

# Relatório de Projeto

Soila da Cruz Rita Benguela

Engenharia Informática

fev | 2023

GUARDA  
POLI  
TÉCNICO



# POLI TÉCNICO GUARDA

**Escola Superior de Tecnologia e Gestão**

---

## **APPT-ÁGUA POTÁVEL PARA TODOS**

---

PROJETO DE FIM DE CURSO  
PARA OBTENÇÃO DO GRAU DE LICENCIADO(A) EM ENGENHARIA  
INFORMÁTICA

**Soila Benguela**  
**Fevereiro / 2023**

# POLI TÉCNICO GUARDA

**Escola Superior de Tecnologia e Gestão**

---

## **APPT-ÁGUA POTÁVEL PARA TODOS**

---

PROJETO DE FIM DE CURSO  
PARA OBTENÇÃO DO GRAU DE LICENCIADO(A) EM ENGENHARIA  
INFORMÁTICA

Professor(a) Orientador(a): Maria Clara Silveira

**Soila Benguela**  
**Fevereiro / 2023**

## Agradecimentos

Primeiramente agradeço a Deus e aos meus pais por me terem dado a vida e por tudo que têm feito por mim. Os meus agradecimentos vão também para todos professores do Instituto Politécnico da Guarda (IPG). As minhas estimas a todos os professores e professoras desta instituição por me terem ensinado com zelo e dedicação. A minha mais alta consideração a minha Orientadora Doutora Maria Clara Silveira por ter me orientado nesta espinhosa tarefa. Um bem-haja a mim mesma e a todos aqueles que de uma forma direta ou indireta ajudaram-me nesta árdua missão.

## Ficha de Identificação

### **Aluno**

**Nome:** Soila da Cruz Rita Benguela

**Número:** 1701061

**Licenciatura:** Engenharia Informática

### **Estabelecimento de Ensino**

Instituto Politécnico da Guarda (IPG)

Escola Superior de Tecnologia e Gestão (ESTG)

### **Docente Orientador**

**Nome:** Maria Clara Silveira

**Grau académico:** Doutoramento em Engenharia Eletrotécnica e Computadores

**Email:** mclara@ipg.pt

## Resumo

O presente relatório visa descrever a implementação do sistema, Água Potável para Todos "APPT", de controle residencial da água, de baixo custo, para aproveitamento de água da chuva em regiões perto de Ecossistemas Lênticos e Lóticos. A água potável, com o tempo, torna-se um recurso escasso devido aos efeitos combinados do aumento da população e das mudanças climáticas. A monitorização da água é, portanto, fundamental, especialmente para uso doméstico. O sistema APPT baseia-se numa "interface web" para visualizar os dados lidos a partir de sensores conectados a um microcontrolador. Permite aos utilizadores monitorizar e visualizar em tempo real os dados enviados pelos sensores ao "Firebase Realtime Database", utilizado como base de dados na nuvem. A aplicação possui uma "interface" intuitiva e fácil de usar, que exhibe os dados de forma clara e organizada. A integração com o "Firebase" permite que os dados sejam armazenados de forma segura e acessíveis a partir de qualquer lugar.

Palavras-chave: Monitorização da água, Arduino, Firebase, Aplicação Web.

## Abstract

*The present report aims to describe the implementation of the system, Drinking Water for All (Água Potável para Todos "APPT"), of low-cost residential water control, for the use of rainwater in regions close to Lentic and Lotic Ecosystems. Drinking water, over time, becomes a scarce resource due to the combined effects of population growth, pollution and climate change. Water monitoring is therefore essential, especially for domestic use. The APPT system is based on a "web interface" to visualize data read from sensors connected to a microcontroller. Which allows users to monitor and visualize in real time the data sent by the sensors to the "Firebase Realtime Database", used as a database in the cloud. The application has an intuitive and easy-to-use "interface", which displays data in a clear and organized way. Integration with "Firebase" allows data to be stored securely and accessible from anywhere.*

*Keywords: Water monitoring, Arduino, Firebase; Web Application.*

# Índice

Agradecimentos .....	iii
Ficha de Identificação.....	iv
Resumo.....	v
Abstract .....	vi
Índice de Figuras .....	ix
Índice de Tabelas.....	x
Lista de Siglas e Acrónimos .....	xi
1. Introdução .....	1
1.1 Enquadramento e Motivação.....	1
1.2 Descrição do Problema .....	1
1.3 Objetivos .....	2
1.4 Estrutura do Relatório.....	2
2 Estado da Arte .....	3
2.1 Smart Water Meter .....	3
2.2 Buoy and Energy Systems .....	5
2.3 Smartroll Multiparameter .....	8
2.4 Análise Crítica.....	9
3 Metodologia .....	11
3.1 Atividades Scrum.....	11
3.2 Síntese das Atividades Realizadas .....	12
4 Análise de Requisitos .....	14
4.1 Reuniões Realizadas .....	14
4.2 Tabela de Casos de Uso.....	14
4.3 Diagrama de Casos de Uso .....	16
5 Implementação .....	18
5.1 Tecnologias Utilizadas .....	18
5.2 Sensores e Atuador .....	19
5.3 Arquitetura do Sistema .....	25
5.4 Conexão ao Firebase .....	27
5.5 Aplicação Web.....	32
6 Verificação e Validação .....	40
7 Conclusão .....	41
Bibliografia .....	42



