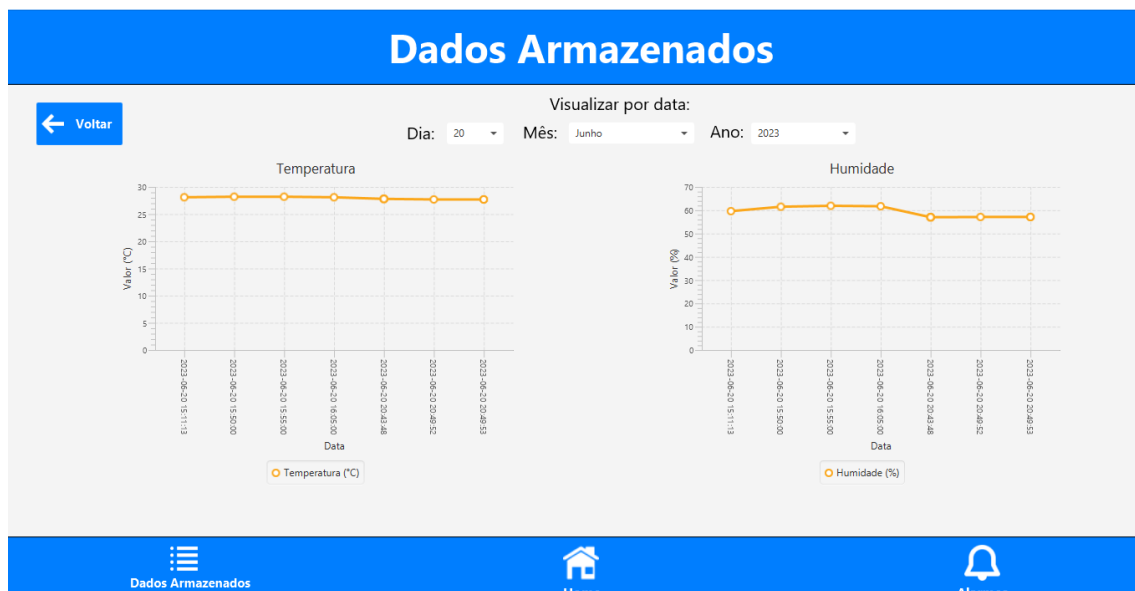
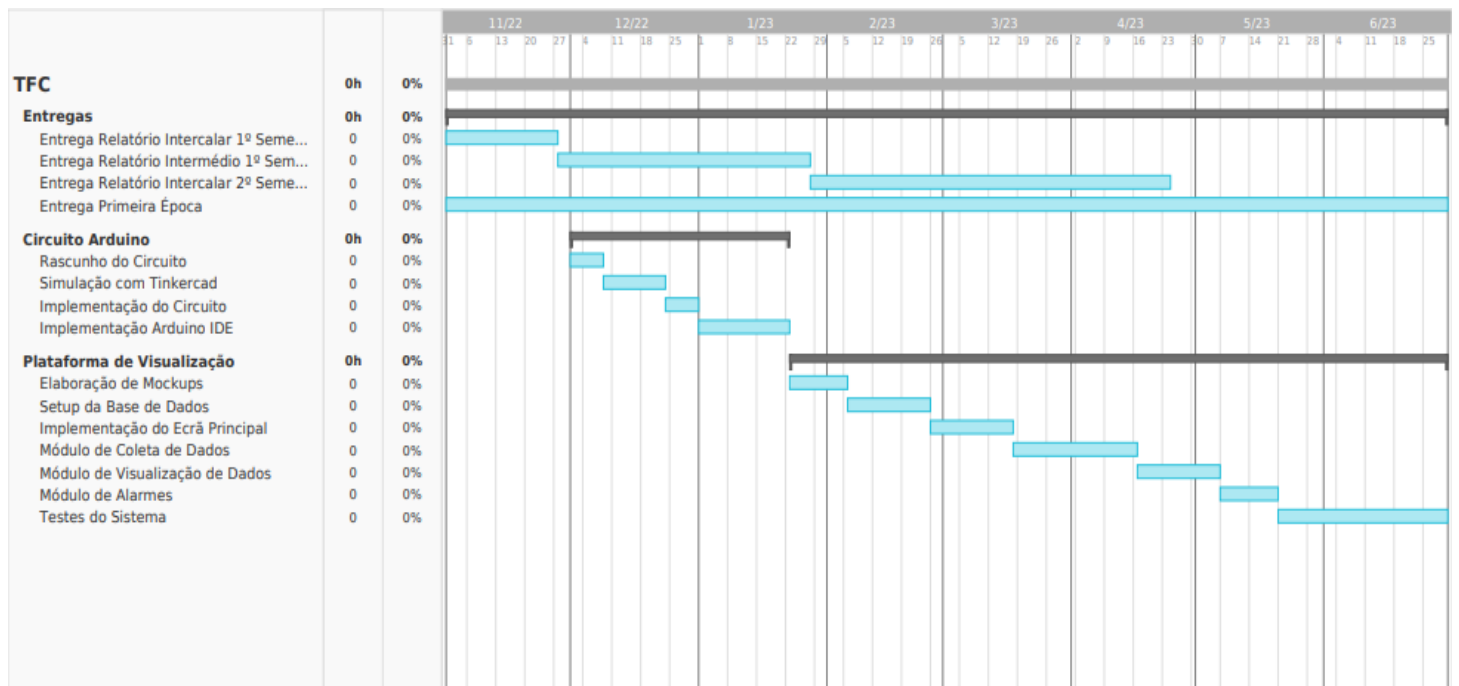


Figura 32 - Manual: gráficos com filtros aplicados

## Progresso de Trabalho – Janeiro/Fevereiro



### GitHub do Projeto:

<https://github.com/felipegbandeira/TFC-DEIS1318-SistemaDeMonitorizacaoDeEdificios>

### - Circuito Arduino

Sendo uma das principais partes do projeto, a implementação do circuito do Arduino foi dividida em duas partes: simulação e implementação do circuito. A simulação foi feita utilizando o site Tinkercad, onde é possível simular o circuito e o código do Arduino.

