



UNIVERSIDADE
LUSÓFONA

Sistema para Monitorização de Plantas

Trabalho Final de curso

Relatório Final

Nome do Aluno: Tiago Bento do Ó Rodrigues - 22004242

Nome do Orientadores: João Pedro Leal Abalada de Matos Carvalho

João Pedro Calado Barradas Branco Pavia

Trabalho Final de Curso | LEI | 30/06/2023

Direitos de cópia

(*Sistema para Monitorização de Plantas*), Copyright de (*Tiago Bento do Ó Rodrigues*), ULHT.

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

Resumo

A expansão e a evolução das redes de sensores sem fios no âmbito da *Internet of Things* (IoT), que se tem verificado nos últimos anos mais, tornou este tipo de tecnologia apelativa e para a realização de tarefas de monitorização e controlo em diversas áreas, como a ambiental (interior e/ou exterior), da saúde e bem-estar.

A pesquisa bibliográfica efetuada permitiu identificar quais são as necessidades básicas de uma planta nomeadamente a água, luz e por fim o calor. Os fatores críticos do crescimento de uma planta são a luz, a humidade e a temperatura.

O presente trabalho final de curso visa propor o desenvolvimento de um sistema de controlo e monitorização de plantas baseado numa plataforma de IoT. O sistema assenta numa arquitetura de redes sem fios híbrida que integra dispositivos Bluetooth Low Energy (BLE), bem como nós sensores e nós atuadores. A comunicação entre as várias entidades da plataforma é assegurada através do protocolo MQTT (Message Queuing Telemetry Transport) tendo por base o modelo publish/subscribe.

A concretização deste projeto está dividida em várias etapas, das quais se destacam: a especificação da arquitetura enquadrada com as necessidades e funcionalidades desejadas; a seleção e programação de dispositivos para controlo e monitorização de equipamentos ligados a um Raspberry Pi; a implementação da base de dados local para o armazenamento de dados relevantes; a implementação de uma *gateway* (BLE-Wi-Fi), de forma a permitir a comunicação entre dispositivos com diferentes protocolos e por último o desenvolvimento de uma aplicação para controlar e monitorizar em tempo real os equipamentos através de uma interface gráfica.

O levantamento de requisitos técnicos definiu as grandezas físicas a ser monitorizadas nomeadamente luz, temperatura, humidade e fertilidade, tendo a escolha do sensor Xiaomi Flora ir ao encontro dos requisitos necessários. Para integrar o sensor e criar uma rede IoT foi necessário recorrer a hardware que suporte dispositivos Bluetooth LE como é o caso do minicomputador Raspberry Pi e a software sendo programas em linguagem Python com intuito de desenvolver comunicação com o hardware e obter visualização de dados com a utilização de protocolos como MQTT.

Palavras-Chave: IoT, Bluetooth LE, MQTT, Raspberry Pi, Python.

Abstract

The expansion and evolution of wireless sensor networks in the context of the Internet of Things (IoT), which has occurred in recent years, has made this type of technology appealing and to perform monitoring and control tasks in various areas, such as environmental (indoor and/or outdoor), health and welfare.

The bibliographical research carried out allowed us to identify the basic needs of a plant, namely water, light and finally heat. The critical factors for the growth of a plant are light, humidity and temperature.

This final course work aims to propose the development of a plant control and monitoring system based on an IoT platform. The system is based on a hybrid wireless network architecture that integrates Bluetooth Low Energy (BLE) devices, as well as sensor nodes and actuator nodes. The communication between the various entities of the platform is ensured through the MQTT protocol (Message Queuing Telemetry Transport) based on the publish/subscribe model.

The implementation of this project is divided into several stages, of which we highlight: the specification of the architecture framed with the needs and desired features; the selection and programming of devices to control and monitor equipment connected to a Raspberry Pi; the implementation of the local database for storing relevant data; the implementation of a gateway (BLE-Wi-Fi), to allow communication between devices with different protocols and finally the development of an application to control and monitor in real time the equipment through a graphical interface.

The survey of technical requirements defined the physical quantities to be monitored, namely light, temperature, humidity and fertility, having chosen the Xiaomi Flora sensor to meet the necessary requirements. To integrate the sensor and create an IoT network it was necessary to use hardware that supports Bluetooth LE devices such as the Raspberry Pi minicomputer and software in Python language in order to develop communication with the hardware and obtain data visualization using protocols such as MQTT.

Keywords: IoT, Bluetooth LE, MQTT, Raspberry Pi, Python.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores, Professor João Carvalho e Professor João Pedro Pavia, por toda a orientação, conselhos, disponibilidade que prestaram ao longo do desenvolvimento desta dissertação sempre com boa disposição. Também queria agradecer pela oportunidade de lecionar a cadeira Trabalho final de curso.

Em segundo lugar, um agradecimento à Escola Profissional Val do Rio em Oeiras por ter prestado apoio na distribuição de material técnico e pela colaboração na aquisição de novos requisitos considerados para este projeto.

À Universidade Lusófona Humanidades e Tecnologias e a todos os Professores que tive o gosto de conhecer e com quem tive o privilégio de aprender ao longo do meu percurso académico nesta instituição.

A todos os meus amigos e companheiros de curso que conheci durante todo este percurso, pelas aventuras passadas e pela grande amizade criada que me ajudou a ultrapassar os piores momentos destes anos. De todas as amizades criadas, quero destacar os meus grandes colegas e amigos com quem passei estes anos: Gonçalo Pedroso, Tiago Silva, Rodrigo Simões, João Antas, Tiago Águeda, Ruben Silva, Simão Bento, Tomás Fonseca e Gonçalo Soares. Em cada um de vocês levo uma amizade para a vida.

À minha família, que dirijo um dos meus maiores agradecimentos, pela educação e todo o apoio prestado. Em especial aos meus pais, agradeço por todos os sacrifícios e preocupações que tiveram de ultrapassar para que eu pudesse seguir com o meu percurso académico.

Índice

Resumo	3
Abstract.....	4
Agradecimentos	5
Índice	6
Lista de Figuras.....	8
Lista de Tabelas.....	10
Introdução.....	11
1 Identificação do Problema.....	13
2 Viabilidade e Pertinência	15
3 Benchmarking.....	17
4 Engenharia	22
4.1 Levantamento e análise dos Requisitos	22
4.1.1 Requisitos Funcionais	23
4.1.2 Requisitos Não Funcionais	25
4.1.3 Requisitos Técnicos	25
4.2 Diagramas de Casos de Uso	26
4.3 Modelos Revelantes	27
4.4 Estrutura da Aplicação.....	29
5 Solução Proposta.....	30
5.1 Introdução	30
5.2 Arquitetura	30
5.3 Implementação	32
5.4 Tecnologias e Ferramentas Utilizadas	33
5.5 Protótipo	39
5.6 Abrangência.....	41
6 Plano de Testes e Validação.....	42
7 Método e Planeamento.....	46
7.1 Descrição do Cronograma.....	47
7.1.1 Elaboração do Relatório	47
7.1.2 Desenvolvimento do Ambiente do Sistema	49

7.1.3	Fase de Testes do Sistema.....	50
7.2	Cumprimento do Planeamento	51
7.2.1	Esforço	54
7.2.2	Avaliação Crítica	55
8	Resultados.....	57
8.1	Estudo das Plantas.....	57
8.2	Medição das Plantas	58
8.3	Plataforma ThingSpeak.....	60
8.4	Decisão do Modelo em Machine Learning	63
9	Conclusão e trabalhos futuros.....	68
	Bibliografia	69
	Glossário	72

Lista de Figuras

Figura 1 – Aplicações de IoT	11
Figura 2 - App de um sistema rega automática	15
Figura 3 – Empresa SPS	17
Figura 4 – Solução Smart Plant Systems	17
Figura 5 - Solução Daisy.si Watering Device	18
Figura 6 - Solução Forja Cloud Plataform	19
Figura 7 - Hardware da Solução IAC Smart Plant	19
Figura 8 - Plataforma da Solução IAC Smart Plant	20
Figura 9 – APP Flower Care	20
Figura 10 – Diagrama de Caso de Uso	27
Figura 11 – Diagrama Entidade-Relação (ER)	28
Figura 12 – Diagrama UML	28
Figura 13 – Diagrama em Árvore	29
Figura 14 – Arquitetura do Sistema	31
Figura 15 – Raspberry Pi 3B	33
Figura 16 – Xiaomi Mi Flora	33
Figura 17 – Raspberry Pi Camera V2	34
Figura 18 – Raspberry OS	34
Figura 19 – Linguagem Programação Python	35
Figura 20 – Biblioteca Python Pandas	35
Figura 21 – Biblioteca Python scikit learn	36
Figura 22 – Módulo Bluez	36
Figura 23 – Exemplo de um ficheiro CSV	37
Figura 24 – MQTT protocolo com Broker	37
Figura 25 – Mosquitto MQTT Broker	38
Figura 26 – Exemplo de um Dashboard da plataforma ThingSpeak	38
Figura 27 – ThingSpeak	38
Figura 28 – Circuito eletrônico	40
Figura 29 – Esquema elétrico do circuito	40
Figura 30 – PCB do Circuito	41
Figura 31 – Cronograma Elaboração do Relatório em formato Gantt	46
Figura 32 – Cronograma Desenvolvimento do Ambiente do Sistema em formato Gantt	46
Figura 33 – Cronograma Fase de Testes do Sistema em formato Gantt	47
Figura 34 - Cumprimento da Elaboração do Relatório em formato Gantt	51
Figura 35 – Cumprimento do Desenvolvimento do Ambiente do Sistema em formato Gantt	52
Figura 36 – Cumprimento da Fase de Testes em Formato Gantt	52
Figura 37 – Planta 1	57
Figura 38 – Planta 2	57
Figura 39 – python planta1.py	58
Figura 40 – python planta2.py	58
Figura 41 – Dados da planta 1	59
Figura 42 – Dados da planta 2	59
Figura 43 – Canal da Planta 1 (Luz, Temperatura, Humidade e Fertilidade)	60
Figura 44 – Canal da Planta 1 (Bateria e Crescimento)	60
Figura 45 - Canal da Planta 2 (Luz, Temperatura, Humidade e Fertilidade)	61
Figura 46 – Canal da Planta 2 (Bateria e Crescimento)	61
Figura 47 – Crescimento da Planta 1	62
Figura 48 – Crescimento da Planta 2	62
Figura 49 - Alerta da Planta 2	63

Figura 50 – Gráfico Decision Tree	64
Figura 51 – Gráfico Gradient Tree Boosting	64
Figura 52 – Guassian Naive Bayes	65
Figura 53 – Gráfico Nearest Neighbors	65
Figura 54 – Gráfico Random Forest	66
Figura 55 – Gráfico Redes Neurais MLP	66
Figura 56 – Gráfico Support Vector Machine	67

Lista de Tabelas

Tabela 1 - Análise Benchmarking	21
Tabela 2 – Requisitos Funcionais Previstos	25
Tabela 3 – Requisitos não Funcionais	25
Tabela 4 – Requisitos Técnicos	26
Tabela 5 – Tabela de Testes e Validação	45
Tabela 6 – Cumprimento dos Requisitos	54

Introdução

A Internet das Coisas (IoT) descreve a rede de objetos físicos - "coisas" - que estão incorporados com sensores, software, e outras tecnologias com o objectivo de ligar e trocar dados com outros dispositivos e sistemas através da Internet. Estes dispositivos variam desde objectos domésticos comuns a ferramentas industriais sofisticadas. Com mais de 7 mil milhões de dispositivos de LCI ligados atualmente, os peritos esperam que este número aumente para 10 mil milhões até 2020 e 22 mil milhões até 2025 de acordo com Turber [Turber, S. (no date) *How technology affords organizing value creation in the digital era the evolution of thriving business models in nascent IOT categories.* dissertation.].

Nos últimos anos, a IoT tornou-se uma das tecnologias mais importantes do século XXI. Agora que podemos ligar objetos do quotidiano - aparelhos de cozinha, carros, termóstatos, monitores de bebé - à Internet através de dispositivos incorporados, é possível uma comunicação perfeita entre pessoas, processos e coisas. Através da computação de baixo custo, a nuvem, *big data*, análises e tecnologias móveis, as coisas físicas podem partilhar e recolher dados com um mínimo de intervenção humana. Neste mundo hiperligado, os sistemas digitais podem registar, monitorizar e ajustar cada interação entre as coisas ligadas [Figura 1 – Aplicações de IoT]. O mundo físico encontra o mundo digital - e eles cooperam.



Figura 1 – Aplicações de IoT

Atualmente, a participação de mercado da agricultura inteligente tem crescido exponencialmente em parte devido às rápidas mudanças ambientais, como condições climáticas extremas, deterioração do solo e falta de humidade do solo. Junto com a dinâmica imprevisível de nosso mundo atual, essas situações instáveis estão levando as propriedades agrícolas a lançar projetos de agricultura inteligente. Com sensores inteligentes que medem todos os aspectos do trabalho diário

automaticamente, a tecnologia IoT para agricultura permite que os agricultores automatizem a coleta de dados em tempo real para aumentar os volumes de produção, reduzir custos e gerenciar despesas e melhorar a eficiência geral em muitos diferentes aspectos da agricultura.

Face ao avanço tecnológico perante a agricultura industrial envolvidas na área da robótica, drones, e câmaras vigilância podemos comprovar que a nível de cultivo doméstico ou empresarial (floristas) não contem um grande mercado IoT pelo facto de haver desconhecimento destes conceitos sendo que este projeto pretende-se atingir um público-alvo jovem para o incentivo da utilização de sistemas informáticos que permitem monitorização das suas plantas com “verdadeiro objetivo” de cuidarem das suas plantas.

1 Identificação do Problema

A água é vital para todos os aspectos das nossas vidas e a utilização da água tem continuado a aumentar à medida que a produção de plantas se expande para satisfazer as necessidades alimentares de uma população em crescimento. A água é um bem natural precioso e é desperdiçada a níveis angustiantes. Só em ambientes domésticos, os relatórios da ONU [3] revelam fugas podem desperdiçar quase 1 trilião de garrafões de 30L de água anualmente. Este desperdício tem de ser evitado para assegurar a sustentabilidade e reduzir os custos.

Foi feita uma pesquisa através de um inquérito da AGROPORTUGAL a mais de 1.000 jardineiros portugueses. Esta revelou que todos os jardineiros perdem algumas das suas plantas todos os anos. Em média, os jardineiros disseram que perderam 35% das suas plantas, mas para aqueles que estão apenas a começar a sua jornada de jardinagem, esse número foi duas vezes mais elevado. A AGROPORTUGAL identificou que como fator decisivo para esta percentagem de mortes, a falta de monitorização das plantas onde os jardineiros não regavam com a quantidade de água necessária. Para além deste facto, chegou-se à conclusão que os mesmos não cumpriam com as condições necessárias de exposição a luz.

Vivemos num mundo onde tudo pode ser controlado e operado de forma remota, mas ainda assim existem alguns sectores importantes no nosso país onde a automatização não foi adotada ou não foi utilizada em pleno. A agricultura tem sido uma das principais ocupações do homem desde as primeiras civilizações e ainda hoje as intervenções manuais na agricultura são inevitáveis. A monitorização de plantas constitui uma parte importante dos sectores da agricultura e horticultura no nosso país, uma vez que podem ser utilizadas para cultivar plantas em condições climáticas controladas para uma produção ótima. Automatizar a monitorização e o controlo dos parâmetros climáticos que direta ou indiretamente regem o crescimento das plantas é a próxima evolução na área IoT.

Com este projeto pretende-se criar um sistema de monitorização de plantas que permita obter um maior controlo de o crescimento de plantas de forma evitar o desperdício de água que foi referido anteriormente na área doméstica. Para além disso este sistema tem também como objetivo reduzir a percentagem de mortes em plantas por falta de monitorização das mesmas.

Após a implementação do sistema e a realização de testes, os resultados obtidos foram analisados e comparados com as metas propostas inicialmente cumprindo com a redução do desperdício de água

pela monitorização numa plataforma simples como também a redução da taxa de mortalidade das plantas uma vez que esta plataforma permite detetar a falta de desempenho no “crescimento” da planta e alertar o utilizador perante a situação indicada.

É também importante mencionar pela obtenção dos resultados que posso comprovar uma boa eficiência de uso da água e a taxa de mortalidade das duas plantas estudadas, no entanto é fundamental reconhecer que existem limitações e desafios inerentes à natureza da agricultura e horticultura, os quais podem resultar em diferenças entre os resultados alcançados e as metas propostas inicialmente. O importante é buscar constantemente melhorias e soluções para maximizar o potencial de crescimento e sustentabilidade das plantas.

2 Viabilidade e Pertinência

O presente projeto dispõe de capacidades para garantir a sua viabilidade de desenvolvimento para além do âmbito académico, visto que existe a possibilidade deste projeto ser utilizado de forma conjunta com outros sistemas informáticos que procurem automatizar tarefas relacionadas com o cultivo de plantas.

Como referido anteriormente, nesta fase o sistema tem como principal objetivo responder às necessidades de monitorização e controlo de plantas reduzindo assim o esforço e o tempo despendido em interações entre os utilizadores perante o seu dia a dia no cuidado das suas plantas. Também teremos um objetivo secundário de obter uma melhor performance no crescimento das plantas do utilizador, ou seja, melhorar o tempo de crescimento médio de uma planta doméstica ao cuidado de uma pessoa segundo os estudos de White [White, A.C. (no date) Why do some plants grow faster than others? A source-sink perspective. dissertation.] assim como aumentar a esperança de tempo de vida destas. Com isto, os utilizadores poderão aplicar este tempo otimizado na realização das suas tarefas pessoais e assim obter um melhor aproveitamento no crescimento das plantas.

A metodologia aplicada no desenvolvimento deste projeto permite que o mesmo seja altamente personalizável e adaptável a qualquer planta estudada permitindo assim um uso mais universal. A utilização de um sistema informático com servidor local pode também ser um conceito a ser melhorado, existindo a possibilidade de exportar todos os dados para os serviços *Cloud* e para melhorar a eficiência da arquitetura recomenda-se o desenvolvimento de uma APP para a visualização do controlo das plantas em qualquer lugar.



Figura 2 - App de um sistema rega automática

Em relação ao âmbito proposto inicialmente, o sistema alcançou os objetivos principais de monitorização e controle das plantas, reduzindo o

esforço e o tempo necessário para interagir com elas no dia a dia. Além disso, também se buscou melhorar o desempenho do crescimento das plantas, reduzindo o tempo médio de crescimento com base em estudos anteriores e aumentando sua expectativa de vida.

Perante a integração de sistemas informáticos conclui que a solução tem a capacidade se integrar com outros sistemas, é importante mencionar que a implementação real dessa integração pode depender de fatores como compatibilidade tecnológica, disponibilidade de recursos e requisitos específicos dos sistemas a serem integrados exigindo sempre adaptações adicionais no código principal.

No desenvolvimento inicial propôs o uso de um servidor local para armazenar os dados do sistema. No entanto, na análise comparativa identifico que a plataforma permite a exportação desses dados para serviços em nuvem, visando melhorar a eficiência da arquitetura. Essa evolução tecnológica pode permitir um acesso mais fácil aos dados de qualquer lugar por meio de uma aplicação (APP) dedicada, proporcionando maior conveniência e flexibilidade para os utilizadores como podemos verificar pela secção de resultados uma ferramenta externa que consegue guardar esses mesmo dados e ilustrar para os utilizadores em forma de “gráficos”.