Anexo A .....	44
Anexo B .....	47
Anexo C .....	49

## Índice de Figuras

Figura 1-Fluxograma do Sistema Smart Water Mete.....	4
Figura 2-Handheld Embedded System.....	4
Figura 3-Aplicação Android do sistema Smart Water Meter .....	5
Figura 4-Visão geral do sistema Buoy and Energy Systems .....	6
Figura 5- Fluxograma do Sistema Buoy and Energy Systems.....	7
Figura 6-Interface web do Sistema Buoy and Energy Systems .....	8
Figura 7-Dados enviados por SMS Buoy and Energy Systems .....	8
Figura 8- Scrum Overview .....	11
Figura 9- Planeamento do Sprint Backlog.....	12
Figura 10-Atividades do Sprint Backlog aplicado ao presente projeto .....	13
Figura 11-Diagrama de casos de uso do sistema APPT.....	17
Figura 12-Microcontrolador Wemos D1 mini Pro ESP8266 .....	19
Figura 13-Imagem é do Wemos D1 mini Pro, a versão 3.0.0 .....	20
Figura 14-Sensores DHT11 e os pinos de ligação .....	21
Figura 15-Diagrama de tempo do SRF04.....	22
Figura 16- Sensor HC-SR04.....	22
Figura 17-Sensor de PH.....	23
Figura 18-Servomotor FS90R POLOLU .....	24
Figura 19-Ligações dos Sensores com o Esp8266 .....	25
Figura 20- Arquitetura do Sistema APPT.....	26
Figura 21-Diagrama Geral de Alto Nível.....	27
Figura 22 Função para Controlar o Estado do Doseador .....	28
Figura 23-Função para Calcular a Temperatura.....	29
Figura 24-Função para Calcular o ph.....	30
Figura 25-Função para Calcular a Distância .....	31
Figura 26-Imagem dos dados recebidos no Firebase.....	32
Figura 27-Função para gerir LOGIN/LOGOUT UI.....	33
Figura 28-Código para Gerir a Atualização dos Dados na Interface web.....	34
Figura 29-Função para criar a Tabela.....	35
Figura 30-Tela de login da aplicação web .....	36
Figura 31-Imagem da Interface web .....	37
Figura 32-Função para controlar o doseador na Interface Web.....	37
Figura 33-Imagem do doseador do cloro ligado e desligado .....	38
Figura 34-Imagem da tabela exibida na Interface web.....	39
Figura 35- Sensor DS18B20 .....	45
Figura 36-Sensor DFRobot SEN0204 .....	46
Figura 37-Notificação do nível da água.....	50

## Índice de Tabelas

Tabela 1-Tabela comparativa de Funcionalidade .....	10
Tabela 2-Tabela de Atores e respectivos objetivos .....	15
Tabela 3-Sumário de resultados dos testes realizados .....	40

## Lista de Siglas e Acrónimos

<b>API REST</b>	Interface de Programação de Aplicação Restful
<b>APPT</b>	Água Potável Para Todos
<b>CDN</b>	<i>Content Delivery Network</i>
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>EPA</b>	<i>Environmental Protection Agency</i>
<b>ESTG</b>	<i>Escola Superior de Tecnologia e Gestão</i>
<b>GSM</b>	<i>Global System for Mobile Communications</i>
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>ID</b>	<i>Identity</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IoT</b>	<i>Internet of Things</i>
<b>IPG</b>	Instituto Politécnico da Guarda
<b>JDK</b>	<i>Java Development Kit</i>
<b>MP</b>	<i>Smartroll Multiparameter</i>
<b>OD/DO</b>	Oxigênio Dissolvido/ <i>Dissolved Oxygen</i>
<b>ORP</b>	<i>Oxidation-Reduction Potential</i>
<b>RTC</b>	<i>Real-Time Clock</i>
<b>SDK</b>	<i>Software Development Kit</i>
<b>SMS</b>	<i>Short Message Service</i>
<b>SQL</b>	<i>Structured Query Language</i>
<b>SWM</b>	<i>Surface Water Management</i>
<b>TDS</b>	<i>Total Dissolved Solids</i>
<b>UID</b>	<i>Unique ID</i>

<b>UML</b>	<i>Unified Modeling Language</i>
<b>URL</b>	<i>Uniform Resource Locator</i>
<b>WSN</b>	<i>Wireless Sensor Network</i>

# 1. Introdução

O presente relatório descreve o projeto desenvolvido pela aluna Soila Benguela, no âmbito da unidade curricular Projeto de Informática, da Licenciatura em Engenharia Informática no Instituto Politécnico da Guarda, sob a orientação da Professora Doutora Maria Clara Silveira. No decorrer do desenvolvimento do projeto foi implementada o sistema, Água Potável para Todos "APPT", de controle residencial da água, de baixo custo. O sistema APPT foi desenvolvido usando HTML, JavaScript, CSS e Firebase. Permite ao utilizador ter acesso a conteúdo dinâmico através da *internet*, um dos muitos benefícios da "Internet of Things (IoT)." A ideia base da IoT é que os objetos do dia a dia sejam equipados com tecnologia de computação e conectividade, tornando-os inteligentes e capazes de recolher e partilhar dados. Isso permite a criação de sistemas mais eficientes e automatizados, que podem ser usados em muitas áreas, como saúde, transporte, agricultura, entre outras.

## 1.1 Enquadramento e Motivação

A ideia principal deste projeto surgiu pela motivação de desenvolver um sistema básico de gestão dos recursos hídricos em São Tomé e Príncipe, país de origem da autora. O país é rico em biodiversidade, tanto na fauna como na flora, com uma floresta densa, rios e nascentes que brotam um pouco por todo o lado. Mesmo com tanta abundância em recursos hídricos, a produção de água não é suficiente para satisfazer as necessidades da população. Um dos muitos problemas é a falta de recursos humanos capacitados e de ferramentas para a gestão da água. Com esta preocupação em mente, surgiu a ideia de construir um sistema de baixo custo para automatização de um purificador de água caseiro, que possa minimizar estes constrangimentos e que seja útil à sociedade.

## 1.2 Descrição do Problema

Ter água potável em casa é fundamental e é uma necessidade para toda a população, no entanto, ainda existem pessoas que não contam com esse tipo de benefício. Aqueles que não recebem água potável, podem também buscar por alternativas para contornar esse problema, a criação de um purificador de água caseiro. Pode parecer algo impossível, mas com poucos materiais, baratos, pode-se criar um purificador de água caseiro. O principal problema que este projeto tenta resolver é a realização da automação de um purificador de água caseiro de uma forma económica.

### 1.3 Objetivos

O objetivo deste trabalho é mostrar que é possível automatizar um purificador de água caseiro com a construção de um projeto de baixo custo, onde com apenas uma aplicação e um microcontrolador consegue-se controlar dispositivos remotos numa residência, assim como ler os dados a partir de diversos sensores, criar um doseador de cloro e o sistema de alertas que avisa o utilizador do nível de água e o estado do doseador. Também foi desenvolvida uma aplicação Web para exibir todos os dados lidos dos sensores, guardados na “Firebase Realtime Database”, em forma de medidores e uma tabela onde é possível gerir todos os dados registados. Os mesmos são protegidos e acessíveis apenas pelo utilizador que os criou ou que possui as permissões necessárias usando e-mail e *password*.

