



**Politécnico
de Viseu**

Escola Superior
de Tecnologia
e Gestão de Viseu

Implementação de um sistema de domótica Análise exploratória dos dados recolhidos

Júlio Filipe da Silva Rocha

Trabalho de Projeto

Mestrado em Engenharia Informática - Sistemas de Informação

Trabalho efetuado sob a orientação de
Professor Doutor Carlos Alberto Torres Quental

Outubro de 2023



**Politécnico
de Viseu**

Escola Superior
de Tecnologia
e Gestão de Viseu

Implementação de um sistema de domótica

Análise exploratória dos dados recolhidos

Júlio Filipe da Silva Rocha

Trabalho de Projeto

Mestrado em Engenharia Informática - Sistemas de Informação

Trabalho efetuado sob a orientação de

Professor Doutor Carlos Alberto Torres Quental

Outubro de 2023

Não é triste mudar de ideias, triste é não ter ideias para mudar

Francis Bacon

Agradecimentos

Quero expressar a minha sincera gratidão a todas as pessoas que contribuíram para o desenvolvimento desta dissertação. Durante esta jornada, recebi o apoio e encorajamento de muitas pessoas valiosas, às quais dedico os meus agradecimentos.

Em primeiro lugar, quero agradecer à minha família e amigos, pelo constante apoio, compreensão e incentivo ao longo desta jornada. As palavras de encorajamento e apoio moral foram fundamentais para enfrentar os desafios que surgiram durante o desenvolvimento do projeto.

Quero expressar, também, a minha profunda gratidão ao corpo docente do Departamento de Engenharia Informática da Escola Superior de Tecnologia e Gestão de Viseu. O conhecimento, orientação e ensinamentos que me transmitiram foram inestimáveis para o meu crescimento académico e profissional. Agradeço, ainda, ao Município de Vouzela pela oportunidade e voto de confiança no desenvolvimento deste projeto de domótica aplicado nas instalações da piscina municipal.

Um agradecimento especial vai para o meu orientador, o Professor Doutor Carlos Quental. A sua orientação especializada, apoio constante e valiosos *insights* foram essenciais para o sucesso deste projeto. Agradeço pela disponibilidade, paciência, compromisso e, tão importante, a amizade ao longo destes anos. A sua orientação foi fundamental para moldar esta dissertação e aprimorar as minhas habilidades na área de domótica.

Também gostaria de agradecer aos meus colegas de turma e colaboradores do Município de Vouzela que contribuíram, de alguma forma, para o desenvolvimento deste projeto. As ideias e sugestões partilhadas enriqueceram significativamente o trabalho realizado.

Por fim, gostaria de expressar a minha gratidão a todas as fontes de conhecimento e referências utilizadas neste projeto. Agradeço aos autores, investigadores e programadores, cujo trabalho foi fundamental para a fundamentação teórica e prática deste projeto.

A todos os mencionados e a todos aqueles que, de alguma forma, contribuíram para esta dissertação e para o sucesso do projeto de domótica na piscina municipal de Vouzela, o meu mais sincero agradecimento.

Resumo

O constante crescimento das tecnologias e melhoria em arquitetura de sistemas vem resolver muitos dos problemas dos cidadãos e da sua qualidade de vida. A diversidade de aplicações da Internet das Coisas tem vindo a revolucionar a forma como nos relacionamos com a tecnologia. Redes de sensores industriais sem fio, casas inteligentes controladas remotamente utilizam arquiteturas de rede e protocolos com um nível de automação impensável há alguns anos.

Este projeto propõe a implementação de um sistema de domótica que pretende resolver ou minimizar problemas nas instalações de uma piscina municipal, tais como: assaltos às instalações; controlo da qualidade do ar; fuga de gás no espaço envolvente à caldeira; redução do consumo de energia e o controlo da temperatura e humidade no Cais da Piscina. Esta proposta, como solução de baixo custo, contempla a montagem de placas Arduino, de forma a que os responsáveis da organização possam obter informações numa aplicação Android, em tempo real, e, dessa forma, poderem tomar decisões de acordo com a interpretação dos dados recolhidos.

O projeto divide-se em duas fases: 1) Montagem de três microcontroladores e respetivos sensores e atuadores nas instalações da piscina; 2) Análise dos dados gerados pelo sistema de domótica, recolhidos pelos microcontroladores e armazenados na base de dados.

A primeira fase será dedicada à codificação dos microcontroladores, processamento dos dados recolhidos, armazenamento numa base de dados para lidar com dados de séries temporais, e apresentação dos dados numa aplicação Android bem como na aplicação *desktop* Grafana. Na fase de codificação, destacam-se algumas funcionalidades, como o envio de alertas e o trabalho dos atuadores (relés), cruciais para o bom serviço prestado.

A segunda fase será a análise dos dados e apresentação estatística, através de bibliotecas, como *Seaborn*. O conjunto de dados deriva de uma série temporal, ou seja, os dados sofrem alterações ao longo do tempo, por isso os dados históricos serão analisados e explorados por diversas funções da biblioteca *Pandas*, bem como a produção de uma matriz de correlação, onde será verificada uma correlação negativa, entre os dados da temperatura e humidade e, se possível, a correlação bem visível entre as variáveis do sensor magnético, aplicado na porta, e do sensor de movimento.

PALAVRAS CHAVE: Domótica, Internet Of Things (IoT), Arduino, Android, Influxdb, Análise de Dados, Python, Web API, PHP

Abstract

The constant growth of technologies and improvement in system architecture solve many of the citizens problems and their quality of life. The diversity of Internet of Things applications has been revolutionizing the way we relate to technology. Wireless industrial sensor networks, remotely monitored and controlled smart homes use network architectures and protocols with a level of automation unthinkable a few years ago.

This work proposes the development and implementation of a low-cost home automation system, which intends to solve or minimize problems in the installations of a municipal swimming pool, such as: assaults on the installations; air quality control; gas leakage in the space surrounding the boiler; reduction of energy consumption and control of temperature and humidity on the pool deck. This proposal, as a low-cost solution, addresses the assembly of Arduino with microcontrollers, so that those responsible for the organization can obtain information in an Android application, in real time, and, in this way, be able to make decisions according to the interpretation of the collected and analyzed data.

The project is divided into two phases: 1) Assembly of three microcontrollers and respective sensors and actuators in the pool facilities; 2) Analysis of the data generated by the home automation system, collected by the microcontrollers and stored in the database.

The first phase will be dedicated to Arduino coding, processing the collected data, storing it in a database to handle time series data, and presenting the data in an Android application as well as in the Grafana desktop application. In the coding phase, some features stand out, such as sending alerts and the work of the actuators, crucial for the good service provided.

The second phase will be for data analysis and statistical presentation, through some libraries, such as Seaborn. The data set derives from a time series, that is, the data changes over time, so historical data will be analyzed and explored by various functions of the Pandas, as well as the production of a correlation matrix, where a negative correlation will be verified between the temperature and humidity data and, if possible, the correlation between the variables of the magnetic sensor, applied on the door, and the motion sensor.

KEYWORDS: Home Automation, Internet of Things (IoT), Arduino, Android, InfluxDB, Data Analysis, Python, Web API, PHP

Índice

Agradecimentos.....	i
Resumo.....	ii
Abstract.....	iii
Índice.....	iv
Índice de Figuras.....	vii
Índice de Gráficos.....	x
Índice de Tabelas.....	xi
Lista de Abreviaturas, Siglas e Acrónimos.....	xii
1 Introdução.....	1
1.1 Enquadramento.....	2
1.2 Apresentação do problema e motivação.....	2
1.3 Objetivos.....	3
1.4 Metodologia.....	4
1.5 Planeamento do trabalho.....	5
1.6 Estrutura do documento.....	6
2 Revisão da literatura.....	7
2.1 Análise de ofertas no mercado.....	10
2.1.1 PoolCop.....	10
2.1.2 Zodiac Aqualink.....	11
2.1.3 Legrand MyHome.....	11
2.1.4 Niko Home Control.....	12
2.2 Análise e justificação da solução.....	13
3 Hardware, Ferramentas, Tecnologias e Serviços.....	14
3.1 Hardware: Placas Arduino, sensores e atuadores.....	15

3.1.1	Arduino Mega, sensores e atuadores instalados na Recepção	17
3.1.2	Arduino Uno, sensores e atuadores instalados Casa das Máquinas	20
3.1.3	Hardware comum às placas Arduino instaladas.....	26
3.2	Ferramentas e tecnologias.....	30
3.2.1	Arduino IDE e a programação dos microcontroladores.....	32
3.2.2	PHP Storm no desenvolvimento da API	33
3.2.3	InfluxDB como escolha no armazenamento dos dados	34
3.2.4	Android Studio na concepção da aplicação mobile	35
3.3	Conclusão.....	37
4	Trabalho Experimental	38
4.1	Codificação e montagem do hardware em fase de prototipagem	38
4.1.1	Atualização do <i>firmware</i> da placa ESP8266-01.....	40
4.2	Servidor e Web API.....	43
4.2.1	Configurações iniciais no servidor.....	44
4.2.2	Web API: Fluxo de dados do Arduino ao Android.....	46
4.3	Apresentação dos dados no Grafana	57
4.4	Conclusão.....	59
5	Análise Exploratória de Dados	61
5.1	Espaço da Recepção.....	62
5.2	Espaço do Cais da Piscina.....	67
5.3	Espaço da Casa das Máquinas	69
5.4	Conclusão.....	70
6	Conclusão e trabalho futuro.....	71
6.1	Sugestões para trabalho futuro.....	72
7	Referências	74
	ANEXOS.....	81

Anexo A – Planeamento do trabalho.....	82
Anexo B – Esquema do espaço Receção.....	83
Anexo C – Esquema do espaço Cais da Piscina.....	84
Anexo D – Esquema do espaço Casa das Máquinas.....	85

Índice de Figuras

Figura 1 – Domótica (Fonte: Bolzani C. , 2010).....	1
Figura 2 - Algum hardware reunido para o projeto.....	4
Figura 3 - Arquitetura do projeto de domótica.....	5
Figura 4 - Aplicação Vouzela IoT – Piscina.....	6
Figura 5 - PoolCop.....	11
Figura 6 - Zodiac - AquaLink TRi.....	11
Figura 7 - Legrand My Home.....	12
Figura 8 - Niko Home Control.....	12
Figura 9 - Arduino Mega (Fonte: (Arduino Mega, 2021).....	16
Figura 10 - Arduino Uno (Fonte: (Arduino UNO, 2021).....	16
Figura 11 - Arduino Mega instalado no espaço da Recepção.....	18
Figura 12 - Sensor MC-38 instalado na porta de acesso à Recepção.....	18
Figura 13 - Sensor magnético MC-38.....	19
Figura 14 - Sensor PIR.....	19
Figura 15 - Sensor DHT, modelo 22.....	19
Figura 16 - Sensor MG-811.....	20
Figura 17 - Sensores instalados no hall da Recepção.....	20
Figura 18 - Arduinos Uno instalados no espaço da Casa das Máquinas.....	21
Figura 19 - Sensor DHT instalado no Cais da Piscina.....	22
Figura 20 - Sensor DHT, modelo 22 (DFRobot).....	22
Figura 21 - Sensor LDR instalado na Casa das Máquinas.....	23
Figura 22 - Sensor LDR.....	23
Figura 23 - Sensor PIR instalado no espaço da Casa das Máquinas.....	24
Figura 24 - Sensores MQ-6 instalados no espaço da caldeira.....	25

Figura 25 - Sensor MQ-6.....	25
Figura 26 - Ecrã LCD	26
Figura 27 - Atuador módulo relé de 1 canal.....	27
Figura 28 - Microcontrolador ESP8266-01 com adaptador	28
Figura 29 - Alimentador MB-102, Breadboard e conectores	29
Figura 30 - Arduino IDE	31
Figura 31 - Adição de repositório de placas ESP8266	39
Figura 32 - Instalação de novas placas, concretamente ESP8266.....	40
Figura 33 - Esquema de ligação do ESP8266-01 ao computador, através do Arduino.....	40
Figura 34 - Processo de atualização do firmware no ESP8266-01	41
Figura 35 - Versão atualizada do firmware instalado no ESP8266-01	42
Figura 36 - Conexão e teste entre os alguns componentes de hardware	43
Figura 37 - Medidas criadas na base de dados, através da API.....	46
Figura 38 - Envio de dados do Arduino (Hall/Receção) para a API.....	47
Figura 39 - API regista de dados no InfluxDB.....	48
Figura 40 - App Android requisita à API	48
Figura 41 - API processa pedido de consulta do último registo pelo Android.....	49
Figura 42 - API devolve JSON, de dados brutos, a pedido do Android.....	49
Figura 43 - App Android interpreta JSON e envia dados para interface.....	50
Figura 44 - App Android apresenta dados, em tempo real.....	50
Figura 45 - API processa pedido de consulta pelo Android para apresentação de gráficos.....	51
Figura 46 - API devolve JSON, para gráficos, a pedido do Android.....	52
Figura 47 - App Android interpreta JSON e envia dados processados para os arrays representados pelos eixos dos gráficos.....	52
Figura 48 - App Android apresenta dados, em formato gráfico.....	53
Figura 49 - API processa pedido de consulta pelo Android para apresentação de histórico....	54

Figura 50 - API devolve dados no formato JSON, para histórico, a pedido do Android.....	54
Figura 51 - App Android interpreta JSON e envia dados processados para os arrays que representam cada parâmetro	55
Figura 52 - App Android apresenta dados históricos, em formato lista	55
Figura 53 - App Android permite configurar notificações	56
Figura 54 - App Android recebe notificações	56
Figura 55 - Grafana na configuração da conexão.....	57
Figura 56 - Grafana na configuração de uma visualização	58
Figura 57 - Grafana no ambiente de dashboard com várias visualizações.....	59
Figura 58 - Matriz de Correlação entre a Temperatura e Humidade, na Receção	67
Figura 59 - Matriz de Correlação entre a Temperatura e Humidade, no Cais da Piscina	69
Figura 60 - Cronograma previsto para desenvolvimento do projeto	82
Figura 61 - Esquema circuito do sistema no espaço Receção	83
Figura 62 - Esquema circuito do sistema no espaço Cais da Piscina	84
Figura 63 - Esquema circuito do sistema no espaço Casa das Máquinas.....	85

Índice de Gráficos

Gráfico 1 - Registo estatístico de CO2	64
Gráfico 2 - Registo estatístico de Temperatura e Humidade na Receção	65
Gráfico 3 - Variação da Temperatura e Humidade na Receção	66
Gráfico 4 - Registo estatístico de Temperatura e Humidade no Cais da Piscina	67
Gráfico 5 - Variação da Temperatura e Humidade no Cais da Piscina	68

Índice de Tabelas

Tabela 1 - Registo diário de abertura de porta, na Receção	63
Tabela 2 - Registo diário de movimentos, na Receção.....	63
Tabela 3 - Valor médio obtido da Temperatura e Humidade na Receção.....	66
Tabela 4 - Valor médio obtido da Temperatura e Humidade no Cais da Piscina.....	68
Tabela 5 - Registo diário de movimentos, na Casa das Máquinas	70

Lista de Abreviaturas, Siglas e Acrónimos

API - Application Programming Interface

CSV - Comma-Separated Values

DHT - Digital Humidity and Temperature

IDE - Integrated Development Environment

IOT - Internet of Things

HTTP - HyperText Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

JSON - JavaScript Object Notation

LCD - Liquid Crystal Display

LDR - Light Dependent Resistor

PHP - Hypertext Preprocessor

PIR - Passive Infrared

RISC - Reduced Instruction Set Computer

SSH - Secure Socket Shell

VPS - Virtual Private Server

URL - Uniform Resource Locator

USB - Universal Serial Bus

1 Introdução

Durante os últimos anos, temos vindo a assistir a um grande crescimento do volume de dados que, muitas vezes, geram informação fundamental para auxílio às tomadas de decisão. No caso da Internet das Coisas, concretamente na área da domótica, há, cada vez mais, a preocupação de monitorizar espaços e controlar equipamentos remotamente, através da Internet, bem como consultar o histórico dos dados gerados por sensores e atuadores conetados a dispositivos que suportam microcontroladores. Automação residencial (domótica) (Figura 1) pode ser entendida como uma rede que integra e controla dispositivos eletrónicos de uma habitação, de forma a adequar as necessidades das pessoas e otimizar as funções elétricas, tecnológicas e sustentáveis (Shah & Yaqoob, 2016).



Figura 1 – Domótica (Fonte: Bolzani C. , 2010)

A automação é realizada para a função de ligar ou desligar dispositivos elétricos por meio de placas que incorporam um microcontrolador. Os atuadores, ligados às placas que integram microcontroladores, são geridos através de um servidor, que guarda um histórico de eventos e determinam o funcionamento destes através de codificação (Alves, Florian, & Farina, 2022). A automação residencial oferece vários benefícios, incluindo maior comodidade devido à automação de várias tarefas, maior segurança por meio de sistemas de vigilância, monitorização de espaços concretos e melhor gestão de energia, permitindo haver um maior controlo nos custos. Tudo isto pode ser conseguido à distância, através da Internet.

Em suma, domótica e a Internet das Coisas (IoT) são tecnologias relacionadas, mas têm algumas diferenças fundamentais. A domótica é uma área da tecnologia dedicada ao controlo e automação de sistemas e dispositivos domésticos, de humidade, iluminação, vigilância, entretenimento, entre outros que podem ser monitorizados e controlados remotamente. A Internet das Coisas (IoT, na sigla em inglês, *Internet of Things*) é um sistema que permite que dispositivos sejam conetados e monitorizados remotamente pela Internet. A IoT permite o processamento e o uso dos dados obtidos por esses dispositivos para melhorar a eficiência, a

segurança e a conveniência do dia a dia e teve uma forte evolução, sendo atualmente utilizado em vários domínios como casas inteligentes, telemedicina, ambientes industriais, entre outros (El-hajj, 2019).

1.1 Enquadramento

Este projeto enquadra-se na área de domótica, aplicado às instalações da piscina municipal de Vouzela. Embora muitas das definições encontradas sobre domótica nos levem diretamente à automação residencial, isto é, monitorização e controlo remoto de habitações, o trabalho proposto neste documento foca-se numa infraestrutura municipal, no entanto, todo o *hardware*, ferramentas e tecnologias exploradas são transversais a qualquer infraestrutura, seja uma habitação, piscina municipal, centro social, escola, entre outros.

A piscina municipal de Vouzela é um espaço com várias valências desportivas, com cerca de 1500 utilizações por mês. Para além do tanque (piscina), com oferta de escola de natação e hidroginástica, o espaço dispõe de sauna e banho a vapor. Dispõe, ainda, de uma sala de desporto, onde são desenvolvidas atividades como karaté, dança, teatro, entre outras, praticadas pelas várias associações do concelho.

1.2 Apresentação do problema e motivação

O interesse pelos dispositivos IoT, tanto do consumidor como da comunidade científica e académica, tem crescido ao longo dos últimos anos, apontando para um crescimento na utilização de milhares de milhões de dispositivos e a um mercado que já vale milhares de milhões de euros (EUROPEAN COMMISSION, 2022).

As instalações da piscina municipal são muito utilizadas pela população em geral, desde crianças aos séniores e tem aquecimento conduzido a gás propano. Contudo, levantam-se alguns problemas que devem ser minimizados ou resolvidos, como, por exemplo, a inexistência de sensores de fuga de gás, no espaço da caldeira ou a monitorização alternativa da temperatura e humidade do espaço envolvente à piscina. Verifica-se, também, a inexistência de um controlo da qualidade do ar no espaço da Receção, onde, a certas horas do dia, durante a primeira semana de cada mês, devido ao pagamento das mensalidades, acolhe muitos utentes. Ainda, e não menos importante, existem ocorrências registadas devido a assaltos, durante o período noturno. Todos os problemas encontrados motivaram o levantamento de necessidades, bem como um processo de investigação, e à proposta de monitorização de alguns espaços, para

que exista um controlo em tempo real, em alguns casos alternativo, que zele pela segurança dos utentes e registre a entrada não prevista de pessoas, fora do horário laboral.

1.3 Objetivos

Este projeto tem como objetivo o desenvolvimento de uma solução de baixo custo, nas instalações da piscina municipal, através da implementação de microcontroladores em espaços distintos, que suportarão sensores e atuadores, bem como a monitorização, em tempo real, através de uma aplicação para Android desenvolvida para o efeito.

A primeira fase passará por estudar algumas divisões dentro das instalações da piscina municipal, concretamente o espaço da caldeira, Cais da Piscina e Receção. Com o estudo dos espaços mencionados, obteremos, com mais rigor, os pontos estratégicos onde serão instalados os sensores, na fase seguinte. Posteriormente serão instalados três microcontroladores, um em cada divisão, e respetivos sensores e atuadores, nos locais definidos aquando do estudo de cada espaço. Estes microcontroladores, antes de serem estrategicamente colocados nas divisões já mencionadas, serão alvo de programação. A título de exemplo, os sensores de fuga de gás propano serão instalados próximo de algumas juntas/emendas em curvas nas tubagens de passagem de gás e/ou próximo do ponto de ignição.

Seguidamente será desenvolvida uma aplicação para Android, onde serão apresentados dados, em tempo real, recolhidos pelos vários sensores, bem como recebidos alertas no caso de alguma anomalia detetada.

Na fase final do projeto, haverá uma análise dos dados recolhidos, durante determinado período de tempo, de forma a auxiliar algumas tomadas de decisão, como por exemplo, nova definição, no controlador geral, da temperatura e humidade relativa no Cais da Piscina; aquisição de equipamento de ar condicionado e/ou purificação de ar ou, até, necessidade de obras de vários tipos.

Resumidamente, os objetivos pretendem, com a implementação de um sistema de microcontroladores, fazer a análise de dados e monitorização, em tempo real, através de uma aplicação Android, para resolver um conjunto de necessidades prioritárias, que apontem para a prestação de um serviço público mais eficiente, como a prevenção de assaltos, fuga de gás, controlo de temperatura e humidade, qualidade do ar, redução do custo energético, entre outros aspetos que, durante o desenvolvimento do projeto venham a ser detetados, e que contribuam

para a segurança e bem-estar dos utentes e colaboradores do serviço da piscina municipal de Vouzela.

1.4 Metodologia

Nesta secção é abordada a metodologia explorada para a possível resolução do conjunto de problemas encontrados: 1) Pesquisa bibliográfica; 2) Revisão da literatura e estado da arte; 3) Estudo das tecnologias a utilizar; 4) Desenvolvimento do sistema; 5) Recolha de dados dos dispositivos / sensores; 6) Análise e tratamento dos dados.

Após levantamento das necessidades inerentes ao bom funcionamento dos espaços da piscina municipal de Vouzela e identificação dos problemas, realizou-se, em primeiro lugar, uma pesquisa bibliográfica, em teses, dissertações e artigos científicos sobre domótica, analisando como outros autores resolveram problemas semelhantes.

Após revisão da literatura, concluímos que a utilização da plataforma Arduino seria uma solução de baixo custo e fiável, devido à grande oferta de sensores e atuadores existentes no mercado, para além de ser uma plataforma de código aberto.

Em segundo lugar, e depois de estudados os espaços da piscina municipal, foi elaborada a lista de hardware necessário para cada divisão da infraestrutura, como se mostra na Figura 2. A título de exemplo, serão necessários dois sensores magnéticos e um microcontrolador que envie, em tempo real, o atual estado de cada sensor, para os responsáveis das instalações, o que permitirá perceber a entrada não prevista de pessoas nas instalações, minimizando o impacto dos assaltos registados até à presente data.

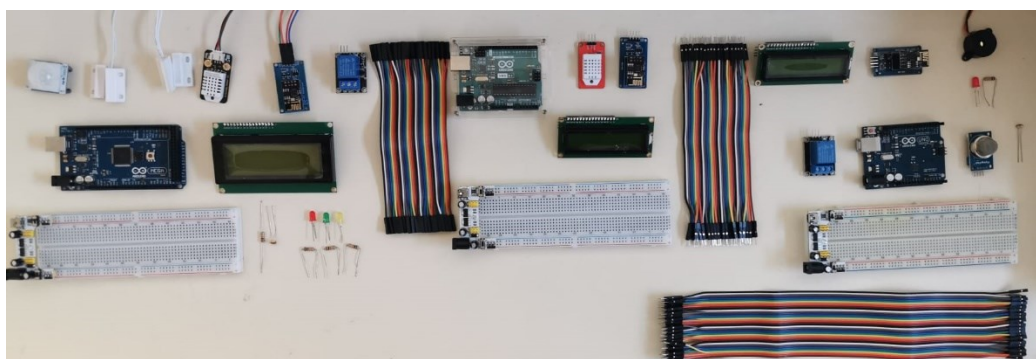


Figura 2 - Algum hardware reunido para o projeto

Com a lista completa do conjunto de hardware necessário (microcontroladores, sensores e atuadores), foram estudadas, com auxílio da revisão da literatura, as ferramentas de software e tecnologias mais indicadas para o desenvolvimento e implementação deste projeto. Escolheu-se

um servidor Linux, primeiro por ser uma solução *Open Source*, segundo por ser de baixo custo, com recursos suficientes para o desenvolvimento de uma *API (Application Programming Interface)* em *PHP (Hypertext Preprocessor)*, que processe os dados enviados pelos microcontroladores. Para o armazenamento dos dados será utilizado o motor de base de dados InfluxDB, por ser, além de gratuito, indicado para projetos IoT e a apresentação da informação na aplicação Grafana. A Figura 3 ilustra a arquitetura do projeto de domótica implementado na piscina municipal de Vouzela.

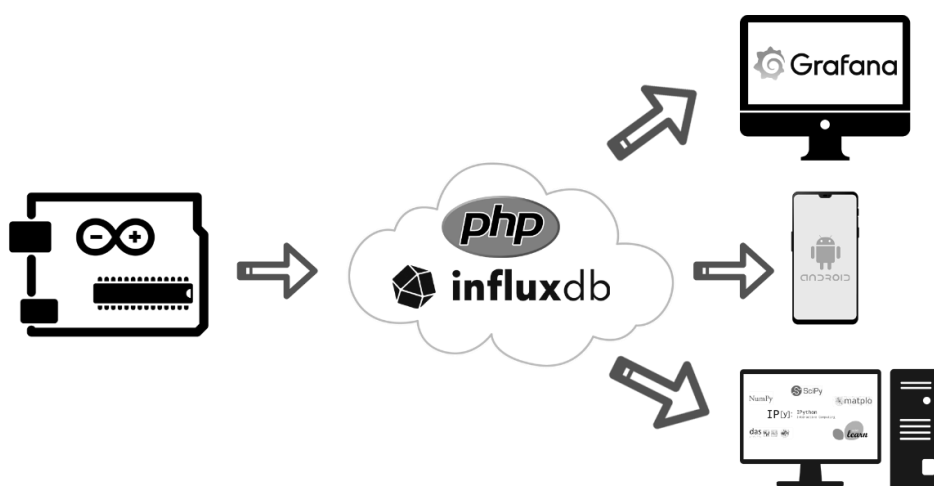


Figura 3 - Arquitetura do projeto de domótica

De seguida, foram analisadas as várias possibilidades de ferramentas e tecnologias a utilizar para observação dos dados em tempo real, bem como a receção de alertas se verificadas determinadas condições, optando-se pelo desenvolvimento de uma aplicação Android, devido à experiência já adquirida com aplicações Android e por considerarmos que é mais prático, cómodo e rápido consultar a informação num *smartphone* (Figura 4).