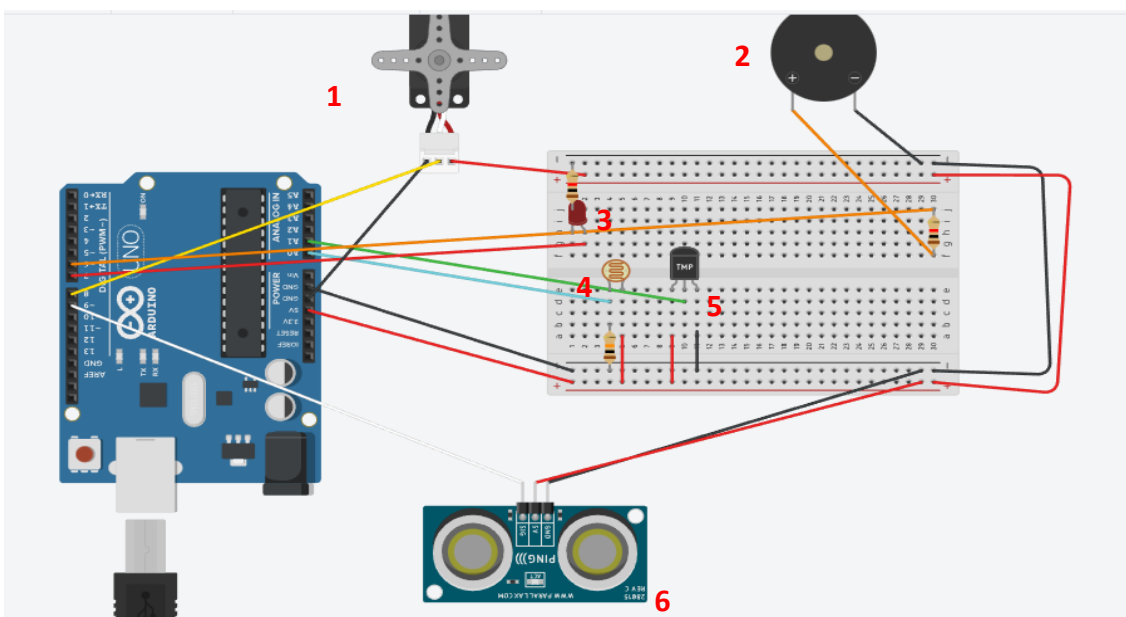


Figura 33 - Simulador do Tinkercad

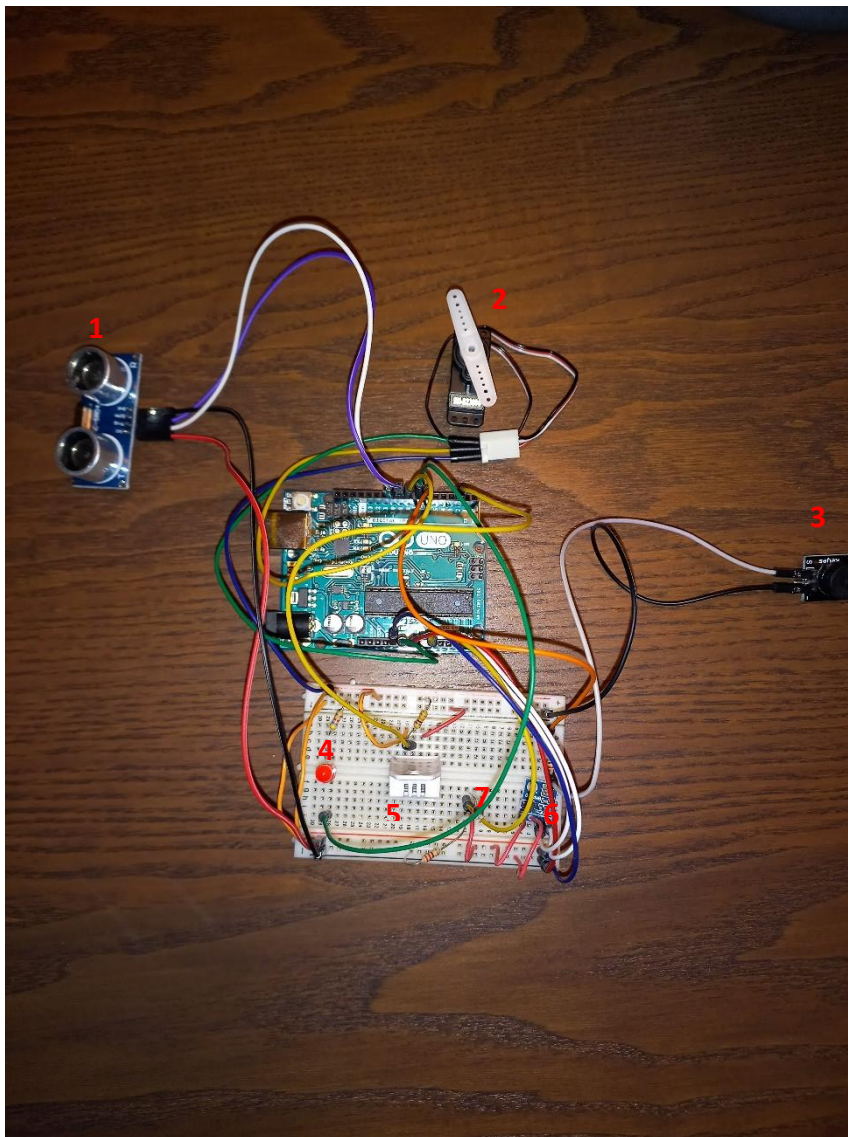
#### Legenda:

1. Micro-servo motor
2. Buzzer
3. LED
4. Fotorresistor
5. Sensor de temperatura
6. Sensor ultrassônico

O atual estado da simulação encontra-se na imagem acima. Alguns sensores e componentes não estão disponíveis para simular neste site. Os componentes simulados foram:

1. Micro-servo motor
2. Buzzer
3. LED vermelho
4. Fotorresistor
5. Sensor de temperatura
6. Sensor ultrassônico
7. Resistores

Não foi possível simular o sensor de humidade e o sensor de temperatura utilizado no site é diferente do que foi implementado no circuito real. Apesar disso, a simulação serviu como base e como ferramenta de prática e estudo para a posterior implementação do circuito real. Atualmente, o circuito encontra-se desta maneira:

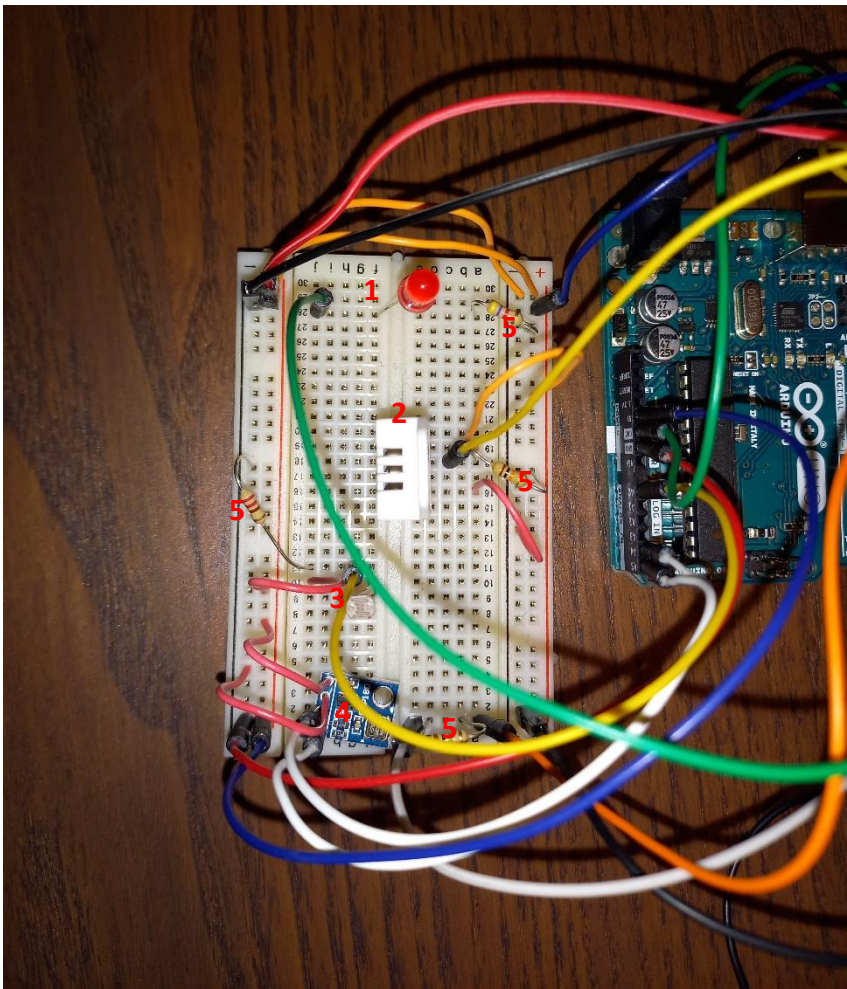


**Figura 34 - Foto circuito**

**Legenda:**

1. Sensor Ultrassônico
2. Micro-servo motor
3. Buzzer
4. LED Vermelho
5. Sensor de Humidade
6. Sensor de temperatura
7. Fotorresistor

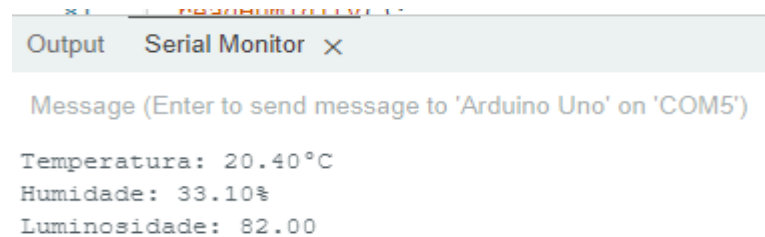


**Legenda:**

- 1.** LED vermelho
- 2.** Sensor de Humidade
- 3.** Fotoresistor
- 4.** Sensor de temperatura
- 5.** Resistores

**Figura 35 - Foto circuito (2)**

Foi utilizado o ambiente de desenvolvimento do Arduino para escrever o código necessário para realizar a leitura dos sensores e a ativação de outros componentes, como o micro-servo e o buzzer (o código pode ser encontrado no [github](#) do projeto, no ficheiro “Arduino”). Os dados coletados são enviados para a serial port, e podem ser visualizados no ambiente de desenvolvimento.

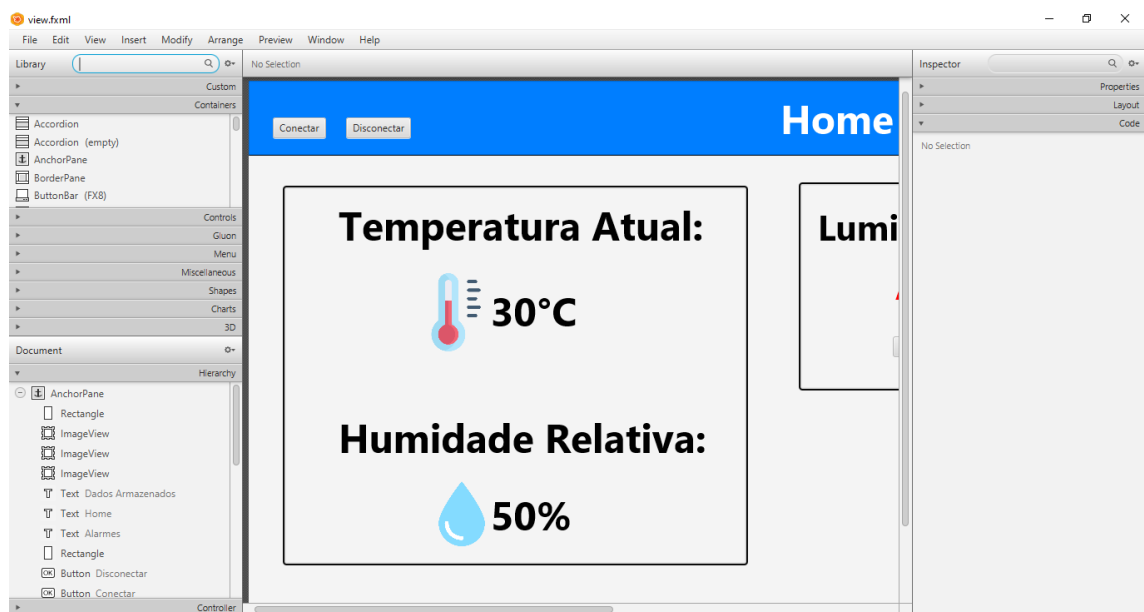
**Figura 36 - Exemplo de output dos sensores**