### 1.4 Estrutura do Relatório

O presente documento está dividido em sete capítulos distintos, designadamente a introdução, estado da arte, metodologia, análise de requisitos, implementação, verificação e validação, e conclusão. O primeiro capítulo apresenta a introdução ao projeto, bem como os objetivos e a motivação. No segundo capítulo é realizado um estudo das aplicações existentes e a seguir uma análise crítica de cada uma delas. O terceiro capítulo trata a metodologia utilizada neste projeto. No quarto capítulo é feita uma análise de requisitos. O quinto capítulo trata a implementação, mais precisamente as tecnologias utilizadas e o desenvolvimento do sistema APPT. No sexto capítulo, apresenta-se a verificação e validação do sistema. O sétimo e último capítulo apresenta a conclusão e trabalho futuro relativo a este projeto.

## 2 Estado da Arte

O Estado da Arte é uma das partes mais importantes de todo trabalho científico, uma vez que faz referência ao que já se tem descoberto sobre o assunto pesquisado. Além disso, auxilia na melhoria e desenvolvimento de novos postulados, conceitos e paradigmas. Trata-se de uma atividade árdua por ser crítica e reflexiva. Para o trabalho presente foi realizado, pesquisas em buscas de aplicações, com características inovadoras e semelhantes com os objetivos propostos ao projeto. O resultado dessas pesquisas, foi, alguns artigos que serão apresentados.

### 2.1 Smart Water Meter

Os autores [1] apresentam um sistema com foco na qualidade da água nas áreas rurais e difícil de monitorizar devido à falta de conectividade de diferentes laboratórios de água. O sistema desenvolvido mede o pH, Sólidos Dissolvidos Totais (TDS) e temperatura de amostras da água por meio de sensores disponíveis no mercado. O conjunto de dados medidos de pH e TDS foram utilizados para saber outros parâmetros como salinidade, ORP<sup>1</sup> e condutividade por relações matemáticas padrão. O esquema do sistema integrado e o fluxograma de funcionalidade estão representados na Figura 1.

---

<sup>1</sup> Um ORP (Potencial de Redução da Oxidação) é uma medida da carga molecular que se refere à capacidade de uma substância para oxidar ou reduzir a oxidação e é medida em milivolts (mV). Uma carga positiva de ORP indica que a substância é um oxidante. Quanto maior a carga positiva, mais forte é seu poder oxidante.



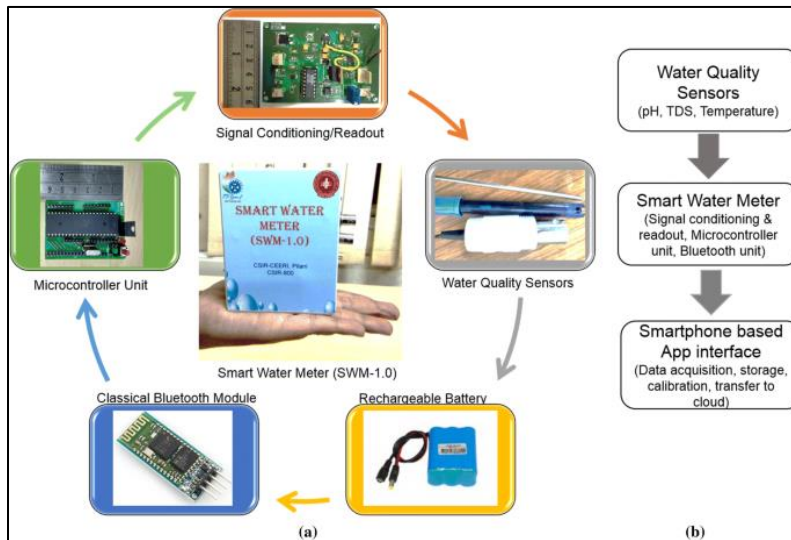


Figura 1-Fluxograma do Sistema Smart Water Meter [1]

A Figura 1 inclui dois esquemas:

- Esquema de vários componentes do protótipo do medidor da qualidade da água inteligente;
- Esquema de integração do medidor de qualidade da água inteligente (SWM-1.0).

O sistema embestado é composto por vários módulos, como a unidade do circuito condicionador de sinal, unidade do microcontrolador, módulo Bluetooth e uma bateria recarregável. O sistema funciona com bateria de 7,5 V por mais de 8–10 h continuamente sem carregar, como é descrito na Figura 2.

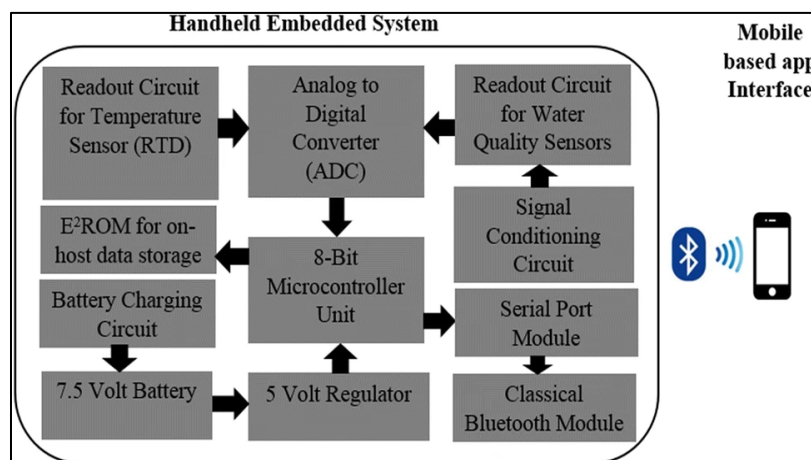


Figura 2-Handheld Embedded System [1]

O sistema é de baixo custo e fácil de operar por uma aplicação Android, ilustrado na Figura 3, sendo desenvolvido para comunicar com o dispositivo usando o protocolo Bluetooth. A aplicação Android exibe os parâmetros de qualidade água e o nível de bateria. Também fornece as informações da água de um determinado local e o armazena em formato de ficheiro Excel.