Prevemos, também, para complementar a *app*, fazer a apresentação da informação na aplicação Grafana. Por fim, será realizada uma análise exploratória dos dados recolhidos, através da linguagem Python, com recurso a algumas bibliotecas disponíveis para esse efeito. Através dessa análise, será possível perceber se será necessário reprogramar alguns dispositivos internos na piscina municipal, como o *set-point* da temperatura e humidade ou a aquisição de equipamentos de tratamento do ar.

1.5 Planeamento do trabalho

Na Figura 60 (Anexo A) é apresentado o cronograma do projeto, através do gráfico de Gantt, com as tarefas previstas e concretizadas nos meses indicados para o ano 2023.

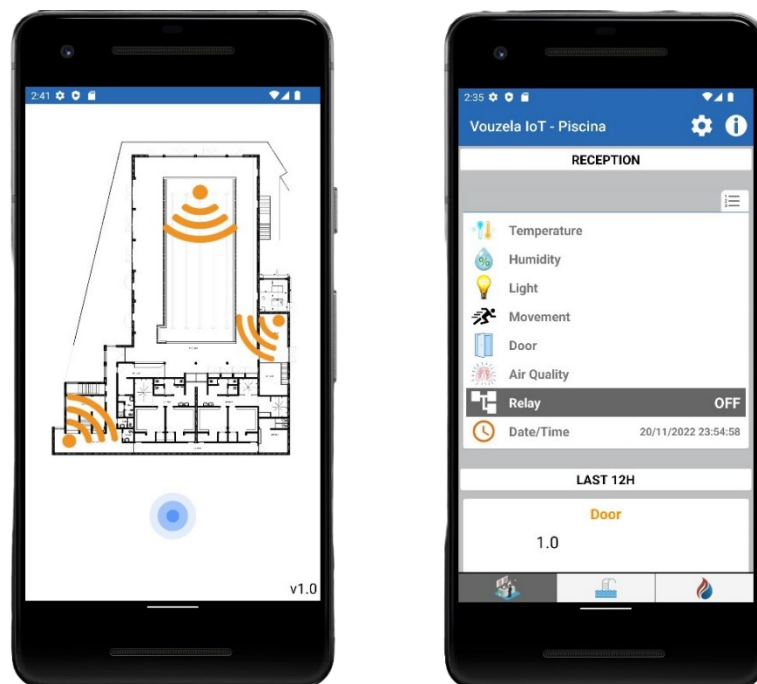


Figura 4 - Aplicação Vouzela IoT – Piscina

1.6 Estrutura do documento

Neste capítulo, introdução, é apresentado um breve enquadramento do trabalho, o problema encontrado e a solução proposta, a calendarização prevista e organização do documento.

No capítulo 2 é apresentada uma revisão da literatura, onde são abordados alguns estudos de outros autores sobre o mesmo tema, ou seja, soluções encontradas para problemas semelhantes.

O capítulo 3 descreve o hardware escolhido para a implementação do projeto, bem como as ferramentas de desenvolvimento utilizadas e tecnologias aplicadas.

O capítulo 4 descreve o trabalho experimental, desde a aquisição do servidor até à apresentação dos dados em tempo real numa aplicação Android e Web, passando pela montagem do hardware, desenvolvimento da Web API (API) e conceção da aplicação para smartphone.

O Capítulo 5 aborda a análise exploratória dos dados recolhidos, que poderão auxiliar em tomadas de decisão.

No Capítulo 6 são apresentadas as conclusões e sugestões para trabalho futuro.

2 Revisão da literatura

A Internet mudou a vida do ser humano permitindo conectividade em qualquer lugar e a qualquer hora e, nas últimas duas décadas, impulsionou avanços extraordinários da tecnologia, permitindo ligar a Internet aos sistemas informáticos e a objetos do mundo real, tais como eletrodomésticos, dispositivos “*smart*”, veículos, entre outros. Esta conexão entre dispositivos e Internet foi chamada de *Internet of Things* (IoT)¹.

A Internet das Coisas já não é uma visão de futuro, está presente no dia a dia e tem um impacto profundo no desenvolvimento tecnológico e na vida das pessoas, permitindo que todos os dispositivos comuniquem e se possam configurar de forma independente e sem intervenção humana. Permite monitorizar espaços físicos, em tempo real, melhorando a tomada de decisão baseada nos dados recolhidos, levando-nos a um nível de automação impensável há pouco tempo.

Dispositivos inteligentes, casas inteligentes, domótica são termos cada vez mais em uso. O termo “domótica” resulta da junção das palavras *domus* (casa, em latim) e robótica. Pode ser definido como um conjunto de serviços integrados em um sistema para satisfazer necessidades básicas dos ocupantes do ambiente (Bolzani C. A., 2004), tendo como objetivo o controlo e automação de quase tudo presente numa habitação, desde iluminação, aquecimento, ventilação e ar condicionado, segurança, vigilância, bem como outro tipo de dispositivos, quer seja realizado remotamente ou no próprio local (Carvalho, 2019).

A importância da monitorização em tempo real é referida por Liu (2016) num artigo, em que propôs um sistema de ar condicionado num *datacenter*, suportado por tecnologia na *cloud*, com o objetivo de reduzir o consumo de energia, sem comprometer o funcionamento dos equipamentos dentro do *datacenter*. O sistema de monitorização ambiental do *datacenter*, que opera sobre o protocolo *Zigbee* (Gupta, 2021; Nardo MD, 2021), incluiu um equipamento de ar condicionado, ventilação e controlo de temperatura, através de sensores, enquanto o trabalho na *cloud* consistiu no armazenamento e recolha de dados.

Ramphela (2020) definiu um sistema integrado, também num *data center*, que inclui uma base de dados para registo de atividades(*logs*), alertas via *email* e um programa de controlo de vários sensores, que assume a responsabilidade pela recolha de dados das variáveis monitorizadas,

¹ O termo foi cunhado por Kevin Ashton: <https://www.rfidjournal.com/that-internet-of-things-thing>

como temperatura, humidade, localização, movimento, fumo, água e variação da tensão. À semelhança de outros estudos, o objetivo é monitorizar, em tempo real, variáveis que comprometam o bom funcionamento de um *datacenter*.

Misra (2018) descreve a utilização de um sistema para monitorização de um aterro sanitário, onde sensores (*mq-135* e *mq-136*) são usados para detetar gases nocivos no espaço e limite máximo de resíduos. Esse sistema, que utiliza um microcontrolador Arduino e módulo esp8266 wi-fi para ligação à Internet, envia dados diretamente à entidade municipal, que atua de acordo com os valores recebidos numa aplicação para smartphone.

Saha (2017) apresenta a monitorização da temperatura em vários pontos de um *datacenter*, através do microcontrolador esp8266 wi-fi e sensor (*dht11*), de forma a que quando atingido um determinado valor, sejam lançados alertas via SMS e *email*, de forma a identificar e resolver o problema atempadamente, minimizando qualquer impacto.

Embora o estudo esteja publicado há alguns anos, a arquitetura em camadas do *Smart Home System* baseado em IoT, e descrito por Bing (2011), continua bem presente. O sistema doméstico inteligente, tal como proposto neste projeto, é dividido em três camadas: 1) camada de aplicação; 2) camada de rede; 3) camada de deteção. A camada de deteção é responsável pela recolha de dados de todos os eletrodomésticos e envia esses dados para a camada intermediária, a camada de rede. A camada de rede usa a Internet para enviar dados para a camada de aplicação, que possui funções diferentes em níveis diferentes para finalidades diferentes.

Valente (2022) também dá ênfase ao uso da tecnologia IoT para o uso correto e controlado da água numa zona vinhateira em Portugal. O uso do protocolo LoRaWan (LoRa Alliance, 2022), para transmissão de dados numa longa distância, em detrimento do Wi-fi, é especialmente importante para aplicações em áreas remotas, onde as redes GSM têm pouca cobertura. Os dados recolhidos por sensores sofisticados são armazenados numa base de dados específica para séries temporais (*Influxdb*) (InfluxDB is the time series platform for developers, s.d.) e é aplicado um modelo preditivo de *machine learnig* de forma a determinar e prever, entre outros fatores, o stress hídrico.

Molino (2017) destaca a importância do uso da tecnologia IoT para prevenção de acidentes, concretamente de afogamentos, em piscinas, através de um microcontrolador e respetivos

sensores, que possibilitam, através de uma plataforma, a emissão de alertas sobre o perigo no ambiente aquático, através da análise, em tempo real, do estado das ondas.

Gresa (2021) levantou a questão da preocupação na manutenção de piscinas e os efeitos positivos do uso da tecnologia IoT para redução de custos. A proposta de solução baseia-se numa aplicação web, desenvolvida em *PHP* (PHP, 2022), que recebe dados, também em tempo real, de um microcontrolador Arduino, para controlo do nível da água, temperatura ambiente e pressão atmosférica.

Outro estudo, que servirá de base para esta proposta de projeto, foca-se numa solução de baixo custo para monitorização de um *datacenter*, enquanto infraestrutura de tecnologias de informação para recuperação de desastres, onde é utilizado apenas um microcontrolador Arduino, versão uno, sensores de temperatura, humidade, luz, movimento e um sensor magnético, este último, para verificação de violação do espaço (entrada indesejada/não prevista no espaço) (Rocha, 2022). Os dados são armazenados numa base de dados *MySQL* (MYSQL, 2022) e apresentados numa aplicação Android, onde ainda há a possibilidade da receção de alertas via *email*, programadas numa API desenvolvida em *PHP*.

O projeto que propomos diferencia-se e vem completar o atrás referenciado (Rocha, 2022), apesar da semelhança no desenvolvimento da solução, assim como noutras investigações referenciadas, há um novo espaço em estudo, que é a piscina municipal de Vouzela, onde se prevê a utilização de dois novos sensores (deteção de gás propano e qualidade do ar); utilização de três microcontroladores Arduino, em vez de apenas um, com apresentação de dados num ecrã LCD; recodificação, nos microcontroladores, dos tempos de recolha de dados; recodificação da aplicação Android para integrar os novos espaços; uma análise estatística mais completa dos dados, através da linguagem *Python* e, para além dos alertas via *email*, também serão programados alertas via serviço *Onesignal* (OneSignal, s.d.) para sistema Android.

Os vários sensores, atuadores e, coerentemente, os microcontroladores, serão instalados em três espaços distintos nas instalações da piscina municipal. Um microcontrolador e sensor de temperatura e humidade (*dht22*) serão instalados no espaço do Cais da Piscina; outro microcontrolador, bem como outro sensor de temperatura e humidade (*dht22*), sensor de luz (*ldr*), movimento (*pir*), magnético (*mc-38*), sensor de qualidade do ar (*mq-135*) e relé (este como atuador), serão instalados junto à entrada secundária de acesso ao exterior e, por fim, um terceiro microcontrolador, sensor de gás (*mq-6*) e outro relé (este também como atuador), serão instalados

no interior da “Casa das Máquinas”. Todos estes microcontroladores, bem como os respetivos sensores e atuadores terão propósitos diferentes, tais como, a recolha de dados em tempo real em espaços diferentes, o envio de alertas e a análise desses dados para tomadas de decisão. Os dados serão armazenados numa base de dados de séries temporais (TSDBs - *Time Series Database*), que parece ser a solução para armazenar uma grande quantidade de dados gerados por aplicações IoT (InfluxDB is the time series platform for developers, s.d.), onde está comprovada a performance superior, comparativamente a outros motores de base de dados mais convencionais (Naqvi, 2017). Para além disso será elaborada uma análise descritiva dos dados armazenados, através de bibliotecas *Python*.

2.1 Análise de ofertas no mercado

O mercado está, cada vez mais, consciente da necessidade de reduzir custos energéticos e de monitorizar e automatizar espaços desportivos, privados ou públicos, que zelem pela segurança dos utentes e qualidade de serviço prestado aos mesmos. Nas secções seguintes apresentam-se algumas soluções encontradas no mercado, direcionadas para piscinas ou habitações.

2.1.1 PoolCop

PoolCop (IoT POOL) é uma solução de domótica focada na manutenção de piscinas (Figura 5), contemplando alguns parâmetros como:

- 1) Filtração: Eficiência e duração, limpeza automática, temperatura da água;
- 2) Nível de água: Consumo, impacto da previsão do tempo, recarga e redução, deteção de vazamentos;
- 3) Tratamento: Controlo de PH, balanço hídrico, desinfeção (cloro, sal, UV, remanescente, entre outros.)
- 4) Equipamentos: Aquecimento, luzes, limpador, fontes, irrigação, entre outros.
- 5) Conexão: Vídeo, visualização e controle, emails, alertas, no computador, tablet e smartphone.



Figura 5 - PoolCop

2.1.2 Zodiac Aqualink

Com uma grande experiência em automação nos EUA, a solução da Zodiac AquaLink TRi, mostrada na Figura 6, permite controlar os diferentes equipamentos da piscina, através do smartphone, tablet ou computador. Foca-se na automação de filtração, tratamento de água, aquecimento, iluminação, limpeza, entre outros, mas vocacionado para piscinas de privados (DR. POOLS, s.d.).



Figura 6 - Zodiac - AquaLink TRi

2.1.3 Legrand MyHome

Esta solução, simples e eficiente, responde às necessidades de segurança, conforto, distribuição de energia e comunicação de qualquer projeto, seja habitação particular ou estabelecimento comercial, graças à compatibilidade com uma ampla gama de dispositivos (LEGRAND, s.d.). Este sistema assume a automação e controlo remoto ou local de iluminação, estores, cortinas, portões, bem como gestão de temperatura, energia e visualização dos consumos.

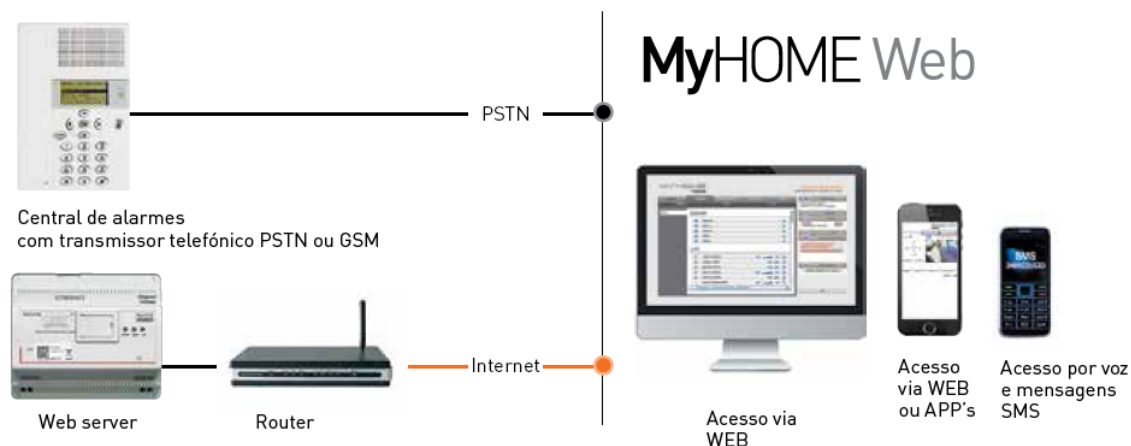


Figura 7 - Legrand My Home

2.1.4 Niko Home Control

Niko Home Control, mostrado na Figura 8, é uma solução escalável dedicada a habitações, permitindo que os equipamentos relacionados com a iluminação, aquecimento, estores, videoporteiro, áudio, entre outros, sejam controlados através de um smartphone ou um interruptor digital central, garantindo também um consumo energético mais eficiente (LD PLAN, s.d.). O sistema Niko Home Control liga-se ao Google Assistant (GOOGLE, s.d.) ou ao Amazon Alexa (AMAZON, s.d.) e possibilita o controlo dos equipamentos da casa com comandos de voz.



Figura 8 - Niko Home Control

2.2 Análise e justificação da solução

Os produtos apresentados no ponto 2.1 respondem a alguns dos requisitos necessários, tais como escalabilidade, e possuem dispositivos específicos com funções de vigilância, automação de portões, iluminação, estores, entre outros.

Contudo, o projeto que se propõe diferencia-se dos apresentados em vários aspetos: é uma solução escalável, de baixo custo, que contempla o controlo e automação residencial aplicado às instalações de uma piscina municipal, pretendendo resolver ou minimizar alguns problemas, tais como: assaltos às instalações; controlo da qualidade do ar; fuga de gás no espaço envolvente à caldeira; redução do consumo de energia e o controlo da temperatura e humidade no Cais da Piscina. Os sistemas conhecidos respondem, apenas, a alguns dos requisitos e oneram consideravelmente a solução.

Além disso, o desenvolvimento e atualização das aplicações de visualização dos dados ou a implementação de novo hardware (sensores, atuadores e microcontroladores), é trabalhado à medida das necessidades encontradas, sem recorrer a entidades externas, permitindo no futuro aumentar facilmente os dispositivos e as funções pretendidas

3 Hardware, Ferramentas, Tecnologias e Serviços

A automação residencial, também conhecida como domótica, tem sido amplamente adotada como uma solução eficiente para melhorar a segurança, o conforto e a eficiência energética dos ambientes. Com o avanço das tecnologias e o surgimento de dispositivos interconectados, como os sensores, atuadores e microcontroladores acoplados em placas Arduino, tornou-se possível desenvolver sistemas inteligentes capazes de controlar e monitorizar diversos aspetos de uma residência ou, neste caso, um estabelecimento público.

Neste contexto, o presente capítulo dedica-se a explorar o projeto de domótica implementado na piscina municipal de Vouzela, com ênfase nos componentes de hardware, nas ferramentas de desenvolvimento de software e nas tecnologias utilizadas.

A escolha adequada dos componentes de hardware, tendo em conta o seu custo-benefício, é fundamental para garantir o funcionamento eficiente e seguro do sistema de automação. O hardware selecionado para o desenvolvimento deste projeto de domótica poderá ser encontrado facilmente no comércio online nacional.

Neste capítulo serão, também, abordadas as placas Arduino, utilizadas no projeto da piscina municipal de Vouzela, bem como os sensores e atuadores utilizados para recolher dados ambientais, monitorizar o estado dos dispositivos e controlar as diferentes funcionalidades do espaço.

Para além disso, será discutido o conjunto de ferramentas de desenvolvimento de software utilizadas durante o projeto. O Arduino IDE (*Integrated Development Environment*), reconhecido pela sua simplicidade e flexibilidade, desempenha um papel central na programação das placas Arduino e na criação de lógica de automação personalizadas.

Adicionalmente, o PHP Storm (JET BRAINS, s.d.), como ambiente de desenvolvimento integrado amplamente utilizado na programação PHP (PHP, 2022), o InfluxDB (InfluxDB is the time series platform for developers, s.d.), como o sistema de gestão de bases de dados escolhido para o armazenamento dos valores lidos pelos sensores e atuadores, e, igualmente importante, o Android Studio (GOOGLE, s.d.), como plataforma de desenvolvimento para aplicações Android.

Por fim, serão apresentadas as tecnologias associadas às ferramentas utilizadas no projeto. A linguagem de programação C, utilizada na programação dos microcontroladores embutidos nas

placas Arduino, permite implementar condições para controlo direto dos componentes de hardware, enquanto o PHP possibilita a criação de uma API, traduzindo-se num conjunto de regras e protocolos que permitem a comunicação e interação entre diferentes sistemas de software, concretamente entre o Arduino e a aplicação Android.

A linguagem de programação Java é explorada para a criação de aplicações móveis que permitam aos responsáveis interagir com o sistema de domótica da piscina municipal de Vouzela de forma intuitiva e prática.

Ao longo deste capítulo serão apresentados alguns detalhes sobre o hardware, as ferramentas de desenvolvimento de software e as tecnologias utilizadas no projeto de domótica implementado na piscina municipal de Vouzela. O objetivo é oferecer uma visão abrangente sobre as soluções adotadas e seus respectivos papéis na criação de um ambiente automatizado, seguro e eficiente.

É importante salientar que, no presente capítulo, os componentes não são abordados de forma tão detalhada, referindo, apenas, qual o hardware utilizado e o seu papel no projeto de domótica.

No capítulo seguinte, “Trabalho Experimental”, serão explorados todos os passos, de forma mais pormenorizada, para melhor compreensão, desde a aquisição de servidor, onde se encontra a API e Base de Dados, até à apresentação dos dados na aplicação para smartphone Android, já disponível na loja da Google (Play Store).

3.1 Hardware: Placas Arduino, sensores e atuadores

Nesta secção serão abordadas as placas Arduino Mega e Uno, juntamente com os microcontroladores acoplados a estas placas, utilizadas no projeto de domótica implementado na piscina municipal de Vouzela.

O Arduino Mega foi instalado no espaço da Recepção, onde estão conetados quatro sensores para monitorizar e controlar diferentes aspetos do ambiente.

O microcontrolador ESP8266-01S serve para enviar os dados recolhidos para uma base de dados através da Internet.

As placas Arduino Mega e Arduino Uno são amplamente reconhecidas como plataformas de desenvolvimento eletrónico e são escolhas populares para projetos de domótica. Essas placas são baseadas em microcontroladores Atmel AVR, que oferecem recursos poderosos e flexíveis

para a implementação de sistemas de automação residencial. As características abaixo indicadas poderão ser confirmadas no site oficial arduino.cc (ARDUINO, 2022).

O Arduino Mega (Figura 9), equipado com o microcontrolador ATmega2560, destaca-se pela sua capacidade expandida de entrada e saída, sendo ideal para projetos que reclamam um maior número de conexões.

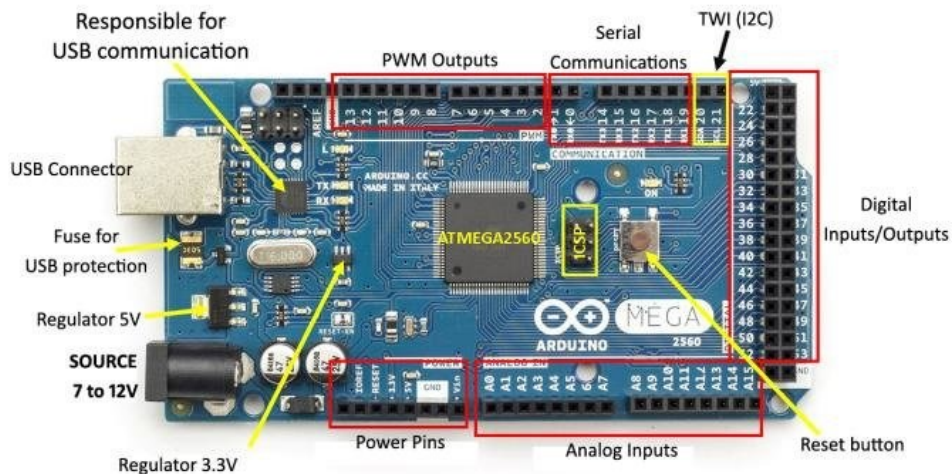


Figura 9 - Arduino Mega (Fonte: (Arduino Mega, 2021))

O Arduino Uno (Figura 10), possui um microcontrolador ATmega328P, e é caracterizado pela sua simplicidade e facilidade de uso. Esta placa possui uma quantidade suficiente de pinos de entrada e saída, tornando-o adequado para projetos de menor escala.

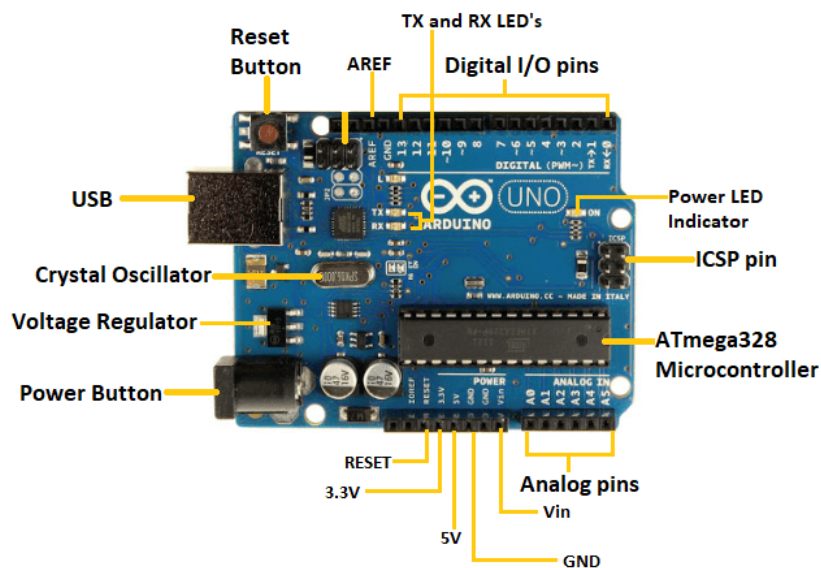


Figura 10 - Arduino Uno (Fonte: (Arduino UNO, 2021))

Estas placas Arduino possibilitam a ligação de diversos sensores e atuadores, permitindo a monitorização e o controlo de variáveis como temperatura, humidade, movimento, luminosidade, entre outros. Os microcontroladores acoplados nestas placas são programáveis através da linguagem baseada em C/C++, o que facilita a implementação da lógica de automação e a interação com outros dispositivos (Banzi, 2011).

Os microcontroladores Atmel AVR presentes nas placas Arduino Mega e Uno possuem características que contribuem para a sua eficiência e versatilidade. Esses microcontroladores são responsáveis por executar as instruções e controlar o fluxo de dados dentro do sistema de automação residencial.

O microcontrolador ATmega2560, presente no Arduino Mega, possui uma arquitetura RISC de 8 bits, frequência de operação de 16 MHz, mas uma memória de programa de 256 KB. Essa capacidade de memória expandida é especialmente útil para projetos mais complexos, que requerem o armazenamento de um maior volume de código e dados.

Já o microcontrolador ATmega328P, utilizado no Arduino Uno, apresenta, também, uma arquitetura RISC de 8 bits, uma frequência de operação de 16 MHz e uma memória de programa de 32 KB. Essas especificações permitem o processamento rápido e eficiente das tarefas do sistema.

Graças aos microcontroladores incorporados nas placas Arduino Mega e Uno, é possível implementar algoritmos, lógica de controlo e comunicação entre dispositivos, tornando o sistema de automação residencial na piscina municipal de Vouzela altamente personalizável e adaptável às necessidades específicas.

3.1.1 Arduino Mega, sensores e atuadores instalados na Receção

No espaço da Receção da piscina municipal de Vouzela, o Arduino Mega desempenha um papel crucial no sistema de automação residencial. Esta placa, conhecida pela sua ampla capacidade de entrada e saída de dados, permite a conexão e o controle de diversos dispositivos, incluindo sensores e atuadores. Na Figura 11 podemos visualizar o Arduino Mega instalado no espaço da Receção da piscina municipal. Abaixo desta encontram-se os sensores conetados ao Arduino Mega, no espaço da Receção.