3 Benchmarking

Com uma breve análise de mercado é possível averiguar que atualmente existe alguns concorrentes no que diz respeito a sistemas de monitorização / controlo de plantas. Uma vez que é uma área pouco desenvolvida procuramos analisar os aspetos mais diferenciadores dos sistemas procurados e refletir melhorias na nossa solução.

Vamos tomar como exemplo soluções como a “SPS (Smart Plant Systems)”, a “Daisy.si Watering Device”, “Florja Smart Plant Care” , a “IAC Smart Plant Remote Monitor” e a “Flower Care”

“SPS” é uma empresa que disponibiliza uma sala inteligente onde os utilizadores contendo as plantas dessa sala podem obter as informações das suas plantas em tempo real.



Figura 3 – Empresa SPS



Figura 4 – Solução Smart Plant Systems

A solução oferecida aposta muito na representação visual dos dados [Figura 4] juntamente com o facto da solução estar adaptada para o mobile e suporta para os serviços Cloud faz com que esta solução seja muito forte

“Daisy.si” é uma empresa que disponibiliza um “regador” automático para plantas exterior e interiores funcionando como rega automática de flores/plantas. Este dispositivo inteligente lê a humidade do solo, temperatura, brilho e rega através de um algoritmo avançado, capaz de definir a quantidade de água ideal que a planta necessita.

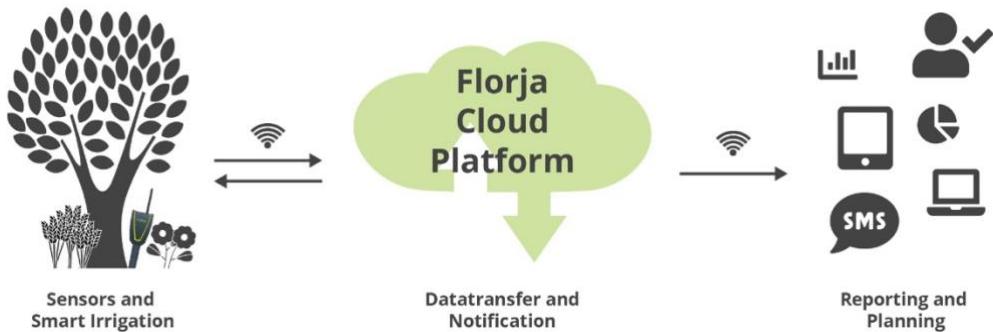
A solução oferecida aposta muito na área da automatização sendo uma solução muito diferenciada das outras soluções propostas pelo facto automatizar a rega das plantas do utilizador facilitando o controlo das mesmas. Não esquecer que esta solução contém uma restrição que o dispositivo da “Daisy.si Watering Device” [Figura 5] necessita estar ligado a um tanque de água para o seu funcionamento de rega.



Figura 5 - Solução Daisy.si Watering Device

“Florja” é uma empresa que disponibiliza uma plataforma de gerenciamento de plantas baseada em nuvem que permite aos usuários monitorizar o estado atual de suas plantas de qualquer lugar e a qualquer momento. A requisição dos sensores para a leitura das plantas não pertence ao sistema apenas a Florja Smart Plant Care funciona uma plataforma externa para visualização dos dados das plantas em tempo real.

A solução oferecida aposta muito na representação visual dos dados [Figura 6] juntamente com o facto da solução estar adaptada para o mobile e suporta para os serviços Cloud mas necessita de um sistema externo que envolva sensores compatíveis com a plataforma fornecida para o seu funcionamento.

**Figura 6 - Solução Forja Cloud Platform**

“IAC” é uma empresa que disponibiliza é um sistema de IoT industrial baseado em nuvem, que permite ao pessoal monitorizar remotamente as suas plantas em todo mundo. Este sistema conhecido como o “IAC Smart Plant” envolve a instalação de um dispositivo [Figura 7] que integra vários sensores de temperatura, humidade, fertilidade e entre outros que envia os dados recolhidos em tempo real a uma *Cloud* que pode ser acedida em qualquer lugar por uma plataforma online.

A solução oferecida aposta muito na representação visual dos dados [Figura 8] juntamente com o facto da solução estar adaptada para o mobile e suporta para os serviços Cloud como também usa apenas um pequeno dispositivo para extração de dados o que faz com que esta solução seja muito forte.

**Figura 7 - Hardware da Solução IAC Smart Plant**

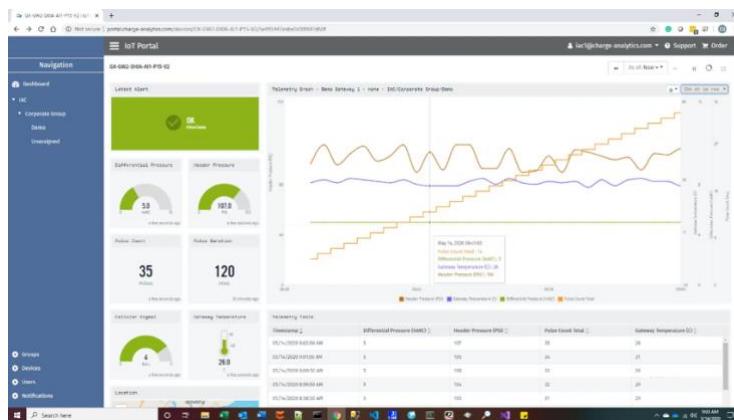


Figura 8 - Plataforma da Solução IAC Smart Plant

“Flower Care” é uma aplicação para dispositivos móveis desenvolvida pela Beijing HHCC Plant Technology Co., Ltd [Figura 9] sendo disponível nas seguintes lojas oficiais : Google Play e AppStore que permite monitorizar e controlar plantas domesticas.

Esta solução na área de computação móvel contém um grande destaque por ser uma solução diferente dos outros concorrentes até mesmo do TFC pois permite para além de fornecer informações sobre o nível de humidade do solo, fertilidade, temperatura e luz também ajustar as condições de cultivo para atender às necessidades específicas de cada planta. Para ajudar estas condições de cultivo o utilizador deve inserir a espécie da sua planta sendo a que app contém uma larga base dados de espécies de plantas beneficiando na adaptação da atribuição dos valores ambientais mínimos para o crescimento da planta em si.

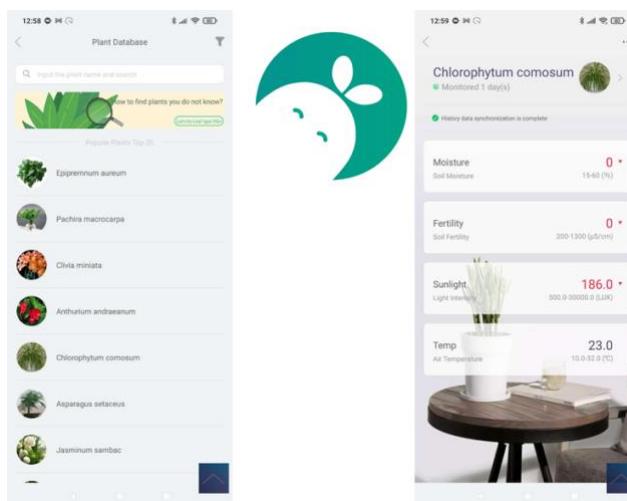


Figura 9 – APP Flower Care

É necessário ter em consideração que existem mais soluções no mercado, mas que a maioria, tal como as apresentadas acima tem um custo associado ao uso para além da aquisição. Por vezes algumas destas soluções são fornecidas apenas como um serviço o que requer, muitas das vezes, que haja necessidade de aquisição de serviços de manutenção ou serviços extras funcionalidades destas plataformas.

Com a solução apresentada por nós tentamos responder às necessidades de utilizadores finais domésticos, dando uso do que de melhor estes tipos de soluções oferecem, permitindo assim com que o produto final seja um sistema simples de usar e com baixos custos associados.

Na seguinte tabela [Tabela 1] é possível analisar de uma forma mais sistemática algumas funcionalidades dos softwares existentes no mercado, acima detalhados, versus o que é proposto que a aplicação web a ser desenvolvida venha a ter.

	Visualiza ção Dados das Plantas em tempo real	Visualizaç ão do “Crescim ento” das plantas através	Configuraç ão da rega “automáti ca” a” pela análise de dados	Sistema com Sensores de Leitura das Plantas Incluído	Transmiss ão dos dados das plantas por Bluetooth	Protocolo MQTT comunicação entre servidor e os sensores	Aplicação de Modelo de Classificação de Machine Learning
SPS	✓	✗	✗	✓	✓	✓	✗
Daisy	✗	✗	✓	✓	✗	✗	✗
Florja	✓	✓	✗	✗	✓	✓	✗
IAC	✓	✓	✗	✓	✓	✓	✗
SMP TFC	✓	✓	✓	✓	✓	✓	✓
Flower Care	✓	✓	✓	✓	✓	✓	✗

Tabela 1 - Análise Benchmarking

4 Engenharia

Os sistemas informáticos atuais têm como objetivo atender às necessidades dos utilizadores individuais ou de organizações. Para que esses sistemas sejam bem-sucedidos, é essencial criar uma ligação profunda entre os desenvolvedores e os utilizadores. Por isso, é crucial definir corretamente os requisitos do sistema, pois eles servem como uma orientação para gerentes de projeto, programadores, especialistas em sistemas e a equipa de qualidade durante o desenvolvimento do projeto

4.1 Levantamento e análise dos Requisitos

Portanto, no início deste projeto, procurei identificar quais requisitos precisariam ser implementados e cumpridos para que o sistema atendesse às necessidades dos nossos utilizadores e realizasse plenamente a visão que eu tinha concebido para o seu estado final.

De seguida foi enumerado os Requisitos Funcionais [Tabela 2], Requisitos Não Funcionais [Tabela 3] e os Requisitos Técnicos [Tabela 4] que atendem tanto aos objetivos estabelecidos discutidos por mim e pelos orientados como também às necessidades previsíveis perante os utilizadores atribuindo uma escala de tamanhos , uma escala ‘Have’ e por último os Story Points.

4.1.1 Requisitos Funcionais

Requisito	Escala de Tamanhos	Escala "Have"	Story Points (0-10)	Implementado (✓ ou X)
RF1 - O sistema deverá permitir a conectividade de sensores e câmeras num circuito eletrónico.	XS	must have	1	✓
RF2 - O sistema deverá permitir que os scripts de programação tenham acesso direto aos componentes ligados tais como sensores e câmeras ao circuito eletrónico.	M	must have	2	✓
RF3 - O sistema deverá a realizar a medição dos parâmetros ambientais das plantas tais como a temperatura, luz, humidade e fertilidade.	S	must have	1	✓
RF4 - O sistema deverá tirar diariamente fotografias das plantas consoante a medição dos dados ambientais delas.	L	could have	4	✓
RF5 - O sistema deverá permitir a utilização de scripts de automatização para inicialização sem intervenção do utilizador.	S	nice to have	1	✓
RF6 - O sistema deverá armazenar os dados ambientais medidos das plantas e as respetivas localizações das fotografias mantendo um histórico de dados num ficheiro CSV.	L	must have	5	✓
RF7 - O sistema deverá armazenar as fotografias das plantas em formato PNG numa pasta local contendo acesso remoto do mesmo.	S	must have	3	X
RF8 .O sistema deverá permitir que os scripts de programação tenham acesso ao ficheiro CSV e a pasta local que contem as fotografias.	XS	should have	1	✓
RF9 - O sistema deverá permitir a implementação de algoritmos de machine learning para o analise de dados do crescimento das plantas.	XL	should have	10	✓
RF10 - O sistema deverá calcular uma estimativa de restante tempo de vida das plantas consoante a espécie da planta.	XL	nice to have	10	✓
RF11 - O sistema deverá calcular a quantidade de água de necessária para a rega das plantas para um bom desenvolvimento no crescimento das mesmas.	L	should have	7	✓

RF12 - O sistema deverá calcular a temperatura ideal para um bom crescimento das plantas consoante a espécie da planta.	L	should have	5	✓
RF13 - O sistema deverá calcular o tempo necessário de exposição das plantas ao sol diariamente consoante a espécie da planta.	XL	could have	9	✓
RF14 - O sistema deverá calcular um intervalo de valores da humidade necessários para que as plantas continuem vivas.	L	nice to have	6	✓
RF15 - O sistema deverá calcular um intervalo de valores da fertilidade do solo necessários para que as plantas continuem vivas.	XL	nice to have	8	✓
RF16 - O sistema deverá permitir a exportação dos dados plantas analisados para uma plataforma de monitorização remota.	L	must have	6	✓
RF17 - O sistema deverá permitir a exportação dos cálculos feitos pelos algoritmos para uma plataforma de monitorização remota.	M	must to have	4	✓
RF18 - O sistema deve permitir criar avisos sobre as temperaturas das plantas se estas estiverem abaixo do recomendado na plataforma de monitorização de plantas.	M	nice to have	4	✓
RF19 - O sistema deve permitir criar avisos sobre a humidade das plantas se esta estiver abaixo do recomendado na plataforma de monitorização de plantas.	M	nice to have	5	✓
RF20 - O sistema deve permitir criar avisos sobre a fertilidade do solo das plantas se esta estiver abaixo do recomendado na plataforma de monitorização de plantas.	M	nice to have	5	✓
RF21 - O sistema deve permitir criar avisos sobre o tempo exposição de luz das plantas se esta estiver abaixo do recomendado na plataforma de monitorização de plantas.	M	nice to have	5	✓
RF22 - Os utilizadores devem ter uma conta registada na plataforma ThingSpeak associado ao sistema.	S	must have	2	✓
RF23 - Os utilizadores devem poder visualizar os dados ambientais das suas plantas em tempo real através de ilustração de valores.	L	must have	6	✓
RF24 - Os utilizadores devem poder visualizar o histórico dos dados ambientais	XL	should have	7	✓

das suas plantas através de utilização de gráficos.				
-----------------------------------------------------	--	--	--	--

Tabela 2 – Requisitos Funcionais Previstos**4.1.2 Requisitos Não Funcionais**

Requisito	Escala “Have”
RNF1 - O sistema deverá ter a capacidade de ler e armazenar os dados do sensor com velocidade adequada.	should have
RNF2 - O sistema deverá ter a capacidade de tirar fotografias com uma qualidade aceitável.	must have
RNF3 - O sistema deve ser confiável e estável, com uma taxa de falhas minoritária.	should have
RNF4 - O sistema deve ser seguro para proteger contra acesso não autorizado e violações de privacidade.	could have
RNF5 - O sistema deve ser projetado para ser facilmente transportável e instalado em outros locais.	should have
RNF6 - O sistema deve conter facilidade de utilização com uma interface intuitiva e fácil de navegar.	nice to have
RNF7 - O sistema deve conter a compatibilidade com outros dispositivos informáticos que tenham acesso a internet com recurso a interface gráfica.	must have
RNF8 - O sistema deve ser projetado para ser fácil de manter e atualizar.	could have
RNF9 - O sistema deve ser projetado para ter uma disponibilidade elevada, de forma a garantir que ele está disponível quando é necessário	must have

Tabela 3 – Requisitos não Funcionais**4.1.3 Requisitos Técnicos**

Requisito	Escala “Have”
RT1 - O sistema necessita obrigatoriamente de conexão à internet para se conectar a plataforma de monitorização de plantas Thingboard.	must have
RT2 - O hardware do sistema deve ser composto por um Raspberry Pi com conectividade WiFi e Bluetooth, por um sensor Xiaomi mi Flora e por uma câmara module 2 Raspberry	must have
RT3 - O sistema operativo do sistema deverá ser Raspberry Pi OS com a linguagem de programação Python instalada devidamente.	must have
RT4 - O sistema deverá ter instalado um MQTT broker para envio de dados através do protocolo MQTT à plataforma de monitorização.	must have
RT5 - O sistema deve incluir todas as bibliotecas e módulos dependentes do Python para o funcionamento do sensor Xiaomi Mi Flora e a câmera Pi v2.	must have
RT6 - O sistema deve incluir todas as bibliotecas e módulos dependentes do Python para o funcionamento dos algoritmos de machine learning utilizados.	must have
RT7 - O sistema deve incluir uma capacidade armazenamento suficiente sendo no mínimo 8GB para armazenar os dados e as fotografias das plantas.	should have
RT8 - O sistema deve incluir a última versão atualizado da linguagem de programação Python	should have
RT9 - O sistema deve incluir medidas de segurança para proteger contra o acesso não autorizado e violações de privacidade mantendo os dados da conta principal do sistema com a conta principal da plataforma de monitorização.	nice to have
RT10 - O sistema deverá estar a recorrer automaticamente o software desenvolvido após ser iniciado pela ferramenta system dameon service que esta incluído nos sistemas operativos Linux.	nice to have
RT11 - Os utilizadores devem configurar as suas contas na plataforma ThingSpeak a ligação ao sistema de monitorização de plantas pelo um token genérico do sistema.	must have

Tabela 4 – Requisitos Técnicos

4.2 Diagramas de Casos de Uso

Pelo diagrama de casos de uso é possível documentar que o sistema faz do ponto de vista do utilizador, por outras palavras, este diagrama descreve as principais funcionalidades do sistema a interação dessas funcionalidades com os utilizadores do mesmo sistema. Contudo foi pedido um diagrama caso de uso para demonstrar as interações entre o sistema e os utilizadores

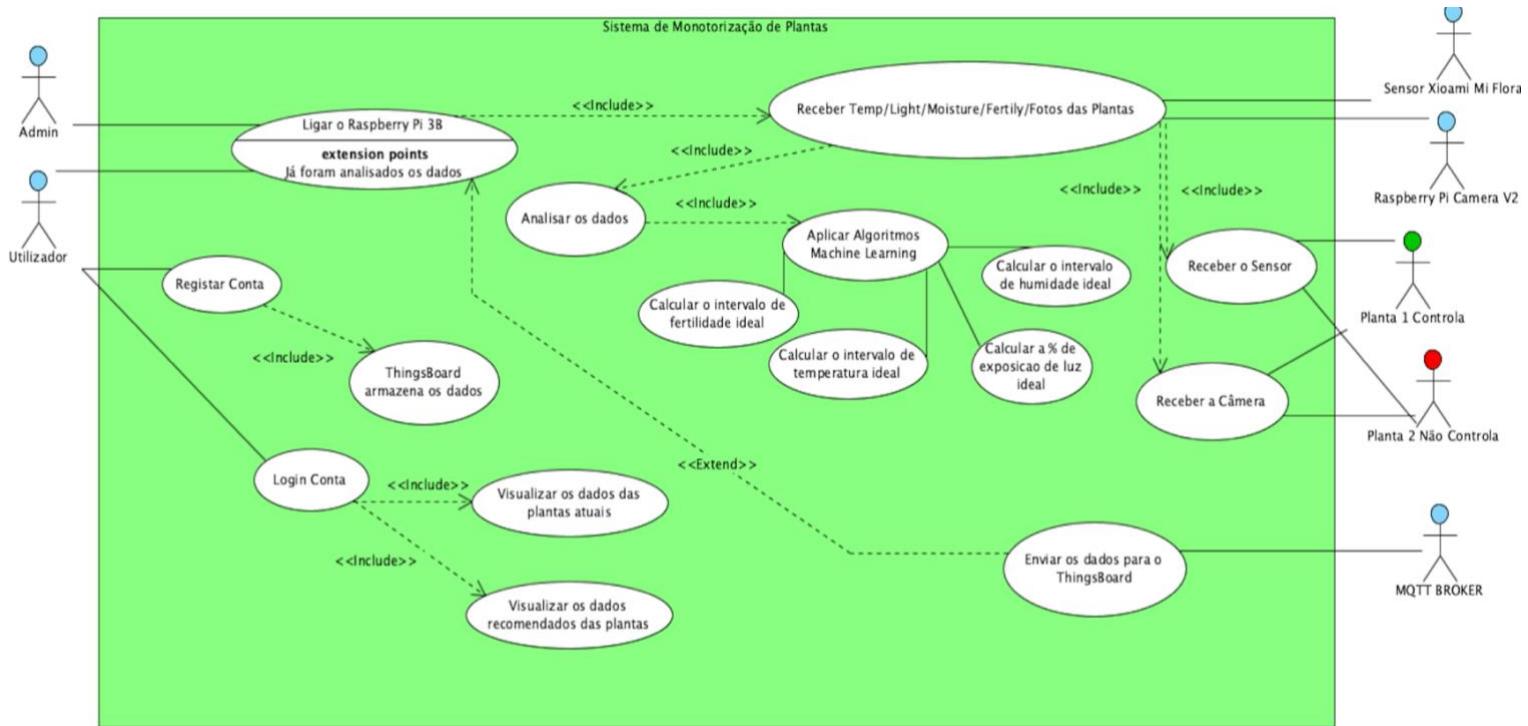


Figura 10 – Diagrama de Caso de Uso

4.3 Modelos Revelantes

Um diagrama entidade-relação (ER) consiste num tipo de fluxograma que ilustra como “entidades” onde podem ser pessoas, objetos e entre outros que se relacionam entre si dentro de um sistema. Também conhecidos como modelos ER, usam um conjunto definido de símbolos, tais como retângulos, diamantes, ovais e linhas de conexão para representar a conectividade de entidades, relacionamentos e seus atributos. Eles espelham estruturas gramaticais, onde entidades são substantivos e relacionamentos são verbos.