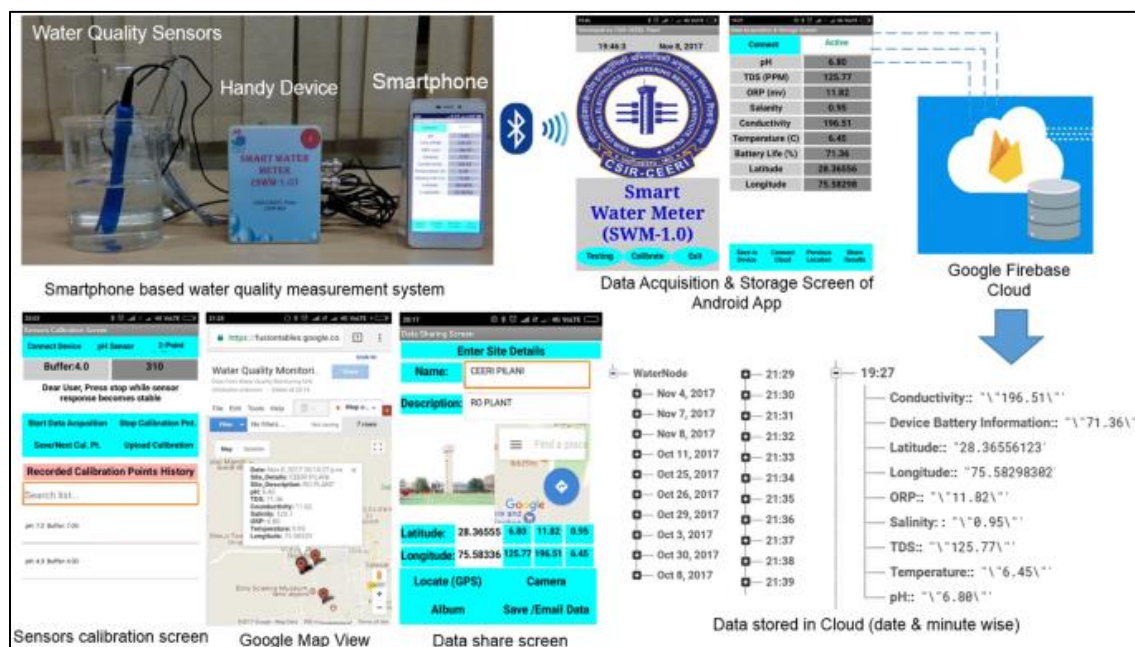


Figura 3-Aplicação Android do sistema Smart Water Meter [1]

A aplicação Android oferece várias opções para compartilhar dados e avisos de qualidade da água baseados na localização usando várias opções, como SMS, WhatsApp e e-mail e salva o histórico de calibração no próprio “smartphone” para fins de registo.

## 2.2 Buoy and Energy Systems

Os autores do [2] por sua vez, apresentam um sistema de monitorização da qualidade da água em tempo real e de baixo custo que pode ser aplicado a rios remotos, lagos, áreas costeiras e outros corpos d'água. O “hardware” principal do sistema consiste em sensores eletroquímicos disponíveis no mercado, um microcontrolador, um sistema de comunicação sem fio e uma boia adaptada. Deteta a temperatura da água, oxigénio dissolvido<sup>2</sup> na água e pH num intervalo de tempo pré-programado. O protótipo desenvolvido divulga as

<sup>2</sup> Oxigénio dissolvido (OD ou DO), ou por vezes saturação em oxigénio, é uma medida relativa da quantidade de oxigénio dissolvido num determinado fluido ou é por ele transportado. A unidade padrão é, em geral, miligramas por litro (mg/l) ou partes por milhão (ppm).

informações recolhidas em gráficos e tabelas por meio de uma “Interface web”, a visão geral do sistema é mostrada na Figura 4.

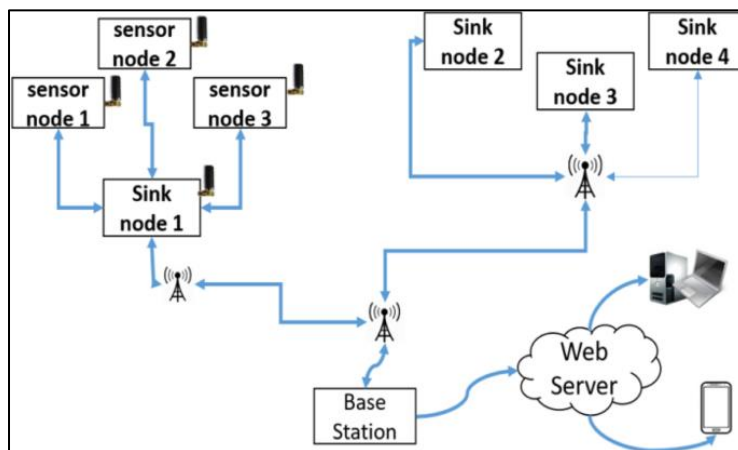


Figura 4-Visão geral do sistema Buoy and Energy Systems [2]

A função mais importante do sistema é garantir que os dados recolhidos dos sensores sejam lidos em tempo real. Os valores serão enviados via SMS para um número de telemóvel predefinido e fornece gráficos para facilitar a monitorização como é demonstrado a Figura 6 e Figura 7, e também fornecer folhas de dados e ficheiro PDF para fins de pesquisa.

Portanto, o sistema foi projetado para fornecer o seguinte:

1. Recolha e armazenamento de dados;
2. Divulgação e exibição de dados (tabelas, gráficos e folhas de dados);
3. Gestão de energia por longos períodos;
4. “Interface” de manutenção para administradores.

A Figura 5 exibe do sistema de monitorização da qualidade da água proposto. Após a execução de um nó “node”, o sistema inicializará o cartão SD e o relógio em tempo real (RTC). Também preparará o módulo GSM para transferência de dados.

Após alguns segundos, o sistema começará a recolher dados através dos sensores de pH, DO e temperatura. Os dados serão enviados utilizando o transceptor GSM para a estação base<sup>3</sup> ou para o número de telemóvel pré-identificado. Além disso, os dados serão armazenados em RTC como “backup” com o mesmo formato da base de dados. Na ausência de um sinal GSM, os dados serão salvos automaticamente no RTC.

<sup>3</sup> Servidor da estação base (ou Base Station Server, em inglês) é um servidor de rede que controla e gerencia a comunicação entre as estações-base numa rede sem fio. As estações-base são os dispositivos que fornecem a conexão sem fio aos dispositivos clientes, como “smartphones”, “tablets”, portáteis, entre outros. Elas são responsáveis por transmitir e receber dados e fornecer conectividade sem fio aos dispositivos.