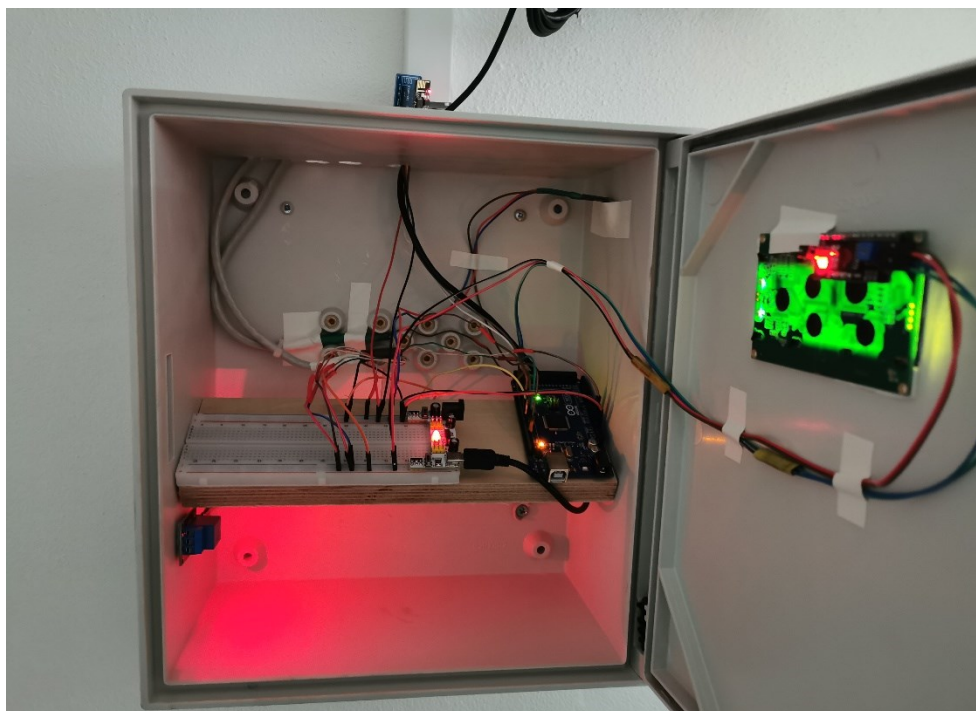


Figura 11 - Arduino Mega instalado no espaço da Receção

Sensor/Interruptor Magnético MC-38

Este sensor foi instalado na porta de acesso ao hall da Receção, como podemos verificar na Figura 12, e tem como objetivo comunicar o estado da porta, ou seja, se está aberta ou fechada.



Figura 12 - Sensor MC-38 instalado na porta de acesso à Receção

O funcionamento do sensor magnético MC-38, como já descrito no próprio nome, usa os conceitos do magnetismo para comutar o seu circuito interno, permitindo o envio de um sinal para o microcontrolador. Podemos dizer que o funcionamento é semelhante a um botão comum: de um lado do sensor existe um contacto aberto, semelhante a um *switch*, e do outro lado existe um íman, que ao aproximarem-se, são atraídos pelo seu campo magnético e, conseqüentemente comutando os contactos no lado do *switch*, como representado na Figura 13.

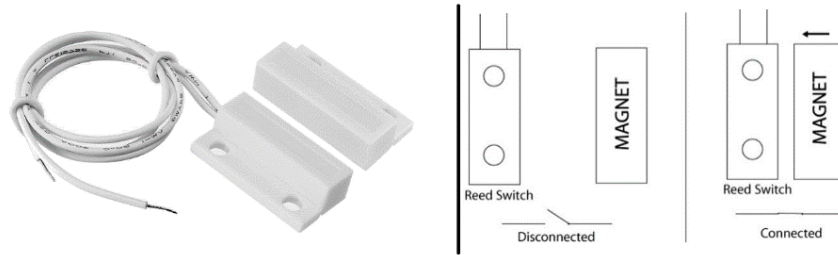


Figura 13 - Sensor magnético MC-38

Sensor PIR (*Passive Infrared*)

O sensor de movimento PIR, mostrado na Figura 14, é responsável pela detecção de movimento no hall da Recepção. Utiliza sensores infravermelhos para identificar variações de calor causadas pelo movimento de pessoas. Quando detecta movimento, envia um sinal para o Arduino Mega, permitindo que o sistema execute ações específicas, concretamente registrar a ação e enviar uma notificação.

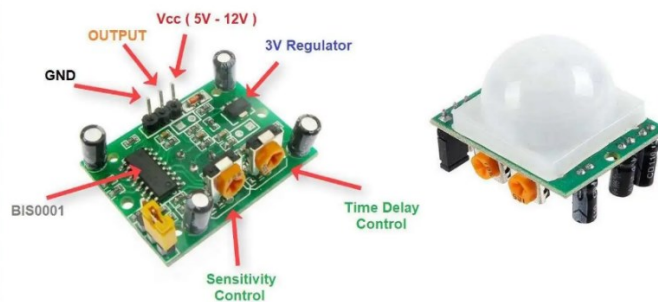


Figura 14 - Sensor PIR

Sensor DHT (*Digital Humidity and Temperature*)

O sensor DHT, modelo 22 da Figura 15, é utilizado para obtenção dos valores de temperatura e humidade no ambiente da Recepção. É capaz de medir com precisão esses parâmetros e envia os dados para o Arduino Mega para posterior processamento e tomada de decisões.

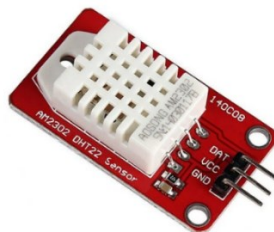


Figura 15 - Sensor DHT, modelo 22

Sensor MG-811

Por razões de saúde e bem-estar dos utentes, o sensor MG-811 (Figura 16) é fundamental no sistema de automação. Esse sensor é responsável por medir os níveis de dióxido de carbono (CO₂) no ambiente. No entanto, estes valores podem variar consoante a temperatura do local em estudo (Saptiani, 2019). Caso se verifique um nível elevado de CO₂, indicando uma má qualidade do ar, ele envia um sinal para o Arduino Mega, que aciona um atuador (relé) para ligar um ventilador, por exemplo. Essa ação visa reciclar o ar e garantir um ambiente mais saudável.



Figura 16 - Sensor MG-811

Este conjunto de sensores foi colocado num local estratégico, junto ao balcão de atendimento, mas de difícil acesso, como ilustrado na Figura 17, de forma a recolher os dados com maior precisão.



Figura 17 - Sensores instalados no hall da Recepção

3.1.2 Arduino Uno, sensores e atuadores instalados Casa das Máquinas

Neste subcapítulo, serão abordadas as placas Arduino Uno instaladas no espaço da Casa das Máquinas da piscina municipal de Vouzela. Essas placas desempenham funções específicas e

estão conectadas a diferentes sensores, permitindo a monitorização e controle de variáveis importantes nesse ambiente. Em particular, um Arduino Uno está conectado ao sensor DHT, modelo 22, localizado no Cais da Piscina, para obter em tempo real os valores de temperatura e humidade, bem como a um sensor de luminosidade (LDR) e um sensor de movimento (PIR), instalados na Casa das Máquinas. Enquanto isso, um segundo Arduino Uno está conectado a dois sensores MQ-6 para deteção de fuga de gás propano, instalados no espaço da caldeira. Em suma, estas duas placas Arduino Uno estão instaladas num espaço denominado Casa das Máquinas, como ilustrado na Figura 18, mas gerem sensores que estão noutros espaços próximos, concretamente no Cais da Piscina e na divisão da caldeira, mostrado no Anexo C e D.

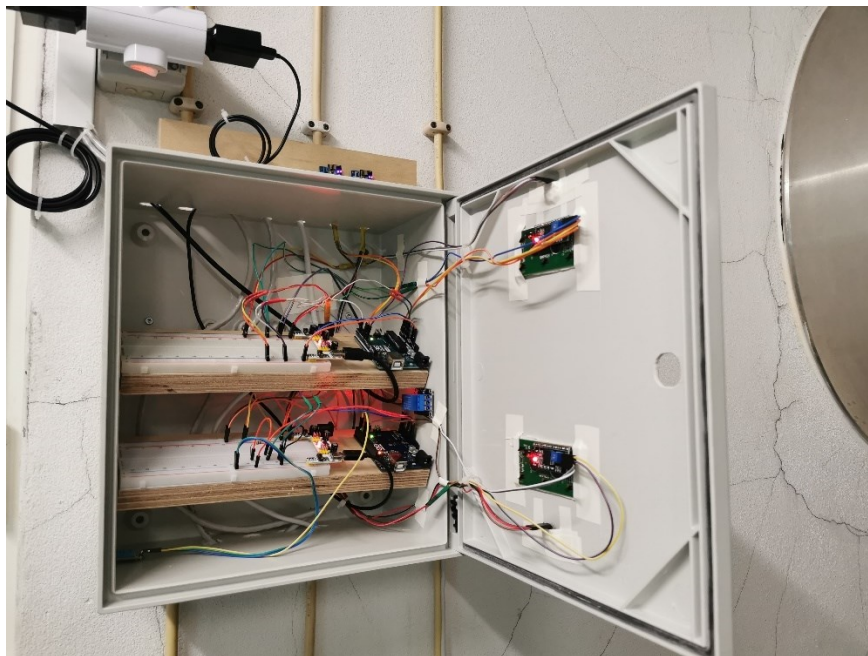


Figura 18 - Arduinos Uno instalados no espaço da Casa das Máquinas

3.1.2.1 Arduino Uno e a conexão ao DHT22 no Cais da Piscina

No espaço da Casa das Máquinas, uma placa Arduino Uno foi dedicada para a monitorização das condições ambientais no Cais da Piscina. Para esse fim, conectou-se ao Arduino Uno, através de fio de cobre, presente nos cabos de rede, um sensor DHT (Figura 19), modelo 22, da DFRobot (DFRobot, s.d.), um sensor preciso e confiável para medir temperatura e humidade.

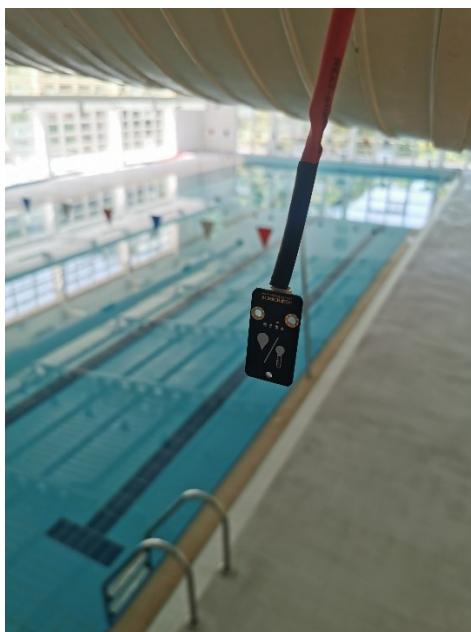


Figura 19 - Sensor DHT instalado no Cais da Piscina

Sensor DHT (*Digital Humidity and Temperature*)

Estrategicamente posicionado no Cais da Piscina, fornece leituras em tempo real dessas variáveis importantes. O Arduino Uno, por sua vez, recebe esses dados e pode executar ações com base neles, como ajustar sistemas de ventilação ou alertar os responsáveis para terem atenção às condições ambientais relacionadas com estes parâmetros.

Dessa forma, a placa Arduino Uno em conjunto com o sensor DHT22 (Figura 20) contribui para a monitorização e manutenção adequadas das condições de temperatura e humidade no Cais da Piscina, garantindo um ambiente confortável e seguro para os utentes.



Figura 20 - Sensor DHT, modelo 22 (DFRobot)

Sensor LDR (*Light Dependent Resistor*)

O sensor de luminosidade mostrado na Figura 21 foi instalado na Casa das Máquinas e conetado ao primeiro Arduino Uno, onde também está ligado o sensor de temperatura e humidade

relativa, instalados no Cais da Piscina. Este sensor é responsável por monitorizar a intensidade de luz no ambiente.

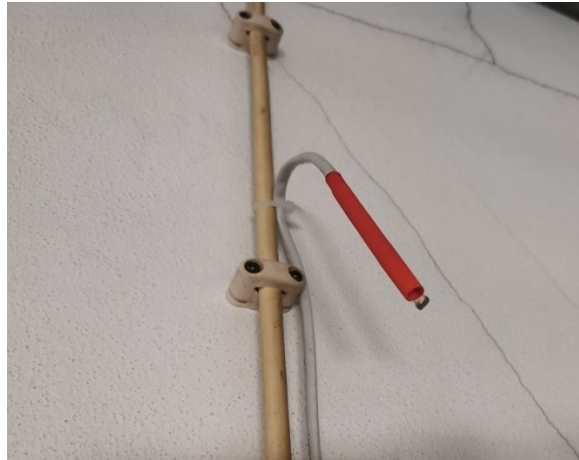


Figura 21 - Sensor LDR instalado na Casa das Máquinas

O princípio de funcionamento do sensor de luz LDR (Figura 22) (resistência dependente de luz) mudará com a variação da intensidade da luz, em que num ambiente mais escuro, a resistência LDR é de cerca de $10\text{M}\Omega$ e num ambiente mais claro é de $1\text{K}\Omega$ ou menos. O LDR é feito de materiais semicondutores, como sulfeto de cádmio. Com este material, quando há menos intensidade de luz, há menos resistência e, assim, permite mais passagem de corrente (Putri, 2018). Com base nas leituras desse sensor, o sistema de domótica pode detetar se alguma luz permanece ligada no período pós-laboral, indicando a necessidade de desligá-la para economia de energia.

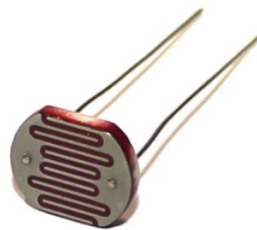


Figura 22 - Sensor LDR

Sensor PIR (*Passive Infrared*)

Na Casa das Máquinas, um sensor PIR (Infravermelho Passivo) está conectado ao mesmo Arduino Uno, concretamente instalado na porta da caixa que alberga os microcontroladores e *breadboards*, como ilustra a Figura 23, de forma a detetar movimento naquela divisão. Com as mesmas características do sensor PIR instalado no espaço da Receção da piscina municipal, este sensor é capaz de detetar movimento através de variações de calor emitidas por objetos ou

pessoas (J. Andrews, 2020). A presença de movimento em horários não previstos pode indicar uma situação anómala.



Figura 23 - Sensor PIR instalado no espaço da Casa das Máquinas

Nesse caso, o Arduino Uno regista a alteração do estado do sensor e envia notificações, email e via aplicação, para os responsáveis diretores das instalações, tal como acontece com o sensor instalado na Recepção.

Em suma, a utilização dos sensores MQ-6, LDR e PIR em conjunto com a placa Arduino Uno, na Casa das Máquinas, permite uma monitorização eficiente e uma resposta rápida a situações anómalas. Isso contribui para o aumento da segurança, economia de energia e eficiência operacional do sistema de domótica implementado na piscina municipal de Vouzela.

3.1.2.2 Arduino Uno e a conexão aos MQ-6 no espaço da Caldeira

Ainda no espaço da Casa das Máquinas, dentro da mesma caixa, outra placa Arduino Uno foi instalada, desempenhando várias funções de monitorização e controlo. Essa placa está conectada aos sensores de deteção de gás propano.

Sensor MQ-6

Dois sensores MQ-6 foram conetados ao Arduino Uno para detetar qualquer possível fuga de gás propano no espaço da caldeira, como podemos verificar na Figura 24. Esses sensores utilizam um material de óxido de metal, que apresenta propriedades de condutividade elétrica que variam em resposta à presença de gases específicos (Ajiboye, 2021).



Figura 24 - Sensores MQ-6 instalados no espaço da caldeira

Quando expostos a determinados gases, os óxidos metálicos modificam a sua condutividade, permitindo a deteção e quantificação da concentração dos gases presentes. Concretamente, os sensores MQ-6 (Figura 25), são altamente sensíveis a gases inflamáveis, como o propano, e, caso seja detetada alguma concentração anormal, o Arduino Uno pode acionar alarmes, tomar medidas de segurança apropriadas, como interromper o fornecimento de gás ou ligar um extrator. Concretamente, foram programados e instalados dois sensores de deteção de gás, de forma a despistar possíveis falsos alarmes, possibilitando a verificação e correção mais tarde, isto é, caso os valores de apenas um sensor seja, sem razão aparente, maior que o valor predefinido no microcontrolador, não são acionados alertas ou ativada a porta onde está ligado o atuador relé.

Para este projeto foi ligado um relé, como atuador, à placa Arduino, de forma a permitir a ligação de um extrator de gás para o exterior, uma sirene ou outro dispositivo que os responsáveis entendam instalar. Aquando da alteração do estado do relé, também é enviada uma notificação para a aplicação, bem como para um endereço de email definido para o efeito.



Figura 25 - Sensor MQ-6

Nos subcapítulos seguintes, relacionados ainda com o hardware utilizado, serão abordados componentes comuns a todas as placas Arduino, sendo os componentes responsáveis pelo acionamento de dispositivos aquando da deteção de gases nocivos, ligação à Internet e a alimentação e conexão do hardware.

3.1.3 Hardware comum às placas Arduino instaladas

Ecrã LCD (*Liquid Crystal Display*)

Adicionalmente, aos sensores e atuadores mencionados anteriormente, é importante realçar que em cada placa Arduino está conectado um ecrã LCD, que desempenha um papel crucial na visualização de informações em tempo real. Estes ecrãs LCD (Figura 26) fornecem *feedback* visual aos utilizadores, apresentando dados relevantes sobre os valores recolhidos e processados pelos microcontroladores, sem que seja necessária a consulta na aplicação Android.



Figura 26 - Ecrã LCD

Uma característica interessante desses ecrãs é a capacidade de controlar a luz de fundo de forma inteligente. Por meio dos sensores PIR, previamente instalados, é possível detetar movimento de objetos junto das caixas onde estão instalados os Arduinos e *breadboards*. Quando um responsável se aproxima, os sensores PIR detetam o movimento e acionam a luz de fundo dos ecrãs LCD, tornando as informações visíveis num ambiente mais escuro. Essa funcionalidade economiza energia, desligando a luz de fundo quando não é necessária, e proporciona uma experiência intuitiva e conveniente para quem pretende visualizar a informação.

A integração entre os sensores PIR e os ecrãs LCD demonstra a versatilidade e adaptabilidade do sistema de domótica, tornando-o mais eficiente e amigável para os responsáveis. A combinação de hardware, sensores, atuadores e interfaces de visualização contribui para uma experiência completa e interativa no controle e monitorização dos diferentes aspetos da domótica implementada na piscina municipal de Vouzela.

Atuador Relé

O módulo relé de um canal (Figura 27), com a tensão de funcionamento a 5 volts, desempenha um papel fundamental como atuador. O relé é um dispositivo eletromecânico que funciona como uma chave controlada eletricamente. Possui dois estados possíveis: normalmente aberto (NA) e normalmente fechado (NF). Quando o relé é ativado por um sinal elétrico, o estado da chave é alterado, permitindo a passagem de corrente elétrica ou interrompendo-a. Além disso, é facilmente integrado ao Arduino, permitindo que o sistema de automação controle dispositivos de elevada potência de forma segura e eficiente.



Figura 27 - Atuador módulo relé de 1 canal

Na piscina municipal são utilizados dois módulos de relé de um canal como atuadores para ativar dispositivos importantes no contexto da domótica. Concretamente, quando o sensor MG-811 identifica níveis elevados de dióxido de carbono no espaço da Receção, o Arduino Mega envia um sinal para o relé, que, por sua vez, liga um ventilador. Essa ação tem como objetivo reciclar o ar, melhorando a qualidade do ambiente para os utentes. Outro módulo de relé de um canal está ligado a um dos Arduino Uno instalados na Casa das Máquinas, de forma a ligar um extrator ou uma sirene, aquando da deteção de gás propano, na divisão da caldeira.

Os Arduinos Mega e Uno atuam como o cérebro do sistema, recebendo os sinais dos sensores e tomando decisões com base nesses dados, acionando os atuadores quando necessário. Além disso, estas placas possuem capacidade de ligação à Internet por meio de módulos WiFi ou Ethernet, permitindo o envio dos dados recolhidos para uma base de dados externa.

Microcontrolador ESP8266-01

Para enviar os dados lidos pelos sensores e atuadores para uma base de dados pela Internet, foi ligado, a cada placa Arduino, um microcontrolador ESP8266-01S. Este módulo WiFi, compacto e de baixo consumo de energia, é amplamente utilizado para permitir conectividade à Internet a

dispositivos eletrônicos. A placa ESP8266-01, mostrada na Figura 28, é altamente compacta, utilizada como periférico para qualquer placa via série (RX/TX) e também como placa autônoma, compatível com redes sobre os protocolos 802.11 b/g/n (Satapathy, 2018). A ligação à Internet é possível atualizando o *firmware*, que permite a execução de comandos AT (Oliveira, 2022). Os comandos AT do ESP8266-01 seguem uma sintaxe específica e são precedidos pela sequência "AT". Após o comando "AT", é possível adicionar um comando específico, como "AT+RST" para reiniciar o módulo ou "AT+CWMODE" para configurar o modo de operação WiFi, ou seja, desempenhar a função de estação, ponto de acesso ou ambos. É de realçar que estas operações são executadas em segundo plano, no caso deste projeto de domótica, através da biblioteca "WiFiEsp" (WiFiEsp, s.d.), integrada no código carregado nos Arduino.

Um ESP8266-01S foi conectado a cada placa Arduino, permitindo a transmissão dos dados colhidos para uma base de dados remota, através de uma conexão sem fio. Essa comunicação é realizada utilizando o protocolo HTTP para o envio das informações.

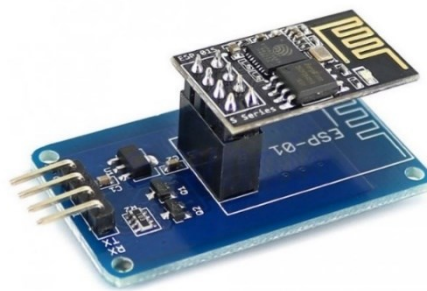


Figura 28 - Microcontrolador ESP8266-01 com adaptador

Com a utilização do microcontrolador ESP8266-01S, o sistema de domótica implementado na piscina municipal de Vouzela torna-se capaz de enviar os dados recolhidos para uma base de dados central, possibilitando a análise, monitorização e controlo remoto das condições do ambiente, visando aprimorar a segurança, o conforto e a eficiência energética do local.

Módulo de alimentação e breadboards

As placas Arduino são alimentados por meio de um transformador de 5 volts, com capacidade de fornecer corrente até 2 Amperes, idêntico à maioria dos carregadores para smartphone. Este transformador fornece a energia necessária para o funcionamento adequado dos microcontroladores, através da interface USB (*Universal Serial Bus*).

Para distribuir a alimentação de forma conveniente, é utilizado um módulo de alimentação (MB-102) para *breadboard* (Figura 29). Esse módulo, ligado diretamente ao transformador por

meio de uma interface USB, e acoplada manualmente à *breadboard*, permite fornecer energia às placas Arduino e sensores conectados.

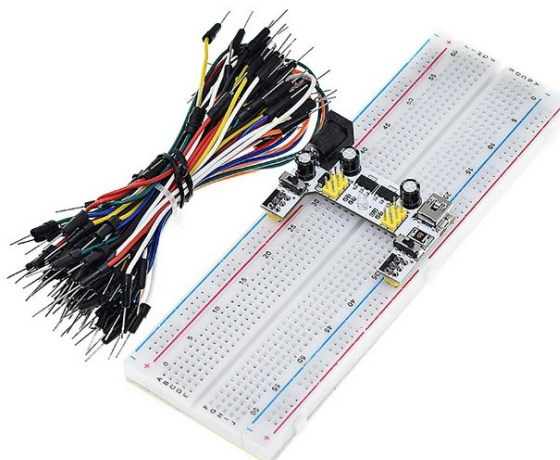


Figura 29 - Alimentador MB-102, Breadboard e conectores

Para estabelecer a ligação entre a *breadboard* e o Arduino, são utilizados conectores macho-macho. Um fio conector é inserido na porta de 5V do Arduino, fornecendo a alimentação necessária para todos os componentes, incluindo o microcontrolador, enquanto outro é ligado à porta terra (GND), garantindo uma conexão adequada entre o Arduino e a *breadboard*.

A alimentação para os sensores é fornecida diretamente pela *breadboard*, através de fios de cobre presentes nos cabos de rede, vulgarmente utilizados nas telecomunicações.

Por sua vez, os dados gerados pelos sensores são transmitidos ao Arduino através dos conectores de dados. Cada sensor possui um conector específico, que é ligado diretamente às portas correspondentes do Arduino, igualmente por meio de cabo de rede, à semelhança da alimentação (5V e GND). Dessa forma, os dados recolhidos pelos sensores podem ser processados e utilizados pelo microcontrolador para tomar decisões e executar as ações adequadas.

Esta configuração de alimentação e conexão dos componentes é essencial para garantir o correto funcionamento do projeto de domótica implementado na piscina municipal de Vouzela, permitindo a interação entre os sensores, os microcontroladores, presentes nas placas Arduino, e os atuadores.

No capítulo “Trabalho Experimental” serão descritos os principais passos durante a montagem do sistema de domótica, com registo de imagens reais, obtidas durante o processo.

Com a conclusão deste subcapítulo sobre o hardware utilizado no projeto, fica evidente a importância dos componentes Arduino Uno, Arduino Mega e ESP8266-01S, juntamente com os diversos sensores e atuadores conectados a eles, através de placas *breadboards* e fios de cobre. No próximo subcapítulo, será apresentado o conjunto de ferramentas de desenvolvimento de software utilizadas, bem como as tecnologias utilizadas na programação, para implementar a lógica e a interface de apresentação dos dados do sistema de domótica.

3.2 Ferramentas e tecnologias

Neste subcapítulo abordaremos as ferramentas e tecnologias fundamentais que foram utilizadas no projeto de domótica implementado na piscina municipal de Vouzela. Enquanto o subcapítulo anterior se focou no hardware, este subcapítulo destaca a importância do software como uma parte essencial do projeto. Ao longo dos subcapítulos seguintes, iremos abordar cada uma das ferramentas e tecnologias associadas, descrevendo as suas funcionalidades e contribuições para o projeto. A linguagem Python será explorada no capítulo “Análise Exploratória dos Dados”.

O desenvolvimento de software desempenha um papel crucial na implementação da automação residencial na piscina municipal de Vouzela. Diversas ferramentas foram utilizadas para facilitar o processo de programação e integração dos componentes do sistema. A seguir, apresentaremos as principais ferramentas utilizadas.

Fritzing: O Fritzing (Fritzing, s.d.) é uma ferramenta de software amplamente utilizada no campo da eletrónica e prototipagem, que permite criar esquemas de circuitos de forma visual e intuitiva. Este software desempenha um papel fundamental no projeto de domótica implementado na piscina municipal de Vouzela, pois foi utilizado para desenhar as ligações entre as placas Arduino, sensores e atuadores, para o espaço da Receção (Figura 61-Anexo B), Cais da Piscina (Figura 62 – Anexo C) e Casa das Máquinas, onde, neste caso, podemos incluir a divisão da caldeira (Figura 63-Anexo D).