Para ilustrar a relação entre as entidades do nosso sistema foi feito um diagrama de Entidade-Relação como se pode verificar na [Figura 11]

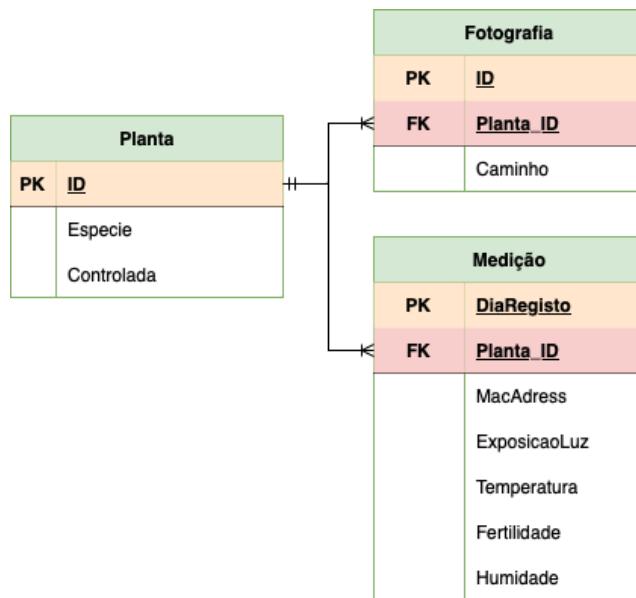


Figura 11 – Diagrama Entidade-Relação (ER)

O diagrama de classe UML (Unified Modeling Language) é uma representação gráfica da estrutura de uma aplicação, mostrando as classes, atributos, métodos e relações entre eles. Ele é amplamente utilizado na modelagem de sistemas orientados a objetos e ajuda a visualizar e comunicar a arquitetura do sistema. Este sistema inicialmente não contém uma estrutura de programação de objetos orientados, mas para o melhor entendimento da programação deste sistema foi realizado um diagrama UML na mesma [Figura 12]



Figura 12 – Diagrama UML

4.4 Estrutura da Aplicação

Um diagrama em árvore é uma representação gráfica que mostra a estrutura hierárquica de um sistema [Figura 13]. Ele é composto por nós e ramos, onde cada nó representa um componente ou elemento do sistema e os ramos mostram as relações entre esses componentes. Os diagramas em árvore são amplamente utilizados para representar e visualizar a arquitetura de sistemas complexos. Ele pode ser útil para entender como os componentes se relacionam e como eles trabalham juntos para atingir o objetivo geral do sistema.

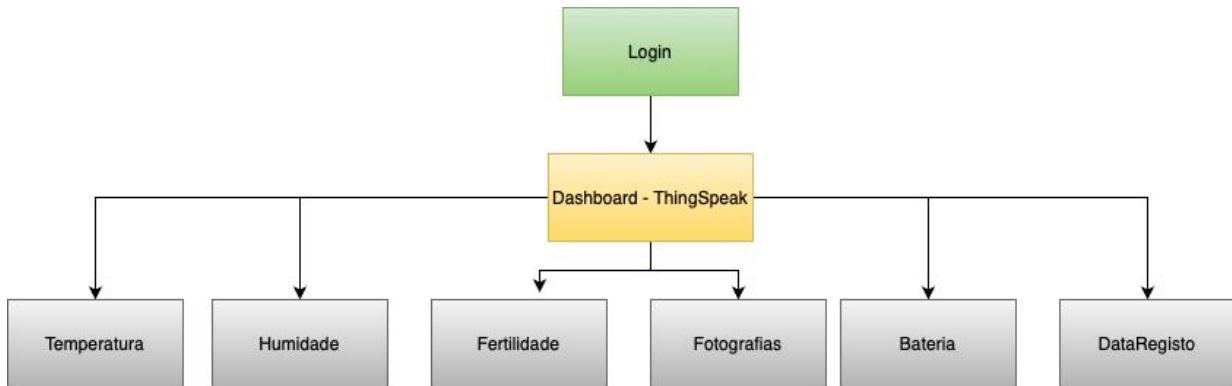


Figura 13 – Diagrama em Árvore

5 Solução Proposta

5.1 Introdução

A solução que se propõe consiste no desenvolvimento de um circuito eletrónico de baixo custo e de fácil utilização, que se capaz de funcionar como plataforma de monitorização de plantas, mas em simultâneo trazer funcionalidades de análise face aos dados recolhidos das plantas.

A solução será dividida em:

1. Montagem do Circuito Eletrónico
2. Desenvolvimento do Programa em Python
3. Incorporação de uma base de dados local
4. Implementação de Machine Learning
5. Desenvolvimento da Plataforma IoT

Para o entendimento da solução proposta é apresentado um link com acesso a um vídeo demonstrativo para explicar o funcionamento do ambiente de desenvolvimento como também para quem quer saber mais do desenvolvimento de programação contem também um link direcionado ao repositório git.

Vídeo demonstrativo no Youtube:

https://www.youtube.com/watch?v=e0ltE9nw_BA

Repositório git no Github:

<https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI270-Sistema-Monitorizacao-Plantas>

5.2 Arquitetura

A arquitetura de um sistema define a articulação lógica dos componentes e serviços em software sem estabelecer uma escolha definitiva sobre a maneira do sistema. Trata-se da definição da base estrutural. A arquitetura de um sistema é importante porque influencia no desempenho, escalabilidade, manutenibilidade e segurança do sistema. Uma boa arquitetura permite que o sistema seja escalado facilmente para lidar com cargas crescentes, seja fácil de manter e atualizar, e seja seguro contra ameaças externas. Além disso, uma arquitetura bem projetada pode ajudar a garantir que o sistema atenda às necessidades dos utilizadores e seja fácil de usar. Por outro lado, uma má arquitetura pode levar a problemas de desempenho, dificuldades na manutenção e problemas de segurança, o que pode afetar negativamente a experiência do utilizador e a confiabilidade do sistema.

Tendo em conta a definição e importância de uma arquitetura de um sistema, desenvolvi um desenho [Figura 14] que ilustra a implementação da solução referida anteriormente.

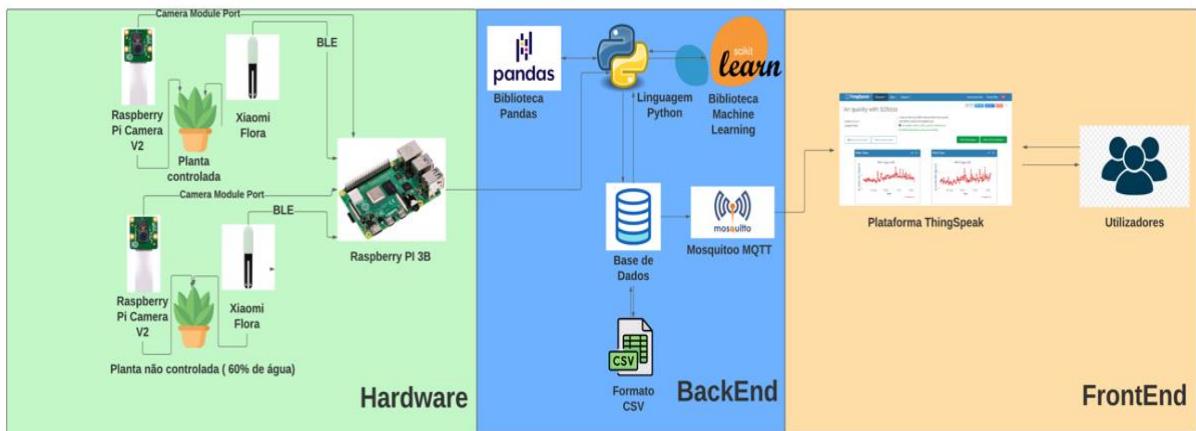


Figura 14 – Arquitetura do Sistema

Podemos observar que a arquitetura do sistema de monitorização de plantas é baseada em 3 camadas:

- **Hardware:** esta camada refere-se aos componentes físicos do sistema, incluindo sensores, dispositivos de aquisição de dados, computadores, dispositivos de comunicação e outros equipamentos necessários para extraír, processar e transmitir informações sobre as plantas monitoradas. Estes componentes trabalham juntos para fornecer informações precisas e em tempo real sobre as condições das plantas, permitindo que as próximas camadas desta arquitetura possam tomar as melhores decisões perante o crescimento das plantas.
- **BackEnd:** esta camada responsabiliza-se pelo software do sistema atrás da interface do utilizador com capacidade de gerir e processar os dados extraídos dos dispositivos na camada do hardware. Inclui uma base dados para armazenar os dados processados das aplicações e serviços que consideraram como uteis, ou seja, são aplicadas técnicas para extração/processamento dos dados necessários. É também uma camada que inclui logicas de negócios e interações com outros sistemas em tempo real para visualização dos mesmos na próxima camada.
- **FrontEnd:** é uma camada responsável por fornecer a interface do utilizador para o sistema incluído aparência e a funcionalidade da própria interface tal como menus, gráficos e outros elementos iterativos que permitem os utilizadores interagirem com os dados processados na camada anterior. O FrontEnd pode também incluir recursos de notificações ou relatórios perante os dados

visualizados uma vez que é muito importante obter informações sobre o crescimento das suas plantas.

5.3 Implementação

A solução vai ser desenvolvida recorrendo à tecnologia Python [Figura 19]. Esta uma linguagem de alto nível lançada no começo dos anos 1990, muito popular atualmente, e que suporta tanto programação orientada a objetos como programação estruturada. A escolha desta tecnologia, para além da sua versatilidade vai também permitir a utilização/configuração do hardware necessário para o acompanhamento das plantas e futuramente implementação de IoT e Data Science com as bibliotecas dedicadas.

Sendo que a solução se baseia no conceito de um sistema de monitorização de plantas, o Raspberry Pi 3 Model [Figura 15] torna-se um excelente candidato para tirar o conceito do papel e trazer para o mundo prático. Este minicomputador com todos os componentes integrados numa só placa contém as ligações do Sensor Xiaomi mi Flora [Figura 16] e do Raspberry Pi Camera Module2 [Figura 17] como também software pré-instalado para o desenvolvimento deste projeto nomeadamente a linguagem de programação Python referido anteriormente.

Uma vez que o conceito desta solução é baseado em Python, a utilização dos componentes adicionais ao Raspberry Pi referidos anteriormente é estritamente necessária a aquisição das bibliotecas miflora e bluepy.

A plataforma de monitorização em si vai ser desenvolvida com a maior personalização possível utilizando a tecnologia ThingsSpeak [Figura 27] como o exemplo da [Figura 26] que provém da implementação de Machine Learning dos dados guardados num ficheiro local enviados por um servidor Mosquito MQTT [Figura 25] através dos métodos Publish/Subscribe configurados em Python.

5.4 Tecnologias e Ferramentas Utilizadas

Nesta seção irei aprofundar o conhecimento sobre as tecnologias utilizadas na solução proposta como também as ferramentas utilizadas. A apresentação das tecnologias e ferramentas é por de construção do projeto em questão cumprindo assim com o desenvolvimento da arquitetura definida.

Hardware:

- **Raspberry Pi 3B**: é um computador de placa única desenvolvido pela Fundação Raspberry Pi. Tem um CPU ARM Cortex-A53 quad-core de 64 bits de 1.2GHz, 1GB de RAM, e 802.11n sem fios e Bluetooth 4.1 incorporado. Tem também 40 pinos GPIO, 4 portas USB, uma porta HDMI, e uma porta Ethernet. Pode executar o sistema operativo Raspbian, bem como outras distribuições Linux, e é normalmente utilizado para projectos IoT.

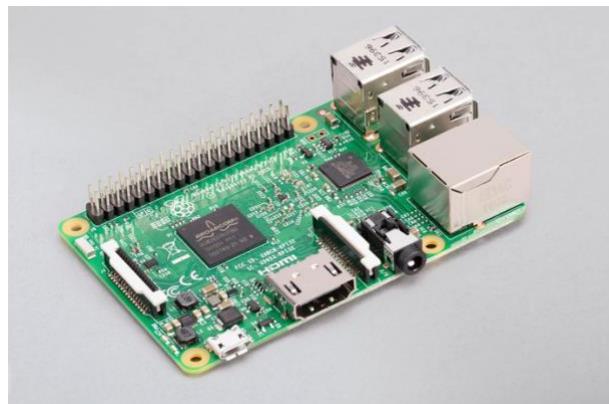


Figura 15 – Raspberry Pi 3B

- **Xiaomi Mi Flora**: O Xiaomi Flora é um dispositivo Bluetooth Low Energy que monitora o ambiente de uma planta. Ele mede: intensidade da luz solar, temperatura da planta, humidade do solo, fertilidade do solo (via condutividade elétrica).



Figura 16 – Xiaomi Mi Flora

- **Raspberry Pi Camera v2:** É um módulo de câmera que possui um sensor Sony IMX219 de 8 megapixéis com uma ligação direta ao raspberry pela entrada CSI.



Figura 17 – Raspberry Pi Camera V2

- **Raspberry OS:** É um sistema operativo do tipo Unix baseado na distribuição Debian Linux para a família Raspberry Pi de computadores compactos de placa única.



Figura 18 – Raspberry OS

BackEnd:

- **Linguagem Programação Python:** O Python é uma linguagem de programação amplamente usada em aplicações da Web, desenvolvimento de software, ciência de dados e machine learning (ML). Os desenvolvedores usam o Python porque é eficiente e fácil de aprender e pode ser executada em muitas plataformas diferentes



Figura 19 – Linguagem Programação Python

- **Biblioteca Pandas:** Pandas é uma biblioteca para uso em Python, open-source e de uso gratuito (sob uma licença BSD), que fornece ferramentas para análise e manipulação de dados



Figura 20 – Biblioteca Python Pandas

- **Biblioteca scikit learn :** O scikit-learn é uma biblioteca da linguagem Python desenvolvida especificamente para aplicação prática de Machine Learning. Esta biblioteca dispõe de ferramentas simples e eficientes para análise preditiva de dados, é reutilizável em diferentes situações, possui código aberto, sendo acessível a todos e foi construída sobre os pacotes NumPy, SciPy e matplotlib.



Figura 21 – Biblioteca Python scikit learn

- **Biblioteca miflora:** biblioteca que permite ler os dados a partir do sensor Xiaomi Flora
- **Module bluez:** modulo de extensão Bluetooth Low que permite que os desenvolvedores usem os recursos do sistema BLE para ligar qualquer dispositivo que usem a conexão por BLE

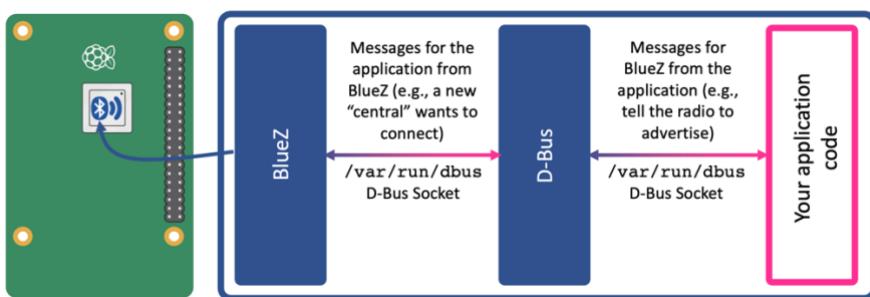


Figura 22 – Módulo Bluez

- **Formato CSV:** é um ficheiro sem formatação em que os valores estão separados por vírgulas, delimitados por aspas e cada linha tem um registo diferente.

```

January 2019.csv - Notepad
Report generated on 01-01-2020,,
Created by: user9284,,
Company XYZ,,,
Date,Country,Units,Revenue
2019-01-08,USA,343,15461.36
2019-01-04,Panama,93,4681.26
2019-01-07,Panama,42,2220.36
2019-01-16,Brazil,103,1853.78
2019-01-17,USA,28,286.3
2019-01-24,Canada,372,24826.98
2019-01-26,Canada,61,1592.42
2019-01-28,Canada,264,3228.11
2019-01-13,Canada,27,257.97
2019-01-28,Brazil,323,3024.25

```

Figura 23 – Exemplo de um ficheiro CSV

- **MQTT:** um protocolo de mensagens leve para sensores e pequenos dispositivos móveis otimizado para redes TCP/IP. Este protocolo é útil para transferências de dados de dispositivos IoT que utilizam Bluetooth Low que é o caso dos sensores utilizados neste projeto.
- **Mosquitto MQTT:** é uma ferramenta que permite gerir as publicações ou subscrições do protocolo MQTT permitindo a comunicação entre um servidor MQTT com um cliente de uma plataforma de IoT que é o caso da ThingSpeak.