Após a transferência e “backup” de dados através do RTC, o sistema desligará automaticamente e aguardará o intervalo de tempo de 30 minutos para economizar energia. Um circuito temporizador despertará o sensor se atingir o intervalo de tempo necessário de 30 minutos. O ciclo continuará até que a tarefa necessária seja alcançada.

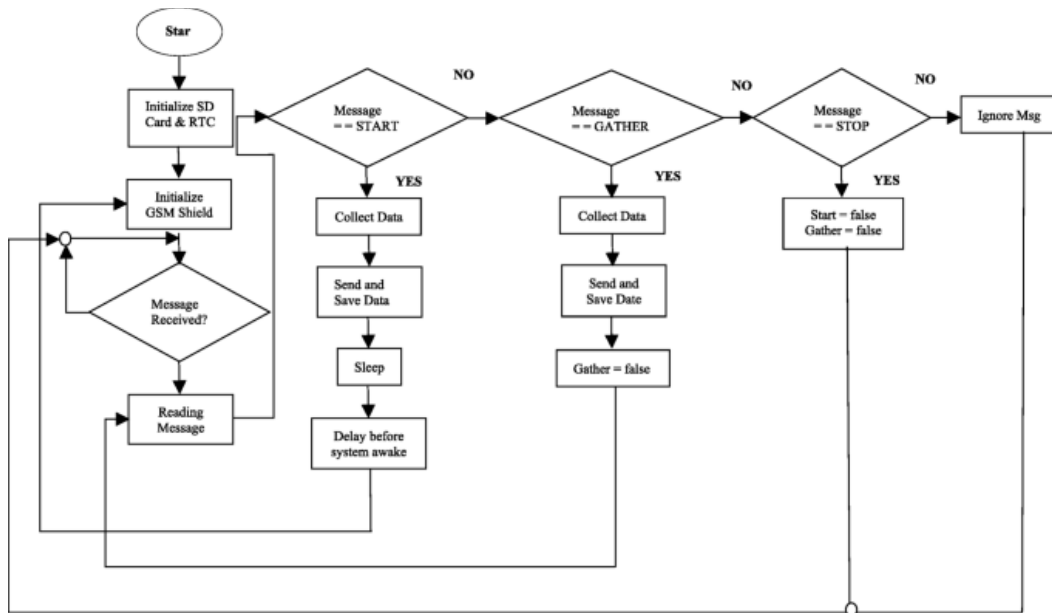


Figura 5- Fluxograma do Sistema Buoy and Energy Systems [2]

O sistema foi implantado em dois ribeiros, durante duas semanas, de modo a verificar a estabilidade e eficácia da boia, fonte de energia e armazenamento. Os dados detetados foram enviados através dos transdutores XBee<sup>4</sup> para o nó sorvedouro que agrega dados do nó sensor e os seus próprios dados. Após o envio dos dados, os sensores voltam ao modo hibernação para economizar energia da bateria, formando outro ciclo do sistema.

<sup>4</sup> Os transdutores XBee são módulos sem fio confiáveis que podem ser utilizados numa ampla variedade de projetos de IoT para fornecer comunicação sem fio confiável e em malha entre dispositivos. Os transdutores XBee são configuráveis e podem ser programados para se comunicarem utilizando diferentes protocolos de comunicação sem fio, como Wi-Fi, ZigBee, Bluetooth, Low Energy (BLE) e outros.

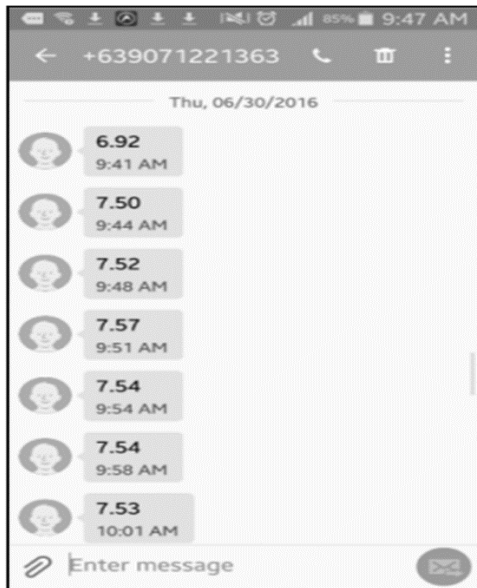


Figura 7-Dados enviados por SMS Buoy and Energy Systems [2]

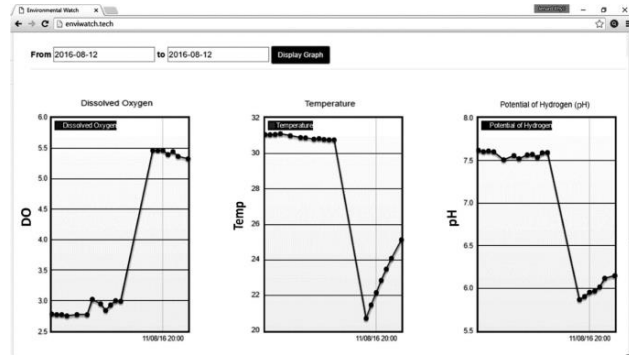


Figura 6-Interface web do Sistema Buoy and Energy Systems [2]

Os gráficos de dados podem ser visualizados na Interface “web” como mostrado na Figura 6. Isso poderia dar tempo suficiente para a gestão e as partes interessadas tomarem ações para prevenir acidentes relacionados ao aumento ou diminuição significativa dos valores monitorizados do parâmetro de qualidade da água.

### 2.3 Smartroll Multiparameter

Os atores [3] desenvolveram um sistema de monitorização de água, para uma aplicação móvel Smartroll Multiparameter (MP) da empresa In-Situ. Ideal para verificações no local da qualidade da água, o MP pode ser utilizado para qualquer aplicação de monitorização de águas superficiais. Não é necessário tempo de treino para utilizar a sonda, a aplicação intuitiva para “smartphone”. A In-Situ diz que os técnicos podem ler instantaneamente os resultados de 14 parâmetros de qualidade da água utilizando simplesmente um “smartphone”, sem a necessidade de um medidor portátil. A aplicação é projetado para ser executável num iPhone, iPod touch ou iPad, facilita o envio instantâneo de dados por e-mail ao colaborador e o registo de dados específicos do site num “smartphone”. O sistema MP inclui uma sonda compacta com sensores inteligentes testados em campo e uma bateria de encaixe com a tecnologia sem fio Bluetooth.