**Observações:**

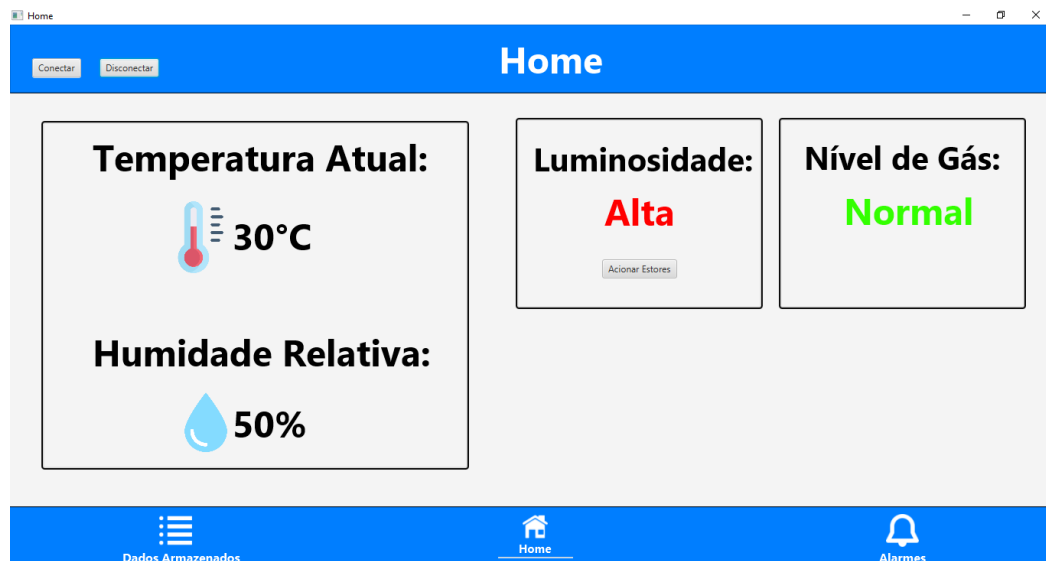
- Algumas funções, como acionar o buzzer, só funcionam quando o Arduino recebe um valor no serial port. Por exemplo, caso seja enviado “1” ao Arduino, será acionado o servo motor.
- As comunicações com o Arduino, neste momento, são feitas através do cabo USB e a serial port, posteriormente isso será alterado para funcionar com um módulo de WI-fi para comunicações sem fio

## - Plataforma de Visualização

A plataforma de visualização está sendo escrita em Java, utilizando a framework JavaFX e a ferramenta SceneBuilder (ver figura abaixo) para auxiliar na construção das GUIs. A comunicação com o Arduino é feita através da biblioteca JSerialComm, que simplifica e facilita a conexão e comunicação com o microcontrolador. A plataforma terá, inicialmente, 3 ecrãs principais, sendo eles: a página principal, página de alarmes e página de dados armazenados. Os ecrãs encontram-se em um estado inicial de desenvolvimento, ainda precisando implementar algumas funções importantes.



**Figura 37 - Ferramenta SceneBuilder para construção das interfaces com JavaFX**

**- Página principal:****Figura 38 - Estado atual da página principal**

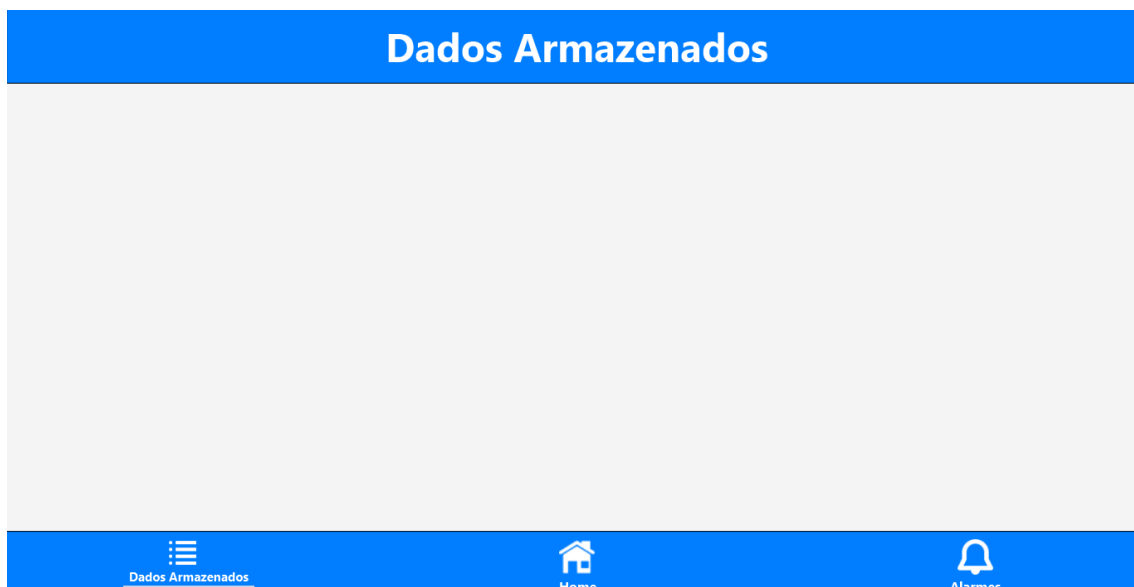
Atualmente, a página principal possui os principais elementos visuais que foram propostos nos Mockups criados anteriormente, porém ainda não é possível visualizar os dados reais coletados pelo Arduino. No canto superior esquerdo, o botão “Conectar” é utilizado para se conectar com o Arduino através da abertura da porta serial do mesmo, o botão “Desconectar” fecha a comunicação com o microcontrolador. Nesta página há também o botão “Acionar Estores”, que envia um sinal ao Arduino para que o servo-motor seja acionado. A barra de navegação também se encontra funcional.