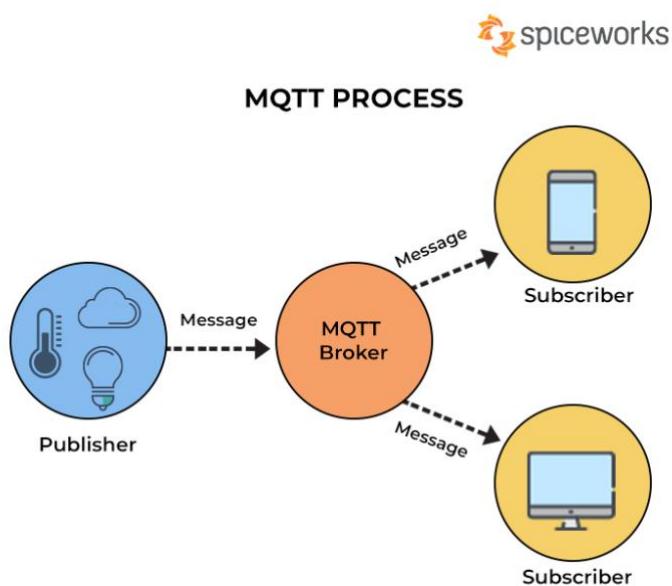


Figura 24 – MQTT protocolo com Broker



Figura 25 – Mosquitto MQTT Broker

FrontEnd:

- **ThingSpeak:** é uma plataforma IoT de código aberto que permite o rápido desenvolvimento, gestão e dimensionamento de projetos IoT. O objetivo é fornecer a solução pronta a usar para a nuvem IoT ou no local que irá permitir a infraestrutura do lado do servidor para as aplicações IoT.

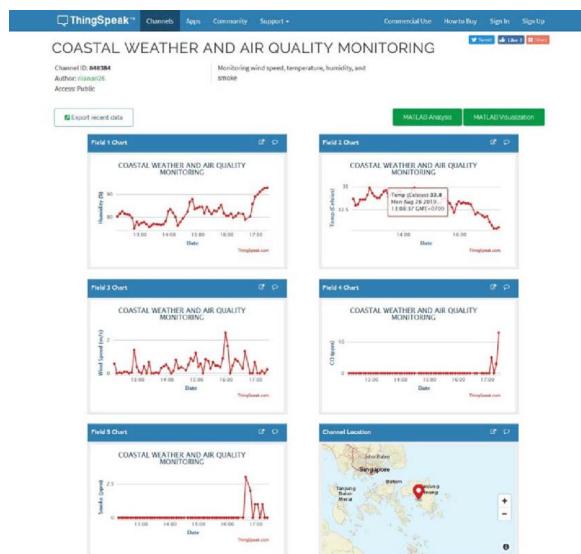


Figura 26 – Exemplo de um Dashboard da plataforma ThingSpeak



Figura 27 – ThingSpeak

5.5 Protótipo

Um protótipo especifica um modelo, ou o primeiro exemplar de um objeto designando assim um produto em fase de testes ou de projeto. Este pequeno roteiro trata da prototipagem de circuitos eletrônicos, onde a principal preocupação deve ser com o conhecimento da pinagem do componente e/ou da sua apresentação física. Com a atual disseminação de fontes de conhecimento pela internet, esta tarefa ficou um pouco mais fácil, pois as informações aqui necessárias são encontradas nas *datasheets* (folhas de dados - contém as características elétricas e outras informações) dos componentes, que são normalmente disponibilizadas por seus fabricantes.

Em muitos casos, a prototipagem eletrônica normalmente é feita com os próprios componentes do circuito definitivo, conectados através de uma matriz de contatos, mais conhecida como protoboard. Assim como a maioria dos processos industriais, este também conta com uma diversidade de ferramentas de CAD e neste exemplo será usado o Fritzing, que mesmo não simulando a execução do circuito, tem como vantagens a sua usabilidade, além de sua rica e adaptável biblioteca de componentes.

Para exemplificar o processo de desenvolvimento de um protótipo, a representação de um circuito eletrônico será usada [Figura 28] para exemplificar o contexto de hardware do sistema referido anteriormente na arquitetura e na listagem de ferramentas e tecnologias utilizadas. Também foi desenvolvido um esquema elétrico [Figura 29] que permite a representação gráfica de circuitos eletrônicos com objetivo de fornecer informações úteis perante a interpretação elétrica de um projeto.

Para além de disponibilizar as representações de circuito eletrônico e esquema elétrico foi também criado a montagem de um PCB do protótipo, ou seja, representação gráfica de uma placa de circuito impresso [Figura 30].

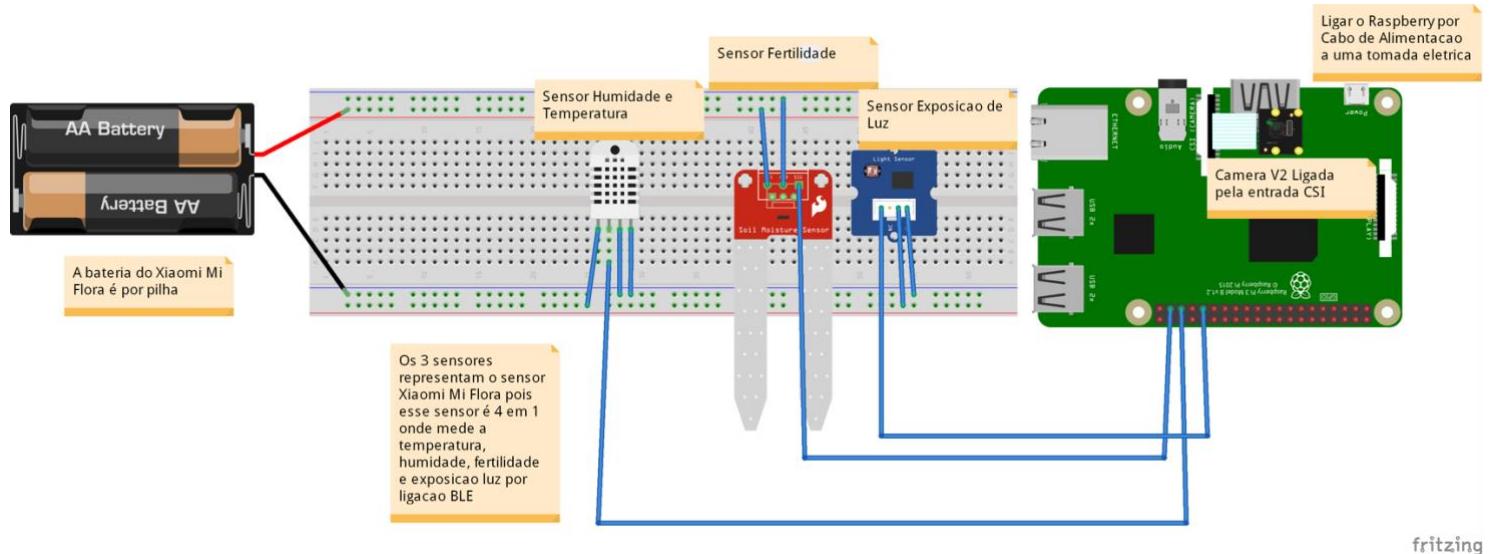


Figura 28 – Circuito eletrônico

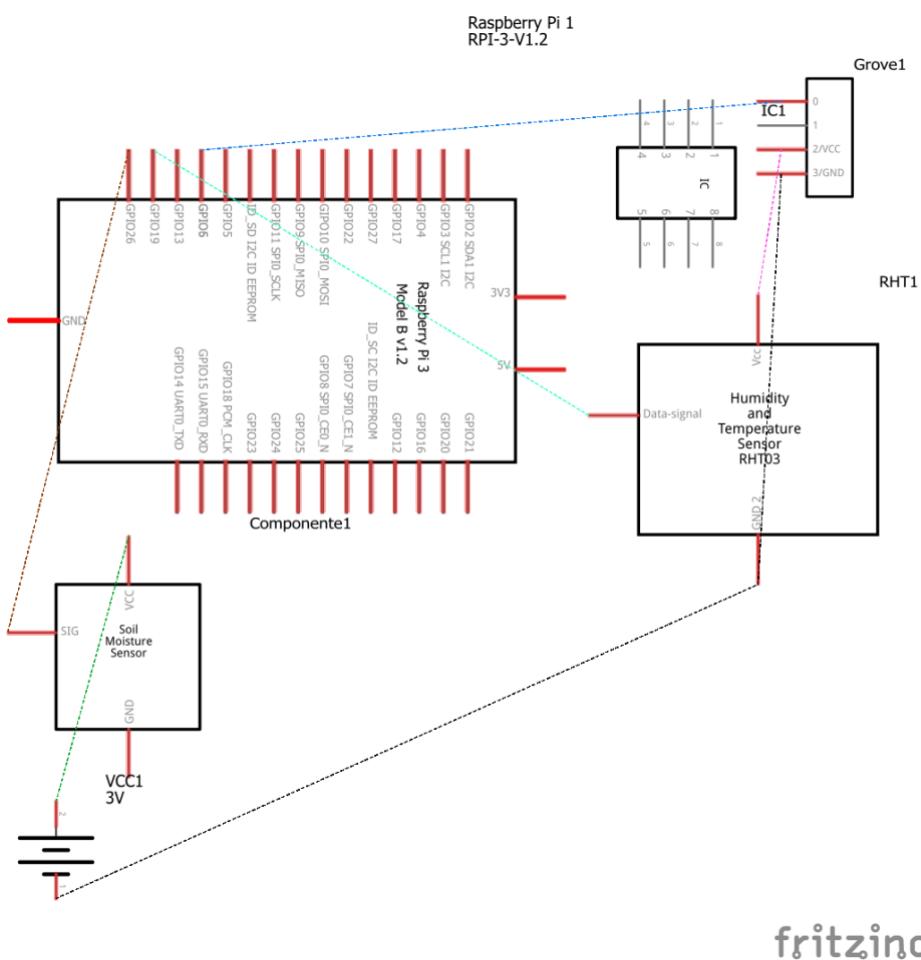


Figura 29 – Esquema elétrico do circuito

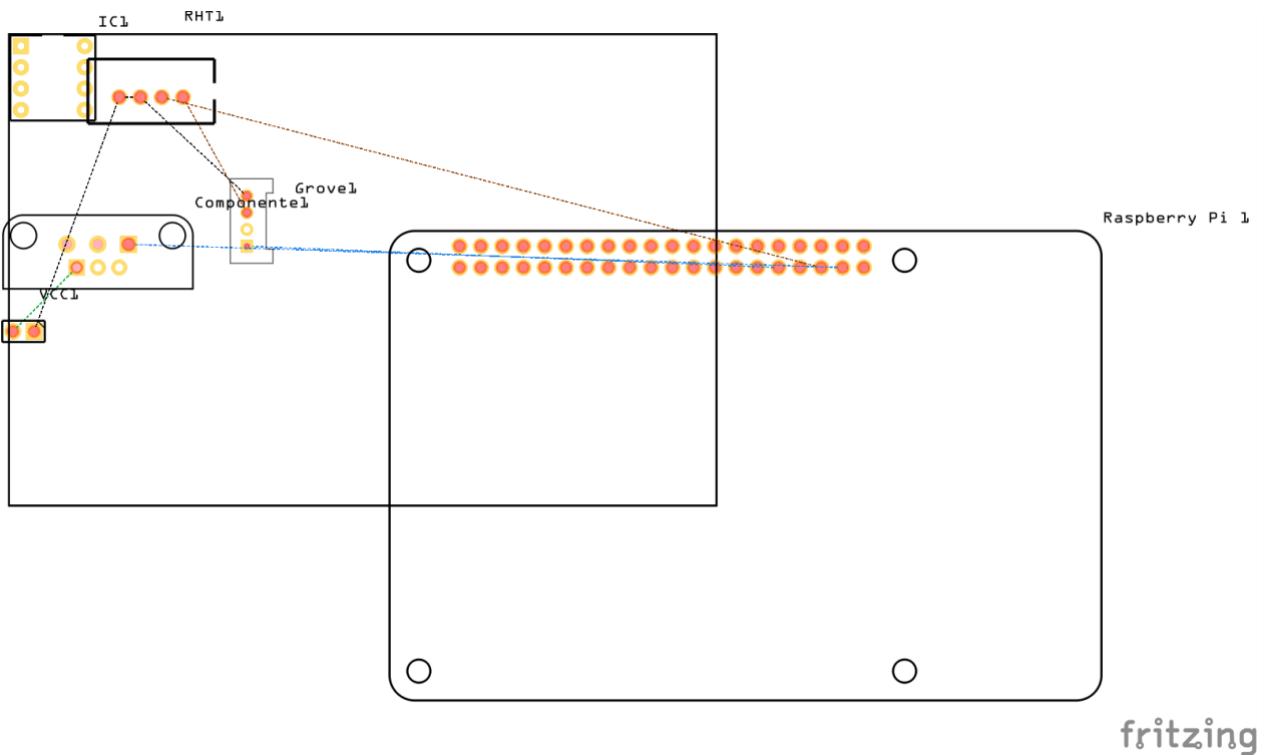


Figura 30 – PCB do Circuito

5.6 Abrangência

Na vertente académica, esta solução vai envolver metodologias já trabalhadas em algumas áreas curriculares já lecionadas até à data, sendo estas:

- Data Science
- Base de Dados
- Engenharia de Requisitos e Testes
- Engenharia de Software
- Redes de Computadores
- Sistemas Operativos
- Computação Distribuída
- Arquitetura Avançadas de Computadores
- Fundamentos de Programação
- Inteligência Artificial

6 Plano de Testes e Validação

No âmbito do meu TFC, desenvolvi um plano de testes para validar a solução construída e demonstrar a sua aplicabilidade, pertinência e relevância para a resolução de um problema real. A minha abordagem de teste é baseada em modelos formais de análise de riscos e de impacto, que permitem identificar e mitigar potenciais problemas antes da sua implementação em contexto produtivo.

Na tabela abaixo [Tabela 5], apresento um guião detalhado dos testes que serão realizados, incluindo descrição dos cenários e os resultados esperados. O plano de testes é coerente com os requisitos definidos e incide sobre a qualidade da solução, validação de funcionamento e de operação em contexto produtivo.

Título	Descrição	Requisitos	Resultado Esperado	Resultado Obtido
Conectividade dos dispositivos	O sistema arranca o ficheiro. python que contém todo o código para o reconhecimento do sensor e da câmera	RF1, RF2, RF8	No decorrer do programa. python que o utilizador não receba nenhuma mensagem de erro em relação a conexão dos dispositivos	Ao introduzir para decorrer o ficheiro.py é corrido sem problemas e realiza todos os passos do que é suposto
Medição das Plantas	O sistema arranca o ficheiro. python que contém todo o código para a medição da planta acompanhada pelos dispositivos	RF3	O sistema consiga gerar um JSON dos dados ambientais das plantas e pronto para enviar para plataforma por um MQTT Broker	Na pasta do TFC no ambiente do raspberry é encontrado um ficheiro Plants1.csv e Plants2.csv com todas as medições até o momento. Na plataforma Thingspeak é possível visualizar os dados obtidos em tempo real em vários fields (gráficos) de cada dado como

				também o histórico dos dias anterior apontado com o rato
Automatização	O sistema consegue rodar o scrypt sem o utilizador execute manualmente por uma linha de comando	RF4,	Após o arranque do sistema seja executado o ficheiro python pelo serviço daemon e avise ao utilizador que está a executar	O sistema Raspbian OS permite automatização do .py quando inicia, mas encontra-se desativado para uma melhor demonstração na apresentação do TFC.
Armazenamento	O sistema arranca o ficheiro. pyhton que contém todo o código para leitura e escrita do ficheiro csv que guarda os dados das plantas	RF5, RF6, RF7	O Ficheiro CSV tenha sido criado na pasta do projeto	Na pasta do TFC no ambiente do raspberry é encontrado um ficheiro Plants1.csv e Plants2.csv com todas as medições até o momento. É permitido reescrever o csv permitindo adicionar novas medições.
Implementação Machine Learning	O sistema extrai os dados retirados do caderno jupyter desenvolvido para analise de dados das plantas retirados diariamente para o estudo do crescimento das mesmas nas quais foram calculadas estimativas de valores	RF9, RF10, RF11, RF12, RF13, RF14, RF15,	O sistema contenha o valor do modelo classificação calculado em memória diferente de “0”.	A plataforma ThingSpeak recebe o valor de “0” ou “1” consoante o cálculo do modelo de machine learning e é ilustrado como um field na plataforma.

	para um bom crescimento.			
Envio Dados a Plataforma de Monitorização	O sistema arranca o ficheiro python que contém todo o código para extração dos dados da planta em tempo real e envia para o canal criado no ThingSpeak.	RF16, RF17, RF18	O sistema atribui uma mensagem para o utilizador que foi enviado com sucesso os dados	Utilizador visualiza uma mensagem no terminal do Raspbian OS que os dados foram enviados para o thingspeak podendo confirmar na própria plataforma
Acesso a Plataforma de Monitorização	O Utilizador faz login na aplicação ThingSpeak e procura pelo canal publico	RF23	O utilizador consiga encontrar o canal publico através do ID	O utilizador dentro da secção dos public channels ao procurar pelo user ID:mwa0000029941 421 onde encontra os dois canais das respetivas duas plantas teste
Notificações da Plataforma de Monitorização	O ThingSpeak notifica aos utilizadores por email caso algum dos parâmetros ambientais atinja um valor inferior ao um conjunto de valores pelo Machine Learning	RF19, RF20, RF21, RF22,	O utilizador recebe mails onde o assunto é uma mensagem com o parâmetro ambiental que esteja fora do normal	Após uma medição da planta onde valor do field "crescimento" obteve 0 de seguida a plataforma ativa o react enviado um mail para a conta registada do canal
Visualização dos Dados Ambientas	O Utilizador visualiza os dados em tempo real do sensor e da câmera	RF24	O utilizador consegue observar dados através de gráficos no canal do ThingSpeak	Na plataforma Thingspeak é possível visualizar os dados obtidos em tempo real em vários fields (gráficos) de

				cada dado após a medição da planta
Histórico dos Dados Ambientais	O Utilizador visualiza os dados anteriormente retirados pela ilustração de gráficos	RF25	O utilizador consiga visualizar dados das plantas anterior e veja a data retirada ao apontar com o rato	Na plataforma Thingspeak é possível visualizar os dados obtidos em tempo real em vários fields (gráficos) de cada dado como também o histórico dos dias anterior apontado com o rato

Tabela 5 – Tabela de Testes e Validação

7 Método e Planeamento

De forma que seja possível parametrizar e controlar que as sucessivas implementações se mantenham dentro do prazo previsto, nas imagens abaixo segue a minha proposta cronológica de ordem de trabalho.