Com o Fritzing é possível criar representações gráficas de componentes eletrónicos, incluindo os diversos modelos de placas Arduino, sensores, atuadores e outros dispositivos. Esses componentes são organizados num ambiente de trabalho, onde é possível conectá-los e estabelecer as ligações necessárias para o funcionamento correto do circuito.

No contexto deste projeto de domótica, o Fritzing foi utilizado para representar as placas Arduino Uno e Mega, bem como os sensores, atuadores e outros componentes utilizados. Esta aplicação permitiu visualizar as ligações entre esses elementos, garantindo a correta conexão e

interação entre as partes. Estes diagramas foram importantes para garantir a correta montagem e instalação dos componentes, bem como para orientar eventuais futuras manutenções e modificações no sistema.

Arduino IDE: A IDE Arduino é uma plataforma de desenvolvimento que permite a programação dos microcontroladores Arduino e ESP8266, através da tecnologia C, como linguagem de programação (Figura 30). Disponível de forma livre, no site oficial Arduino (ARDUINO, s.d.), apresenta-se com uma interface intuitiva e recursos específicos para a linguagem Arduino. Esta ferramenta é fundamental para a criação e carregamento dos programas nos microcontroladores, acoplados nas placas Arduino, utilizados no projeto.

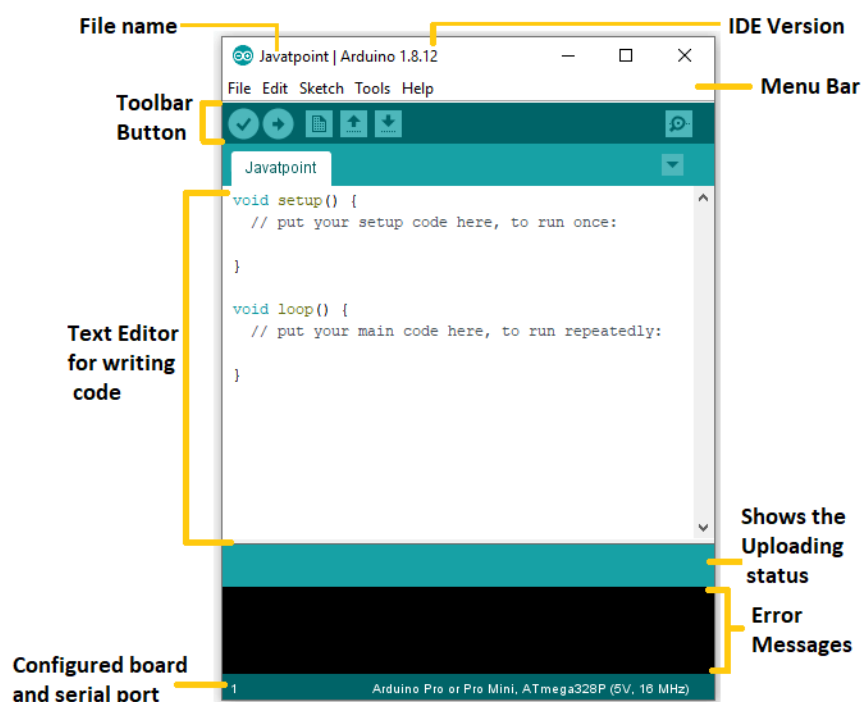


Figura 30 - Arduino IDE

PHP Storm: O PHP Storm (JET BRAINS, s.d.) é uma poderosa IDE voltada para o desenvolvimento de aplicações Web, disponível gratuitamente para a comunidade acadêmica. Neste projeto de domótica, o software desempenha um papel essencial no desenvolvimento da API (*Application Programming Interface*), responsável por conectar o hardware à base de dados e enviar os dados obtidos pelos sensores para a aplicação mobile, em tempo real. Com recursos avançados de depuração e integração com *frameworks* PHP, o PHP Storm oferece uma experiência de desenvolvimento eficiente e produtiva. Um exemplo de *framework* utilizada é a PHPMailer (PHP Mailer, s.d.), que permite o envio de emails, de forma automática, quando

cumpridas as condições. Estas condições serão explicadas e exemplificadas no capítulo “Trabalho Experimental”.

Esta interface (API) foi desenvolvida com a tecnologia PHP (**PHP: *H*ypertext *P*reprocessor**), como a linguagem de programação usada na lógica implementada no servidor, com sistema operativo Ubuntu, adquirido para o efeito.

Android Studio: O *Android Studio* (GOOGLE, s.d.) é a principal IDE utilizada para o desenvolvimento de aplicações móveis para dispositivos com sistema operativo Android, software proprietário da Google. Neste contexto, a ferramenta é utilizada na criação da aplicação Android que apresenta os dados armazenados na base de dados, proporcionando uma interface amigável e intuitiva para os utilizadores. Com ferramentas de design, emulação e depuração, o *Android Studio* possibilita a criação de aplicações Android robustas e eficientes.

Grafana: A aplicação web Grafana (GRAFANA, s.d.) é uma poderosa ferramenta de visualização e monitorização de dados. No contexto do projeto de domótica implementado na piscina municipal de Vouzela, o Grafana desempenha um papel fundamental na apresentação dos dados obtidos pelos sensores. Através do Grafana, é possível criar *dashboards* personalizadas, gráficos interativos e tabelas informativas, permitindo uma visualização clara e intuitiva das informações relacionadas, por exemplo, ao ambiente do Cais da Piscina. Além disso, o Grafana oferece recursos avançados de personalização, permitindo que os responsáveis configurem alertas, filtrem os dados, gerem relatórios e realizem análises estatísticas. No capítulo “Trabalho Experimental” será demonstrado o uso desta aplicação.

Ao longo dos próximos subcapítulos, descreveremos, cada uma dessas ferramentas e tecnologias associadas, destacando algumas das suas funcionalidades específicas e como foram aplicadas para alcançar os objetivos deste projeto.

3.2.1 Arduino IDE e a programação dos microcontroladores

O Arduino IDE é uma ferramenta de desenvolvimento amplamente utilizada para programar e configurar os microcontroladores Arduino, incluindo os modelos Arduino Uno, Mega e ESP8266-01 utilizados neste projeto. Este software desempenha um papel fundamental ao permitir a criação da lógica de funcionamento destes dispositivos, controlando as interações entre sensores, atuadores e outras partes do sistema.

O principal objetivo do Arduino IDE é fornecer um ambiente de programação simples e acessível, mesmo para aqueles sem experiência prévia em programação. A sua interface

intuitiva e amigável facilita o processo de desenvolvimento de software para os microcontroladores Arduino. Com o Arduino IDE, os programadores têm acesso a uma ampla gama de bibliotecas, exemplos de código e ferramentas úteis que ajudam na implementação de projetos eletrônicos.

O Arduino IDE foi utilizado para programar a lógica dos microcontroladores embutidos nos dispositivos Arduino Uno, Mega e ESP8266-01, através da linguagem de programação C. Este IDE permitiu o desenvolvimento do código necessário para a leitura dos sensores, o acionamento dos atuadores e a comunicação com outros componentes do sistema.

No capítulo "Trabalho Experimental", que detalhará os passos executados na programação de cada microcontrolador, serão abordados procedimentos como o reconhecimento de novas placas no Arduino IDE, utilizando o gestor de placas disponível na ferramenta. Um exemplo disso é o reconhecimento do ESP8266-01 como uma placa programável no IDE, permitindo a sua utilização como parte integrante do projeto.

Além disso, durante o desenvolvimento do projeto, foram adicionadas bibliotecas recomendadas ao editor do Arduino IDE, para facilitar o trabalho com os diversos sensores conectados. Essas bibliotecas fornecem funções pré-programadas que simplificam a leitura e o processamento dos dados provenientes dos sensores, agilizando o desenvolvimento e garantindo maior confiabilidade no funcionamento do sistema.

Assim, o Arduino IDE desempenha um papel essencial neste projeto, permitindo a criação da lógica dos microcontroladores Arduino e ESP8266-01, bem como a integração de bibliotecas e exemplos para a gestão dos sensores e atuadores, com um estudo mais aprofundado no capítulo "Trabalho Experimental".

3.2.2 PHP Storm no desenvolvimento da API

Nesta secção abordaremos a utilização do PHP Storm como ferramenta de desenvolvimento no projeto de domótica implementado na piscina municipal de Vouzela. Em particular, destacaremos o desenvolvimento da API, que desempenha um papel fundamental na interligação do hardware com a base de dados e na conexão com a aplicação móvel Vouzela IoT - Pool².

² Disponível na loja Google (Play Store) em <https://play.google.com/store/apps/details?id=com.monitoringpm>

O PHP Storm é uma poderosa IDE voltada para o desenvolvimento de aplicações, através da tecnologia PHP, como linguagem de programação. Neste projeto utilizamos o PHP Storm para a criação da API, uma interface de programação de aplicações web que permite a comunicação e o intercâmbio de dados entre diferentes sistemas.

A API, desenvolvida na ferramenta PHP Storm, tem como principal função processar os dados enviados, através da Internet, pelos microcontroladores ESP8266-01, conetados aos Arduino Mega e Uno, e armazená-los numa base de dados de séries temporais, InfluxDB. Esta base de dados permite o armazenamento e a organização dos dados recolhidos, permitindo um histórico de informações relevantes para análise e tomada de decisões futuras.

Além disso, a API estabelece a conexão com a aplicação móvel Vouzela IoT - Pool, desenvolvida para dispositivos Android. Através dessa conexão, a aplicação móvel é capaz de ler diretamente os dados armazenados na base de dados de séries temporais e apresentá-los na sua interface. Isso proporciona aos utilizadores uma visualização em tempo real das informações relacionadas ao sistema de domótica da piscina municipal, como temperatura, humidade, deteção de movimento, estado da porta, fuga de gás, e outros dados relevantes.

A API, desenvolvida no PHP Storm, também possui a capacidade de interpretar os dados recebidos e acionar alertas através de email e notificações na própria aplicação Android. Esses alertas são acionados com base em condições predefinidas, que podem ser configuradas de acordo com as necessidades específicas do projeto. Dessa forma, é possível tomar ações rápidas e eficientes com base nos dados recolhidos, contribuindo para a segurança, o conforto e a eficiência energética do sistema de domótica implementado.

As condições de acionamento de alertas, bem como o armazenamento e a segurança dos dados, serão detalhados no capítulo "Trabalho Experimental".

O PHP Storm mostrou-se uma ferramenta fundamental para o desenvolvimento da API no projeto de domótica da piscina municipal de Vouzela. Através dessa ferramenta, foi possível criar uma interface de comunicação eficiente entre o hardware, a base de dados e a aplicação móvel.

3.2.3 InfluxDB como escolha no armazenamento dos dados

Nesta secção, iremos destacar a importância do InfluxDB (InfluxDB is the time series platform for developers, s.d.). Esta ferramenta é um motor de base de dados otimizado para aplicações IoT, concretamente no contexto do projeto de domótica implementado na piscina municipal de

Vouzela. Também abordaremos o papel do PHP Storm na relação com o InfluxDB, ressaltando que todos os detalhes e procedimentos relacionados à base de dados serão apresentados em maior profundidade no capítulo "Trabalho Experimental".

O InfluxDB é um sistema, não relacional, de gestão de base de dados de séries temporais (*Time Series Database Management System*) que foi selecionado como o motor de base de dados para o projeto de domótica em questão. Esta ferramenta foi escolhida devido à sua eficiência e capacidade de lidar com grandes volumes de dados que são gerados em aplicações IoT (Naqvi, 2017).

Como estamos a trabalhar com um projeto IoT, é essencial ter um sistema de gestão de base de dados que seja capaz de armazenar, processar e consultar dados temporais de forma eficiente. O InfluxDB destaca-se nesse sentido, oferecendo recursos específicos para lidar com dados de séries temporais, como a capacidade de inserir e consultar dados com taxas consideravelmente elevadas, comparativamente a outros motores de bases de dados (Naqvi, 2017).

A integração do InfluxDB na API, desenvolvida no PHP Storm, é fundamental para a correta manipulação dos dados obtidos pelos microcontroladores e a inserção na base de dados, através da tecnologia SQL (*Structured Query Language*). Através da API, desenvolvida na tecnologia PHP, é possível estabelecer a conexão com o InfluxDB e utilizar as suas funcionalidades para inserir os dados recebidos dos sensores nas devidas medidas ("tabelas") na base de dados.

No capítulo "Trabalho Experimental", será abordado, com maior detalhe, todo o processo envolvido na criação e configuração da base de dados InfluxDB, através da API, bem como a implementação das consultas necessárias para obter as informações relevantes para o projeto de domótica.

Em suma, o InfluxDB, como um motor de base de dados de séries temporais voltado para aplicações IoT (InfluxDB is the time series platform for developers, s.d.), desempenha um papel crucial no projeto de domótica da piscina municipal de Vouzela. A sua eficiência e capacidade de lidar com dados temporais tornam-no uma escolha adequada para o armazenamento e processamento dos dados provenientes dos sensores.

3.2.4 Android Studio na conceção da aplicação mobile

Nesta secção abordaremos o papel fundamental do Android Studio no desenvolvimento da aplicação Vouzela IoT - Pool. Também destacaremos algumas características específicas da aplicação, como a conexão com o servidor, o uso da biblioteca *OkHttp* (OkHttp, s.d.) para

comunicação sobre o HTTP, a ativação de alertas via serviço *OneSignal* e a apresentação de gráficos utilizando a biblioteca *MPAndroidChart* (MPAndroidChart, s.d.).

O Android Studio é o ambiente de desenvolvimento oficial para a criação de aplicações Android. Este software fornece uma ampla gama de recursos que facilitam a programação, depuração e teste de aplicações móveis e oferece suporte completo para a linguagem de programação Java, tecnologia utilizada no desenvolvimento da aplicação Vouzela IoT - Pool.

Java é uma tecnologia amplamente utilizada no desenvolvimento de aplicações Android. Possui uma sintaxe clara e orientada a objetos, facilitando a criação de aplicações robustas e eficientes. Foi usada neste projeto, aproveitando os recursos da linguagem para criar uma experiência de utilizador fluida e interativa.

Para estabelecer a conexão com o servidor e obter os dados do sistema de domótica, foi criada uma classe de ligação ao servidor no Android Studio, onde é declarado o caminho (hiperligação) da API desenvolvida no servidor. Essa classe utiliza a biblioteca *OkHttp*, uma tecnologia que oferece recursos eficientes para realizar requisições HTTP, possibilitando a comunicação com a API desenvolvida para o efeito. Através do método GET, os dados são obtidos do servidor e utilizados para atualizar as informações na interface da aplicação.

A aplicação Vouzela IoT - Pool também oferece a funcionalidade de ativação de alertas para os utilizadores. Esses alertas são enviados através do serviço *OneSignal*, que permite o envio de notificações *push* para dispositivos móveis. Embora a configuração do horário das notificações, para os diferentes parâmetros, seja configurada na API (servidor), a aplicação está pronta para receber essas notificações e apresentá-las aos utilizadores de acordo com as condições estabelecidas.

Uma característica importante da aplicação Vouzela IoT - Pool é a apresentação visual da variação de valores ao longo do tempo. Para isso, foi utilizada a biblioteca *MPAndroidChart*, que permite criar e exibir gráficos interativos na interface da aplicação. Esses gráficos são gerados a partir dos dados recolhidos e permitem aos responsáveis visualizar a evolução dos valores nas últimas doze horas, proporcionando uma compreensão clara das tendências e variações.

Em suma, o Android Studio desempenha um papel crucial no desenvolvimento da aplicação Vouzela IoT - Pool, oferecendo um ambiente completo e poderoso para a criação de aplicações Android.

3.3 Conclusão

Neste capítulo “Hardware, Ferramentas, Tecnologias e Serviços”, exploramos os elementos fundamentais do projeto de domótica implementado na piscina municipal. Começamos por abordar o hardware utilizado, destacando os dispositivos utilizados para monitorizar e controlar diferentes aspetos do ambiente. Em seguida, discutimos as ferramentas e tecnologias utilizadas no desenvolvimento da API, em PHP, que foi responsável pela comunicação e processamento dos dados provenientes dos sensores e atuadores, bem como da Arduino IDE para a programação dos microcontroladores através da linguagem C. Além disso, destacamos o uso de sistemas como o InfluxDB, que permitiram o armazenamento dos dados de forma eficiente. Por fim, foram abordadas outras ferramentas e tecnologias para o desenvolvimento da aplicação Android, como o Android Studio e Java.

O capítulo proporcionou uma visão abrangente sobre os componentes e recursos utilizados, demonstrando como as diferentes tecnologias e ferramentas se complementaram para criar um sistema de domótica eficiente e funcional na piscina municipal.

No capítulo seguinte, “Trabalho Experimental”, serão detalhados alguns pontos importantes do desenvolvimento da aplicação, bem como de todo o processo inerente a este projeto, fornecendo uma visão mais aprofundada das etapas e processos envolvidos.

4 Trabalho Experimental

O trabalho experimental retrata a componente prática do projeto. No capítulo anterior abordámos o hardware e apresentámos imagens reais dos componentes já instalados nas divisões da piscina municipal, e neste capítulo iremos detalhar o trabalho experimental realizado no projeto de domótica implementado na piscina municipal de Vouzela. Abordaremos o processo, desde a seleção e aquisição do hardware necessário, como placas Arduino, módulos WiFi, sensores, atuadores e outros componentes de conexão e alimentação, como *breadboards*, cabos de rede e conectores, até à apresentação dos dados na aplicação Vouzela IoT – Pool.

O capítulo também descreverá a configuração do servidor, com o sistema operativo Ubuntu, onde a API foi desenvolvida com recursos à linguagem PHP. Esse servidor suportará também o sistema de base de dados InfluxDB, responsável pelo armazenamento dos dados do projeto. Serão abordadas as etapas de utilização das ferramentas necessárias para a programação dos microcontroladores, como a Arduino IDE, bem como o desenvolvimento da API, integrada no servidor por meio da ferramenta PHP Storm. Além disso, serão apresentadas algumas configurações iniciais no motor de base de dados InfluxDB, como a definição da política de retenção de dados.

Outro ponto importante a ser abordado é a utilização da ferramenta Android Studio para o desenvolvimento da aplicação móvel Vouzela IoT - Pool. Serão apresentados os passos para o desenvolvimento das principais funcionalidades da aplicação mobile, assim como a conexão com a API para o acesso e apresentação dos dados na interface da aplicação.

Por último será demonstrado o processo para apresentação dos dados na aplicação web Grafana, onde serão configuradas visualizações para exploração dos dados no formato de gráfico e tabela, semelhante ao apresentado na aplicação Vouzela IoT – Pool.

Ao longo das secções seguintes, serão detalhadas as etapas executadas no trabalho experimental, proporcionando uma visão abrangente do processo de implementação do projeto de domótica.

4.1 Codificação e montagem do hardware em fase de prototipagem

Após a identificação das necessidades de avaliação e correção nas instalações, bem como a aquisição do hardware necessário e a instalação dos softwares correspondentes, deu-se início à etapa de codificação e montagem do hardware.

A prototipagem da montagem do hardware é uma etapa fundamental no desenvolvimento do projeto de domótica. Esta fase envolve a criação de protótipos funcionais que permitem testar e validar as funcionalidades do sistema antes da implementação definitiva. Neste contexto, a montagem do hardware é realizada de forma temporária e flexível, permitindo ajustes e correções conforme necessário. A prototipagem permite identificar possíveis problemas e aprimorar o design do sistema, garantindo uma implementação final mais eficiente e eficaz.

Nesta fase, cada espaço físico descrito no capítulo "Hardware, Ferramentas e Tecnologias" foi cuidadosamente analisado. Com base nos requisitos específicos de cada espaço, os microcontroladores Arduino Uno, Arduino Mega e ESP8266-01 foram programados utilizando a linguagem C e a ferramenta Arduino IDE.

Os sensores e atuadores foram conectados aos microcontroladores de acordo com as necessidades de cada local. A montagem do hardware envolveu a utilização de *breadboards*, conectores e outros componentes de ligação e alimentação, por meio de transformadores com tensão de 5 volts em corrente contínua (DC). Para garantir uma conexão adequada, foram realizados testes de integração para verificar o funcionamento correto de cada dispositivo.

No que diz respeito ao software, concretamente ao Arduino IDE, foram realizadas configurações adicionais para reconhecimento de novas placas, concretamente para o microcontrolador ESP8266-01, por meio do gestor de placas no Arduino IDE (Figura 31). Também foram adicionadas bibliotecas recomendadas para facilitar o trabalho com os sensores (Figura 32).

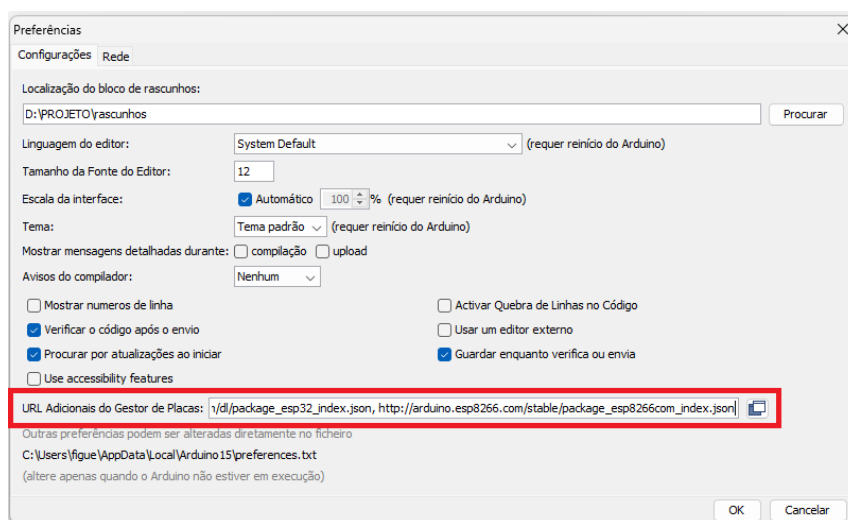


Figura 31 - Adição de repositório de placas ESP8266

A instalação de novas placas no IDE irá permitir o reconhecimento da ESP8266-01, como um módulo WiFi que terá a função de ligar os Arduinos à Internet, para envio, processamento e armazenamento dos dados num servidor externo, através da Internet.

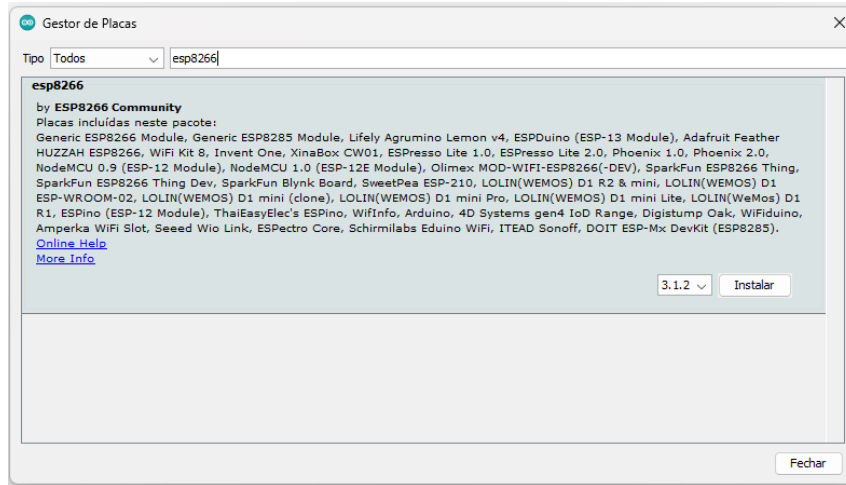


Figura 32 - Instalação de novas placas, concretamente ESP8266

4.1.1 Atualização do *firmware* da placa ESP8266-01

Esta secção descreve o processo da atualização do *firmware* da placa ESP8266-01, utilizando apenas um Arduino Uno e respetivo hardware necessário, como ilustra a Figura 33, para que este módulo WiFi fique funcional a permitir ligação à Internet.

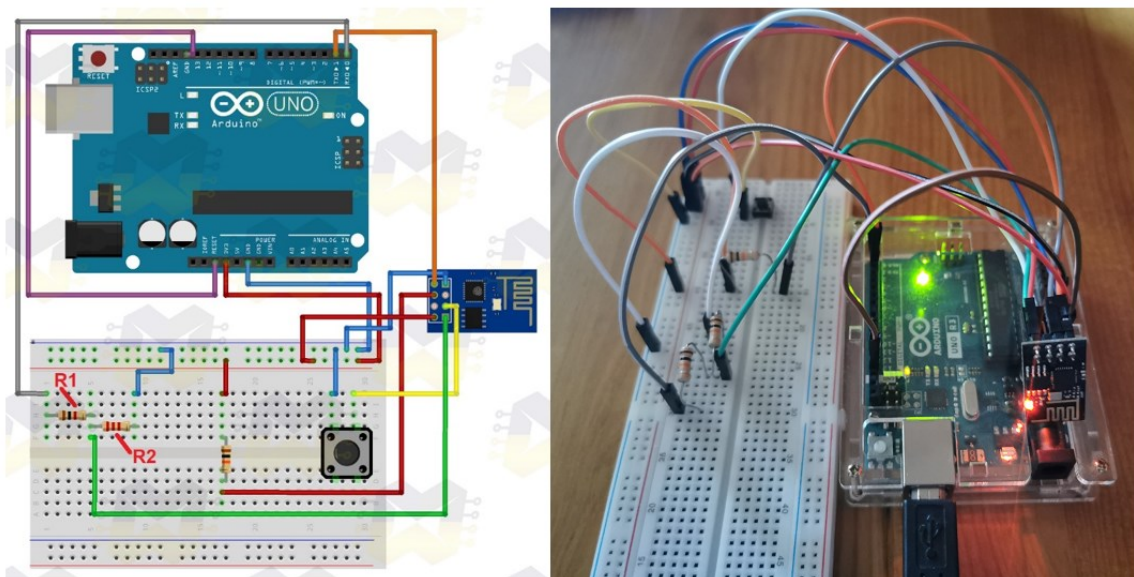


Figura 33 - Esquema de ligação do ESP8266-01 ao computador, através do Arduino

Os principais passos, após as ligações, para essa atualização, de acordo com Oliveira (s.d.), foram os seguintes:

- 1) *Download* e descompactação da ferramenta de atualização e respetivo *firmware*;
- 2) Manter pressionado o botão, que se encontra na *breadboard*, e conetar, via USB, o Arduino ao computador, de forma a colocar o microcontrolador ESP8266-01 em modo de gravação (após ligar o Arduino ao computador, já não é necessário manter o botão pressionado);
- 3) Aceder ao “Painel de Controlo” do Windows, “Gestor de Dispositivos”, e na opção “Portas (COM e LPT)”, verificar qual a porta COM atribuída ao Arduino;
- 4) Executar o software de atualização, descarregado anteriormente, selecionar o ficheiro de *firmware*, a carregar no ESP8266, e definir a porta atribuída ao Arduino, bem como outras opções como mostrado na Figura 34.