O MP pode ser utilizado para verificação no local, monitorização de águas pluviais, perfil vertical e amostragem de águas subterrâneas de baixo fluxo. Mede também o DO, condutividade, ORP, pH, salinidade, sólidos dissolvidos totais, nível de água, pressão

barométrica, temperatura e muito mais. A sonda possui tecnologia de oxigênio dissolvido aprovada pela EPA para determinações instantâneas de OD.

## 2.4 Análise Crítica

No contexto geral dos artigos aqui apresentados, têm uma descrição rigorosa de como o estudo, elaboração e execução dos seus projetos em destaque, foram realizados. Tendo uma explicação das escolhas dos materiais e métodos. Os métodos foram descritos de forma clara, permitindo aferir a reprodutibilidade dos resultados obtidos e avaliação dos mesmos por parte dos leitores.

Os autores [1] tiveram como foco principal, a qualidade da água nas áreas rurais difíceis de monitorizar devido à falta de conectividade de diferentes laboratórios de água. Apresentaram um projeto baseado num sistema mobile Android que armazena parâmetros de pH o TDS e a temperatura das amostras, transfere para a nuvem com informações de localização usando o Google Maps. Prediz o índice de qualidade da água em tempo real e também fornece com facilidade para integração de novos dados de qualidade da água. O sistema em geral é económico, portátil e fácil de utilizar para os moradores ou qualquer pessoa não qualificada.

Os autores [2] apresentam adequadamente o enquadramento do sistema de monitorização da qualidade da água, que pode ser aplicado em rios, lagos, áreas costeiras e outros corpos d'água remotos. Em termos de "hardware", foi feito o uso de sensores eletroquímicos, microcontrolador, possibilitando usar o DO, pH e a temperatura como indicadores de qualidade da água, usaram tecnologias WSN rede de sensores sem fio adequada para monitorizar as características físicas e químicas da água em áreas remotas a um custo menor e reduzir a necessidade de trabalhadores.

Por fim, os autores [3] apresenta o sistema MP que inclui uma sonda compacta com sensores inteligentes testados em campo e uma bateria de encaixe com e utilizando a tecnologia Bluetooth. Além de todos os parâmetros já referidos. Infelizmente não foi fornecido informações detalhadas que possa facilitar a replicação do projeto.

Tabela 1-Tabela comparativa de Funcionalidade

Funcionalidades	Smart Water [1]	Buoy and Energy Systems [2]	Smartroll Multiparameter [3]	APPT
Monitorizar parâmetros da água em tempo real	✓	✓	✓	✓
Aplicação desenvolvida para smartphone ou web	✓	✓	<sup>5</sup> —	✓
Medir diferentes parâmetros da água	✓	✓	✓	✓
Compartilhar dados e avisos usando diferentes opções, como SMS, WhatsApp e E-mail	✓	✓	—	✓
Exibição de informações na web	✓	✓	—	✓
Doseador de cloro (atuador para abrir e fechar)	✗	✗	✗	✓

Depois de analisadas as funcionalidades de cada uma das aplicações, a proposta deste projeto visa a criação de uma aplicação web, para análise dos parâmetros da água tendo em conta os seguintes:

- Condições da temperatura e pH e o nível da água no purificador caseiro de água;
- Controlar o cloro residual com a manipulação do doseador;
- Notificação do nível da água, estado do doseador de cloro e do ph.

<sup>5</sup> Não foi fornecido informações sobre a Funcionalidade

### 3 Metodologia

A metodologia é o estudo dos métodos. Isto é, o estudo dos caminhos para se chegar a um determinado fim. Visando analisar as características dos vários métodos indispensáveis tais como: avaliar capacidades, limitações e criticar os pressupostos quanto a sua utilização. Além de ser uma disciplina que estuda os métodos, a metodologia é também considerada uma forma de conduzir a pesquisa ou um conjunto de regras para ensino de ciência e arte. Fornecendo um processo dinâmico e interativo para o desenvolvimento estruturado de projetos com vista à qualidade do produto final [4].

Neste capítulo é abordado o estudo dos métodos de desenvolvimento de “software” ágil, metodologia adotada neste projeto, em inglês, “Agile development”. E entender a lógica do método ágil de desenvolvimento de “software”, nomeadamente o “SCRUM”.

#### 3.1 Atividades Scrum.

Scrum *framework* da metodologia ágil é, atualmente, a mais utilizada principalmente pela sua simplicidade e facilidade de adoção e adaptação. Scrum caracteriza-se pelas etapas de desenvolvimento, definidas como Sprints. O funcionamento deste modelo pode ser observado na

Figura 8 e na Figura 9. Como pode ser visto na Figura 9, o conjunto de funcionalidades definidas para um produto a ser entregue num projeto são agrupadas no Product Backlog.

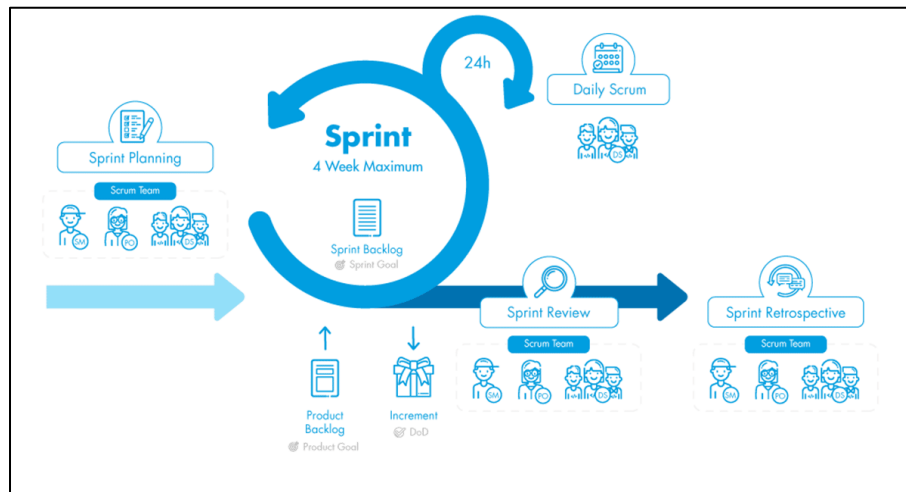


Figura 8- Scrum Overview [5]



A Daily Scrum é uma reunião com a duração máxima de 15 minutos que ocorre no início de cada dia, em que cada membro da equipa informa sobre o que foi feito no dia anterior e o plano para o próprio dia. Nesta reunião também são identificadas as eventuais situações que estejam a bloquear o andamento do processo de desenvolvimento.