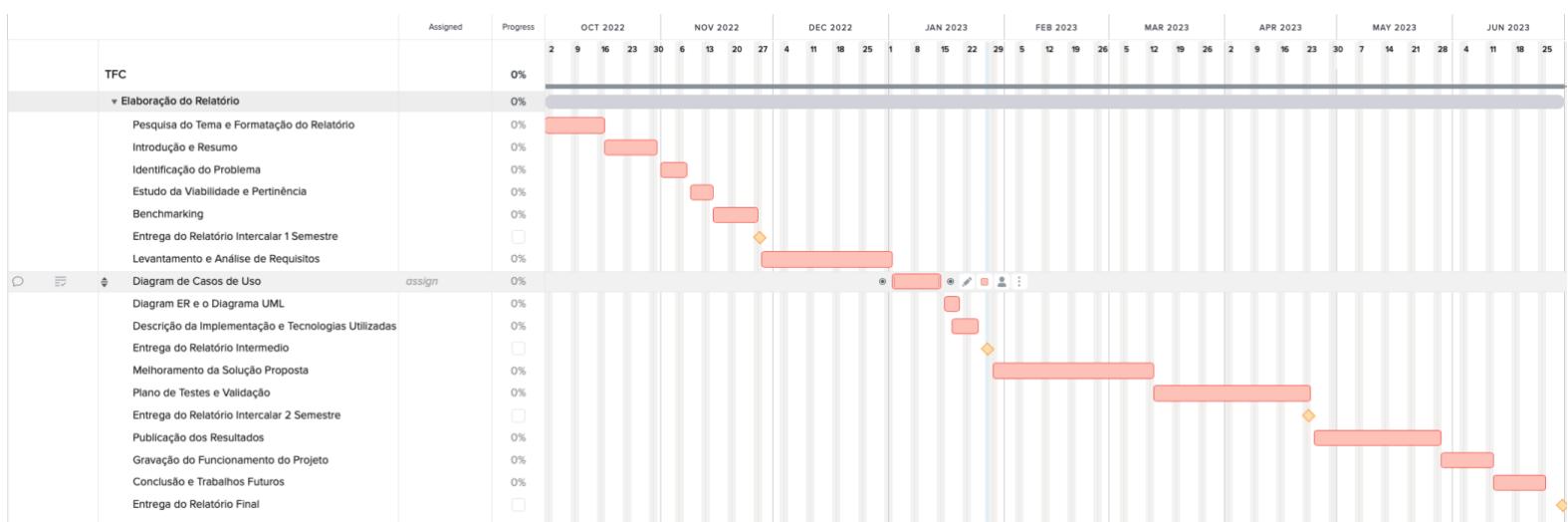


Figura 31 – Cronograma Elaboração do Relatório em formato Gantt

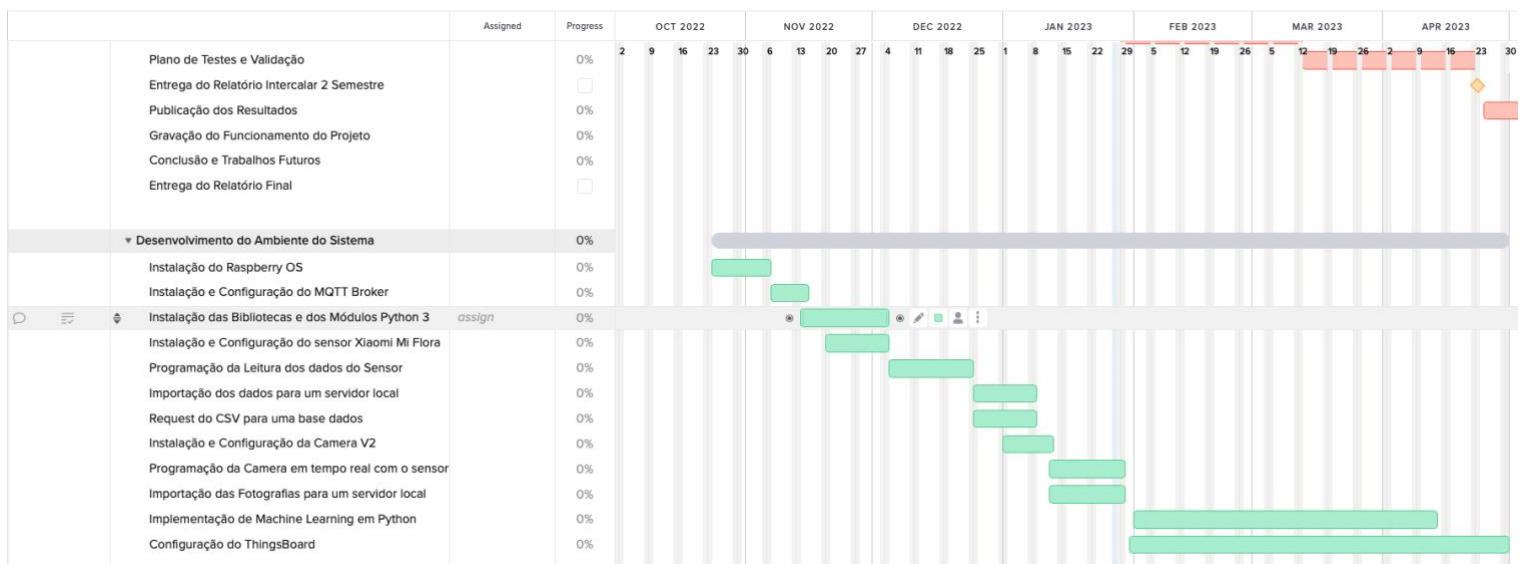


Figura 32 – Cronograma Desenvolvimento do Ambiente do Sistema em formato Gantt

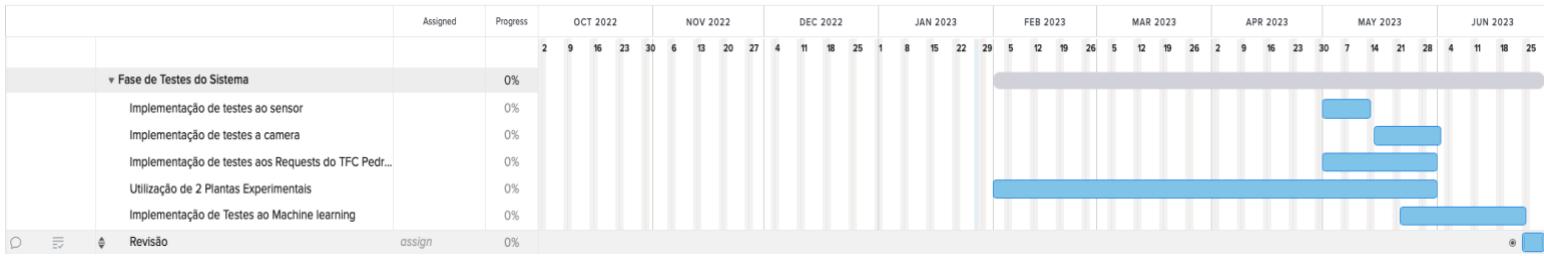


Figura 33 – Cronograma Fase de Testes do Sistema em formato Gantt

7.1 Descrição do Cronograma

É feita uma breve descrição sobre o planeamento do trabalho final de curso proposto dividido por 3 partes onde cada uma delas terá uma os detalhes de cada ação e o tempo previsto. A utilidade desta descrição é entender melhor os tópicos presentes no cronograma de formato Gantt.

7.1.1 Elaboração do Relatório

- Pesquisa do Tema e Formatação do Relatório
 - Sendo o início da UC de Trabalho Final de curso foi pedido inicialmente a elaboração da estrutura de um relatório e definir o que será o trabalho final de curso dado ao estado aceite na atribuição do tema.
- Introdução e Resumo
 - Nesta fase é feito a ideologia do tema sistema monitorização de plantas como a justificação da escolha do tema incluído.
- Identificação do Problema
 - Neste tópico do relatório é elaborado a identificação de problemas atuais e futuros perante a construção deste sistema como também a identificação global dos problemas da agricultura e perda de água na rega de plantas.
- Estudo da Viabilidade e Pertinência
 - Neste tópico do relatório é pretende-se que procure saber mais sobre um sistema de monitorização de plantas e como poderá conter um futuro após a conclusão deste trabalho final ou até mesmo a possibilidade de conjugar com outro sistema existente.
- Benchmarking
 - Após a procura de sistemas de monitorização de plantas existentes deve-se mencionar neste relatório essas mesmas soluções encontradas e comparar com o nosso sistema.
- Entrega do Relatório Intercalar 1 Semestre

- Confirmar a entrega do relatório intercalar do 1 semestre dando uma revisão por completo todos as secções escritas neste relatório.
- Levantamento e Analise de Requisitos
 - Neste tópico deve-se refletir sobre possíveis requisitos do TFC incluído requisitos funcionais , não funcionais e técnicos . Após a reflexão é enumerar os requisitos de uma forma bem estrutura e atribuir possíveis custos de tempo previsíveis bem como relevo de importância neste sistema de monitorização de plantas.
- Diagrama de Casos de Uso
 - Tal como nome indica , deve-se construir diagrama de casos de uso que refletem as interações do sistema entre os atores propostos.
- Diagrama ER e o Diagrama UML
 - Tal como o nome indica , deve-se construir um diagrama de entidade-Relação incluído o próprio diagrama de UML após a conclusão do diagrama de ER tendo de inserir depois no relatório.
- Descrição da Implementação e Tecnologias Utilizadas
 - Neste tópico do relatório deve-se fazer uma breve descrição extensa sobre como o sistema de monitorização será implementado bem como uma lista extensa de ferramentas e tecnologias utilizadas.
- Entrega do Relatório Intermedio
 - Confirmar a entrega do relatório intermedio do 1 semestre dando uma revisão por completo todos as secções escritas neste relatório.
- Melhoramento da Solução Proposta
 - Revisão por completo a solução proposta dando alteração a tecnologia no desenvolvimento do TFC e a fundamentação das principais opções na construção da solução. Não esquecer que a solução proposta deverá ser ajustada aos requisitos identificados.
- Plano de Testes e Validação
 - Neste tópico do relatório deve-se demonstrar o funcionamento da solução , sendo importante que os testes demonstrem que ela cumpre os objectivos que se propôs. Os testes a realizar devem incidir sobre qualidade da solução desenvolvida bem como a validação de funcionamento e de operação em contexto produtivo.
- Entrega do Relatório Intercalar 2 Semestre
 - Confirmar a entrega do relatório intercalar do 2 semestre dando uma revisão por completo todos as secções escritas neste relatório.
- Publicação dos Resultados
 - Neste tópico do relatório deve-se realizar a descrição detalhada de resultados do sistema de monitorização de plantas desde o sucesso de execução dos scripts até a ilustração dos dados das plantas na plataforma ThingsBoard .
- Gravação do funcionamento do projeto

- Nesta altura da tarefa o protótipo tem de estar finalizado para a gravação da execução do sistema para demonstrar as capacidades do sistema aos júris.
- Conclusão e Trabalhos Futuros
 - Neste tópico do relatório tal como nome indica deve-se escrever uma conclusão detalhada sobre o TFC incluído os pros e contras deste sistema no mundo real bem como skills que adquiriu ao longo desta UC. Caso seja possível terei de escrever um manual de instalação e de utilização deste sistema.
- Entrega do Relatório Final
 - Confirmar a entrega do relatório final dando uma revisão por completo todos as secções escritas neste relatório.

7.1.2 Desenvolvimento do Ambiente do Sistema

- Instalação do Raspberry OS
 - Pesquisar e consultar o manual de Raspberry Pi 3B para aprendizagem da instalação do sistema operativo.
- Instalação e Configuração do MQTT Broker
 - Instalação da ferramenta Mosquitto MQTT Broker para proceder mais tarde com o desenvolvimento de código no envio de dados do sensor Xiaomi para um CSV e mais para o thingsboard pelo protocolo MQTT.
- Instalação das Bibliotecas e dos Módulos dependentes para o Python 3
 - Instalação das bibliotecas necessárias para uso na linguagem Python bem como a instalação dos módulos dependentes para identificação de dispositivos no Python.
- Instalação e Configuração do sensor Xiaomi Mi Flora
 - Confirmação da instalação dos módulos dependentes instalados anteriormente para prosseguir a procura do sensor pelo protocolo BLE. Verificar a listagem de dispositivos ligados e identificar o sensor pelo nome e o macAddress.
- Programação da leitura dos dados do sensor
 - Desenvolvimento do código que permite a conexão do dispositivo Xiaomi Mi Flora e a extração dos dados lidos das plantas.
- Importação dos dados para um servidor local
 - Envio dos dados guardados numa variável para um ficheiro CSV editável para que o registo das plantas cada dia esteja guardado no mesmo ficheiro.
- Request do CSV para uma base de dados
 - Criar um processo de request para uma base de dados do TFC Pedro Rocha para poder usar os dados lidos numa ferramenta de FrontEnd.
- Instalação e Configuração da Câmera V2
 - A instalação física da camera V2 com muito cuidado sendo um equipamento muito frágil e alteração de lente. Após a instalação física ,

deve-se configurar o software da camera com ajuda de módulos existentes bem como bibliotecas de python para o uso da mesma.

- Programação da Câmera em tempo real com o sensor
 - Desenvolvimento de código que permite a usabilidade da câmera tirando uma fotografia e ao mesmo tempo que o sensor contém os dados extraídos das plantas
- Importação das fotografias para um servidor local
 - Envio das fotografias tiradas no momento para uma pasta específica e alteração do CSV guardando um caminho específico da fotografia que se situa dentro da pasta .
- Implementação de Machine Learning em Python
 - Esta é a fase com mais carga de trabalho com intuito que a máquina aprenda sozinha a analisar os dados registados e construa os gráficos que explicam o crescimento da planta previsto e indicam os valores necessários que a planta deve conter para manter viva.
- Configuração do ThingSpeak IoT
 - Tal como o nome indica consiste em adicionar os dados das plantas na plataforma de monitorização bem como os dados calculados na fase anterior

7.1.3 Fase de Testes do Sistema

- Implementação dos testes ao sensor
 - Aplicar testes fora do comum ao sensor para validar o limite valores entre plantas na espécie escolhida.
- Implementação dos testes da Câmera
 - Aplicar testes fora do comum a camera para validar a qualidade de fotografias entre plantas na espécie escolhida.
- Implementação dos testes dos Requests para o TFC Pedro Rocha
 - Aplicar testes ao requests em python no servidor Django do Pedro Rocha para validar a receção dos dados em tempo real.
- Utilização de 2 Plantas Experimentais
 - Definir as 2 plantas que iremos estudar ao longo do TFC e validar os dados que obtivemos com os previstos pelos valores teoricamente possíveis.
- Implementação de testes do Machine Learning
 - Para testar as nossos algoritmos e modelos linear iremos utilizar valores fora do normal para corrigir caso haja alguma anomalia
- Revisão
 - Rever que o sistema esteja todo funcional até o dia da apresentação

7.2 Cumprimento do Planeamento

Para demonstrar as indicações de cumprimento do TFC nesta secção foi feita a implementação da coluna progress nos respetivos cronogramas da secção [Método e Planeamento] com intervalos de 0 a 100 por percentagem para representar o progresso atual. Caso o leitor contenha dificuldade em visualizar a coluna progress pode observar a cor dos retângulos entre as datas representam o estado/progresso de cada etapa onde a cor mais “viva” estabelece a percentagem correspondida da etapa.

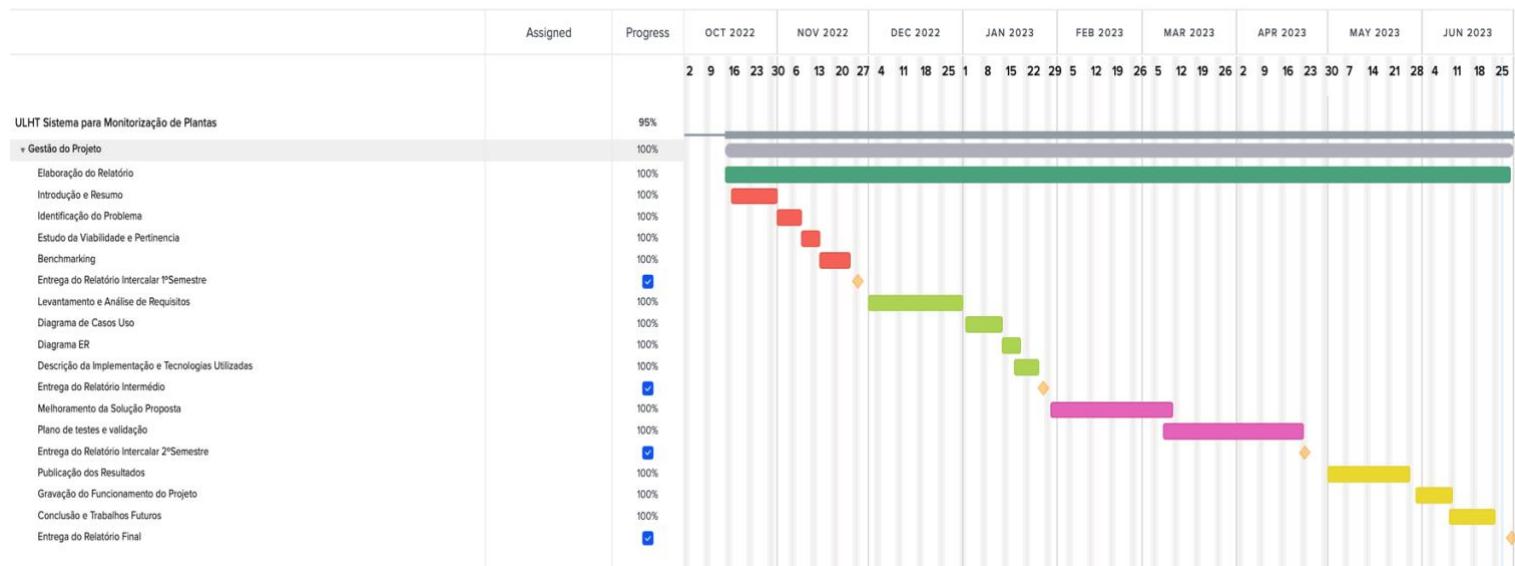


Figura 34 - Cumprimento da Elaboração do Relatório em formato Gantt

Sistema para Monitorização de Plantas

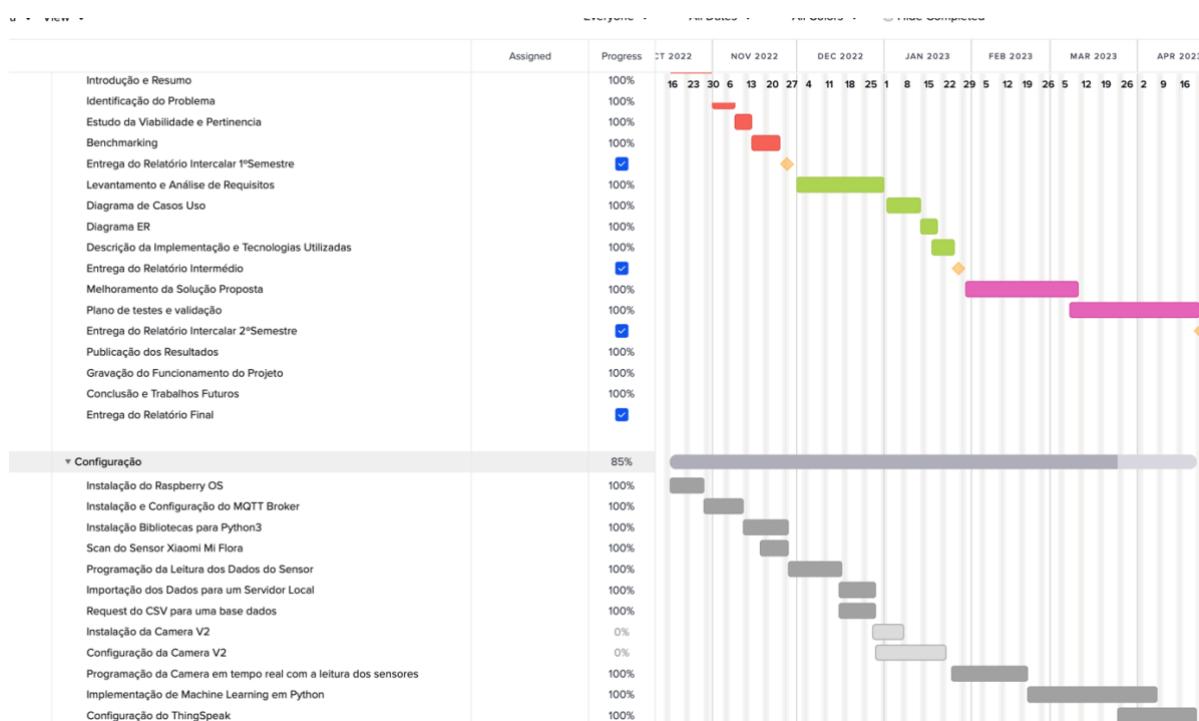


Figura 35 – Cumprimento do Desenvolvimento do Ambiente do Sistema em formato Gantt