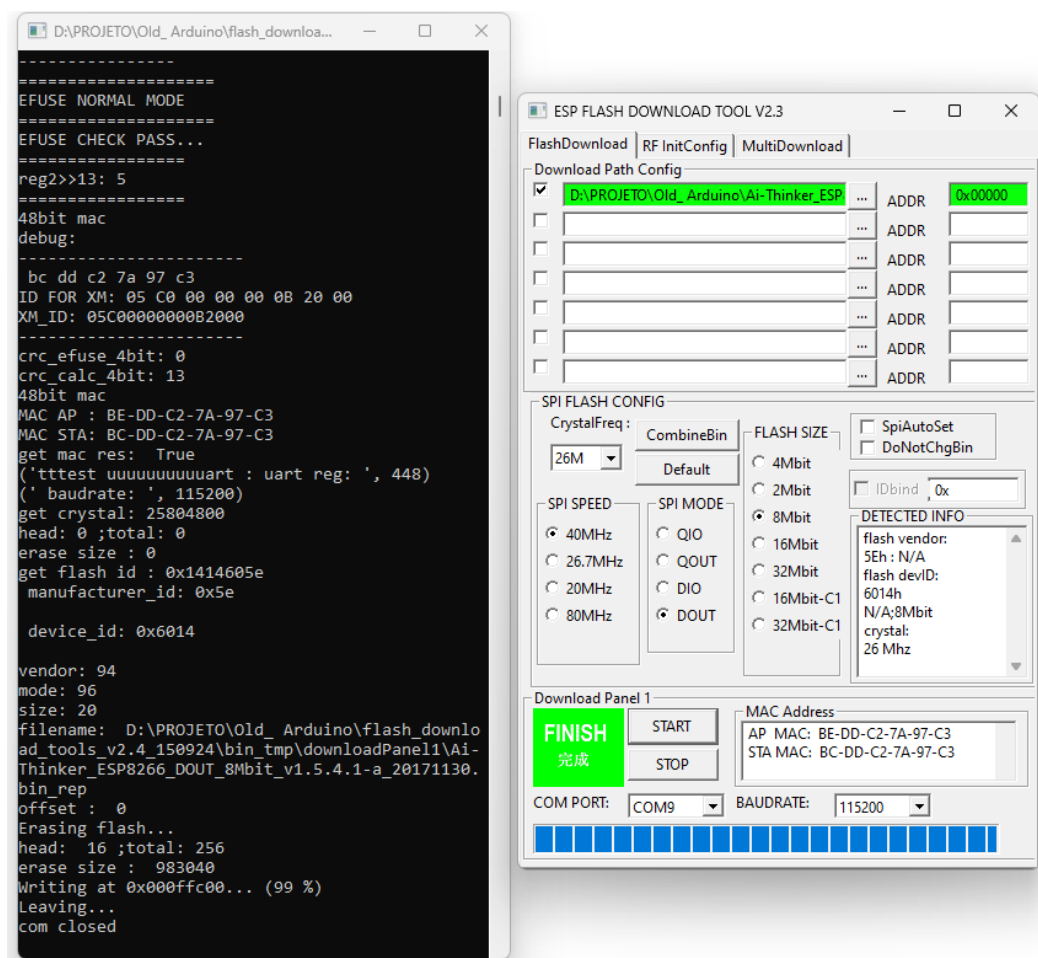


Figura 34 - Processo de atualização do firmware no ESP8266-01

Esses passos foram seguidos para atualizar o *firmware* da placa ESP8266-01, para permitir ativar a função de ligar os Arduinos à Internet. Para verificarmos se a atualização foi executada com sucesso, devemos, através do “monitor série” do IDE do Arduino, executar o comando “AT+GMR” de forma a confirmarmos a versão do atual *firmware*. Caso a informação apresentada corresponda à Figura 35, significa que a atualização finalizou corretamente.



Figura 35 - Versão atualizada do *firmware* instalado no ESP8266-01

Desta forma, o projeto de domótica implementado na piscina municipal de Vouzela é capaz de estabelecer a conexão dos Arduinos à Internet, possibilitando a monitorização e controlo remoto das funcionalidades do sistema.

Durante esta fase inicial, foram implementadas lógicas de controlo e comunicação entre os componentes. Os sensores foram configurados para fornecer leituras precisas de temperatura, humidade, movimento, deteção de gás, estado da porta (aberta ou fechada) e luminosidade, enquanto os atuadores foram programados para realizar ações específicas, como acionar ligar/desligar dispositivos elétricos. Na Figura 36 podemos verificar um dos protótipos, de forma a testar a receção dos valores, apresentados num ecrã LCD, de alguns sensores, bem como o comportamento de um atuador.

A fase de prototipagem foi essencial para testar a funcionalidade e a interação entre o hardware e o software, bem como estudar o processamento e armazenamento dos dados num servidor externo. Durante todo o processo, foram registados os desafios encontrados, ajustes realizados e soluções implementadas para garantir o correto funcionamento do sistema.

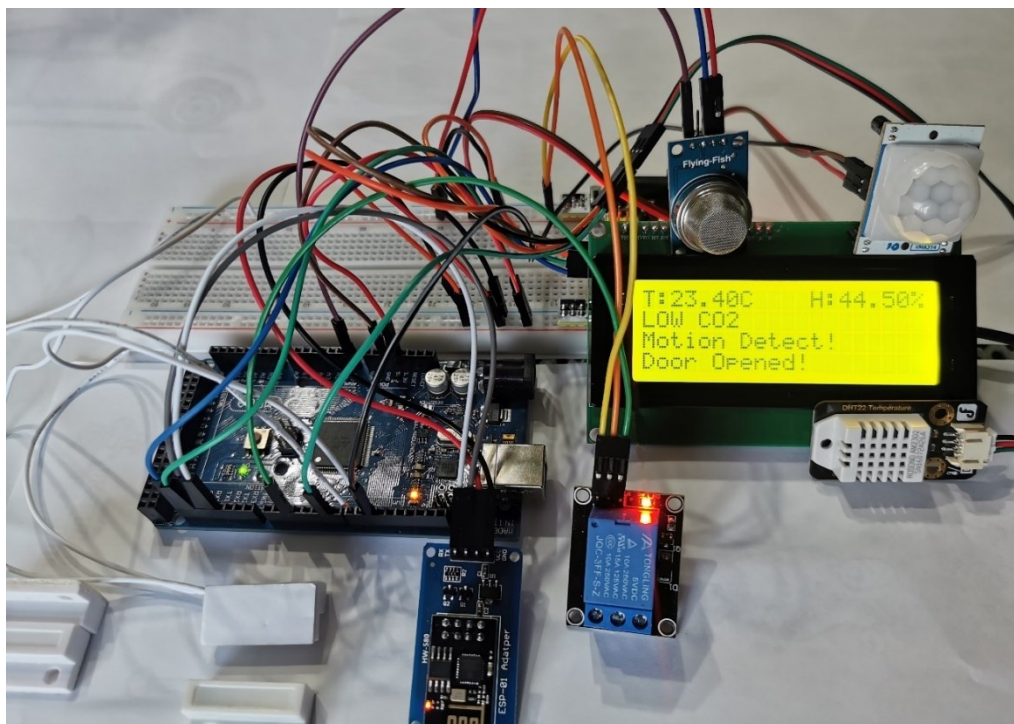


Figura 36 - Conexão e teste entre os alguns componentes de hardware

Na próxima secção detalharemos os procedimentos executados no servidor externo, com sistema operativo Ubuntu, para o desenvolvimento da API, na linguagem PHP, e base de dados, bem como demonstrar o fluxo de dados desde os Arduinos até à aplicação Android, a fim de fornecer uma visão mais aprofundada do trabalho experimental realizado.

4.2 Servidor e Web API

Nesta secção iremos explorar o trabalho realizado no servidor, com sistema operativo Ubuntu (Ubuntu, s.d.). Foi neste ambiente que desenvolvemos a API em PHP, responsável por processar os dados enviados pelos microcontroladores Arduino, através da Internet, utilizando o módulo ESP8266-01.

Além disso, abordaremos em detalhe a instalação e configuração da base de dados da plataforma InfluxDB, que permite guardar os dados processados pela API de forma eficiente para séries temporais. Apresentaremos, também, os passos realizados para configurar o servidor, desenvolver a API em PHP e estabelecer a conexão com o InfluxDB. Discutiremos as configurações iniciais, a definição da política de retenção de dados e outros aspetos importantes para garantir o bom funcionamento do sistema.

No contexto deste projeto de domótica, de forma a ficar mais claro o fluxo de dados, é importante realçar que, embora esta secção esteja mais direcionada para o trabalho realizado no

desenvolvimento da API, concretamente o processamento dos dados, é essencial mostrar a origem dos dados, ou seja, de onde vêm os dados recebidos pelo servidor, bem como o objetivo dos dados processados, isto é, qual o seu destino. Assim, para uma melhor compreensão, também serão demonstradas algumas configurações no Arduino e no Android.

4.2.1 Configurações iniciais no servidor

Para dar início ao desenvolvimento da API em PHP, foram realizadas algumas configurações iniciais no servidor com sistema operativo Ubuntu, versão 20.04, adquirido especificamente para este projeto, por meio de um *Virtual Private Server* (VPS). Cada VPS possui recursos dedicados, como memória, espaço em disco e largura de banda, e funciona como um servidor autónomo. As configurações realizadas, aquando da aquisição do servidor, visam garantir um ambiente adequado e seguro para o desenvolvimento e execução da API. A seguir, descreveremos os passos realizados:

Atualização do sistema operativo

O primeiro passo foi atualizar o sistema operativo Ubuntu, para garantir que o servidor está com as versões mais recentes dos pacotes.

Configuração do *Firewall*

Em seguida foi realizada a configuração da *firewall* no servidor. No Ubuntu 20.04, a *firewall* padrão é o UFW (*Uncomplicated Firewall*). É uma ferramenta simplificada para configurar e gerir regras de *firewall* no sistema. Por padrão, a *firewall* está desativada, mas para garantir a segurança e permitir o tráfego necessário, foram abertas as portas 22 (SSH), 80 (HTTP) e 443 (HTTPS), por meio de comandos “sudo ufw allow X”, sendo o “X” o número da porta correspondente. Essas portas permitem o acesso remoto ao servidor por meio do *Secure Shell* (SSH), um protocolo de rede que permite acesso remoto a sistemas e transferência segura de dados. Depois de definidas as regras, ativou-se a *firewall*, através do comando “sudo ufw enable”.

Instalação do servidor Web Apache

Para suportar a execução da API, foi instalado o servidor web Apache. O Apache é uma solução robusta e amplamente utilizada para hospedar *sites* e aplicações web nos sistemas Linux. Neste servidor web foi desenvolvida a API, cuja principal função é o processamento dos dados enviados pelos microcontroladores Arduino.

Instalação do PHP e Composer

Posteriormente à instalação do servidor web Apache, foi instalado o PHP, uma linguagem de programação amplamente utilizada para desenvolvimento web. O PHP permite a criação de *scripts* e aplicações dinâmicas que serão executadas pelo servidor. A ferramenta de gestão de dependências para PHP, *Composer*, também foi instalada. O Composer facilita a gestão e a instalação de bibliotecas e pacotes necessários para o desenvolvimento da API.

Instalação e configuração da base de dados InfluxDB

Para armazenar e gerir os dados processados pela API, foi instalado o InfluxDB, um sistema de base de dados otimizado para armazenar séries temporais. Este serviço permite o armazenamento eficiente de dados de sensores e facilita a consulta e análise desses dados posteriormente. Depois de instalado, através dos comandos disponibilizados na documentação oficial (Influxdata, s.d.), este sistema de base de dados, depois de instalado e ativado, estará pronto a armazenar os dados processados pela API.

É importante realçar que este sistema de base de dados, ao contrário de alguns motores de base de dados relacionais mais conhecidos, como, por exemplo, Microsoft SQL Server ou MySQL, não permitem a criação de tabelas sem dados. Concretamente, no InfluxDB, as tabelas são denominadas por medidas ou, corretamente, “*measurements*”. Por omissão, aquando da inserção de novos registos no sistema, é gerado, ao mesmo tempo, um novo campo, denominado de “*time*”, que se apresenta no formato segundo. Este campo é responsável por registrar a marca de tempo associada a cada ponto de dados armazenado na base de dados, permitindo a organização, consulta e análise dos dados com base na sua ordem cronológica.

Após instalação do sistema de base de dados, entramos neste sistema, através do comando “influx” e criamos a base de dados “vouzelaïot” para receber os dados processados pela API, através do comando “CREATE DATABASE vouzelaïot”. Outro passo importante, embora não obrigatório, é definir uma política de retenção de dados, de forma a garantir que os dados sejam guardados durante um período de tempo. No caso deste projeto, definiu-se um período de 30 dias, através dos comandos “CREATE RETENTION POLICY data30d ON vouzelaïot DURATION 30d REPLICATION 1” e “ALTER RETENTION POLICY data30d ON vouzelaïot DEFAULT”, para definir essa política como padrão. Na Figura 37 podemos verificar as medidas (“tabelas”) existentes, depois inseridos alguns dados (pontos) de forma automática, através da API.

```
ubuntu@vps-a0770c41:~$ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> use vouzelaiot
Using database vouzelaiot
> show measurements
name: measurements
name
----
piscinaiot_C3H8v1
piscinaiot_C3H8v2
piscinaiot_co2
piscinaiot_door
piscinaiot_humiditycais
piscinaiot_humidityrec
piscinaiot_ldrcm
piscinaiot_pir
piscinaiot_pircm
piscinaiot_relayC3H8State
piscinaiot_relayCo2State
piscinaiot_temperaturecais
piscinaiot_temperaturerec
>
```

Figura 37 - Medidas criadas na base de dados, através da API.

Instalação e configuração do software Fail2Ban

Por questões de segurança, também foi instalado o software Fail2Ban, através do comando “sudo apt-get install fail2ban” (Danischevsky & Brester, 2023). Esta ferramenta é responsável por proteger o servidor contra tentativas de acesso *não identificados* ou *não autorizados*. O Fail2Ban monitoriza os *logs* de autenticação e bloqueia temporariamente os endereços IP que excedem um determinado número de tentativas de login mal-sucedidas. Isso ajuda a proteger o servidor contra-ataques e aumenta a segurança geral do sistema. O serviço, depois de instalado, é automaticamente ativado. Podemos verificar o seu estado através do comando “sudo systemctl status fail2ban”, bem como conhecer o ficheiro de configuração deste software pelo comando “man fail2ban”.

Com todas estas configurações e softwares adicionais implementados, o servidor está pronto para o desenvolvimento da API. Estas etapas iniciais garantem um ambiente estável, seguro e adequado para a implementação e execução da API, proporcionando uma base sólida para o projeto de domótica na Piscina Municipal de Vouzela.

4.2.2 Web API: Fluxo de dados do Arduino ao Android

Uma API é uma interface de programação que permite a comunicação e troca de dados entre diferentes sistemas, nesse caso, entre a aplicação cliente e o servidor. A API define um conjunto de *endpoints* (pontos de acesso) que são chamados pelo cliente para executar determinadas ações ou obter informações específicas.

No contexto desta API, o funcionamento segue os seguintes passos para os dois clientes: o microcontrolador acoplado nas placas Arduino e a app Android. É importante afirmar que serão identificados e explicados os procedimentos principais no trabalho da API, no que respeita ao pedido de registo de alguns dados, por parte do cliente Arduino, bem como a consulta dos dados armazenados, pelo cliente Android. É, com base nestes clientes, que explicaremos o trabalho realizado pela API, tendo em conta o fluxo de dados. Também será explicada a função da API na execução das notificações, sejam por via de correio eletrónico ou *OneSignal*.

A Web API para o cliente Arduino

A título de exemplo, vamos explicar o fluxo de dados entre os clientes da API, no momento do envio dos dados, pelo Arduino, até à apresentação, em tempo real, dos mesmos no Android, depois de processados pela API:

- 1) O cliente (Arduino) faz uma requisição HTTP, pelo método “GET”, para um *endpoint* específico da API, através da classe *WiFiEspClient*, pertencente à biblioteca *WiFiEsp*; Na Figura 38 ilustramos essa requisição, contida numa função, de forma a enviar os dados da temperatura, humidade e CO2 do espaço da Receção para a API, para esta os processar e enviar para a base de dados *InfluxDB*;

```
void registerValues() { // function in loop
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  Serial.print("GET /piscina_iot.php?humidityrec="+String(humidityData)+"&temperaturerec="+String(temperatureData)+"&co2="+String(co2));
  client.print("GET /piscina_iot.php?humidityrec="+String(humidityData)+"&temperaturerec="+String(temperatureData)+"&co2="+String(co2));
  //client.print(url);
  client.print(" "); //SPACE BEFORE HTTP/1.1
  client.print("HTTP/1.1");
  client.println();
  client.println("Host: 51.68.121.218"); // server
  client.println("Connection: close");
  client.println();
  while(client.connected()) {
    while(client.available()) {
      Serial.write(client.read());
    }
  }
} else {
  // if you didn't get a connection to the server:
  Serial.println("Connection: fail...");
}
}
```

Figura 38 - Envio de dados do Arduino (Hall/Receção) para a API

- 2) No servidor, a API recebe a requisição e realiza o devido processamento, concretamente o armazenamento na base de dados, através da biblioteca *InfluxDB-PHP* (*InfluxDB-PHP*, s.d.), que estabelece uma conexão com a base de dados *InfluxDB*;
- 3) A API cria um objeto de ponto de dados (*InfluxDB\Point*) utilizando os dados enviados pelos Arduinos;

- 4) O objeto de ponto de dados contém informações como o nome da medida (“tabela”), os campos e o carimbo de tempo no formato segundo. À medida (“tabela”) também é adicionado um novo campo com a data e hora do sistema Linux, campo esse que será posteriormente utilizado na aplicação Android, aquando da apresentação dos dados em tempo real;
- 5) A API utiliza o método “writePoints”, da biblioteca InfluxDB-PHP, para escrever o ponto de dados na base de dados InfluxDB, criada anteriormente no servidor, pela linha de comandos.

Na Figura 39 podemos verificar parte do código para registo, na base de dados.

```
function registData($type,$value,$item){
    global $database_confs, $database;
    $point = new Point(
        measurement: $database_confs['table_name'].'_'.$type, // measurement name
        $value, // value
        ['type' => $type], // tags
        ['date' => date( format: 'Y-m-d H:i:s')],
        time() // timestamp
    );
    $regist=true;
    if($regist) {
        $database->writePoints([$point], precision: Database::PRECISION_SECONDS);
    }
}
```

Figura 39 - API regista de dados no InfluxDB

Informação em tempo real: Para o cliente Android

O cliente (app Android) faz uma requisição HTTP, pelo método “Request”, para um *endpoint* específico da API, pertencente à biblioteca OkHttp. Na Figura 40 podemos verificar esse pedido, concretamente no parâmetro “operation=singledata”, cuja função é devolver dados brutos, dentro de todas as medidas (“tabelas”) armazenados na base de dados;

```
OkHttpClient client = new OkHttpClient();
public static final String API_URL = "http://51.68.121.218/api/"; //api link
public void getSensorsData(Callback call) throws Exception {
    doGet(endpoint: "", parameters: "operation=singledata", call); // call endpoint web api
}
private void doGet(String endpoint, String parameters, Callback call) throws Exception {
    Request request = new Request.Builder()
        .url(API_URL+endpoint+"?" +parameters)
        .build();
    client.newCall(request).enqueue(call);
}
```

Figura 40 - App Android requisita à API

No servidor, a API recebe o pedido e realiza o devido processamento, concretamente uma consulta à base de dados, para devolver o último registo dos vários parâmetros contidos num

array denominado “accepted_fields”, como ilustrado na Figura 41. O “accepted_fields” é uma variável que contém o nome de todas as medidas (“tabelas”) existentes na base de dados.

```
header('header: Content-Type: application/json');
switch ($operation){
    case 'singledata':{
        foreach ($accepted_fields as $k=>$v) {
            $cons = "SELECT type,value FROM piscinaiot." . $k . " order by time desc limit 1";
            $result = $database->query($cons);
            $points = $result->getPoints();
            $points[0]['time']=date('format: 'H:i:s',strtotime($points[0]['time']));
            $response['data'][] = $points[0];
        }
        break;
    }
}
echo json_encode($response);
```

Figura 41 - API processa pedido de consulta do último registo pelo Android

Com base nessa requisição e no processamento realizado, a API devolve uma resposta HTTP ao cliente, por meio do formato JSON (*JavaScript Object Notation*), com os dados solicitados (Figura 42); JSON é uma formatação utilizada para estruturar dados em formato de texto e transmiti-los de um sistema para outro.



```
{
  code: "200",
  status: "1",
  message: "OK",
  data: [
    - {
      time: "15:30:15",
      type: "temperaturerec",
      value: 27.1
    },
    - {
      time: "15:30:15",
      type: "humidityrec",
      value: 73.6
    },
    - {
      time: "15:30:15",
      type: "co2",
      value: 392
    },
    + { - },
    + { - },
  ]
}
```

Figura 42 - API devolve JSON, de dados brutos, a pedido do Android

O cliente recebe a resposta no formato JSON, interpreta-a e converte-a num objeto Java, através do método “fromJson”, pertencente à biblioteca Gson (Gson, 2023), como representado na Figura 43, e apresenta os dados nas interfaces, como podemos verificar na Figura 44.

No seguimento dos passos acima explicados, demonstramos o fluxo dos dados relacionados com a temperatura, humidade relativa e dióxido de carbono (CO2), desde o envio destes pelo cliente Arduino, até à apresentação dos mesmos, em tempo real, no cliente Android.

```

public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
    try (ResponseBody responseBody = response.body()) {
        if (!response.isSuccessful())
            throw new IOException("Unexpected code " + response);
        ResponseServer resp = gson.fromJson(responseBody.string(), responseType);
        final ArrayList<SingleData> data=resp.data;
        try {
            getActivity().runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    if (data.get(0).value==null) {
                        txt_temperatura.setTextColor(RED);
                        txt_temperatura.setText("NO DATA");
                    } else {
                        txt_temperatura.setText(data.get(0).value);
                    }
                }
            });
        }
    }
}

```

Figura 43 - App Android interpreta JSON e envia dados para interface

De um modo geral, o fluxo de dados entre os clientes da API é semelhante em todos os processos que envolvam a apresentação de dados, variando, apenas, a representação dos mesmos na interface do Android, seja por meio de gráfico ou tabela.

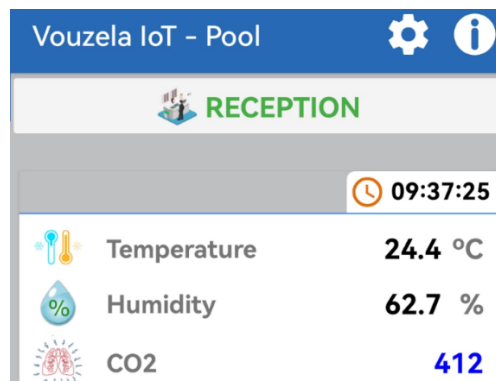


Figura 44 - App Android apresenta dados, em tempo real

Gráficos: Variação dos valores durante as últimas 12 horas.

Os dados obtidos nos dispositivos pelos sensores ligados às placas Arduino e processados no servidor, são enviados em tempo real para a aplicação Android através da API. Esse fluxo contínuo de dados permite que os responsáveis pelas instalações tenham acesso instantâneo às atualizações mais recentes sobre vários parâmetros dentro dos vários espaços da piscina municipal. Além disso, a aplicação oferece uma funcionalidade adicional que permite visualizar gráficos com a representação das últimas 12 horas de dados. Para isso, é utilizada a biblioteca MPAndroidChart, que facilita a criação de gráficos interativos e personalizáveis no Android.

A título de exemplo e de forma a explicarmos o fluxo de dados, desde o Arduino até à apresentação do gráfico no Android, iremos abordar o espaço do Cais da Piscina, onde foi utilizado um sensor DHT, modelo 22, da *DFRobot*, para a recolha dos dados. Nesse gráfico,

apresentado na aplicação, cada ponto representa a média dos dados registados a cada 10 minutos, garantindo que a leitura do gráfico não seja sobrecarregada por uma quantidade excessiva de informações. Dessa forma, os responsáveis podem observar facilmente as tendências de variação de temperatura e humidade relativa ao longo das 12 horas e obter uma perspetiva visual clara do ambiente no espaço do Cais da Piscina, de forma a monitorizar a evolução das condições da piscina e tomar medidas adequadas quando necessário.

O processo para o envio de dados, por parte do cliente Arduino, e registo na base de dados, é semelhante ao explicado anteriormente, quando demonstramos o fluxo de dados sobre a temperatura e humidade relativa no espaço da Receção, mudando apenas o nome dos parâmetros. Neste caso, outro Arduino instalado no espaço Casa das Máquinas envia os dados sobre a temperatura e humidade relativa no Cais da Piscina (tanque) para o servidor.