**Observações:**

- Os botões de conectar e desconectar ainda estão sujeitos à mudança, pois o ideal seria a plataforma se conectar automaticamente ao Arduino ao ser iniciada;
- Já é possível ler a porta serial e enviar dados ao Arduino, porém ainda não consigo tratar destes dados, apenas consigo disponibilizá-los no console do ambiente de desenvolvimento. A prioridade neste momento é encontrar uma maneira de tratar estes dados e disponibilizá-los em tempo real na página inicial.

**- Página alarmes:****Figura 39 - Estado atual da página de alarmes**

A página de alarmes já possui todos os elementos visuais que foram propostos nos mockups. Ainda algumas mudanças serão feitas, mas a ideia é que esta página seja simples. Porém, os botões de acionar alarmes ainda não funcionam.

**- Página de dados armazenados:****Figura 40 - Estado atual da página de Dados**

A página de dados Armazenados ainda não possui conteúdo. Esta página deve disponibilizar uma lista com os dados do edifício que estão em uma base de dados. Ainda não é



possível armazenar os dados coletados, pois, como mencionado anteriormente, apenas consigo ler e fazer *print* dos dados enviados pelo Arduino através da SerialPort. Para dar continuidade a esta página, preciso encontrar a melhor maneira de tratar destes dados.

## - **Próximos passos**

As principais tarefas a serem realizadas nas próximas semanas serão:

- Encontrar uma maneira de tratar os dados recebidos na porta serial
- Visualizar os dados em tempo real na página principal
- Enviar estes dados para a base de dados
- Disponibilizar dados na página “Dados Armazenados”

## Progresso de Trabalho – Março/Abril

### - Circuito Arduino

Não foram realizadas grandes alterações ao circuito em si, porém diversas mudanças foram feitas ao código do Arduino. Agora o Arduino envia os dados ao sistema de uma maneira que facilite a leitura e disponibilização destes dados em tempo real, tendo em vista que anteriormente os dados apresentados por vezes estavam incorretos ou eram enviados fora de ordem. Para solucionar o problema, o Arduino envia os dados no formato: “%tipo%valor;”, onde “%tipo” pode ser “H” para quando for um valor de humidade, “T” para quando for um valor de temperatura e “L” para quando for um valor de luminosidade. Desta forma foi possível garantir melhor precisão ao disponibilizar os dados na plataforma.

```
H45.30;
L46.00;
T26.20;
```

**Figura 41 - Exemplo de dados enviados à plataforma**

Agora o Arduino também é capaz de ler diversos inputs que são feitos à porta Serial. Cada input, que neste caso são números inteiros, realiza uma ação diferente. As ações podem ser:

**Tabela 5 - Comandos do Arduino**

Input (Int)	Ação
1	Ativar Servo Motor
2	Ativar Buzzer
3	Toggle LED vermelho
4	Ler temperatura
5	Ler luminosidade
6	Ler humidade
7	Ler distância de objetos
8	Ativar alarme de temperatura
9	Desligar alarme de temperatura
10	Ativar alarme de luminosidade
11	Desativar alarme de luminosidade
12	Ativar LED vermelho

13	Desativar LED vermelho
----	------------------------

## - Plataforma

Como mencionado no item anterior, a visualização dos dados em tempo real está agora mais precisa e estável. Anteriormente, existia um bug que impedia a visualização dos dados em tempo real quando o utilizador mudava para outra página da plataforma. No entanto, este problema foi resolvido e agora o utilizador pode navegar por todas as páginas sem interromper a visualização dos dados. Foram também realizadas mudanças visuais à página inicial para melhor visualização dos dados.

A página de dados armazenados agora já possui uma tabela com os dados que estão armazenados na tabela 'dados'. A base de dados foi criada no MySQL e até o momento possui a tabela 'Dados' e a tabela 'Utilizadores'. A tabela 'Dados' armazena a temperatura, humidade, luminosidade, data e hora da coleta dos dados. Enquanto a tabela de utilizadores possui um id e três valores Boolean diferentes, um para cada tipo de alarme diferente, que determina se o utilizador acionou ou não um dos alarmes da última vez que utilizou a plataforma.

A página de alarmes sofreu mudanças em relação ao update anterior. Os alarmes de temperatura e luminosidade já estão implementados e funcionais, faltando apenas implementar o terceiro alarme. Os alarmes também devem ter uma função de exibir uma mensagem pop-up na plataforma para quando são acionados, o que ainda não foi implementado.