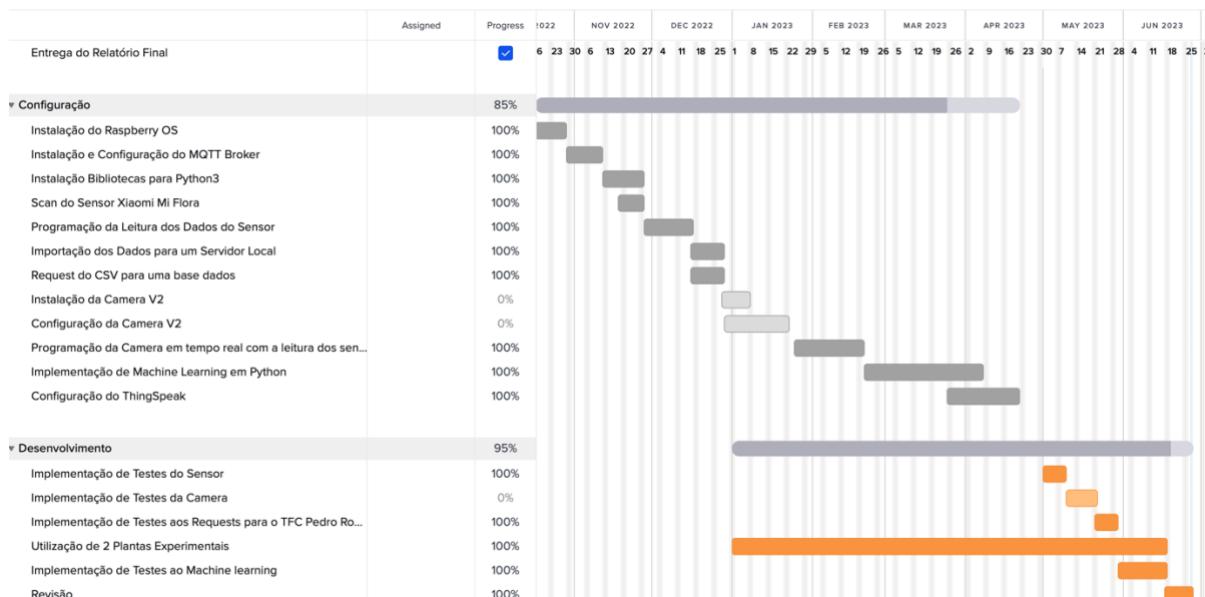


Figura 36 – Cumprimento da Fase de Testes em Formato Gantt

De acordo com o cumprimento calendarização proposta nas figuras [Figura 34, Figura 35 e Figura 36] abaixo é possível ver a data de complementação dos requisitos funcionais, não funcionais e técnicos de acordo com as entregas intercalares e intermédias do projeto.

Requisitos F-Funcional RNF-Não Funcional RT - Técnico	Entrega Intercalar 1ºSemestre	Entrega Intermédia 1ºSemestre	Entrega Intercalar 2º Semestre	Entrega Final
RF 1	<input checked="" type="checkbox"/>			
RF 2	<input checked="" type="checkbox"/>			
RF 3		<input checked="" type="checkbox"/>		
RF 4		<input checked="" type="checkbox"/>		
RF 5	<input checked="" type="checkbox"/>			
RF 6		<input checked="" type="checkbox"/>		
RF 7		<input checked="" type="checkbox"/>		
RF 8		<input checked="" type="checkbox"/>		
RF 9			<input checked="" type="checkbox"/>	
RF 10				<input checked="" type="checkbox"/>
RF 11				<input checked="" type="checkbox"/>
RF 12				<input checked="" type="checkbox"/>
RF 13				<input checked="" type="checkbox"/>
RF 14				<input checked="" type="checkbox"/>
RF 15				<input checked="" type="checkbox"/>
RF 16			<input checked="" type="checkbox"/>	
RF 17			<input checked="" type="checkbox"/>	
RF 18				<input checked="" type="checkbox"/>
RF 19				<input checked="" type="checkbox"/>
RF 20				<input checked="" type="checkbox"/>
RF 21				<input checked="" type="checkbox"/>
RF 22				<input checked="" type="checkbox"/>
RF 23			<input checked="" type="checkbox"/>	
RF 24			<input checked="" type="checkbox"/>	

RNF 1	<input checked="" type="checkbox"/>			
RNF 2		<input checked="" type="checkbox"/>		
RNF 3	<input checked="" type="checkbox"/>			
RNF 4	<input checked="" type="checkbox"/>			
RNF 5	<input checked="" type="checkbox"/>			
RNF 6		<input checked="" type="checkbox"/>		
RNF 7	<input checked="" type="checkbox"/>			
RNF 8			<input checked="" type="checkbox"/>	
RNF 9				<input checked="" type="checkbox"/>
RT 1			<input checked="" type="checkbox"/>	
RT 2	<input checked="" type="checkbox"/>			
RT 3	<input checked="" type="checkbox"/>			
RT 4		<input checked="" type="checkbox"/>		
RT 5	<input checked="" type="checkbox"/>			
RT 6			<input checked="" type="checkbox"/>	
RT 7	<input checked="" type="checkbox"/>			
RT 8	<input checked="" type="checkbox"/>			
RT 9	<input checked="" type="checkbox"/>			
RT 10	<input checked="" type="checkbox"/>			
RT 11			<input checked="" type="checkbox"/>	

Tabela 6 – Cumprimento dos Requisitos

7.2.1 Esforço

Para avaliar o esforço despendido referente ao progresso trabalho foi me recomendo mencionar que semanalmente contenho uma reunião com o orientador e co-orientador na qual o objetivo desta reunião é citar o progresso do trabalho como também o relatório dando assim um feedback positivo ou negativo em relação ao meu esforço perante aos

meus orientadores. Por sua vez orientador e o co-orientador planeiam comigo os “próximos passos” e identificam melhorias ou métodos mais simples do que eu propus ou que já foi realizado.

Assim para avaliação do esforço neste TFC é atribuído uma hora de reunião semanalmente entre os orientadores e autonomamente 6 horas por semana. Sendo assim um total de 28 horas por mês sendo 7 horas por semana excluindo 4 horas que são realizadas o apoio do orientador e co-orientador.

7.2.2 Avaliação Crítica

7.2.2.1 Primeiro Semestre

Pelo feedback dos jurados pela avaliação intercalar do 1ºsemestre e observações feitas pelos orientados que este relatório está muito bem dimensionado com um decorrer a bom ritmo e a existência de um compromisso complementar por parte do aluno sobre o tópico de plantas estudado. Contudo na construção do relatório obtive dificuldades no desenvolvimento da viabilidade e pertinência dado a percepção do que seria de o projeto não estar 100% correta, mas com ajuda dos meus orientadores foi possível chegar a uma conclusão.

Face as tarefas do desenvolvimento do projeto não contive nenhuma dificuldade até ao momento mesmo com surgimento de erros de instalação e na configuração do sensor pois fui encontrado as respetivas soluções com alternativas de linhas código existentes. Apenas surgiu um imprevisto em relação a câmera que já vinha com defeito de fábrica tendo partido o cabo que interliga a interface CSI do Raspberry mas rapidamente os orientadores obtiveram o feedback da solução e apresentaram a solução de aquisição de uma nova câmera sem qualquer problema.

7.2.2.2 Segundo Semestre

Pelo feedback dos jurados pela avaliação intermédia do 1ºsemestre e observações feitas pelos orientadores que este relatório está muito bem dimensionado com um decorrer a bom ritmo e a existência de um compromisso complementar por parte do aluno sobre o tópico de plantas estudado. Em relação a construção do relatório não obtive dificuldades em alterar a minha solução proposta nem demonstrar o progresso do trabalho até a data.

Face as tarefas do desenvolvimento do projeto houve uma alteração na ferramenta da plataforma de monitorização onde irei usar ThingSpeak em vez de a ThingsBoard uma vez que o raspberry mesmo com os requisitos mínimos cumpridos continha uma má interação perante o utilizador (lentidão). Esta alteração provocou um maior tempo dedicado

ao TFC para entender como iria funcionar o envio de dados MQTT para esta plataforma tal como na escolha desta ferramenta foi a maior dificuldade até ao presente pois existem diversas opções para o mesmo objetivo sendo que tive de analisar compatibilidade com o sensor e procurar pelas possíveis limitações entre planos “free” ou custos monetários.

7.2.2.3 Final

Pelo feedback dos jurados pela avaliação intercalar do 2º semestre e as observações feitas pelos orientadores que o relatório está num caminho muito perto do sucesso tendo dificuldades apenas em entender o que devo colocar no capítulo dos resultados tendo indicação orientadores para ilustrar apenas os resultados e talvez construir um manual de instalação e utilização do mesmo.

Face a tarefa do desenvolvimento do projeto foi desenvolvido os modelos de machine learning propostos pelo orientador João Carvalho tendo feito a comparação entre todos até (15/0/2023) para determinar qual o modelo em que devo utilizar para chegar a uma conclusão perante o crescimento das plantas. Foi o escolhido modelo Decision Tree pois é o suficiente para cálculos entre 5 variáveis entre si e o que obteve melhores resultados e de seguida foi configurado o envio não só dos 4 parâmetros (temperatura, luz, humidade e fertilidade) mas também a variável “crescimento” que vai conter valores de 0 ou 1 para definir se a planta está num bom crescimento ou não. De seguida foi implementado um alarme que enviara um email ao utilizador caso a planta esteja “mau crescimento”, ou seja, significa que existe falta de água uma vez que foi o factor de diferença entre as duas plantas estudadas na qual obtive dificuldades pois tive de estudar como criar uma app para o canal das plantas que permitisse enviar um mail.

8 Resultados

8.1 Estudo das Plantas

Como já referido na secção da solução proposta [30], o estudo de duas plantas da mesma espécie é essencial para determinação de resultados numa plataforma de monitorização de dados não só para a ilustração desses mesmos dados como também para a implementação de machine learning com verdadeiro objetivo de determinar se a planta em questão está num caminho de “sucesso” no seu crescimento. Para distinguir as duas plantas foi usado no suporte ao vaso uma base de cor diferente sabendo que a planta 1 [Figura 37] com base laranja era controlada (100% rega de água) enquanto a planta 2 [Figura 38] com base água não era controlada (60% rega de água).



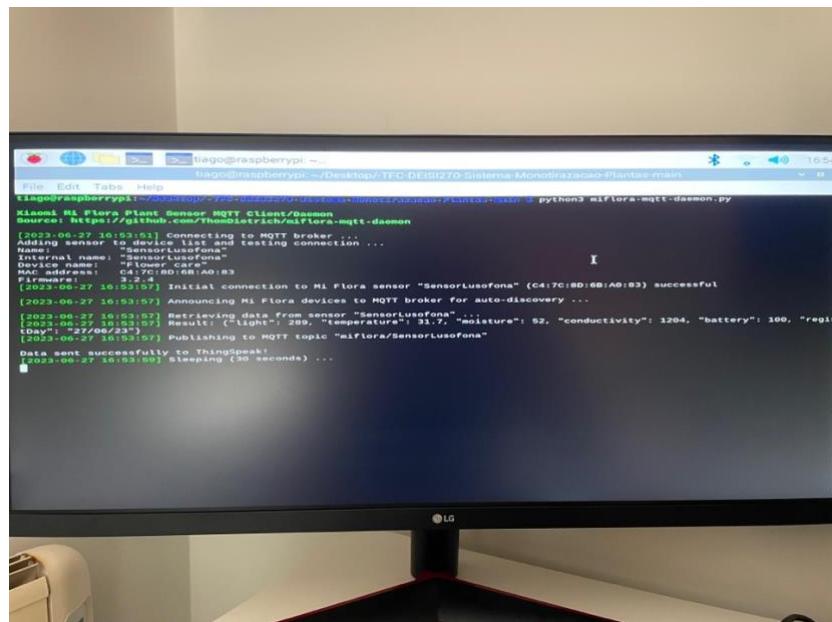
Figura 37 – Planta 1



Figura 38 – Planta 2

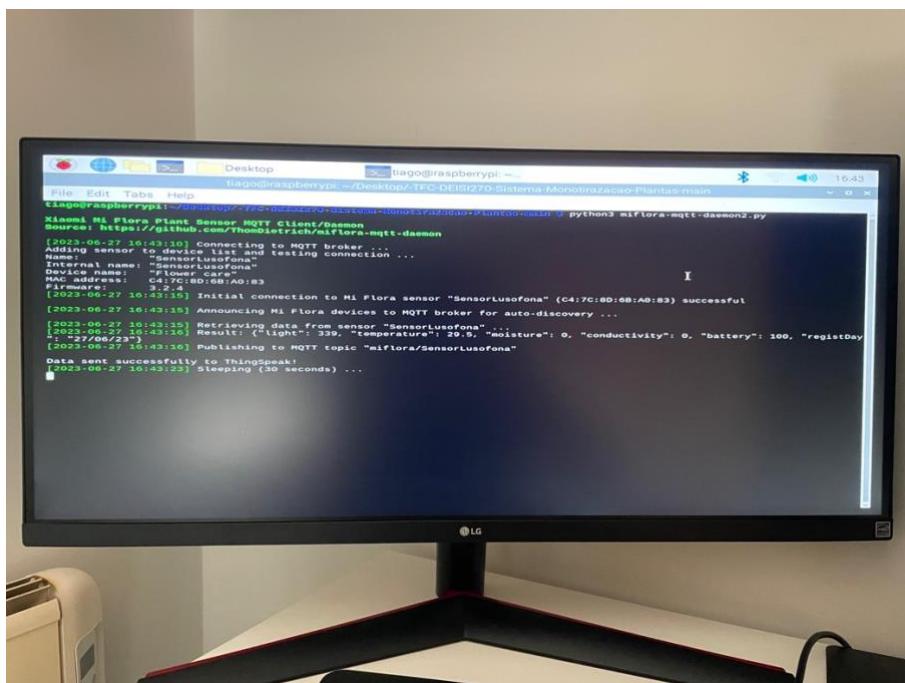
8.2 Medição das Plantas

Para a medição das plantas contenho um ficheiro associado em python para cada planta uma vez que só tenho um sensor para a medição dos valores ambientais e terei de enviar os dados para mesma plataforma em canais diferentes tendo apenas de dar “run” no miflora-mqtt-daemon.py



```
tiago@raspberrypi: ~ Desktop tiago@raspberrypi: ~/Desktop/TFCD-DESI270-Sistema-Monitorizacao-Plantas-main$ python3 miflora-mqtt-daemon.py
Xiaomi Mi Flora Plant Sensor MQTT Client/daemon
Source: https://github.com/ThomazBirich/miflora-mqtt-daemon
[2023-06-27 16:53:53] Connecting to MQTT broker ...
Add Mi Flora sensor to MQTT broker and testing connection ...
Name: "SensorLusofona"
Internal name: "SensorLusofona"
Device name: "Flower Care"
MAC address: C4:7C:80:6B:A0:B3
Firmware: 3.2.0
[2023-06-27 16:53:57] Initial connection to Mi Flora sensor "SensorLusofona" (C4:7C:80:6B:A0:B3) successful
[2023-06-27 16:53:57] Announcing Mi Flora devices to MQTT broker for auto-discovery ...
[2023-06-27 16:53:57] Receiving data from sensor "SensorLusofona" ...
[2023-06-27 16:53:57] Received data: {"light": 209, "temperature": 21.7, "moisture": 52, "conductivity": 1204, "battery": 100, "register": 0}
[2023-06-27 16:53:57] Publishing to MQTT topic "miflora/SensorLusofona"
Data sent successfully to ThingSpeak!
[2023-06-27 16:53:59] Sleeping (30 seconds) ...
```

Figura 39 – python planta1.py



```
tiago@raspberrypi: ~ Desktop tiago@raspberrypi: ~/Desktop/TFCD-DESI270-Sistema-Monitorizacao-Plantas-main$ python3 miflora-mqtt-daemon2.py
Xiaomi Mi Flora Plant Sensor MQTT Client/daemon
Source: https://github.com/ThomazBirich/miflora-mqtt-daemon
[2023-06-27 16:43:10] Connecting to MQTT broker ...
Add Mi Flora sensor to MQTT broker and testing connection ...
Name: "SensorLusofona"
Internal name: "SensorLusofona"
Device name: "Flower Care"
MAC address: C4:7C:80:6B:A0:B3
Firmware: 3.2.0
[2023-06-27 16:43:10] Initial connection to Mi Flora sensor "SensorLusofona" (C4:7C:80:6B:A0:B3) successful
[2023-06-27 16:43:10] Announcing Mi Flora devices to MQTT broker for auto-discovery ...
[2023-06-27 16:43:10] Receiving data from sensor "SensorLusofona" ...
[2023-06-27 16:43:10] Received data: {"light": 209, "temperature": 21.7, "moisture": 0, "conductivity": 0, "battery": 100, "register": 0}
[2023-06-27 16:43:10] Publishing to MQTT topic "miflora/SensorLusofona"
Data sent successfully to ThingSpeak!
[2023-06-27 16:43:23] Sleeping (30 seconds) ...
```

Figura 40 – python planta2.py

Podemos confirmar pela [Figura 39] e pela [Figura 40] que os dados medidos pelo sensor são processados pelo protocolo MQTT para o envio na plataforma ThingSpeak. Para comparação entre valores das duas plantas são guardados em ficheiros CSV diferentes [Figura 41 e Figura 42] para distinção entre si como também para ser usado na implementação de Machine Learning. Os ficheiros .py tem acesso aos csvs e vão reescrevendo novas linhas de medição com os valores medidos e o dia de registo de forma obter uma base dados local.

```

light,temperature,moisture,conductivity,battery,registDay
372,30,3,44,440,100,01/06/23
420,32,3,50,590,100,02/06/23
620,35,2,48,470,100,03/06/23
739,34,2,46,653,100,04/06/23
700,27,1,74,762,100,05/06/23
689,27,4,76,813,100,06/06/23
655,25,2,68,829,100,07/06/23
700,26,1,74,762,100,08/06/23
750,26,6,72,656,100,09/06/23
720,28,0,69,845,100,10/06/23
706,26,7,73,794,100,11/06/23
673,25,5,77,805,100,12/06/23
694,28,7,74,876,100,13/06/23
748,28,2,80,917,100,14/06/23
289,31,7,52,1204,100,27/06/23