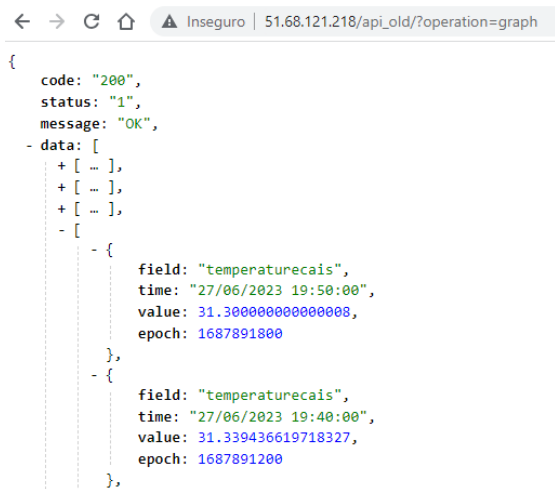
Há, agora, um novo processamento pela API, quando o cliente Android requisita os dados para serem apresentados graficamente. A API identifica o pedido do cliente Android e executa a função específica que devolve, em cada ponto, a média resultante a cada período de 10 minutos, ao longo das últimas 12 horas. Na Figura 45 podemos observar essa função na API.

```
switch ($operation){
  case 'graph':{
    foreach ($accepted_fields as $k=>$v) {
      if($v['graph']) { // '2015-08-18T00:00:00Z'
        $cons = "SELECT value FROM piscinaiot_ . $k . " where type= " . $k . " and time >= " . $date . " order by time desc";
        if($v['aggregation_minutes']>0) {
          $cons = "SELECT MEAN(value) as value FROM piscinaiot_ . $k . " where time >= " . $date . " and time
            <= " . $hora_atual . " GROUP BY time( " . $v['aggregation_minutes'] . " m) order by time desc";
        }
        $result = $database->query($cons);
        $points = $result->getPoints();
        $write=array();
        foreach ($points as $kk=>$vv){
          if($vv['value']!=null){
            $value=$vv['value'];
            $write[]=array(
              'field'=>$k,
              'time'=>date( format: 'd/m/Y H:i:s', strtotime($vv['time'])),
              'value'=>$value,
              'epoch'=>strtotime($vv['time'])
            );
          }
        }
        $response['data'][] = $write;
      }
    }
    break;
  }
}
```

Figura 45 - API processa pedido de consulta pelo Android para apresentação de gráficos

Com base nessa requisição e no processamento realizado, a API devolve uma resposta HTTP ao cliente, no formato JSON, com a média dos dados registados a cada 10 minutos, nas últimas 12 horas. Na Figura 46 encontram-se ilustrados dois de vários pontos devolvidos com a respetiva média. De salientar que, por imposição da biblioteca MPAndroidChart, se criou um

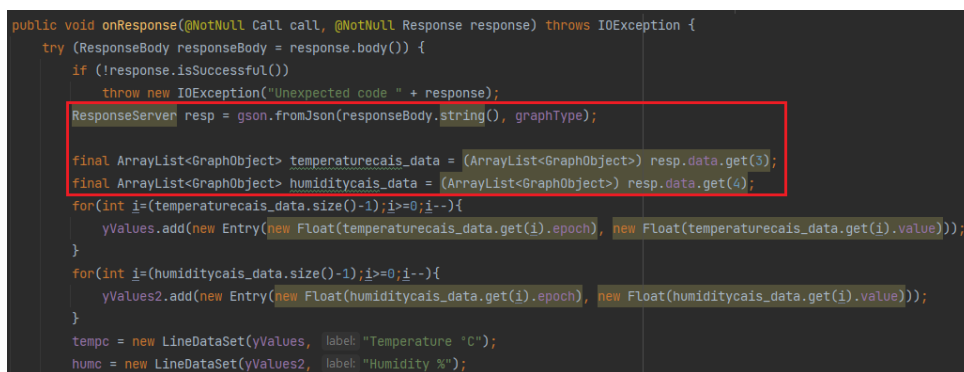
novo campo “epoch” de forma a registar o tempo, em milissegundos, para posterior tratamento aquando da representação gráfica dos dados na interface da aplicação Vouzela IoT – Pool.



```
{
  code: "200",
  status: "1",
  message: "OK",
  data: [
    + [ ... ],
    + [ ... ],
    + [ ... ],
    - [
      - {
        field: "temperaturecais",
        time: "27/06/2023 19:50:00",
        value: 31.300000000000008,
        epoch: 1687891800
      },
      - {
        field: "temperaturecais",
        time: "27/06/2023 19:40:00",
        value: 31.339436619718327,
        epoch: 1687891200
      }
    ]
  ]
}
```

Figura 46 - API devolve JSON, para gráficos, a pedido do Android

À semelhança da interpretação do Android, para os dados em tempo real, este cliente recebe a resposta no formato JSON, interpreta-a e converte-a em objetos Java, em forma de *array*, através do método “fromJson”, pertencente à biblioteca Gson (Gson, 2023), como representado na imagem Figura 47, e apresenta os dados graficamente, como podemos verificar na Figura 48.



```
public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
  try (ResponseBody responseBody = response.body()) {
    if (!response.isSuccessful())
      throw new IOException("Unexpected code " + response);
    ResponseServer resp = gson.fromJson(responseBody.toString(), graphType);

    final ArrayList<GraphObject> temperaturecais_data = (ArrayList<GraphObject>) resp.data.get(3);
    final ArrayList<GraphObject> humiditycais_data = (ArrayList<GraphObject>) resp.data.get(4);
    for(int i=(temperaturecais_data.size()-1);i>=0;i--){
      yValues.add(new Entry(new Float(temperaturecais_data.get(i).epoch), new Float(temperaturecais_data.get(i).value)));
    }
    for(int i=(humiditycais_data.size()-1);i>=0;i--){
      yValues2.add(new Entry(new Float(humiditycais_data.get(i).epoch), new Float(humiditycais_data.get(i).value)));
    }
    tempc = new LineDataSet(yValues, label: "Temperature °C");
    humc = new LineDataSet(yValues2, label: "Humidity %");
  }
}
```

Figura 47 - App Android interpreta JSON e envia dados processados para os arrays representados pelos eixos dos gráficos

No seguimento dos passos explicados acima, demonstramos o fluxo dos dados relacionados com a temperatura e humidade relativa, recolhidos no Cais da Piscina, desde o pedido à API até à apresentação dos mesmos, num gráfico personalizável, na aplicação Vouzela-IoT. Como já referido, o processo para o envio de dados, por parte do cliente Arduino, e registo na base de dados, é semelhante para todos os parâmetros abordados neste projeto de domótica. Esse processo já foi explicado no momento do envio e armazenamento dos dados em tempo real.

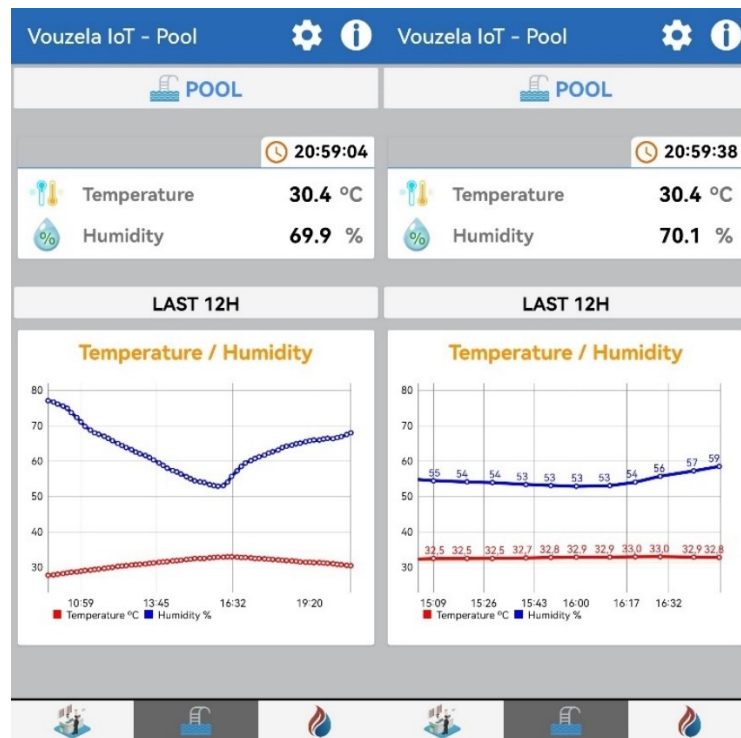


Figura 48 - App Android apresenta dados, em formato gráfico

Tabelas: Histórico dos últimos 10 eventos

A API também desempenha um papel fundamental na monitorização do estado do sensor magnético, que tem como objetivo verificar o estado da porta de acesso à Receção, bem como detetar qualquer movimento na área da Receção e da Casa das Máquinas. A API regista o estado dos relés que estão configurados, no Arduino, para responder, caso sejam detetados níveis elevados de dióxido de carbono no espaço da Receção e gás propano no espaço da caldeira. Sempre que ocorre uma alteração no estado do sensor magnético ou é detetado elevado nível de gás, a API regista essas informações na base de dados, fornecendo um histórico preciso desses eventos. Esses registos são valiosos para monitorizar a segurança e o desempenho dos sistemas instalados, permitindo uma intervenção rápida e apropriada sempre que necessário.

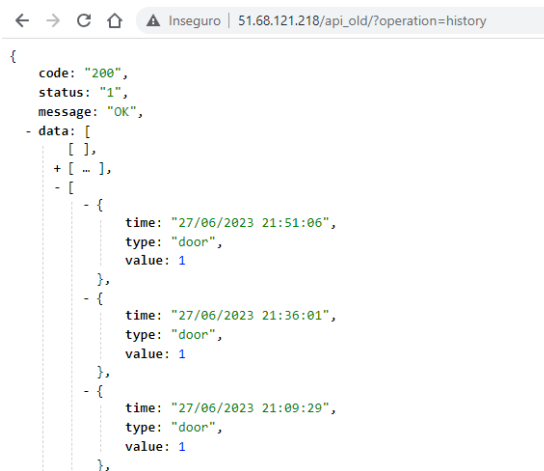
Tal como o processo para envio dos dados, sobre a temperatura e humidade, pelo Arduino e registo desses na base de dados, pela API, os dados relativos ao sensor magnético e de movimento (PIR), os dados sobre os atuadores relé, recebem o mesmo tratamento pela API, mudando, apenas, o nome dos parâmetros, que se traduzem no nome das medidas (“tabelas”) que recebem esses dados. A diferença está, agora, na forma como esses dados históricos são apresentados na aplicação Android. Estes dados são apresentados na forma de uma lista.

Na Figura 49 é ilustrada a função, na API, que identifica o pedido do histórico de dados, pela aplicação *Vouzela IoT - Pool*, cujo valor dos parâmetros é 1 (um), ou seja, a API processa o pedido da aplicação e devolve uma lista com os últimos 10 registos, cujo valor dos parâmetros seja 1, valor esse que se aplica aquando da abertura de porta, deteção de movimento ou fecho do relé, de acordo com a programação deste conjunto de hardware nos microcontroladores e ligação nas portas digitais das placas Arduino.

```
case 'history':{
  foreach ($accepted_fields as $k=>$v) {
    if(!$v['graph']) {
      $cons = "SELECT type, value FROM piscinaiot_1 . $k . " where value=1 ORDER BY time DESC LIMIT 10";
      $result = $database->query($cons);
      $points = $result->getPoints();
      foreach ($points as $kk=>$vv){
        $points[$kk]['time']=date( format: 'd/m/Y H:i:s', strtotime($vv['time']));
      }
      $response['data'][] = $points;
    }
  }
  break;
}
```

Figura 49 - API processa pedido de consulta pelo Android para apresentação de histórico

Com base nessa requisição e no processamento realizado, a API devolve uma resposta no formato JSON, com o histórico dos parâmetros solicitados, como podemos observar na Figura 50, que ilustra os três últimos registos de abertura de porta, na data de 27 de junho.



```
{
  code: "200",
  status: "1",
  message: "OK",
  data: [
    [ ],
    + [ - ],
    - [
      - {
        time: "27/06/2023 21:51:06",
        type: "door",
        value: 1
      },
      - {
        time: "27/06/2023 21:36:01",
        type: "door",
        value: 1
      },
      - {
        time: "27/06/2023 21:09:29",
        type: "door",
        value: 1
      },
    ],
  ],
}
```

Figura 50 - API devolve dados no formato JSON, para histórico, a pedido do Android

À semelhança da interpretação dos dados em tempo real, este cliente recebe a resposta no formato JSON, interpreta-a e converte-a em objetos Java, em forma de *array*, através do método “fromJson”, pertencente à biblioteca Gson (Gson, s.d.), como representado na Figura 51, e apresenta os dados, numa vista reciclada, como ilustrado na Figura 52. Uma vista reciclada (RecyclerView) é um componente do Android que fornece uma maneira eficiente de apresentar

grandes volumes de dados, através de uma lista. A RecyclerView (Alotaibi, Chiou, & Halfond, 2022; DEVELOPERS, s.d.) utiliza o padrão de design ViewHolder para reutilizar a interface dos eventos registados e que estão visíveis de imediato no ecrã, permitindo uma visualização suave e eficiente de novos eventos à medida que vamos navegando pela lista.

```
public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
    try (ResponseBody responseBody = response.body()) {
        if (!response.isSuccessful())
            throw new IOException("Unexpected code " + response);
        ResponseServer resp = gson.fromJson(responseBody.string(), responseTypeHistory);
        final ArrayList<ArrayList<SingleData>> data=resp.data;
        try {
            getActivity().runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    doorhist = new DoorDetailAdapter(getContext(), getActivity(), data.get(2));
                    historicoDoorRecyclerView.setAdapter(doorhist);
                    if(data.get(2).size()>0){
                        btn_expand_historicoDoor.callOnClick();
                    }
                }
            });
        }
    }
}
```

Figura 51 - App Android interpreta JSON e envia dados processados para os arrays que representam cada parâmetro

Assim, demonstramos o fluxo dos dados relacionados com o histórico de dados, concretamente do estado da porta, desde o pedido do Android à API, até à apresentação dos dados, numa lista, neste mesmo cliente. O processo para o envio de dados, por parte do cliente Arduino, e registo na base de dados, é semelhante para todos os parâmetros abordados neste projeto de domótica, já explicado no momento do envio e armazenamento dos dados em tempo real.

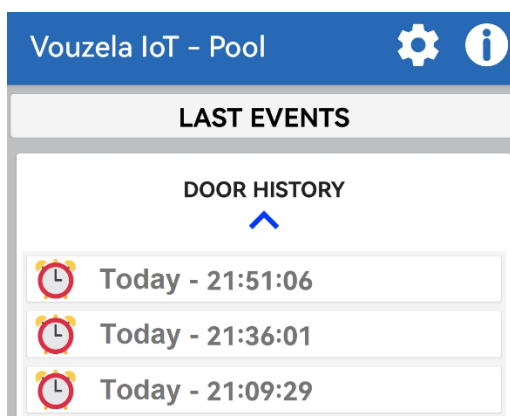


Figura 52 - App Android apresenta dados históricos, em formato lista

Notificações: OneSignal e correio eletrónico (e-mail)

A API também oferece um recurso de notificação personalizável, permitindo que os responsáveis pelas instalações da piscina municipal recebam informações relevantes de forma oportuna. A receção das notificações pode ser configurada, na aplicação, para cada parâmetro

monitorizado (Figura 53), assim como a definição de diferentes horários para esses parâmetros, na API. As notificações são recebidas via email ou através do serviço OneSignal.

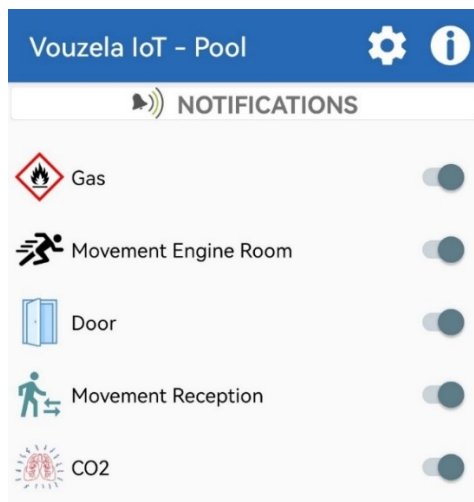


Figura 53 - App Android permite configurar notificações

O OneSignal é uma plataforma de envio de notificações *push* que permite o envio automático de mensagens para os utilizadores. Essas notificações *push* são alertas instantâneos apresentados nos dispositivos móveis e *browsers*, mesmo quando não estão a usar, de forma ativa, a aplicação móvel ou *site*. Através da documentação oficial deste serviço, foi implementada a receção de notificações na aplicação móvel Vouzela IoT – Pool (OneSignal, s.d.). A receção de notificações, seja por via do serviço OneSignal ou correio eletrónico, através da biblioteca PHPMailer (PHP Mailer, s.d.), poderá ser personalizada através da API, para determinadas condições, como, por exemplo, movimento detetado fora do horário laboral. A Figura 54 ilustra a receção de uma notificação de movimento detetado, por ambas as vias, após o horário laboral.

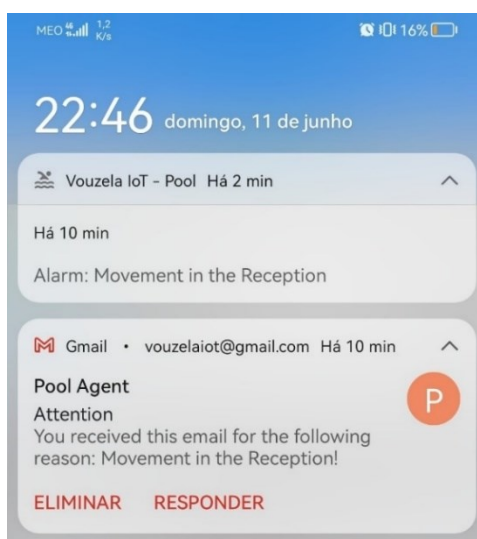


Figura 54 - App Android recebe notificações

4.3 Apresentação dos dados no Grafana

O Grafana desempenha um papel fundamental na visualização dos dados recolhidos pelo sistema de domótica implementado na piscina municipal. Esta plataforma de visualização de dados oferece recursos poderosos para criar painéis personalizados que mostram informações relevantes e em tempo real sobre o funcionamento do sistema.

O Grafana permite criar *dashboards* interativas e informativas que exibem gráficos, tabelas e métricas estatísticas relacionadas com os diversos aspetos do sistema de domótica. Além disso, é possível exibir dados históricos, como a evolução temporal dos parâmetros monitorizados, e também apresentar informações estatísticas, como médias, valores máximos e último registo. Estas *dashboards* permitem monitorizar de forma intuitiva e eficaz o desempenho do sistema, identificar possíveis problemas e tomar decisões para otimizar a sua operação. Nesta secção iremos demonstrar o uso do Grafana para a apresentação dos dados do sistema de domótica.

No momento do registo, com a conta de e-mail `vouzelaio@gmail.com`, é gerada automaticamente uma instância com o URL `vouzelaio.grafana.net`, que identifica a nossa estação de trabalho. Por omissão, o sistema InfluxDB já se encontra instalado, pelo que apenas temos de configurar uma conexão à base de dados. Caso a ligação seja realizada com sucesso, devolve o número de medidas (“tabelas”) encontradas na base de dados, como ilustrado na Figura 55.

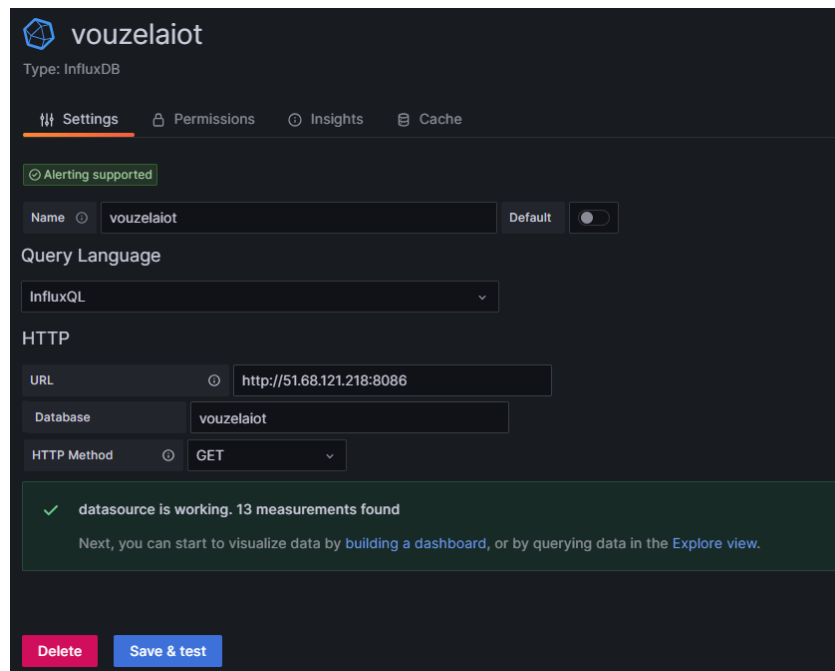


Figura 55 - Grafana na configuração da conexão

Realizada a conexão, a plataforma sugere a criação de *dashboards*, através do desenvolvimento de visualizações. Uma *dashboard* representa uma ou mais visualizações, ao passo que uma visualização representa uma ilustração, por meio de gráfico ou tabela, de uma medida (“tabela”) na base de dados.

A tarefa seguinte passou por tentar reproduzir a informação apresentada nas interfaces da aplicação Android, de forma a termos a mesma informação em ambientes diferentes, isto é, num *smartphone*, através da aplicação Vouzela IoT – Pool, e num computador, através de um navegador de Internet. Na Figura 56 podemos observar o ambiente de configuração de uma visualização, neste caso, a temperatura e humidade relativa do Cais da Piscina, onde foi selecionada a medida (“tabela”), o espaço de tempo para tratamento dos dados, métricas estatísticas como, por exemplo, média ou valor máximo, último valor conhecido, tipo de gráfico, entre outras definições que podem ajudar na melhor compreensão dos dados. Esta visualização, depois de devidamente configurada, destina-se a ser incorporada numa *dashboard*. A visualização pode ser editada em qualquer instante.

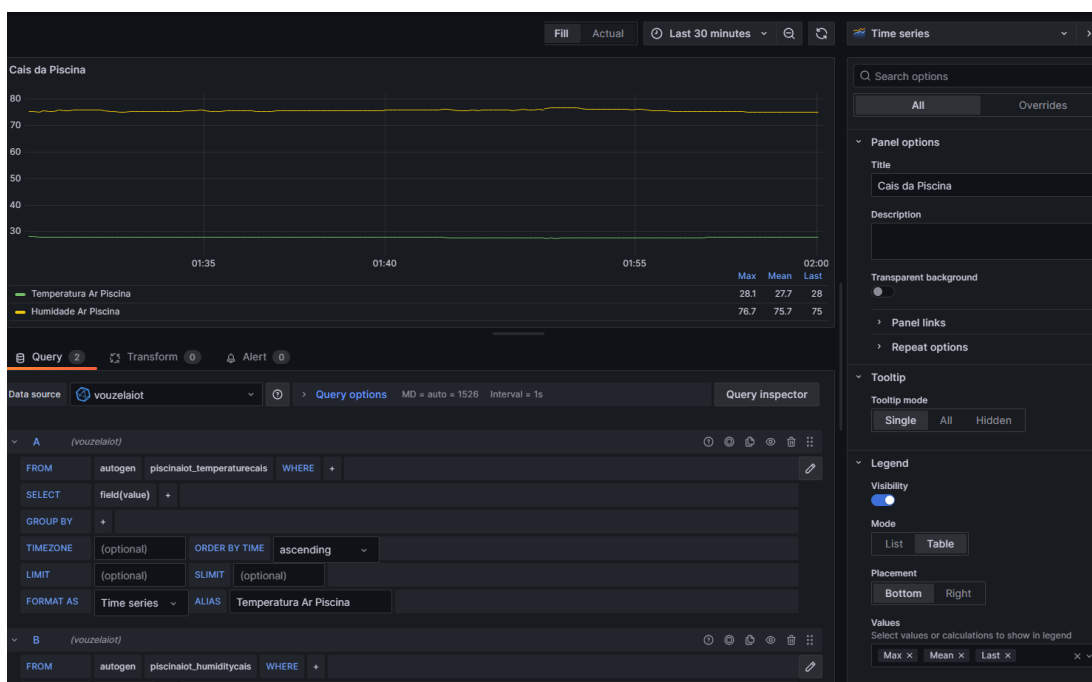


Figura 56 - Grafana na configuração de uma visualização

Após configuradas algumas visualizações, as mesmas integraram, por opção, uma única *dashboard*, como podemos verificar na Figura 57. Esta *dashboard* poderá ser partilhada com outros utilizadores, que estejam registados na plataforma Grafana.



Figura 57 - Grafana no ambiente de dashboard com várias visualizações

Nesta secção, exploramos a criação de visualizações e *dashboards* para o projeto de domótica implementado na piscina municipal, utilizando a plataforma Grafana. Configurámos gráficos, tabelas e métricas estatísticas, apresentando os dados recolhidos pelos sensores ou outros dispositivos do sistema. A criação de *dashboards* personalizadas proporcionou uma visão geral do sistema, possibilitando uma análise abrangente das informações e facilitando a tomada de decisões por parte dos responsáveis pelas instalações da piscina. Com o Grafana foi possível transformar dados em conhecimento e potencializar a eficiência e o controle do sistema de domótica, contribuindo para um ambiente mais confortável, seguro e sustentável.

4.4 Conclusão

Neste capítulo começámos por explicar a montagem e codificação do hardware, essencial para o desenvolvimento da API. Explorámos o fluxo de dados desde os dispositivos Arduino até à aplicação Android, permitindo a apresentação de dados em tempo real. Com essa integração, os responsáveis podem monitorizar os parâmetros importantes da piscina municipal, como temperatura, humidade relativa, níveis de dióxido de carbono e gás propano, movimento e estado da porta, recebendo atualizações instantâneas.

Além disso, implementamos gráficos que apresentam os dados das últimas 12 horas de forma visualmente atraente, garantindo que os responsáveis pelas instalações tenham uma representação clara das tendências de temperatura e humidade no espaço da Recepção e Cais da

Piscina, bem como dos níveis de dióxido de gás propano ao longo do tempo, facilitando a análise e tomada de decisões.

Para além da visualização de vários parâmetros em tempo real ou através de apresentação gráfica, desenvolvemos também um sistema de histórico de eventos. Os responsáveis podem aceder ao histórico de eventos relacionados com a abertura de porta, deteção de movimento fora do horário laboral ou deteção de gases.

Ao combinar todas essas funcionalidades, a API proporciona um sistema de domótica eficiente para a monitorização e controlo da piscina municipal. Os responsáveis podem aceder a dados em tempo real, visualizar gráficos informativos e acompanhar o histórico de eventos importantes, tudo isto com notificações personalizadas por email ou através do serviço OneSignal.

Por fim, foi explorada a aplicação web Grafana, de forma a podermos apresentar os dados, na forma de gráfico ou tabela, noutra ambiente para além de um dispositivo Android.

O próximo capítulo incidirá sobre o trabalho realizado sobre os dados, através da linguagem Python, de forma a extrairmos informação valiosa, através da análise exploratória dos dados.

5 Análise Exploratória de Dados

Neste capítulo será realizada uma análise dos dados obtidos no projeto de domótica implementado na piscina municipal de Vouzela. O objetivo é explorar e compreender os padrões e tendências presentes nos dados coligidos ao longo do tempo.

Utilizando bibliotecas como *Pandas* e *Seaborn* (The pandas development team, 2023; Waskom, 2021), serão aplicados diversos métodos estatísticos e visualizações para estudar os dados históricos. Serão utilizadas funções específicas do *Pandas* para manipular, filtrar e analisar os dados, permitindo extrair informações relevantes e *insights* sobre o comportamento dos sensores e dispositivos do sistema de domótica. Além disso, será gerada uma matriz de correlação entre os dados de temperatura e humidade, com o objetivo de observar uma possível correlação negativa entre essas variáveis.

O objetivo deste projeto contemplava, inicialmente, a correlação entre os dados do sensor magnético, localizado na porta, e do sensor de movimento instalado no *hall* da Receção, de forma a identificar uma correlação significativa entre essas variáveis. Não foi possível fazer a observação, pelo facto do sensor de movimento ser instalado, por razões de segurança definidas pela hierarquia da Organização, num local onde abrangesse outras áreas de acesso à Receção, para além da porta de acesso ao exterior.

Esta análise exploratória permitirá uma melhor compreensão do sistema de domótica e fornecerá informações valiosas para a tomada de decisões e melhorias futuras. A temperatura e humidade relativa de ambos os espaços, dióxido de carbono, deteção de movimento e abertura de porta na área da Receção, serão analisados de forma mais minuciosa, pois são considerados mais pertinentes pelos responsáveis, especialmente no caso da piscina e Receção. Esses parâmetros fornecem informações para avaliar e monitorizar o conforto ambiental, a segurança e o bom funcionamento das instalações. No entanto, a luminosidade no espaço Casa das Máquinas e os valores recolhidos pelos sensores de fuga de gás, não serão objeto de análise, uma vez que esses parâmetros apresentam variações insignificantes ao longo do período de observação.