Em seguida tem-se screenshots das páginas da plataforma:



Figura 42 - Página Inicial (Update de Março/Abril)

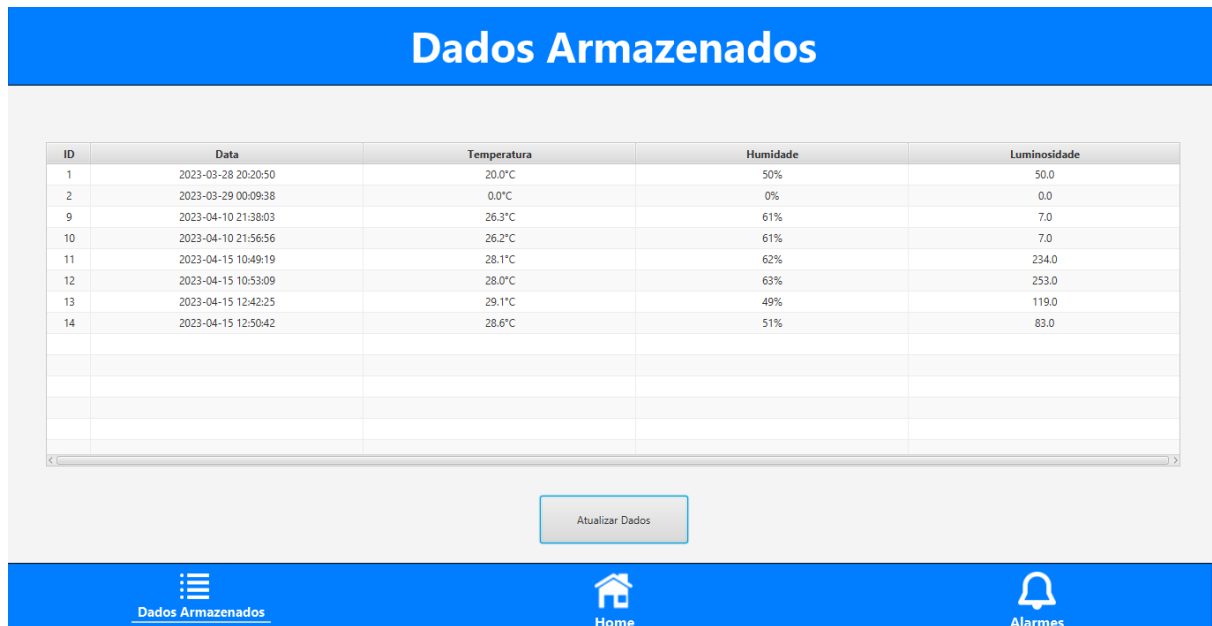


Figura 43 - Página de Dados (Update de Março/Abril)

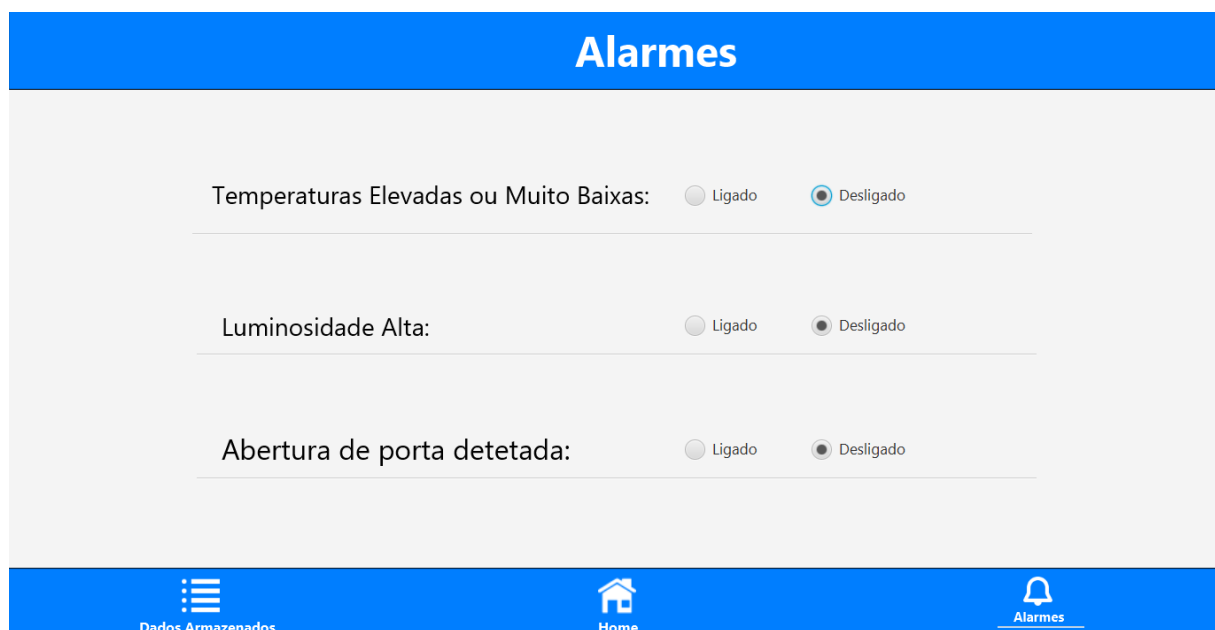


Figura 44 - Página de Alarmes (Update de Março/Abril)

## - Próximos Passos

Por enquanto a comunicação entre o microcontrolador e a plataforma é realizada por um cabo USB e o Serial Port. Como o Arduino UNO não possui função de comunicação wireless, deve ser utilizado o módulo ESP8266. O módulo será incorporado ao circuito, sendo necessário posteriormente adaptar o código da plataforma para funcionar com este módulo.

Alguns requisitos ainda faltam ser implementados, principalmente os requisitos relacionados ao estabelecimento de um período para a coleta automática de dados. Na página de alarmes, é preciso implementar o alarme de abertura de portas e também é necessário que o utilizador receba uma notificação na plataforma quando um dos alarmes for acionado. Na página de dados armazenados ainda é preciso alterar a parte visual da tabela em si e também implementar a função de disponibilizar gráficos que mostrem a evolução dos dados armazenados.

## Progresso de Trabalho – Maio/Junho

### - Finalização do Trabalho

Foram implementados os requisitos restantes e foram feitas diversas alterações em todas as páginas da aplicação. Todos os alarmes estão funcionais, já há sistema de popup, já é possível visualizar os dados através de gráficos, foram adicionados diversos filtros à tabela e foi feita a coleta automática de dados. Esta coleta automática funciona da seguinte forma: ao ativar a opção de coleta, o sistema estabelece três períodos para coletar os dados, um durante a manhã, outro durante a tarde e um no fim da tarde. Isto permite a visualização dos dados durante todo o percurso do dia, podendo avaliar de que forma os dados evoluem.