```

Figura 41 – Dados da planta 1

```

light,temperature,moisture,conductivity,battery,registDay
653,28,3,45,205,100,01/06/23
584,27,7,38,310,100,02/06/23
546,28,0,41,275,100,03/06/23
487,26,7,42,231,100,04/06/23
682,28,1,76,762,100,05/06/23
621,28,0,10,76,100,06/06/23
598,27,8,13,50,100,07/06/23
663,26,9,16,43,100,08/06/23
634,26,7,9,31,100,09/06/23
676,28,4,12,27,100,10/06/23
686,28,2,14,35,100,11/06/23
661,28,1,8,19,100,12/06/23
626,28,0,10,12,100,13/06/23
682,29,4,3,6,100,14/06/23
244,30,3,0,0,100,27/06/23

```

Figura 42 – Dados da planta 2

8.3 Plataforma ThingSpeak

Por fim temos finalmente a plataforma ThingSpeak que representa o front-end deste TFC podendo visualizar os dados enviados pelo sistema tendo assim a possibilidade de visualizar não só os dados em tempo real da planta como também o seu histórico. Os dados das duas plantas estão divididos por “canais” [Figura 43 e Figura 45] para não ter os dados misturados entre si nos gráficos ilustrados

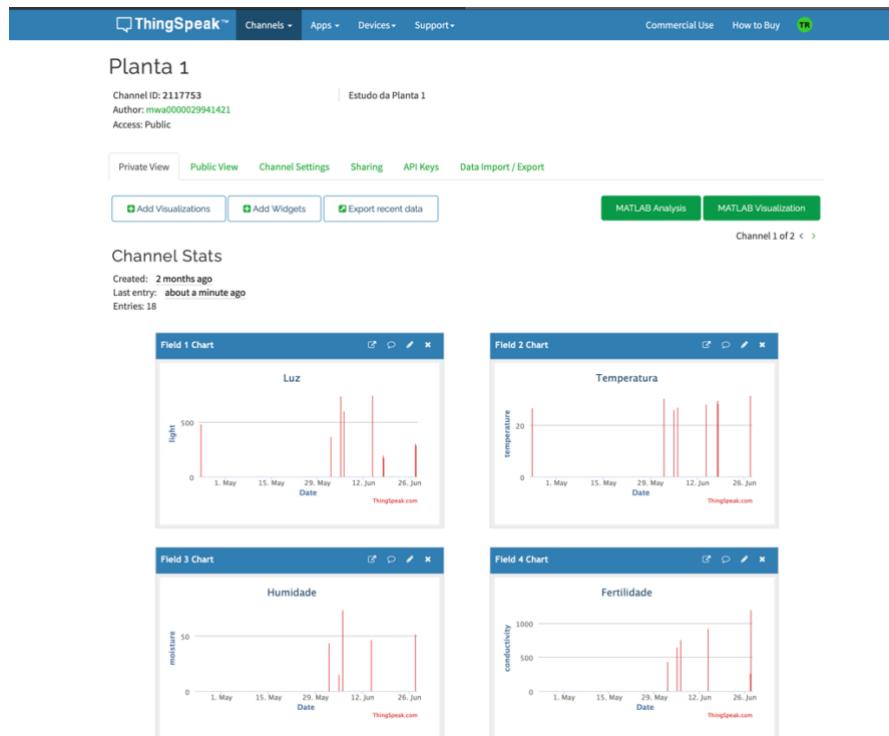


Figura 43 – Canal da Planta 1 (Luz, Temperatura, Humidade e Fertilidade)

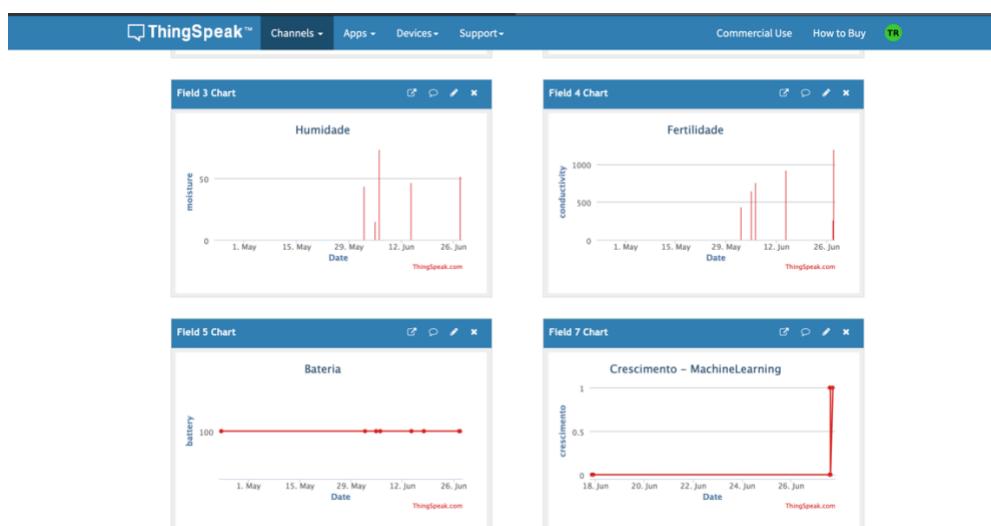


Figura 44 – Canal da Planta 1 (Bateria e Crescimento)

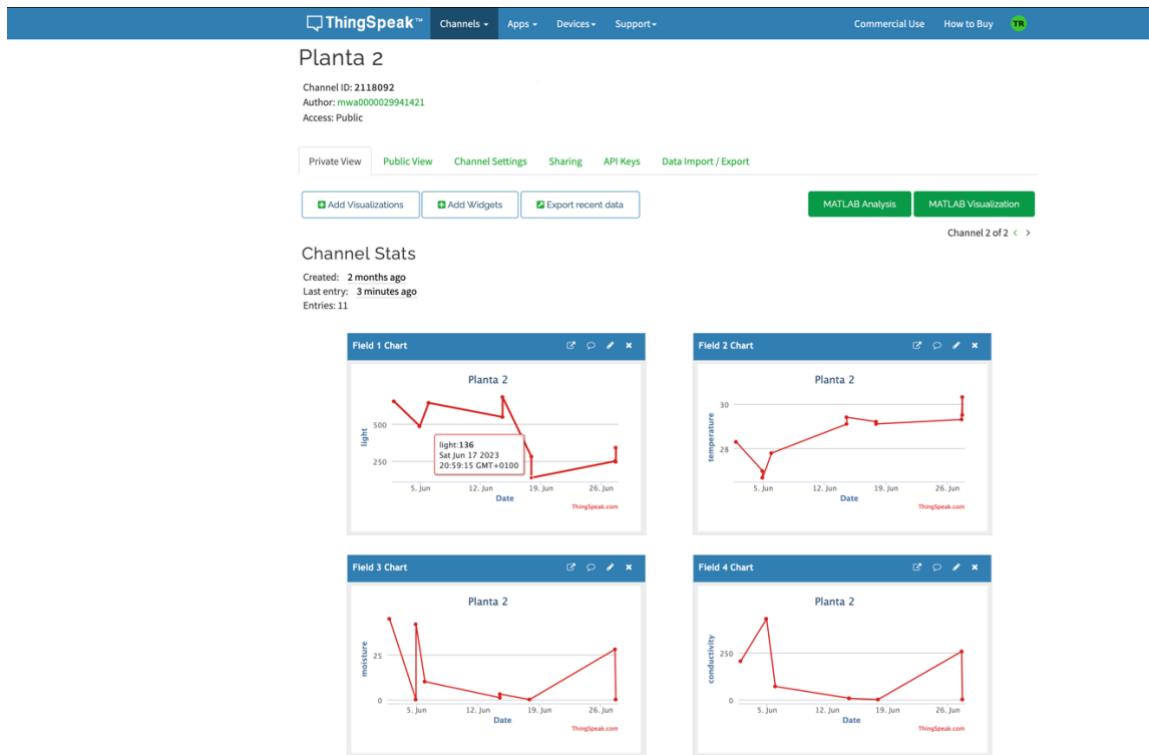


Figura 45 - Canal da Planta 2 (Luz, Temperatura, Humidade e Fertilidade)

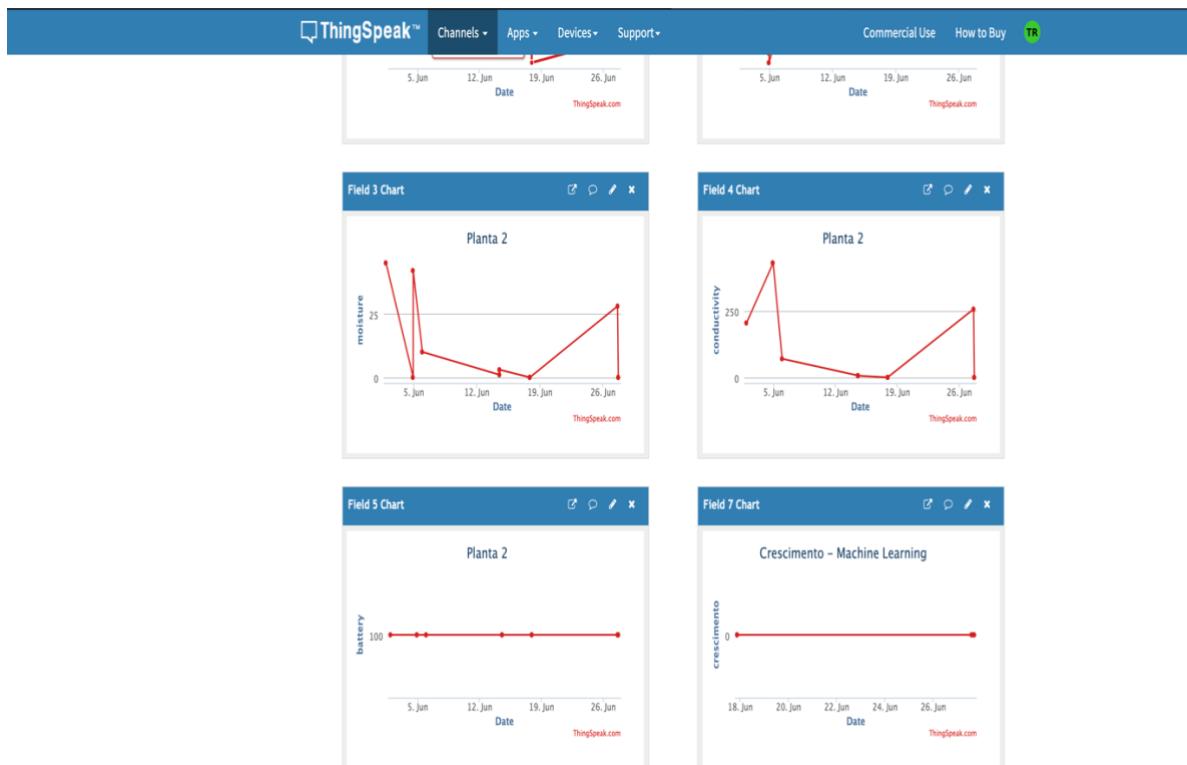


Figura 46 – Canal da Planta 2 (Bateria e Crescimento)

Além dos 5 dados recolhidos das plantas incluído a bateria, a plataforma disponibiliza a característica “Crescimento” sendo uma variável que representa o resultado do modelo de machine learning , ou seja , foi aplicado o modelo Decision Tree para identificar se a planta em questão necessita de água ou de outro factor como a luz tendo valor de “0” ou “1”.

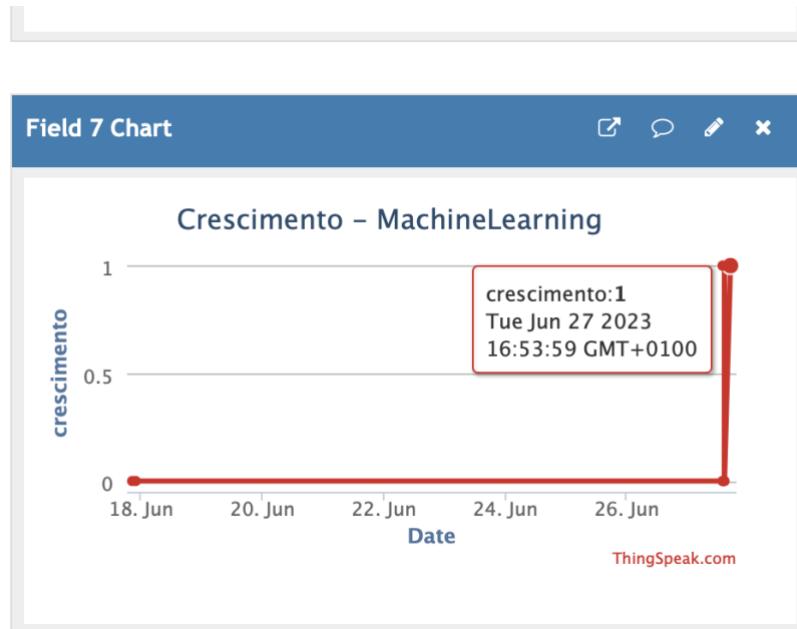


Figura 47 – Crescimento da Planta 1

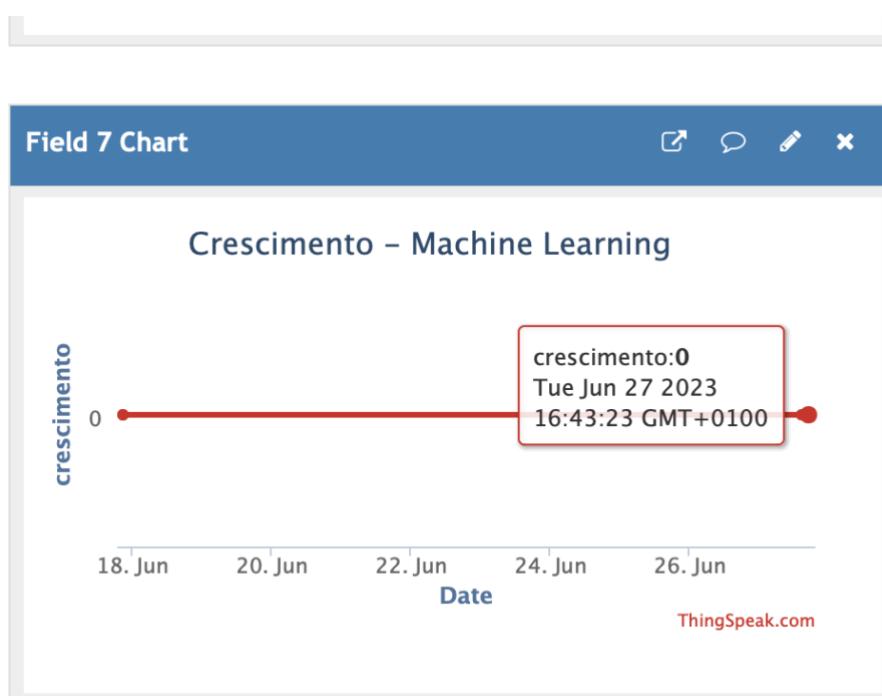


Figura 48 – Crescimento da Planta 2

É percutível que o valor “0” [Figura 48] no crescimento significa que a planta não está em boas condições para o seu crescimento enquanto o valor “1” [Figura 47] afirma que a planta em si está em boas condições. Foi criado um alerta [Figura 49] quando o valor do crescimento fosse “0” para notificar ao utilizador de que a planta necessita de água uma vez que foi único fator de diferenciação entre as plantas estudadas.

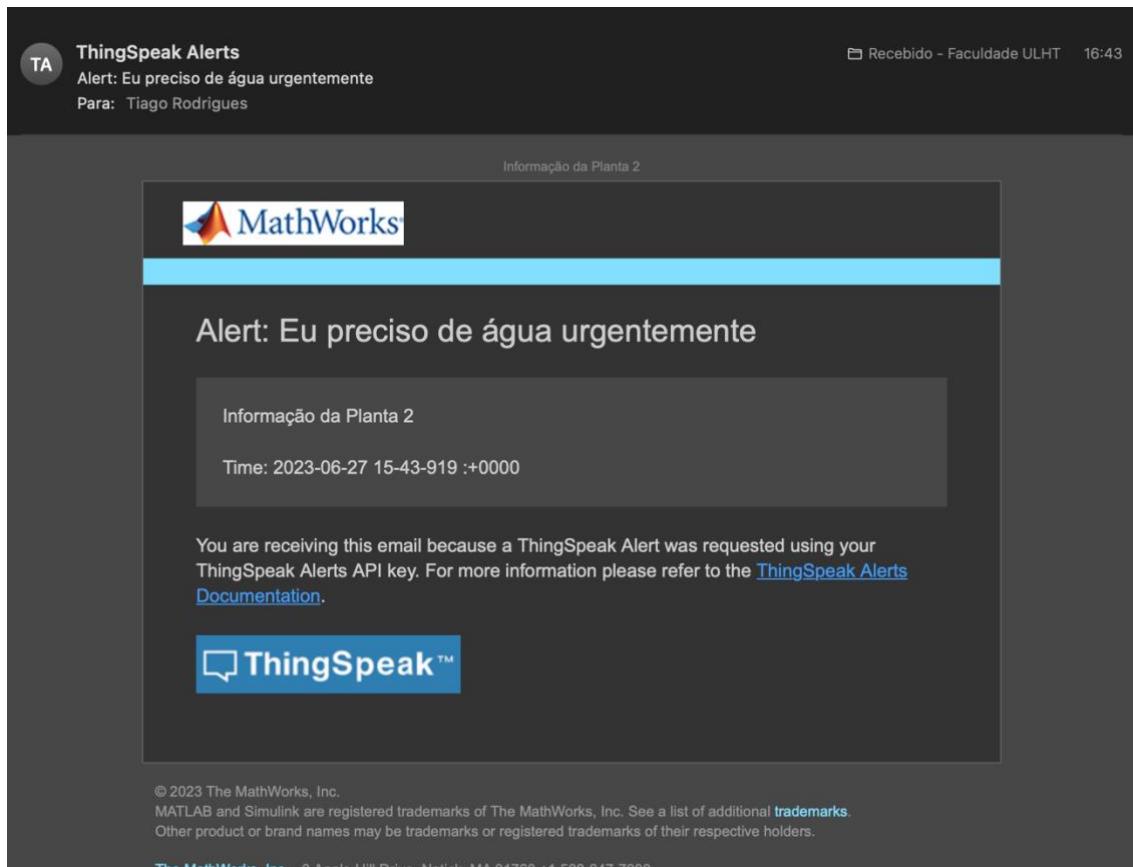


Figura 49 - Alerta da Planta 2

8.4 Decisão do Modelo em Machine Learning

Com a base de dados local em ambiente controlado e não controlado perante as plantas foi realizado um estudo do modelo que poderia classificar uma medição da planta e verificar se está válida para crescer ou necessita de água.

Para pode escolher o melhor modelo em si foi estudado os modelos de classificação que poderiam obter bons resultados no contexto em si sendo será apenas analisado 4 variáveis sendo elas a temperatura, luz, humidade e fertilidade nos seguintes modelos: Decision Tree, Gradient Tree Bost , Guassian Nave Bayes, Nearest Neighbours , RandomForest , Support Vector Machine e por fim redes neurais MLP.