Após a extração dos dados da base de dados InfluxDB, para ficheiros no formato CSV (*Comma-Separated Values*), de forma a serem processados pela linguagem Python, o passo seguinte foi definir um espaço temporal para análise, que se compreendeu entre o dia 5 e 25 de junho de 2023, ou seja, um período de 21 dias de funcionamento normal dos serviços prestados pela piscina municipal de Vouzela. Neste período e para uma análise mais concreta de certos

parâmetros, foi explorado, ainda, um período de 6 horas, num espaço temporal com maior afluência, num dia normal de funcionamento do serviço. O dia selecionado, para o efeito, foi 6 de junho, pelo facto de existirem imagens dos gráficos da aplicação *Vouzela IoT – Pool* referentes à data escolhida, onde se poderá verificar semelhança com os gráficos produzidos através das bibliotecas Python. Cada ficheiro CSV, também denominado de *dataset* ou *dataframe*, gerado automaticamente por meio da extração dos dados do InfluxDB, representa um conjunto de dados no espaço temporal de 21 dias referente a cada medida, ou seja, de cada parâmetro analisado. No contexto da temperatura e humidade relativa, foram produzidos novos *datasets* com a junção dos dados referentes a estas variáveis, ordenados cronologicamente, para cada espaço físico, ou seja, a Receção e o Cais da Piscina.

Esta abordagem permitirá a análise da correlação dos dados específicos de cada área. Inicialmente, serão explorados os dados recolhidos no espaço da Receção, investigando-se as relações e tendências entre a temperatura e humidade nesse ambiente, no espaço temporal de 6 horas seguidas. Em seguida, o foco será direcionado ao Cais da Piscina, analisando-se os padrões e correlações, também entre a temperatura e humidade, naquele espaço físico. Por fim, os dados recolhidos na casa das máquinas também serão examinados de forma a compreender se existiu registo de movimentos previstos durante os 21 dias em análise. Esta abordagem segmentada permitirá uma análise mais detalhada dos dados e *insights* valiosos sobre o funcionamento do sistema de domótica na piscina municipal.

5.1 Espaço da Receção

Abertura de Porta

No espaço da Receção, registaram-se, durante os 21 dias de observação, 1702 alterações de estado do sensor magnético instalado na porta, o que significa que a porta secundária de acesso à Receção abriu precisamente este número de vezes. A aplicação *Vouzela IoT – Pool* permite a receção de alertas aquando da abertura de porta fora do horário laboral, mas não foram registados eventos entre as 22h30 e as 7h00.

Na Tabela 1 podemos verificar o número de eventos em cada dia, entre o dia 5 e 25 de junho, tendo os dias 8, 11, 18 e 25 um número bastante reduzido de eventos, comparativamente às outras datas, pelo facto de o serviço da piscina municipal se encontrar encerrado. Nestes dias o serviço foi acedido, apenas, pelo encarregado. O dia 5 de junho registou mais eventos de abertura de porta.

Tabela 1 - Registo diário de abertura de porta, na Receção

DATA	ABERTURA DE PORTA
05/06/2023	152
06/06/2023	115
07/06/2023	119
08/06/2023	1
09/06/2023	106
11/06/2023	3
12/06/2023	123
13/06/2023	118
14/06/2023	135
15/06/2023	114
16/06/2023	87
17/06/2023	55
18/06/2023	1
19/06/2023	100
20/06/2023	104
21/06/2023	117
22/06/2023	98
23/06/2023	120
24/06/2023	33
25/06/2023	1

Deteção de Movimento

O acesso não previsto às instalações da piscina ou violação do espaço foi registado no parâmetro deteção de movimento, assim como no de abertura de porta. À semelhança da abertura de porta, a aplicação *Vouzela IoT – Pool* também recebe alertas sempre que é detetado movimento fora do horário laboral. Isto permite uma rápida intervenção, aumentando significativamente a segurança do espaço e a redução de custos em caso de assalto.

Nas mesmas datas em que se verifica um reduzido número de eventos de abertura de porta também é possível observar um reduzido número de eventos de deteção de movimento, como podemos observar na Tabela 2, pois, como já referido, o serviço esteve encerrado.

Tabela 2 - Registo diário de movimentos, na Receção

DATA	MOVIMENTOS
05/06/2023	292
06/06/2023	279
07/06/2023	313
08/06/2023	2
09/06/2023	278
10/06/2023	4
11/06/2023	4
12/06/2023	264
13/06/2023	322
14/06/2023	266
15/06/2023	321
16/06/2023	214
17/06/2023	141
18/06/2023	2
19/06/2023	296
20/06/2023	282
21/06/2023	288
22/06/2023	244
23/06/2023	271
24/06/2023	104
25/06/2023	2

Observando as datas em que o serviço esteve em funcionamento, não podemos afirmar que estas variáveis (abertura de porta e deteção de movimento) estejam diretamente relacionadas,

dado que nos dias em que se verifica um maior número de movimentos, não se verifica sempre um maior número de abertura de porta. No espaço de tempo analisado, foram registados 4189 movimentos no hall da Receção.

Dióxido de Carbono (CO2)

O dióxido de carbono é, naturalmente, um parâmetro a analisar, pelo facto de estar diretamente relacionado com a saúde dos utentes. Em determinados períodos do dia existe um maior número de pessoas a partilhar o ar do mesmo espaço, e isso levou-nos à exploração dos dados recolhidos. Sabendo que, à escala global, a média de dióxido de carbono concentrado na atmosfera é de 420 partículas por milhão (ppm), segundo a Administração Nacional Oceânica e Atmosférica (NOAA) dos Estados Unidos (Global Monitoring Laboratory, 2023), e o máximo aceitável no interior dos edifícios é de 650 ppm superiores ao exterior (Afonso, 2018), podemos verificar no Gráfico 1 que o valor máximo atingido no hall de Receção foi de 442 ppm, o que não justifica um investimento imediato num equipamento para a recirculação de ar. No entanto, é importante salientar que os valores observados representam apenas um período de 21 dias no mês de junho, totalizando 654054 registos de CO2.

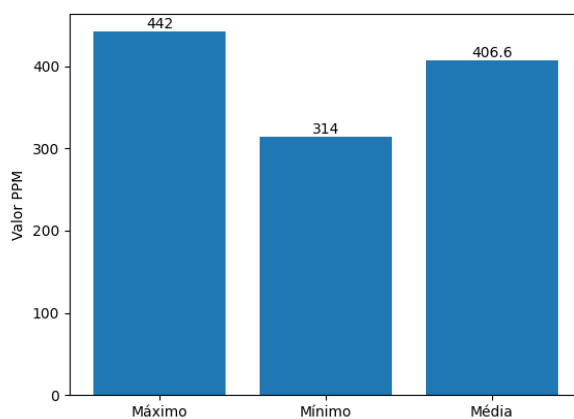


Gráfico 1 - Registo estatístico de CO2

Temperatura e Humidade Relativa

No espaço da receção da piscina municipal, uma análise dos dados recolhidos ao longo de um período de 21 dias revela informações importantes sobre a temperatura e a humidade relativa. No Gráfico 2 podemos observar que a temperatura média registrada foi de 25 °C, atingindo uma máxima de 27.7 °C e uma mínima de 23.5 °C. A humidade relativa teve o valor médio de 62.7%, alcançando um máximo de 74.3% e um mínimo de 46.5%. Estas informações são relevantes

para compreender as condições ambientais no espaço da Receção e, devido à temperatura registada, justificam a aquisição de um equipamento de ar condicionado, de forma a oferecer um maior conforto aos utentes e aos colaboradores, pois a temperatura dos locais de trabalho deve, na medida do possível, oscilar entre 18 °C e 22 °C, salvo em determinadas condições climáticas, em que poderá atingir os 25 °C, e a humidade entre 50% a 70%, segundo a alínea a) do ponto 1, do art. 11º do DL. 243/86, de 20 de agosto, que aprova o Regulamento Geral de Higiene e Segurança do Trabalho nos Estabelecimentos Comerciais, de Escritório e Serviços.

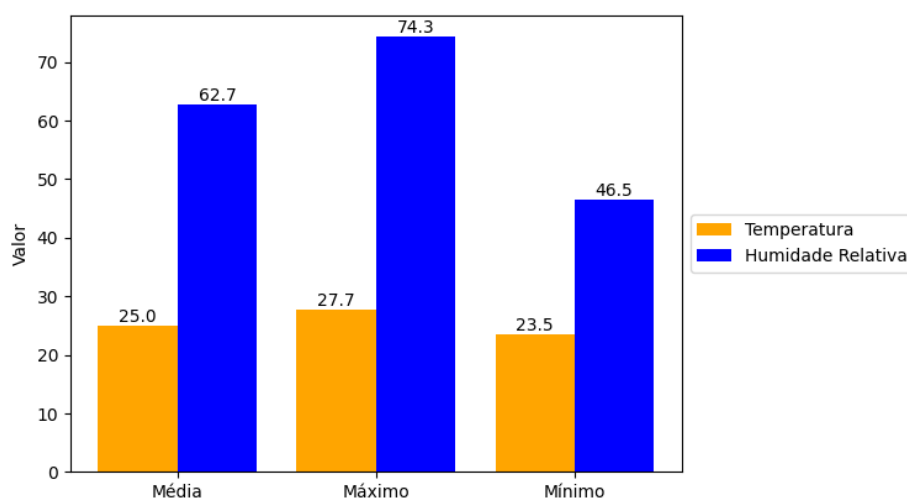


Gráfico 2 - Registo estatístico de Temperatura e Humidade na Receção

Temperatura e Humidade Relativa (12h-18h | 6/06/2023)

Além da análise ao longo do período de 21 dias, também foi efetuada uma análise mais específica da temperatura e humidade num dia normal de funcionamento da piscina municipal. Essa análise teve duração de 6 horas consecutivas, das 12h às 18h, especificamente no dia 6 de junho. Durante esse período, pudemos observar e registrar as variações da temperatura e humidade relativa no ambiente da Receção.

Estes dados são relevantes para uma compreensão mais detalhada das condições ambientais durante o horário de funcionamento, permitindo uma análise mais precisa e focada neste intervalo de tempo. Na Tabela 3 podemos observar a alteração da temperatura e da humidade, através da média obtida a cada hora.

Podemos afirmar que, em quase todo o espaço temporal analisado, existe uma correlação negativa, porque quando o valor da temperatura aumenta, o valor da humidade diminui (e vice-

versa), embora a correlação não seja perfeita, dado que a proporção de valores destas variáveis, durante a variância, não é igual.

Tabela 3 - Valor médio obtido da Temperatura e Humidade na Recepção

Hora	Temperatura (Média)	Humidade (Média)
13:00	25.1	56.7
14:00	25.3	55.7
15:00	25.5	54.2
16:00	25.8	52.2
17:00	25.8	56.1
18:00	25.3	60.6

Estas variações, assim como a correlação negativa, podem, também, ser observadas tanto no gráfico (A) gerado por meio da análise de dados em Python, como no gráfico (B) apresentado na aplicação Vouzela IoT – Pool, representado no Gráfico 3. Estas representações permitem uma visualização mais clara da variância e tendências desses parâmetros ao longo do período de 6 horas em análise.

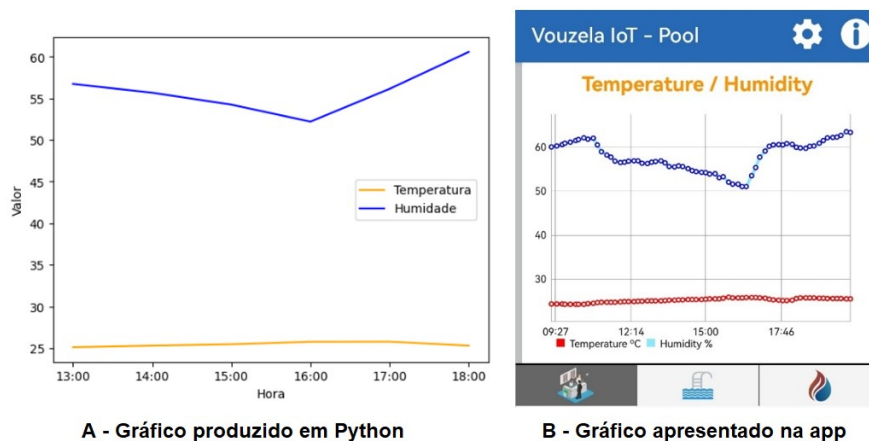


Gráfico 3 - Variação da Temperatura e Humidade na Recepção

Uma matriz de correlação é uma representação tabular que mostra a relação estatística entre pares de variáveis. No contexto da análise de dados de temperatura e humidade, a matriz de correlação permitirá explorar a relação entre essas duas variáveis ao longo de um intervalo de 6 horas, das 12 às 18 horas.

O objetivo da matriz de correlação é identificar se existe alguma relação entre a temperatura e a humidade, e qual é a natureza dessa relação. Através dos coeficientes de correlação presentes na matriz, que variam entre -1 e 1, podemos determinar se há uma correlação positiva (ambas as variáveis aumentam ou diminuem juntas), uma correlação negativa (uma variável aumenta enquanto a outra diminui ou vice-versa) ou se não há correlação aparente entre elas.

Ao analisar a matriz de correlação da Figura 58, podemos observar que existe uma correlação negativa forte, quase perfeita, entre a temperatura e a humidade relativa no espaço da Receção, às 13 e às 16 horas, significando que os valores de ambas as variáveis se aproximaram de uma forma quase proporcional.

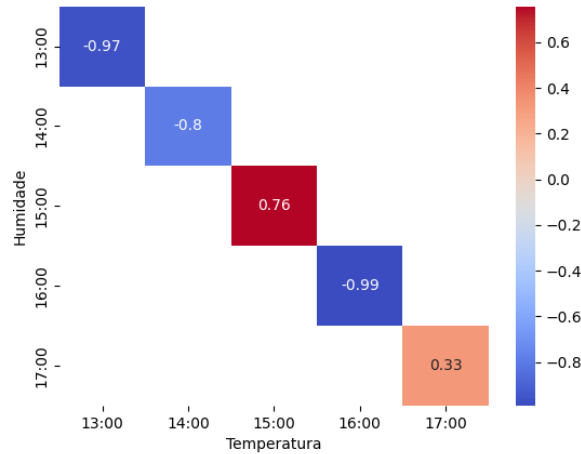


Figura 58 - Matriz de Correlação entre a Temperatura e Humidade, na Receção

5.2 Espaço do Cais da Piscina

As variáveis de temperatura e humidade relativa do ar no cais da piscina também foram objeto de análise, explorando 279672 registos de dados ao longo de um período de 21 dias. Durante esse período registamos os valores diários de temperatura e humidade e calculamos algumas estatísticas relevantes, como ilustrado no Gráfico 4. Em relação à temperatura, a média registrada foi de 27.8 °C, com uma temperatura máxima de 32.8 °C e uma temperatura mínima de 26.3 °C. Quanto à humidade relativa, observamos uma média de 70.7%, com um valor máximo de 81.8% e um valor mínimo de 52.8%.

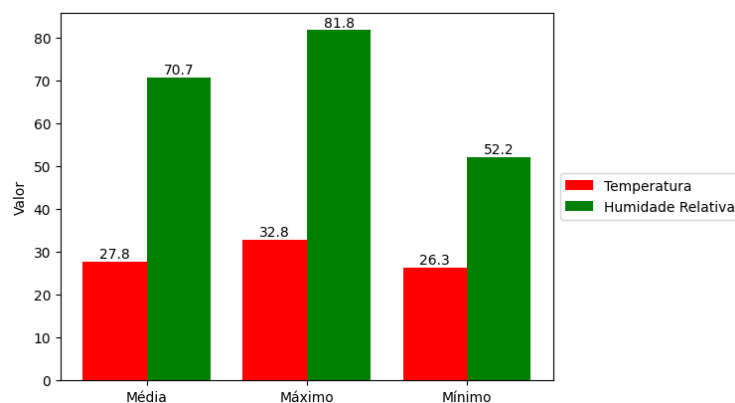


Gráfico 4 - Registo estatístico de Temperatura e Humidade no Cais da Piscina

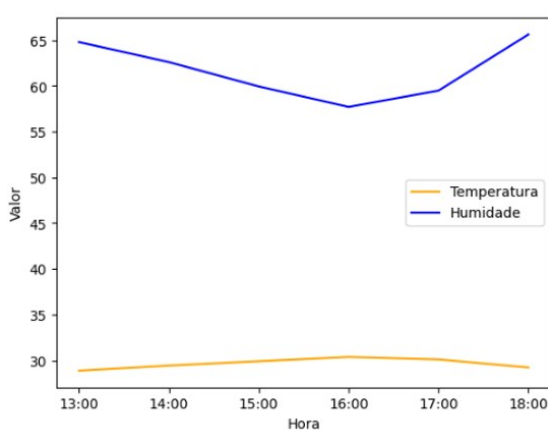
Temperatura e Humidade Relativa (12h-18h | 6/06/2023)

À semelhança da análise efetuada nos dados relativos ao espaço da Receção, também os dados da temperatura e humidade relativa foram analisados num período de 6 horas seguidas, na mesma data. Na Tabela 4 podemos verificar a alteração da temperatura e da humidade, através da média obtida a cada hora, em que podemos afirmar que, em quase todo o espaço temporal analisado, existe uma correlação negativa, porque quando o valor da temperatura aumenta, o valor da humidade diminui (e vice-versa), embora a correlação não seja perfeita, dado que a proporção de valores destas variáveis, durante a variância, não é igual.

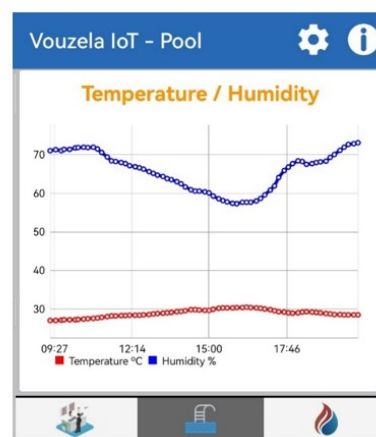
Tabela 4 - Valor médio obtido da Temperatura e Humidade no Cais da Piscina

Hora	Temperatura (Média)	Humidade (Média)
13:00	28.9	64.8
14:00	29.4	62.6
15:00	29.9	59.9
16:00	30.4	57.7
17:00	30.1	59.5
18:00	29.2	65.6

As flutuações e a correlação negativa mencionadas podem ser facilmente observadas tanto no gráfico (A) resultante da análise de dados em Python, quanto no gráfico (B) apresentado na aplicação *Vouzela IoT – Pool*, como ilustrado no Gráfico 5. Essas representações visuais proporcionam uma visualização mais clara das variações e tendências dos parâmetros de temperatura e humidade no Cais da Piscina ao longo do período de análise de 6 horas.



A - Gráfico produzido em Python



B - Gráfico apresentado na app

Gráfico 5 - Variação da Temperatura e Humidade no Cais da Piscina

À semelhança da análise de dados efetuada no espaço da Receção, também foi produzida uma matriz de correlação no contexto da análise de dados de temperatura e humidade, que permitiu explorar a relação entre essas duas variáveis ao longo de um intervalo de 6 horas, das 12 às 18 horas.

Ao analisar a matriz de correlação (Figura 59), podemos observar que existe uma correlação negativa forte, entre a temperatura e a humidade no espaço do Cais da Piscina, às 16 horas, significando que os valores de ambas as variáveis se aproximaram de uma forma quase proporcional. No entanto, também é visível uma correlação positiva às 15 horas. Essa tendência oposta consegue-se verificar no gráfico (B) apresentado na aplicação *Vouzela IoT – Pool*, como ilustrado no Gráfico 5, talvez devido à ativação de alguma máquina controladora do ambiente do Cais da Piscina.

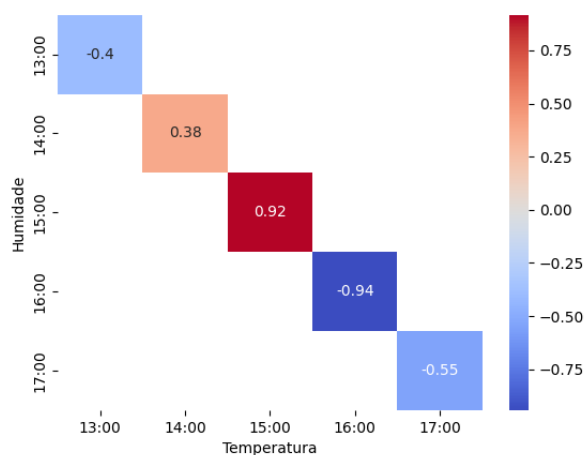


Figura 59 - Matriz de Correlação entre a Temperatura e Humidade, no Cais da Piscina

5.3 Espaço da Casa das Máquinas

Durante o mesmo período de 21 dias, compreendido entre 5 de junho e 25 de junho, também foram analisados 233 registos referentes ao número de movimentos detetados na Casa das Máquinas. O número diário de eventos, mostrado na Tabela 5, comprova a inspeção da maquinaria por parte dos colaboradores. É de referir, ainda, que caso fosse detetado movimento na Casa das Máquinas fora do horário de funcionamento da piscina municipal, os responsáveis do serviço recebiam um alerta informativo sobre este evento, à semelhança do que acontece aquando do movimento detetado fora do horário laboral no espaço da Receção.

Tabela 5 - Registo diário de movimentos, na Casa das Máquinas

DATA	MOVIMENTOS
05/06/2023	21
06/06/2023	18
07/06/2023	8
09/06/2023	25
11/06/2023	21
12/06/2023	13
13/06/2023	12
14/06/2023	6
15/06/2023	11
16/06/2023	15
17/06/2023	7
19/06/2023	19
20/06/2023	11
21/06/2023	14
22/06/2023	7
23/06/2023	16
24/06/2023	8

5.4 Conclusão

Neste capítulo explorámos diversas informações provenientes do sistema de domótica implementado na piscina municipal de Vouzela. Iniciamos a nossa análise no espaço da receção, investigando dados de abertura de porta, movimento, temperatura e humidade. Em seguida, direccionámos a nossa atenção ao cais da piscina, examinando os dados de temperatura e humidade nesse ambiente. Por fim, exploramos os dados de deteção de movimento no espaço da casa das máquinas.

Ao longo dessa exploração, utilizamos dois períodos de tempo distintos para análise. O primeiro período abrangeu um total de 21 dias, permitindo uma avaliação mais abrangente dos padrões e tendências ao longo desse intervalo. Já o segundo período focou-se em um dia normal de funcionamento, com duração de 6 horas consecutivas, das 12h às 18h.

Essa análise detalhada possibilitou-nos identificar padrões e correlações entre as diferentes variáveis em cada espaço. Esses *insights* são fundamentais para compreender as condições ambientais, a atividade nos espaços e auxiliar na tomada de decisões para otimizar o funcionamento da piscina.

6 Conclusão e trabalho futuro

O presente trabalho teve como objetivo principal a implementação de um sistema de domótica na piscina municipal de Vouzela, visando melhorar a gestão, monitorização e controlo dos diversos parâmetros ambientais e de segurança do espaço. Ao longo do desenvolvimento do projeto, foram abordados os principais componentes do sistema, desde o hardware utilizado, como os microcontroladores acoplados em placas *Arduino*, diversos sensores e atuadores, até às ferramentas, como a *Arduino IDE*, *PHP Storm*, *InfluxDB* e o *Android Studio*, assim como as tecnologias utilizadas nestas ferramentas, como as linguagens *C* e *PHP*, *SQL*, *JAVA*. De realçar ainda a importância das bibliotecas, concretamente a *WifiEsp*, *PHPMailer*, *OkHttp* e *Gson*, utilizadas em todo o processo de codificação deste projeto.

Através da integração dos componentes de hardware, foi possível estabelecer a comunicação entre os dispositivos, a API e a aplicação móvel *Android*, permitindo a recolha e o processamento dos dados em tempo real. A utilização do *Grafana*, como plataforma de visualização e monitorização dos dados, proporcionou uma análise exploratória detalhada e uma apresentação estatística dos mesmos, permitindo uma melhor compreensão do ambiente da piscina e a identificação de padrões e tendências. Para além do *Grafana*, a análise de dados também passou pela ferramenta *Colab* (GOOGLE, s.d.), que permitiu uma exploração mais minuciosa através da linguagem de programação *Python* e bibliotecas associadas a esta linguagem para tratamento e apresentação de dados, como a *Pandas* e *Seaborn*.

A partir da análise dos dados, verificou-se que o valor máximo de CO₂ registado no hall de Receção foi de 442 ppm, o que não justifica um investimento imediato num equipamento para a recirculação de ar. No entanto, é importante salientar que os valores observados representam apenas um período de 21 dias no mês de junho, período, esse, onde também se verifica mais de 1700 aberturas de porta, o que proporcionou, inevitavelmente, a recirculação do ar.

Para além do CO₂, também observámos que a temperatura média registrada foi de 25 °C, atingindo uma máxima de 27.7 °C e uma mínima de 23.5 °C, o que justifica a aquisição de um equipamento de ar condicionado, de forma a oferecer maior conforto aos utentes. Outros dados sobre a temperatura e humidade no Cais da Piscina não levantaram, até ao momento, dúvidas que levassem a alteração da programação dos controladores que monitorizam aquele espaço.

Durante a implementação do sistema, enfrentámos desafios técnicos e de configuração, os quais foram superados com dedicação, pesquisa e trabalho em equipa. A colaboração entre os membros colaboradores do Município de Vouzela e o apoio do orientador foram fundamentais para o sucesso deste projeto.

Através deste trabalho, foi possível constatar os benefícios da aplicação da domótica na gestão de espaços públicos, proporcionando um ambiente mais seguro, eficiente e confortável para os utentes. Além disso, a utilização de tecnologias modernas e a análise dos dados recolhidos possibilitaram uma tomada de decisão mais informada e a implementação de medidas preventivas para garantir a qualidade e a sustentabilidade das instalações.

É, ainda, de referir que o objetivo da implementação do sistema de domótica foi superado, e, assim, todos os problemas identificados e mencionados na introdução deste documento, se encontram-se resolvidos.

Este projeto de domótica na piscina municipal de Vouzela representou uma oportunidade única na aplicação dos conhecimentos adquiridos ao longo do curso de mestrado em Engenharia Informática e contribuiu para a melhoria das condições e da eficiência de um espaço público importante para a comunidade. Esperamos que este trabalho possa servir de referência e inspiração para futuros projetos de domótica e contribuir para o avanço da tecnologia na área da automação residencial e gestão inteligente de espaços públicos.

6.1 Sugestões para trabalho futuro

Estão previstas várias melhorias e expansões no sistema de domótica implementado na piscina municipal de Vouzela, sendo uma delas a segurança da comunicação entre os dispositivos Arduino e o servidor. Pretende-se implementar uma comunicação segura utilizando o protocolo HTTPS, o qual garantirá a integridade, confidencialidade e autenticidade dos dados transmitidos. A biblioteca *WiFiEsp* fornecerá as ferramentas necessárias para a implementação desse recurso de segurança.

Está prevista, também, a instalação de novos sensores em diferentes espaços da piscina, bem como o desenvolvimento da aplicação Vouzela IoT – Pool para o sistema IOS, através do sistema multiplataforma Flutter (GOOGLE, s.d.) ou outro. Especificamente, serão adicionados sensores de movimento em locais estratégicos para melhorar a deteção e monitorização de atividades nos espaços da piscina. Também está prevista a integração de um sensor de

temperatura da água, permitindo, assim, obter, em tempo real, a informação sobre a temperatura atual da água da piscina. Esses novos sensores fornecerão dados adicionais e valiosos para a gestão e manutenção eficientes das instalações.