A Sprint Review Meeting é uma reunião realizada no final do ciclo de um Sprint, em que a equipa de desenvolvimento apresenta as funcionalidades que foram implementadas nesse ciclo.

Realiza-se também a Sprint Retrospective com as lições aprendidas durante o ciclo, os pontos positivos e o que é preciso melhorar nos próximos Sprints, é então planeado o próximo Sprint.

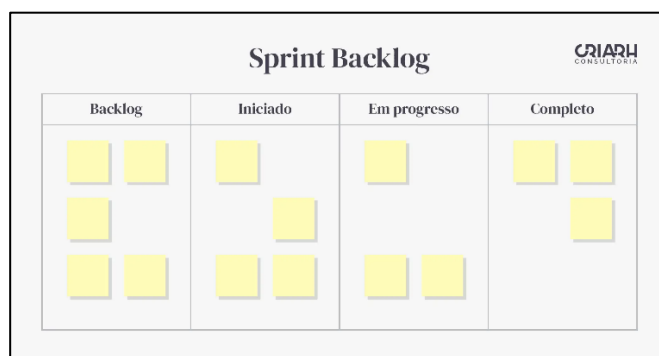


Figura 9- Planeamento do Sprint Backlog [6]

### 3.2 Síntese das Atividades Realizadas

O processo de desenvolvimento neste projeto foi uma adaptação personalizada da metodologia Scrum. A equipa é representada pela aluna e o Scrum Master composto pelo orientador e pelo docente da unidade curricular Projeto de Informática, que em conjunto definem as tarefas a serem desenvolvidas e entregues. O conjunto de tarefas é agrupado num "Product Backlog" que será desenvolvido até à próxima Sprint. As reuniões semanais com o orientador simulam uma Sprint que consistem em planear as tarefas a desenvolver. Estas tarefas a serem implementadas representam a "Sprint Backlog". Na **Erro! A origem da referência não foi encontrada.**, apresenta-se as atividades de um "Sprint Backlog" aplicado no presente projeto. O "Sprint Backlog" é composto por 13 semanas, em que cada semana representa uma reunião realizada com o orientador e ou docente da unidade curricular Projeto de Informática.