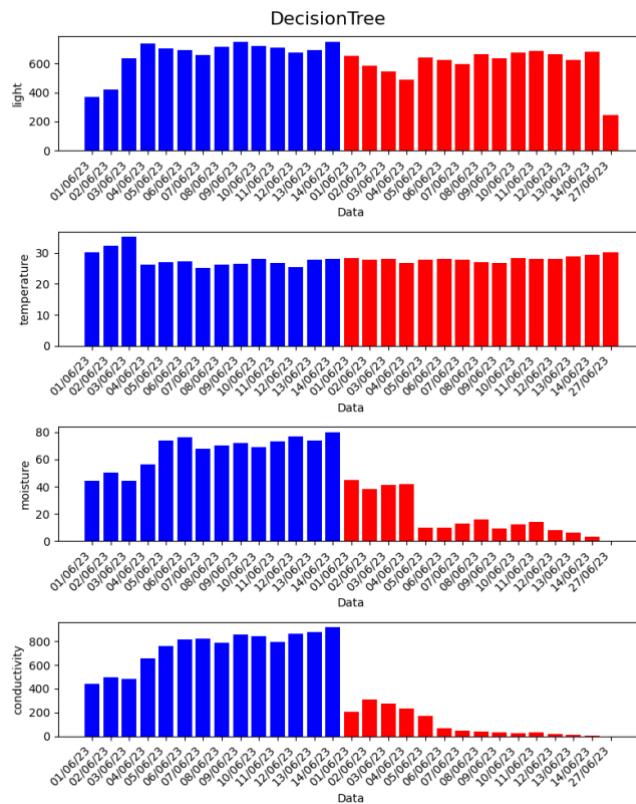


Figura 50 – Gráfico Decision Tree

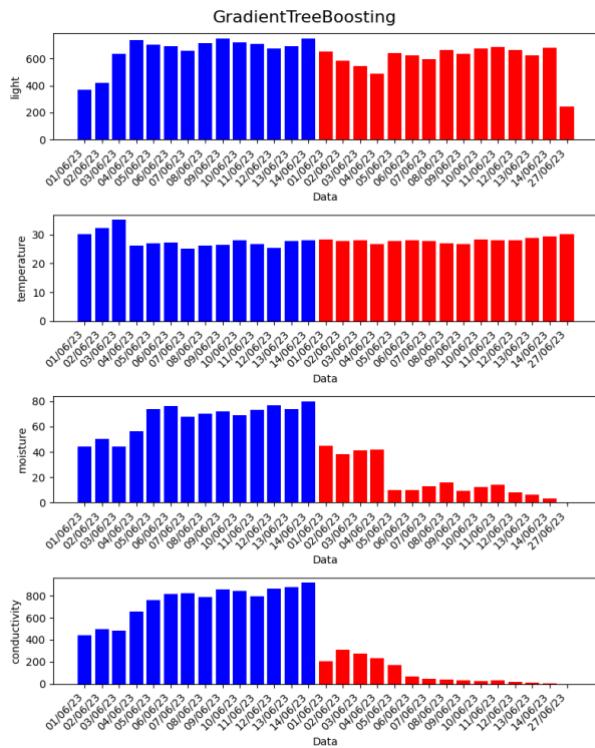


Figura 51 – Gráfico Gradient Tree Boosting

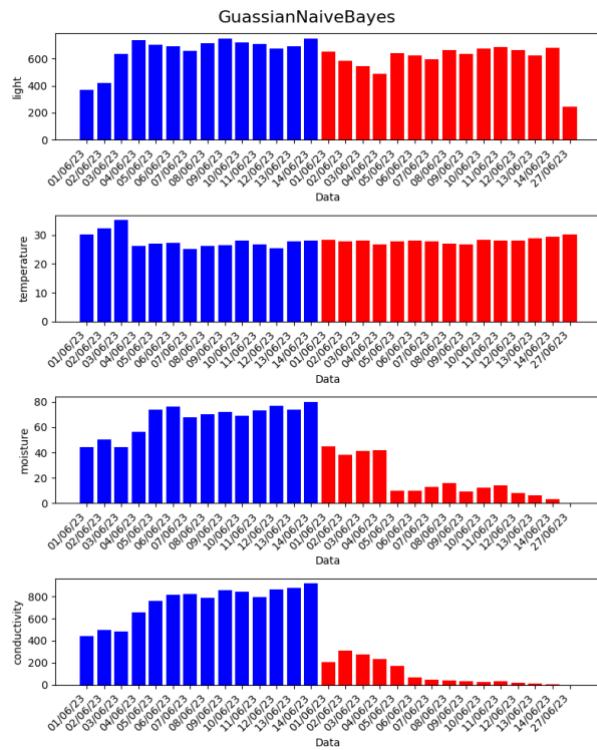


Figura 52 – Guassian Naive Bayes

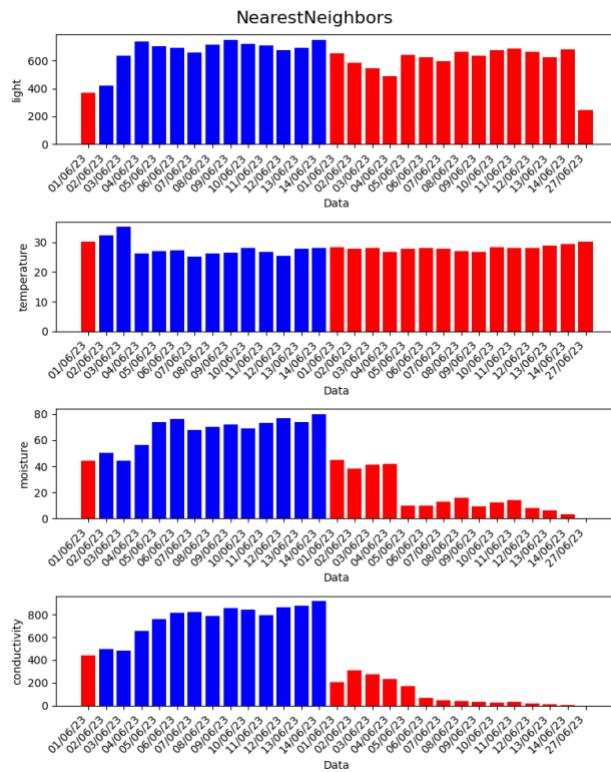


Figura 53 – Gráfico Nearest Neighbors

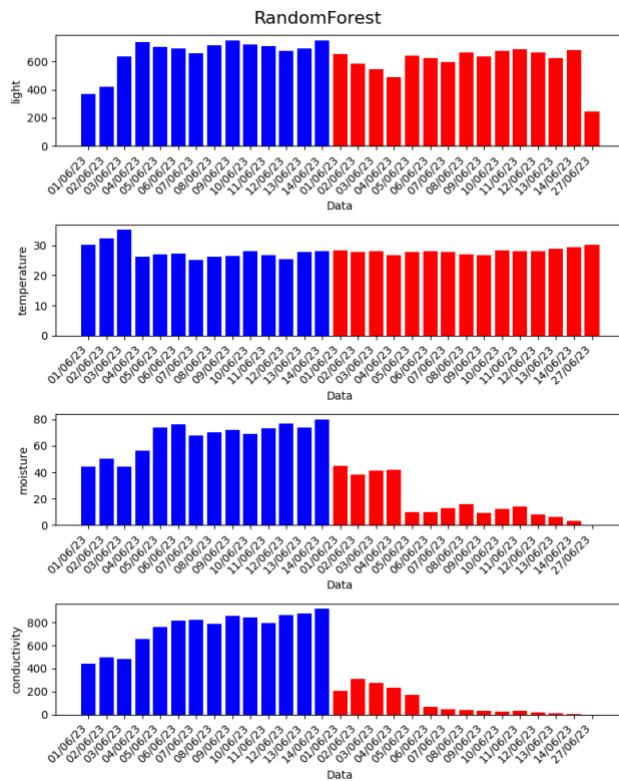


Figura 54 – Gráfico Random Forest

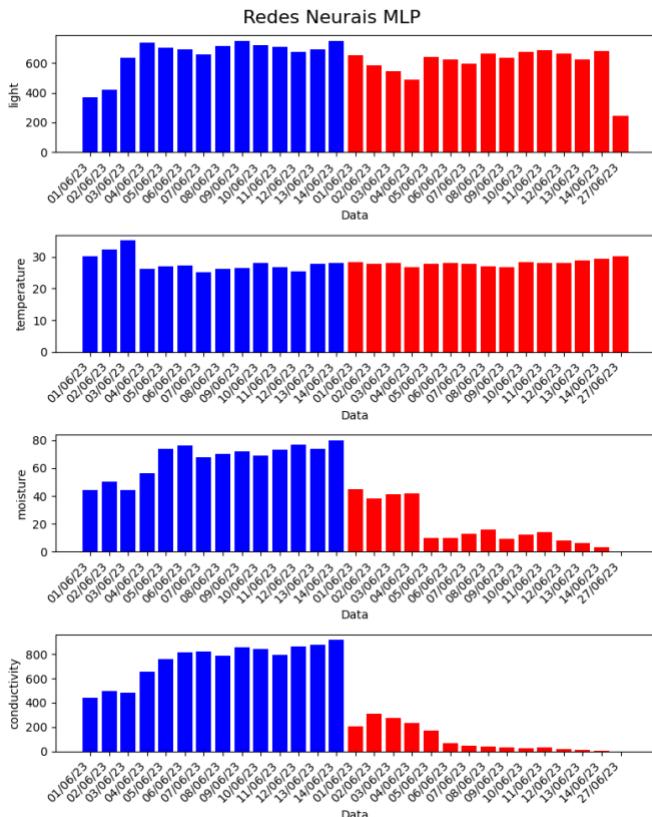


Figura 55 – Gráfico Redes Neurais MLP

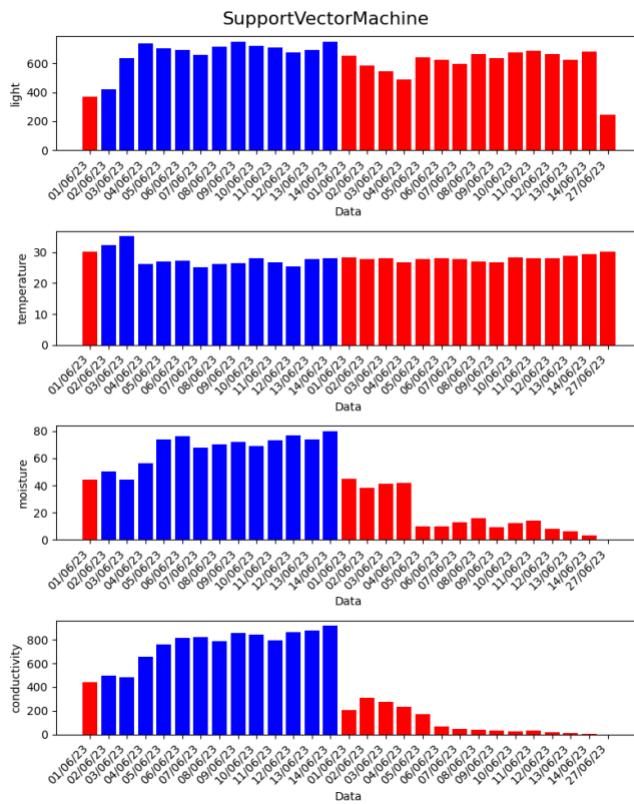


Figura 56 – Gráfico Support Vector Machine

No desenho dos gráficos foi designado a color azul para valor “1” enquanto a color vermelho para o valor “0” tendo assim um destaque mais perceptível numa análise visual podendo verificar que modelos como o Support Vector Machine e o Nearest Neighbours são excluídos pois não contém a sequência prevista pela forma de inserção dos dados. Essa sequência específica de colunas azuis e vermelhas foi criada ao unir os dados das duas plantas na plataforma de monitorização. Os primeiros 14 dias do gráfico representam os dados recolhidos da planta 1, na qual os parâmetros vitais permaneceram em níveis considerados bons. Os 15 dias seguintes do gráfico correspondem aos dados coletados da planta 2, na qual os parâmetros vitais mostraram valores considerados maus.

Observando os gráficos contenham a previsão de valores descrita anteriormente, foi escolhido o modelo Decision Tree pois é a que contem um menor custo de computação em comparação com as restantes com o mesmo resultado tornado assim o programa mais rápido no envio de dados para a plataforma ThingSpeak.

9 Conclusão e trabalhos futuros

Em conclusão, a plataforma de monitoramento de plantas apresenta uma solução abrangente e eficaz para casos domésticos e empresariais. Ao oferecer recursos de acompanhamento em tempo real, alertas e insights valiosos, a plataforma capacita os utilizadores a cuidarem de suas plantas com facilidade e eficiência.

No contexto doméstico, a plataforma permite que os usuários monitorem e controlem o crescimento e a saúde de suas plantas de maneira conveniente. Com notificações sobre rega, exposição à luz solar e nutrientes necessários, os proprietários de plantas podem garantir que suas preciosas vegetações recebam a atenção adequada. Isso resulta em plantas mais saudáveis, mais vibrantes e embelezando os espaços domésticos. Para as empresas, a plataforma de monitoramento de plantas é uma ferramenta valiosa em ambientes como estufas, viveiros ou empresas agrícolas. Com a capacidade de monitorar grandes quantidades de plantas simultaneamente, as empresas podem garantir condições ideais de crescimento, reduzindo os riscos de danos às plantas e maximizando a eficiência operacional. Além disso, a plataforma fornece dados estudados sobre o crescimento das plantas que permitem a tomada de decisões informadas em relação a práticas agrícolas, resultando em um cultivo mais produtivo e rentável.

Com esforço dedicado no desenvolvimento desta solução tive de lidar com pequenos desafios perante o envio para uma plataforma IoT como também implementar modelos de machine learning para decidir qual seria o modelo final para analise de crescimento de uma planta na qual obtive sucesso como poderemos verificar nos resultados e no planeamento. Contudo tal como referido no Capítulo de Viabilidade e Pertinência , não é um projeto que se esgote enquanto projeto académico, podendo ser continuado após a conclusão deste TFC através de um desenvolvimento de uma aplicação no contexto de Computação Móvel permitindo que os utilizadores consigam ver as suas plantas no seus dispositivos moveis sem terem de aceder a um website ou até mesmo simplificar a arquitetura do sistema evitando a utilização de um mini computador como o Raspberry Pi3 mantendo apenas o sensor de leitura de dados conhecido como o Xiaomi Mi Flora.

Bibliografia

- [DEISI21] DEISI, Regulamento de Trabalho Final de Curso, Set. 2021.
- [TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6^a Edição, Prentice Hall, 2020.
- [ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Out. 2021.
- [1] White, A.C. (no date) Why do some plants grow faster than others? A source-sink perspective. dissertation.
- [2] Turber, S. (no date) *How technology affords organizing value creation in the digital era the evolution of thriving business models in nascent IOT categories*. dissertation.
- [3] ONU (ed.) (2019) *Why-waste-water*. Available at:
<https://www.un.org/sustainabledevelopment/blog/2019/05/why-waste-water/>.

[Figura 2 - App de um sistema rega automática] iTimo. (2020, January 5). *App Rega Automático com sistema informático*. AliExpress. <https://pt.aliexpress.com/item/1005002822118029.html>

[Figura 3 – Empresa SPS] SPS. (2022, January 27). *SPS Company*. <https://advanced-uk.com/smart-plant-systems-sps/>

[Figura 4 – Solução Smart Plant Systems] SPS. (2022, January 27). *SPS Company*. <https://advanced-uk.com/smart-plant-systems-sps/>

[Figura 5 - Solução Daisy.si Watering Device] daisy.si. (ca. 2023, January 27). *Solution Daisy.si Watering Device*. <http://daisy.si>

[Figura 7 - Hardware da Solução IAC Smart Plant] IAC. (ca. 2023, January 28). *IAC Smart Plant*. <https://iac-intl.com/iac-smart-plant-remote-monitoring-system/>. <https://iac-intl.com/iac-smart-plant-remote-monitoring-system/>

[Figura 8 - Plataforma da Solução IAC Smart Plant] IAC. (ca. 2023, January 28). *IAC Smart Plant*. <https://iac-intl.com/iac-smart-plant-remote-monitoring-system/>. <https://iac-intl.com/iac-smart-plant-remote-monitoring-system/>

[Figura 15 – Raspberry Pi 3B] *Raspberry Pi*. (ca. 2023, January 27). Raspberry Pi Oficial Site. <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>

[Figura 16 – Xiaomi Mi Flora] *Mi Flower Care - Review of Xiaomi compatible plant monitor.* (ca. 2023, January 27). Smartme.pl. <https://smartme.pl/en/mi-flower-care-plant-monitor-review-compatible-with-xiaomi/>

[Figura 17 – Raspberry Pi Camera V2] *Raspberry Pi*. (ca. 2023, January 27). Raspberry Pi Oficial Site. <https://www.raspberrypi.com/products/camera-module-v2/>

[Figura 18 – Raspberry OS] *Raspberry Pi*. (ca. 2023, January 27). Raspberry Pi Oficial Site. <https://www.raspberrypi.com/software>

[Figura 19 – Linguagem Programação Python] *Python*. (ca. 2023, January 27). Python.org. <https://www.python.org>

[Figura 20 – Biblioteca Python Pandas] PyPi. (ca. 2020, January 27). *Pandas Logo*. <https://pypi.org/project/pandas/>

[Figura 21 – Biblioteca Python scikit learn] scikit-learn. (ca. 2023, January 27). *scikit-learn Machine Learning in Python*. scikit-learn.org. <https://scikit-learn.org/stable/>

[Figura 22 – Módulo Bluez] Lorenz, M. (2021, December 7). *BlueZ 5.50 and D-Bus*. raspberrypi-

bluetooth.com. https://www.raspberrypi-bluetooth.com/bluez-5_5-and-dbus.html

[Figura 24 – MQTT protocolo com Broker] Pinto, P. (2019, March 24). *MQTT: Protocolo de comunicação para pequenos dispositivos móveis*. Pplware. <https://pplware.sapo.pt/tutoriais/networking/mqtt-protocolo-de-comunicacao/>

[Figura 25 – Mosquitto MQTT Broker] Docker. (ca. 2022, January 27). *eclipse-mosquitto*. hub.docker.com. https://hub.docker.com/_/eclipse-mosquitto

[Figura 26 – Exemplo de um Dashboard da plataforma ThingSpeak] Thingsboard. (ca. 2023, January 27). *ThingsBoard Professional Edition Cloud*. ThingsBoard.io. <https://thingsboard.io/installations/>

[Figura 27 – ThingSpeak] - Thingsboard. Ca.(2023, January 27). *ThingsBoard* . ThingsBoard.io. <https://thingsboard.io>

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
BLE	Bluetooth Low Energy
RF	Requisito Funcional
RNF	Requisito Não Funcional
RT	Requisito Técnico
IOT	Internet of Things
ER	Entidade-Relação
ONU	Organização de Nações Unidas
UML	Unified Modeling Language
CSV	Comma Separated Value