Estas melhorias e expansões futuras irão fortalecer ainda mais o sistema de domótica implementado, tornando-o mais seguro, abrangente e eficiente. A incorporação de novos recursos e sensores permitirá uma monitorização mais abrangente e precisa do ambiente da piscina, contribuindo para a tomada de decisões informadas e ações preventivas. Com essas implementações, o sistema continuará a evoluir e a adaptar-se às necessidades e exigências da piscina municipal de Vouzela, proporcionando uma experiência ainda melhor para os utentes e colaboradores e uma gestão mais eficiente do espaço.

7 Referências

- Afonso, C. &. (2018). Qualidade do Ar por Renovação do Ar por Controlo Contínuo de CO2 Versus Injeção Contínua de Ar Novo: Consumos Energéticos Associados e Benefícios. *Revista da Associação Portuguesa de Análise Experimental de Tensões.*, 43-54.
- Ajiboye, A. T. (2021). Analytical determination of load resistance value for MQ-series gas sensors: MQ-6 as case study. (T. (. Control), Ed.) doi:<http://doi.org/10.12928/telkomnika.v19i2.17427>
- Alotaibi, A., Chiou, P., & Halfond, W. (2022). Automated Detection of TalkBack Interactive Accessibility Failures in Android Applications. *IEEE Conference on Software Testing, Verification and Validation (ICST)*. doi:10.1109/ICST53961.2022.00033
- Alves, R. C., Florian, F., & Farina, R. M. (2022). *Domótica: Estudo da contribuição da automação residencial para a acessibilidade de portadores de deficiência física*, 3(12). doi:<https://doi.org/10.47820/recima21.v3i12.2299>
- AMAZON. (s.d.). *Amazon Alexa*. Obtido em março de 2023, de <https://developer.amazon.com/en-US/alexa/>
- Android Developers*. (s.d.). Obtido em abril de 2023, de Criar listas dinâmicas com o RecyclerView: <https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=pt-br>
- ANDROID. (s.d.). *O que é o android*. Obtido em fevereiro de 2023, de Android: https://www.android.com/intl/pt_pt/what-is-android/
- ARDUINO. (2022). *What is Arduino?* Obtido em outubro de 2023, de Arduino: <https://www.arduino.cc/en/Guide/Introduction>
- ARDUINO. (s.d.). *Arduino*. Obtido em março de 2023, de Arduino Downloads: <https://www.arduino.cc/en/software>
- Arduino Mega*. (2021). Obtido em fevereiro de 2023, de JavaTPoint: <https://www.javatpoint.com/arduino-mega>

- Arduino UNO*. (2021). Obtido em fevereiro de 2023, de JavaTPoint: <https://www.javatpoint.com/arduino-uno>
- Banzi, M. &. (2011). *Primeiros passos com o Arduino*. São Paulo: Novatec.
- Bertrand, F. (2022). *A pandas-based library to visualize and compare datasets*. Obtido de Sweetviz: <https://pypi.org/project/sweetviz/>
- Bing, K. e. (2011). Design of an Internet of Things-based smart home system. *2nd International Conference on Intelligent Control and Information Processing*, (pp. 921-924). Harbin, China. doi:10.1109/ICICIP.2011.6008384
- Bolzani, C. (2010). *Análise de arquiteturas e desenvolvimento de uma plataforma para residências inteligentes* (Tese de Doutorado em Engenharia Eletrotécnica ed.). São Paulo. Obtido em outubro de 2023, de https://www.teses.usp.br/teses/disponiveis/3/3142/tde-12082010-112005/publico/Tese_Caio_Augustus_Morais_Bolzani.pdf
- Bolzani, C. A. (2004). *Residências Inteligentes*. Livraria da Física.
- Carvalho, J. D. (2019). *Sistemas de gestão para aplicações de domótica*. Dissertação de Mestrado em Engenharia Electrotécnica e de Computadores, Universidade de Trás-os-Montes e Alto Douro, Vila Real. Obtido de <http://hdl.handle.net/10348/9348>
- Danischevsky, S., & Brester, S. (2023). *Fail2Ban*. Obtido de GitHub: <https://github.com/fail2ban/fail2ban/wiki/How-to-install-fail2ban-packages>
- DFRobot*. (s.d.). Obtido em maio de 2023, de DFRobot - Drive the future.: <https://www.dfrobot.com/product-1102.html>
- DR. POOLS. (s.d.). Obtido em março de 2023, de <https://www.lojadppools.pt/product/aqualink-tri-zodiac>
- DzTechs. (s.d.). *O que é Zigbee, como funciona e quais são seus usos?* Obtido em março de 2023, de DzTechs: <https://www.dz-techs.com/pt/what-is-zigbee>
- El-hajj, M. F. (2019). A Survey of Internet of Things (IoT) Authentication Schemes. (M. AG., Ed.) *Sensors*, 19(5). doi:<http://dx.doi.org/10.3390/s19051141>

- EUROPEAN COMMISSION. (2022). *REPORT FROM THE COMMISSION TO THE COUNCIL AND THE EUROPEAN - Final report - sector inquiry into consumer Internet of Things*. Brussels. Obtido de https://competition-policy.ec.europa.eu/system/files/2022-01/internet-of-things_final_report_2022_en.pdf
- Fail2Ban*. (s.d.). Obtido em maio de 2023, de <https://github.com/fail2ban/fail2ban/wiki/How-to-install-fail2ban-packages>
- Fritzing*. (s.d.). Obtido em fevereiro de 2023, de Fritzing: <https://fritzing.org/>
- Global Monitoring Laboratory*. (2023). Obtido em setembro de 2023, de <https://gml.noaa.gov/ccgg/trends/global.html>
- GOOGLE. (s.d.). *Android Studio*. Obtido em abril de 2023, de Get the official Integrated Development Environment (IDE) for Android app development.: <https://developer.android.com/>
- GOOGLE. (s.d.). *Colaboratory*. Obtido em setembro de 2023
- GOOGLE. (s.d.). *Flutter on Mobile*. Obtido em outubro de 2023, de Flutter: <https://flutter.dev/multi-platform/mobile>
- GOOGLE. (s.d.). *Google Assistant: Hey Google*. Obtido em março de 2023, de <https://assistant.google.com/>
- GRAFANA. (s.d.). *Operational dashboards for your data here, there, or anywhere*. Obtido em setembro de 2023, de Grafana: <https://grafana.com/>
- Gson*. (2023). Obtido de GitHub: <https://github.com/google/gson>
- Gupta, M. S. (2021). A Survey on the ZigBee Protocol, It's Security in Internet of Things (IoT) and Comparison of ZigBee with Bluetooth and Wi-Fi. (X. K. Gao, Ed.) *Applications of Artificial Intelligence in Engineering. Algorithms for Intelligent Systems*. doi:https://doi.org/10.1007/978-981-33-4604-8_38
- Hussain, D. S. (2015). Automated Color Recognition System for Visually Challenged and Achromatopsia People using Arduino and Mobile App. *International Journal of Advanced Research in Electronics and Communication Engineering*. Obtido de

https://www.researchgate.net/publication/281538436_Automated_Color_Recognition_System_for_Visually_Challenged_and_Achromatopsia_People_using_Arduino_and_Mobile_App

Influxdata. (s.d.). Obtido em abril de 2023, de Influxdata documentation: <https://docs.influxdata.com/influxdb/v1.8/introduction/install/>

InfluxDB is the time series platform for developers. (s.d.). Obtido em abril de 2023, de InfluxData: <https://www.influxdata.com/products/influxdb-overview/>

InfluxDB-PHP. (s.d.). Obtido em abril de 2023, de <https://github.com/influxdata/influxdb-php>

IoT POOL. (s.d.). *Best store for your pool*. Obtido em maio de 2023, de <https://www.iot-pool.com/collections/7-2-domotica/products/sistema-de-domotica-automacao-sep>

J. Andrews, M. K. (2020). A motion induced passive infrared (PIR) sensor for stationary human occupancy detection. *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1295-1304. doi:10.1109/PLANS46316.2020.9109909

JET BRAINS. (s.d.). *PHP Storm*. Obtido em abril de 2023, de The Lightning-Smart PHP IDE: <https://www.jetbrains.com/phpstorm/>

LD PLAN. (s.d.). *O Sistema de domótica ideal para sua casa*. Obtido em março de 2023, de <https://ldplan.pt/domotica/residencial/>

LEGRAND. (s.d.). *My Home - Domótica*. Obtido em fevereiro de 2023, de <https://www.legrand.pt/index.php/residencial/itemlist/category/186-my-home-domotica>

Liu, Q. M. (2016). Green data center with IoT sensing and cloud-assisted smart temperature control system. *Computer Networks, 101*, pp. 104-112. Obtido de <https://www.sciencedirect.com/science/article/abs/pii/S1389128615004739>

LoRa Alliance. (2022). *What is LoRaWAN Specification*. Obtido em fevereiro de 2023, de LoRa Alliance: <https://lora-alliance.org>

- MasterWalkerShop*. (s.d.). Obtido em março de 2023, de <https://blogmasterwalkershop.com.br/embarcados/esp8266/upgrade-de-firmware-do-wifi-esp8266-esp-01-atraves-do-arduino-e-conversor-usb-serial>
- McKinney, W. (2010). Data structures for statistical computing in python. Em S. v. Millman (Ed.), *Proceedings of the 9th Python in Science Conference*, 445, pp. 56-61. doi:10.25080/Majora-92bf1922-00a
- Misra, D. D. (2018). An IoT-based waste management system monitored by cloud. *Journal of Material Cycles and Waste Management*, 1574-1582. Obtido de <https://link.springer.com/article/10.1007/s10163-018-0720-y>
- Molino, J. G. (2017). *Diseño e implementación de un dispositivo IoT de bajo coste para entornos acuáticos*. Granada: Escuela Técnica Superior de Ingenierías Informática y deTelecomunicación.
- MPAndroidChart*. (s.d.). Obtido em julho de 2023, de *MPAndroidChart*: <https://github.com/PhilJay/MPAndroidChart>
- MYSQL. (2022). *The world's most popular open source database*. Obtido em fevereiro de 2023, de MySQL: mysql.com
- Naqvi, S. N. (2017). *Time series databases and influxdb*. Studienarbeit, Université Libre de Bruxelles. Obtido de https://cs.ulb.ac.be/public/_media/teaching/influxdb_2017.pdf
- Nasar, M. &. (2019). Suitability Of Influxdb Database For Iot. *International Journal of Innovative Technology and Exploring Engineering*. Obtido de https://www.researchgate.net/publication/335542829_Suitability_Of_Influxdb_Database_For_IoT_Applications
- OkHttp*. (s.d.). Obtido de <https://square.github.io/okhttp/>
- Oliveira, E. (2022). *Upgrade de Firmware do WiFi ESP8266 ESP-01 através do Arduino e Conversor USB Serial*. Obtido de MasterWalker: <https://blogmasterwalkershop.com.br/embarcados/esp8266/upgrade-de-firmware-do-wifi-esp8266-esp-01-atraves-do-arduino-e-conversor-usb-serial>

- OneSignal*. (s.d.). Obtido em julho de 2023, de Customer Messaging Delivered: <https://onesignal.com/>
- OneSignal*. (s.d.). Obtido em julho de 2023, de Guides: Android SDK setup: <https://documentation.onesignal.com/docs/android-sdk-setup>
- PHP. (2022). *A popular general-purpose scripting language that is especially suited to web development*. Obtido em outubro de 2023, de PHP: php.net
- PHP Mailer*. (s.d.). Obtido em julho de 2023, de GitHub: <https://github.com/PHPMailer/PHPMailer>
- Putri, M. A. (2018). Design of security tools using sensor Light Dependent Resistor (LDR) through mobile phone. *International Journal of Innovative Research in Computer and Communication Engineering.*, 168-173.
- Ramphela, M. K. (2020). Internet of things (IoT) integrated data center infrastructure monitoring system. *International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, (pp. 1–6). Obtido de <https://ieeexplore.ieee.org/abstract/document/9183873>
- Rocha, J. L. (2022). A Cost-Effective Framework for Monitoring Disaster Recovery Infrastructures. *New Trends in Disruptive Technologies, Tech Ethics and Artificial Intelligence* (pp. 201-211). Salamanca: de la Iglesia, Daniel H., Paz Santana, Juan F., López Rivero, Alfonso J. Obtido de https://doi.org/10.1007/978-3-031-14859-0_18
- Saha, S. a. (2017). Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system. *Devices for Integrated Circuit (DevIC)*, (pp. 307-310). Kalyani, India. Obtido de <https://ieeexplore.ieee.org/abstract/document/8073958>
- Sanz Gresa, B. (2021). *Diseño e implementación de un sistema de mantenimiento de una piscina basado en Arduino. Doctoral dissertation*. València: Universitat Politècnica de València.

- Saptiani, P. A. (2019). The electrical properties characterization of MG-811 gas sensor toward the temperature alteration of soil testing chamber. *Journal of Physics: Conference Series*, 1280. doi:10.1088/1742-6596/1280/2/022058
- Satapathy, L. M. (2018). Arduino based home automation using Internet of things (IoT). *International Journal of Pure and Applied Mathematics*, 118(17), 769-778.
- Shah, S. H., & Yaqoob, I. (2016). A Survey of Internet of Things (IoT) Technologies and Applications. pp. 381-385. doi:10.1109/SEGE.2016.7589556
- The pandas development team. (2023). pandas-dev/pandas: Pandas (v2.0.3) . (Zenodo, Ed.) doi:10.5281/zenodo.8092754
- Ubuntu*. (s.d.). Obtido em fevereiro de 2023, de <https://ubuntu.com/>
- Valente, A. C. (2022). A LoRaWAN IoT System for Smart Agriculture for Vine Water Status Determination. (D. Dimitrov, Ed.) *Agriculture*, 1695. Obtido de <https://www.mdpi.com/2077-0472/12/10/1695>
- Waskom, M. L. (2021). seaborn: statistical data visualization. (T. O. Journal, Ed.) *Journal of Open Source Software*, 6(60), 3021. Obtido de <https://doi.org/10.21105/joss.03021>
- WiFiEsp*. (s.d.). Obtido em abril de 2023, de Github: <https://github.com/bportaluri/WiFiEsp>

ANEXOS

Anexo A – Planeamento do trabalho

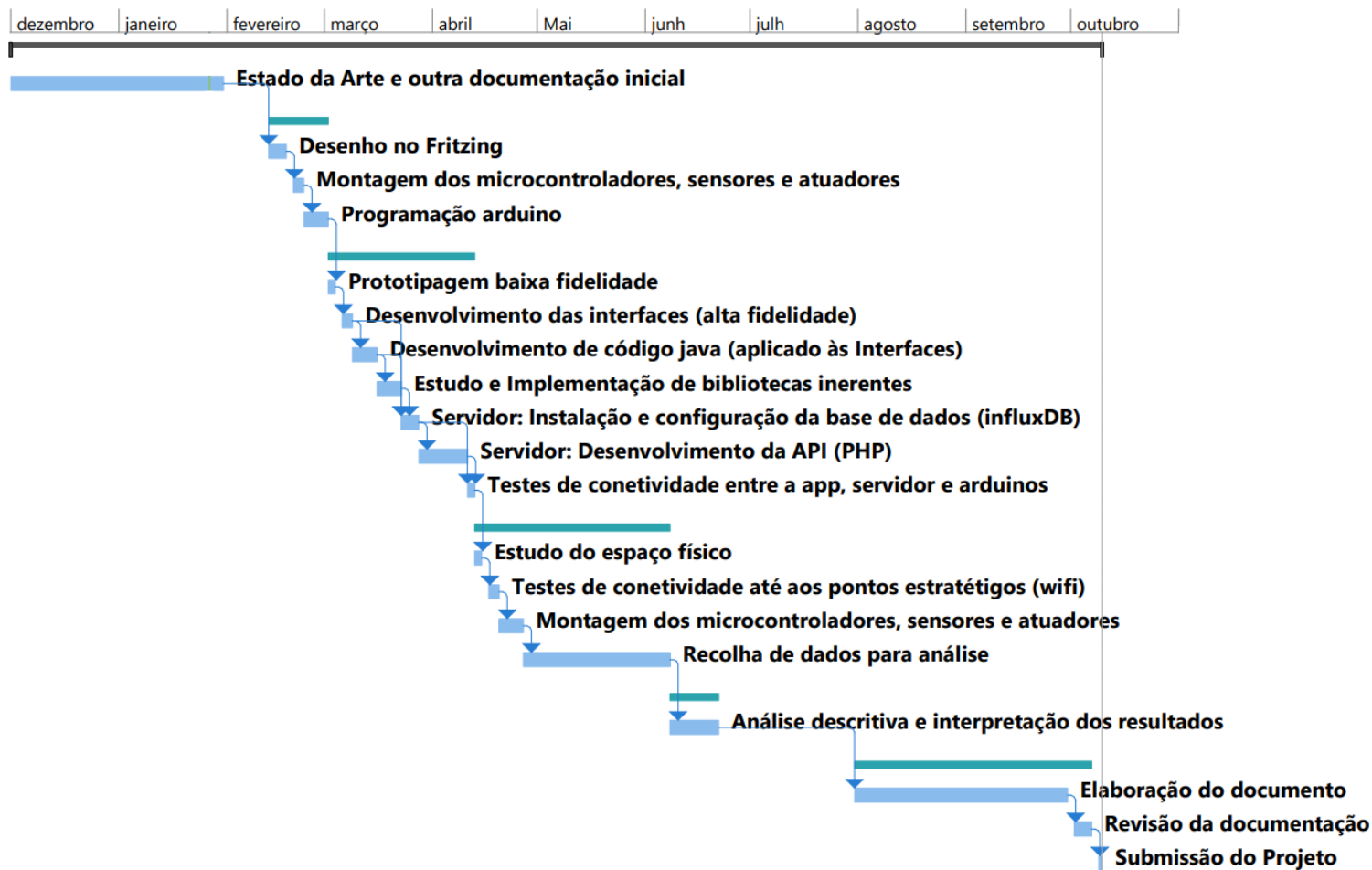


Figura 60 - Cronograma previsto para desenvolvimento do projeto

Anexo B – Esquema do espaço Recepção

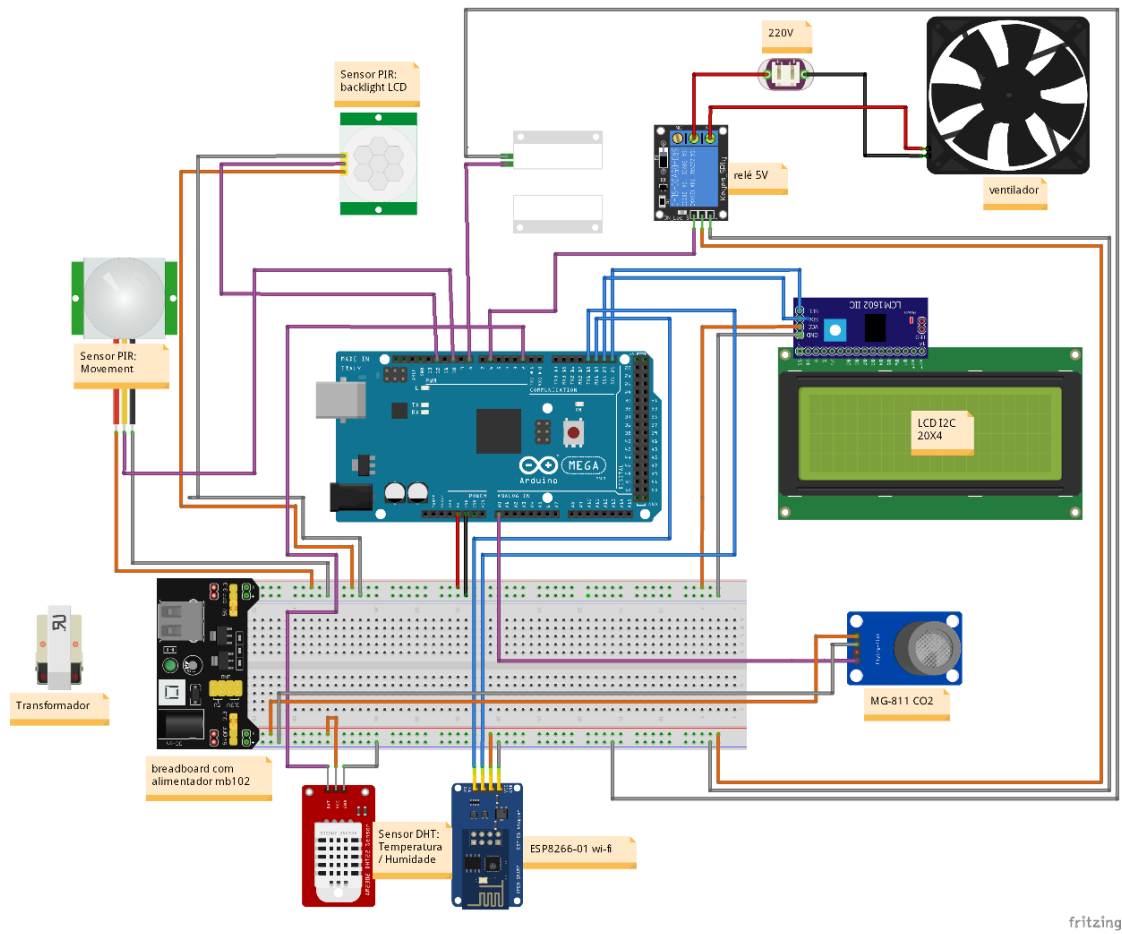
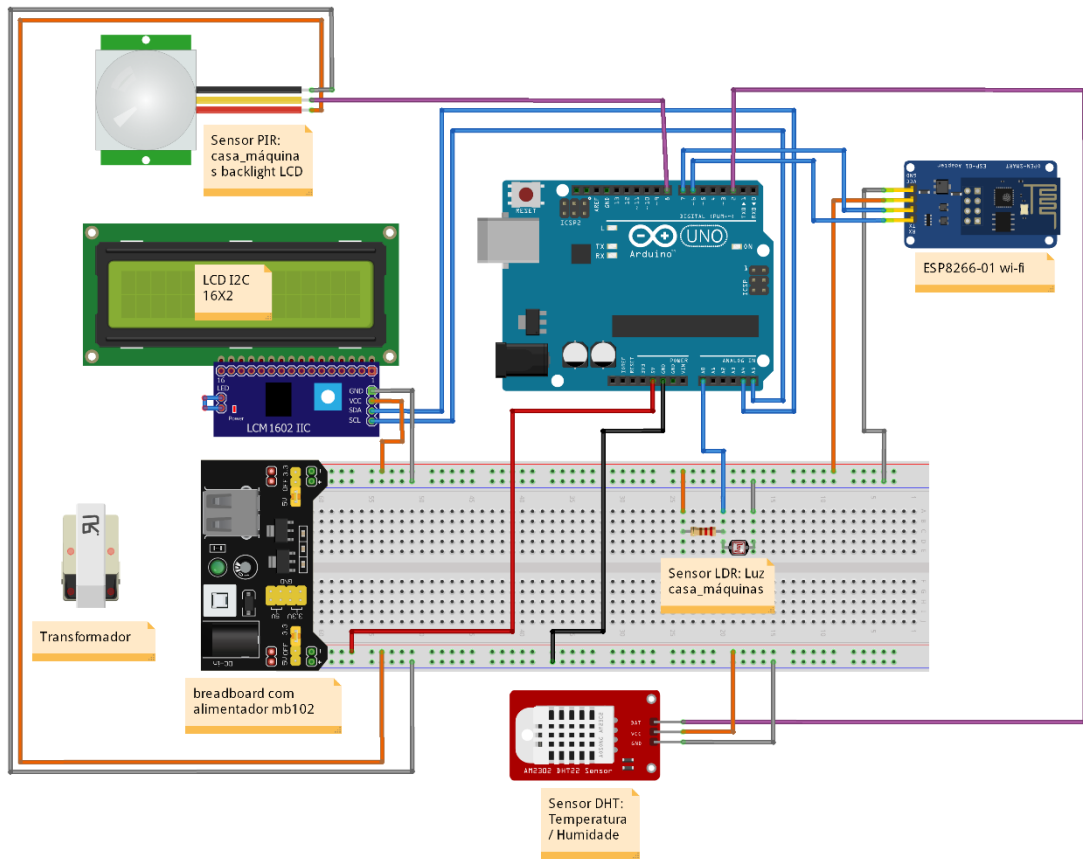


Figura 61 - Esquema circuito do sistema no espaço Recepção

Anexo C – Esquema do espaço Cais da Piscina



fritzing

Figura 62 - Esquema circuito do sistema no espaço Cais da Piscina

Anexo D – Esquema do espaço Casa das Máquinas

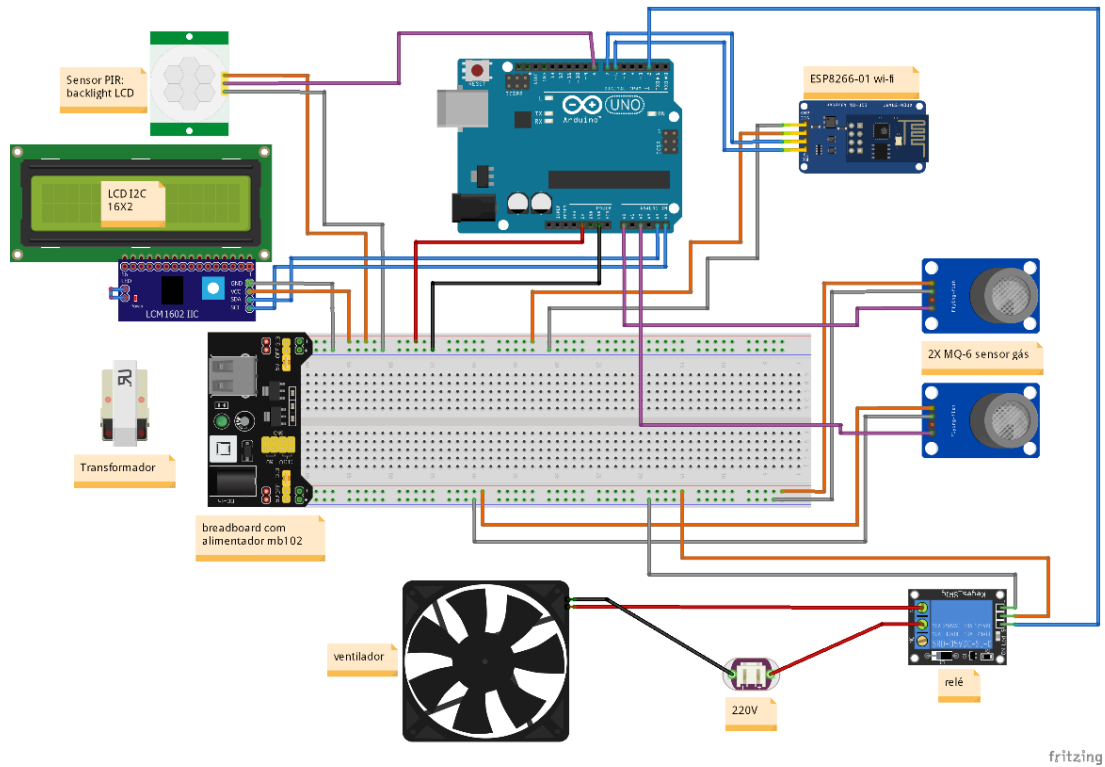


Figura 63 - Esquema circuito do sistema no espaço Casa das Máquinas