A tabela possui diversos filtros que podem ser combinados de diversas formas diferentes. Por exemplo, pode-se saber quais dias tiveram temperatura acima de 30 graus e humidade acima de 50%, ou quais dias do mês tiveram maior temperatura, etc. Os gráficos também podem ser filtrados por dia, sendo possível ver um dia específico ou todos os dados coletados em um mês específico. O funcionamento destes filtros se dá, basicamente, através de queries de SQL que são enviadas à base de dados. Abaixo há um exemplo de como isso funciona no programa.

- Primeiro verifica-se quais filtros foram introduzidos pelo utilizador:

```
// Verificar se as variáveis são nulas e atribuir "" se necessário
dia = (dia == null || dia.equals("Não Especificar")) ? "" : dia;
mes = (mes == null || mes.equals("Não Especificar")) ? "" : mes;
maxTemp = (maxTemp != null) ? maxTemp : "";
minTemp = (minTemp != null) ? minTemp : "";
maxHum = (maxHum != null) ? maxHum : "";
minHum = (minHum != null) ? minHum : "";

if (!mes.equals("")) {
    for (int i = 0; i < meses.length; i++) {
        if (mes.equals(meses[i])) {
            mesNum = i + 1;
        }
    }
}
```

Figura 45 - Exemplo Código de Filtros

- Após isso, é construída a query:

```
// Construir a query SQL
String query = "SELECT * FROM dados WHERE ";

if (!dia.isEmpty()) {
    query += "DAY(data) = " + dia + " AND ";
}

if (!mes.isEmpty()) {
    query += "MONTH(data) = " + mesNum + " AND ";
}

if (!maxTemp.isEmpty()) {
    query += "temperatura < " + maxTemp + " AND ";
}

if (!minTemp.isEmpty()) {
    query += "temperatura > " + minTemp + " AND ";
}

if (!maxHum.isEmpty()) {
    query += "humidade < " + maxHum + " AND ";
}

if (!minHum.isEmpty()) {
    query += "humidade > " + minHum + " AND ";
}

if (query.endsWith("AND ")) {
    query = query.substring(0, query.length() - 4); // Remove o "AND " do final
}
```

Figura 46 - Exemplo Construção da Query

- Por fim, a query é executada em uma função específica que recebe os dados da base de dados e armazena em um ArrayList que é utilizado para construir os gráficos ou a tabela:

```
/**Função para obter os dados ordenados por mês ou dia
 * @param sql: query a ser executada
 * @return lista com os dados*/
protected ArrayList<BuildingData> getSortedData(String sql) throws SQLException {
    ArrayList<BuildingData> data = new ArrayList<BuildingData>();
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/tfc", "user: "root", password: "123");

    try (conn; Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {
        while (rs.next()) { //Executa a query
            data.add(new BuildingData(rs.getString(columnLabel: "id"), rs.getString(columnLabel: "data"),
                Double.parseDouble(rs.getString(columnLabel: "temperatura")),
                Double.parseDouble(rs.getString(columnLabel: "luminosidade")),
                Double.parseDouble(rs.getString(columnLabel: "humidade")))); //Armazena os dados na ArrayList
        }
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }

    return data; //Retorna lista com os novos dados
}
```

Figura 47 - Exemplo código de envio de queries à base de dados

## - Resultado dos Ecrãs:

Abaixo estão o estado final dos ecrãs desenvolvidos:



Figura 48 - Página inicial (versão final)

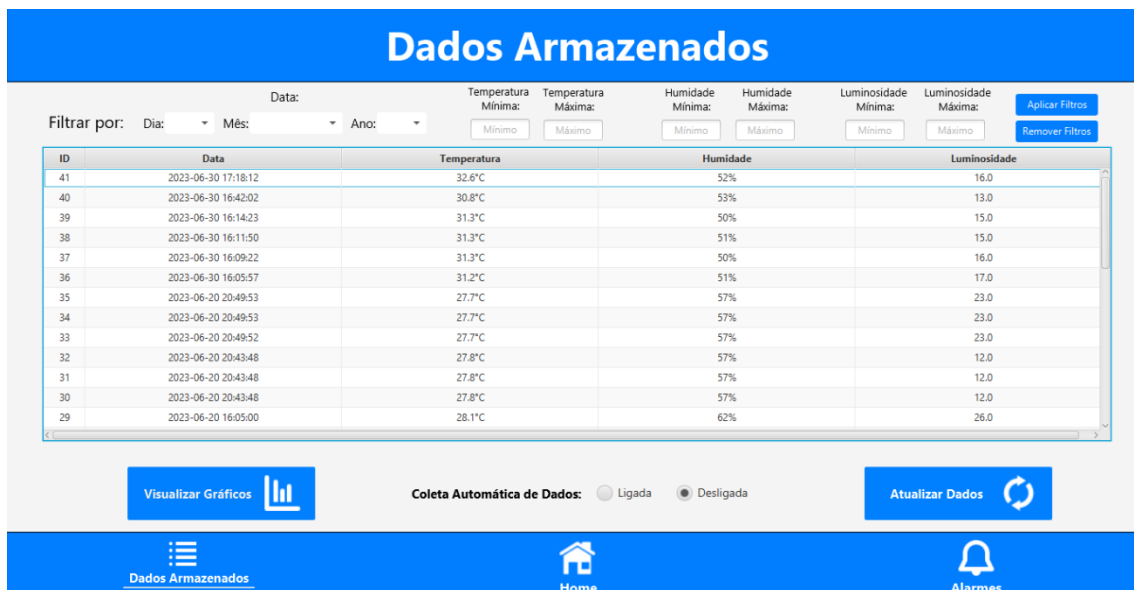


Figura 49 - Página de Dados (versão final)

## Alarmes

Temperaturas Elevadas ou Muito Baixas: ☒ Ligado ☐ Desligado


Temperatura Mínima

Temperatura Máxima

Confirmar

Luminosidade Alta: ☐ Ligado ☒ Desligado

Abertura de porta: ☒ Ligado ☐ Desligado

 Dados Armazenados

 Home


 Alarmes

Figura 50 - Página de Alarmes (versão final)

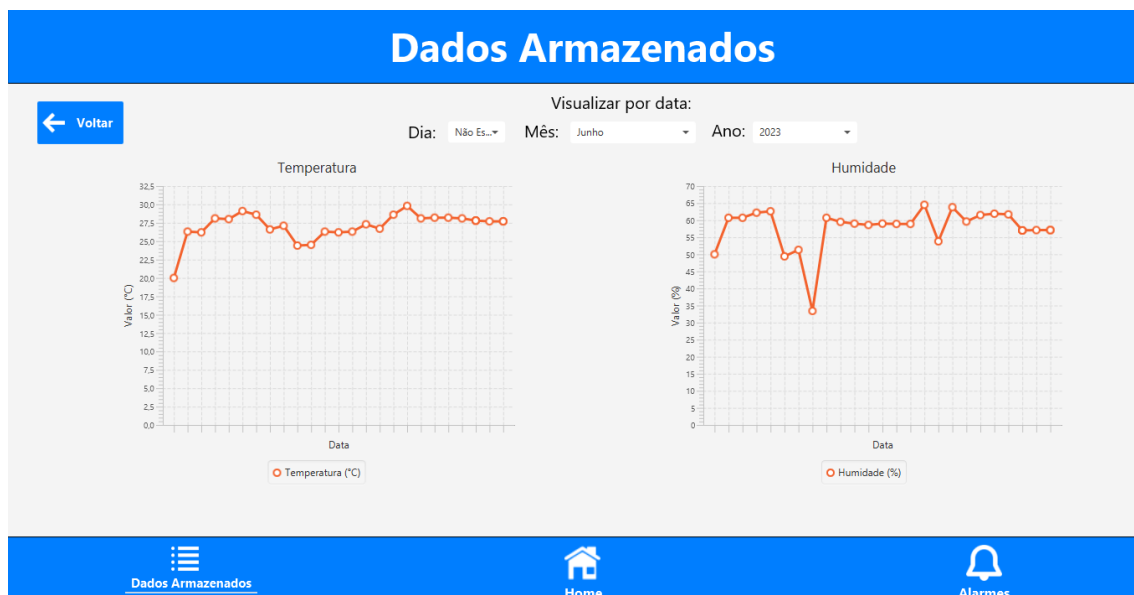


Figura 51 - Página de Gráficos (versão final)



## Teste de Usabilidade

### *Instruções*

O objetivo deste teste é avaliar um protótipo de um sistema de monitorização de edifícios desenvolvido no âmbito de um Trabalho Final de Curso. O sistema tem como objetivo fornecer uma forma de visualizar dados do edifício e fornece algumas interações com os componentes do Arduino.

Será informado(a) caso seja ultrapassada a duração total prevista e daremos o teste por concluído.

***Relembramos que não está a ser testado(a), mas sim a ajudar-nos a melhorar o protótipo.***

Quando der por concluído o teste o/a Administrador(a) pedir-lhe-á para preencher um breve questionário.

O teste é anónimo e confidencial.

**Para darmos início ao teste,** informe o/a Administrador(a) que está preparado(a).

Conclua as seguintes tarefas que testam as principais funcionalidades do protótipo:

**Parte 1 – Testar componentes da página principal:**

1. Verifique que a página inicial está a ser disponibilizada.
2. Verifique se há dados de temperatura, luminosidade e humidade relativa.
3. Teste os botões de Abrir e Fechar os estores.
4. Teste os botões “Desconectar” e “Conectar” na seção “Dispositivo”.
5. Teste a barra de navegação: vá para uma das outras páginas e depois retorne à página principal.

**Parte 2 – Página de Alarmes:**

1. Vá para a página de alarmes.
2. Ative o alarme de temperatura.
3. Escolha a temperatura máxima de 20 graus e carregue em “Confirmar”.
4. Verifique se o alarme foi ativado e se há um popup no ecrã.
5. Selecione a opção “Fechar e Desligar Alarme” do popup
6. Ative o alarme de portas
7. Aproxime sua mão do sensor indicado pelo administrador do teste
8. Verifique o acionamento do alarme
9. Carregue em “Fechar” no popup

**Parte 3 – Dados e Gráficos:**

1. Vá à página “Dados Armazenados”.
2. Verifique se há dados na tabela.
3. Carregue no botão “Atualizar Dados”
4. Vá aos filtros e escolha a data de hoje (dia, mês e ano)
5. Carregue em aplicar filtros e verifique se há um registo na tabela

6. Carregue em “Remover Filtros”
7. Carregue no botão “Visualizar Gráficos”
8. Filtre os gráficos para o dia de hoje
9. Filtre os gráficos para o mês de junho

Caso necessite de mais algum dado contacte o Administrador de teste.

**Obrigada pela sua Colaboração!**

## Questionário de Satisfação

Por favor, leia cada uma das seguintes questões e responda com sinceridade e espontaneidade.

Assinale com uma cruz (X) a opção que considerar mais correcta.

		1	2	3	4	5	
1 Neste protótipo posso facilmente ver as tarefas a realizar	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
2 O nome dos menus está muito adequado	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
3 O desenho gráfico dos ecrãs é	Desagradável	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agradável
4 Clareza do vocabulário utilizado	Pouca	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Muita
5 As cores utilizadas permitem uma fácil leitura	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
6 Facilidade de voltar ao ecrã anterior	Difícil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fácil
7 Os ícones utilizados são de fácil compreensão	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
8 Tamanho da letra	Desadequado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Adequado
9 Aspecto geral dos ecrãs	Desagradável	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agradável
10 Com base apenas naquilo que experimentei e vi, acredito que o Sistema é simples de utilizar	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente

Aspectos mais **negativos**:

---



---

Aspectos mais **positivos**:

---

---

## **Glossário**

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
IOT	Internet of Things