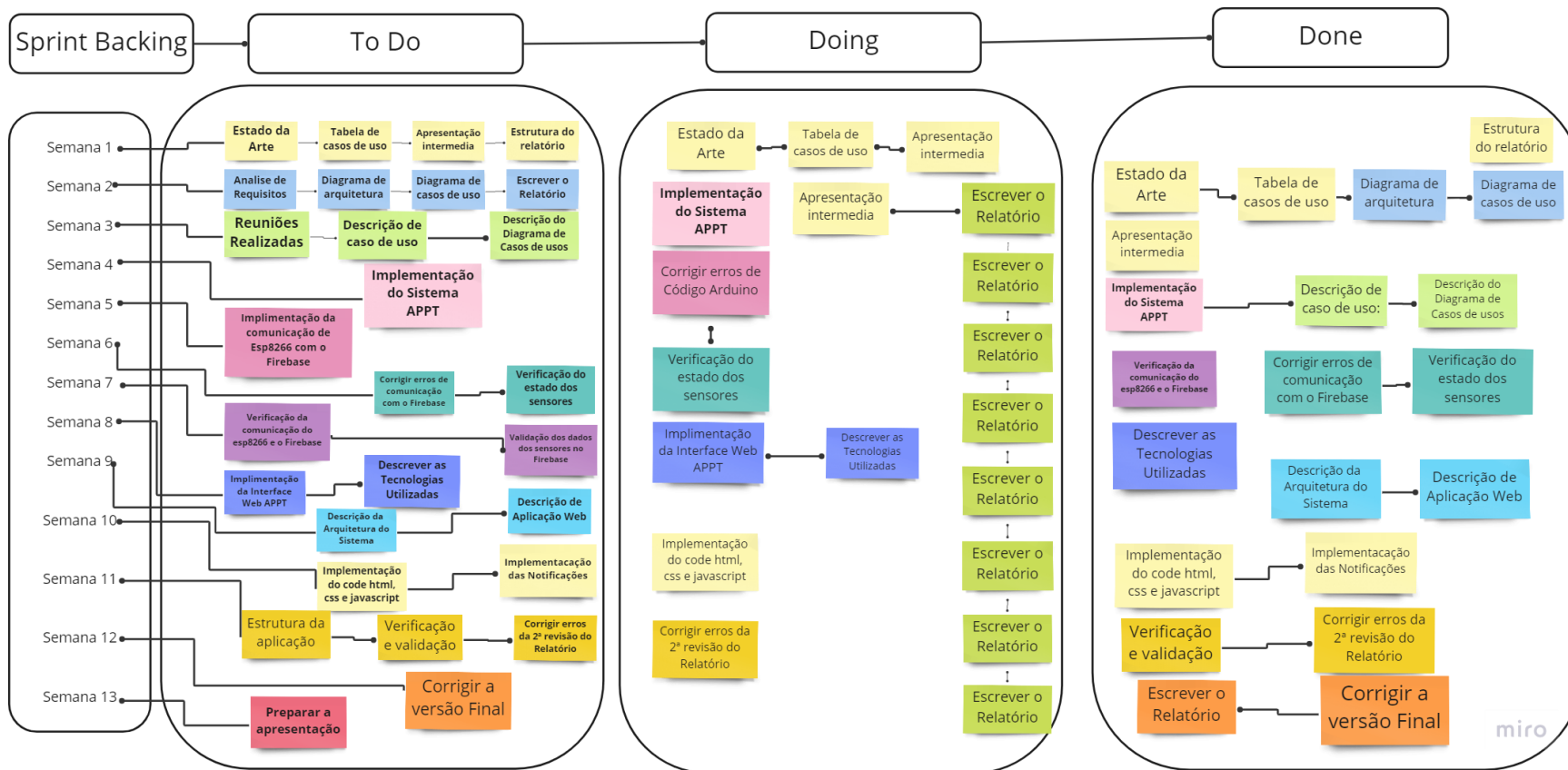


Figura 10-Atividades do Sprint Backlog aplicado ao presente projeto

## 4 Análise de Requisitos

Os requisitos são características que o software ou o sistema a desenvolver deverá cumprir. Definem claramente aquilo que o software terá de realizar para satisfazer as necessidades da organização e dos utilizadores. Os requisitos podem ter dupla função: base de um contrato aberto às interpretações; base do próprio contrato definido em detalhe. Para capturar os requisitos usam-se casos de uso. Um caso de uso é um modo específico de usar um sistema alguma funcionalidade é realizada pelo sistema como resposta a um estímulo do ator. Os casos de uso formam um diálogo entre o ator e o sistema. Os casos de uso providenciam um veículo para: capturar os requisitos sobre o sistema; comunicar com os utilizadores finais e os peritos na matéria; testar o sistema. Ao longo deste capítulo usa-se a linguagem UML (Unified Modeling Language) para modelar o sistema APPT. A UML é uma linguagem de modelação gráfica para especificação, visualização, construção e documentação de sistemas de software ou outros [6].

### 4.1 Reuniões Realizadas

Ao longo do desenvolvimento do projeto, realizou-se algumas reuniões com a presença dos Senhores Professores Pedro Rodrigues<sup>6</sup>, Filipe Caetano e a Senhora Professora Maria Clara Silveira. A preocupação principal era o que deveria ser usado no desenvolvimento do “hardware” do sistema, já que a dúvida central era a determinação dos parâmetros necessários para a classificação correta da qualidade da água. Juntos com os professores, foi-se estipulado o uso do cloro residual como o parâmetro principal. Prosseguindo com o raciocínio, o próximo passo seria a escolha do sensor correto para controlo de cloro de forma barata. Semanalmente houve reunião com a professora doutora Maria Clara Silveira com o intuito de verificar o andamento do relatório e o projeto.

### 4.2 Tabela de Casos de Uso

Como já referido no tópico anterior, usam-se casos de uso para o levantamento de requisitos. Um Caso de Uso “Use Case” é uma sequência de transações relacionadas que são realizadas por um ator do sistema num processo de diálogo. Mais concretamente, um

---

<sup>6</sup> Responsável pelo laboratório LABMIA (LABORATÓRIO DE MONITORIZAÇÃO E INVESTIGAÇÃO AMBIENTAL) O Laboratório de Monitorização e Investigação Ambiental é uma infraestrutura de apoio à Indústria e à Comunidade no domínio do controlo da qualidade química e microbiologia. No laboratório são efetuados trabalhos na área da química analítica e da microbiologia, nomeadamente o controlo químico e microbiológico de matérias-primas e produtos, análise de produtos ambientais, controlo da qualidade de águas, águas residuais e águas de processos, solos e análises foliares. O laboratório colabora com outras instituições nacionais e internacionais no desenvolvimento de projetos de investigação na área ambiental, executa planos de monitorização da qualidade de águas, solos, avaliação de impacto da descarga de efluentes industriais no meio receptor, acompanhamento técnico na exploração de ETA e ETAR e diagnósticos ambientais.

caso de uso é uma “porção” de funcionalidade [5]. Os casos de uso apresentados correspondem aos objetivos dos atores. A tabela de casos de uso representada a seguir, Tabela 2, apresenta todos os atores que fazem parte do sistema APPT bem como os objetivos de cada um.

*Tabela 2-Tabela de Atores e respectivos objetivos*

<b>Atores</b>	<b>Objetivos</b>
<b>Responsáveis pelo tratamento da água</b>	<ul style="list-style-type: none"> <li>– Receber alertar de falta d’água no purificador;</li> <li>– Receber alertar sobre estado do doseador (abertura ou fechado);</li> <li>– Consultar os dados dos sensores de PH, temperatura, nível da água;</li> <li>– Autorizar permissão para receber notificações.</li> </ul>
<b>Administrador</b>	<ul style="list-style-type: none"> <li>– Criar Conta;</li> <li>– Eliminar Conta;</li> <li>– Receber alertar de falta d’água no purificador;</li> <li>– Receber alertar sobre estado do doseador (abertura ou fechado);</li> <li>– Consultar os dados dos sensores de PH, temperatura, nível da água;</li> <li>– Autorizar permissão para receber notificações.</li> </ul>
<b>Sensores: ph, temperatura e nível da água</b>	<ul style="list-style-type: none"> <li>– Enviar e atualizar dados.</li> </ul>
<b>Firestore</b>	<ul style="list-style-type: none"> <li>– Receber e enviar dados do Microcontrolador em tempo real;</li> <li>– Hospedar a aplicação web;</li> </ul>
<b>Atuador</b>	<ul style="list-style-type: none"> <li>– Abrir e fechar o doseador do cloro;</li> <li>– Receber e enviar dados para o Microcontrolador.</li> </ul>

O papel dos atores:

- Responsáveis pelo tratamento da água: é o utilizador final do sistema que pode verificar o nível da água, consultar o estado do doseador no purificador caseiro, receber alerta do sistema caso falte água, receber valores do ph, temperatura e humidade;

- Administrador: é um utilizador final do sistema que pode criar e eliminar conta e herdar todos os benefícios que tem o responsável pelo tratamento da água;
- Sensores: Sensores de ph, HC-SR04 e DHT11 responsável por enviar e atualizar os dados ao Firebase;
- Microcontrolador: o ESP8266 que permite enviar e receber dados para o Firebase;
- Firebase: Fornece ferramentas e serviços de Autenticação, Realtime database, Hosting para ajudar no desenvolvimento e implantar da aplicação “web”;
- Atuador: Micro Servo (SG90) para abrir e fechar o doseador do cloro.

### 4.3 Diagrama de Casos de Uso

Depois de apresentado no tópico anterior os atores que fazem parte do sistema, bem como os objetivos e o papel de cada um, segue-se o diagrama de casos de uso. Um diagrama de casos de uso mostra os casos de uso, atores e as suas interações. Na Figura 11, encontra-se ilustrado num diagrama os casos de uso e os respetivos atores. Este serve para mostrar todas as funcionalidades que o sistema APPT vai ter e quem tem acesso a essas funcionalidades.

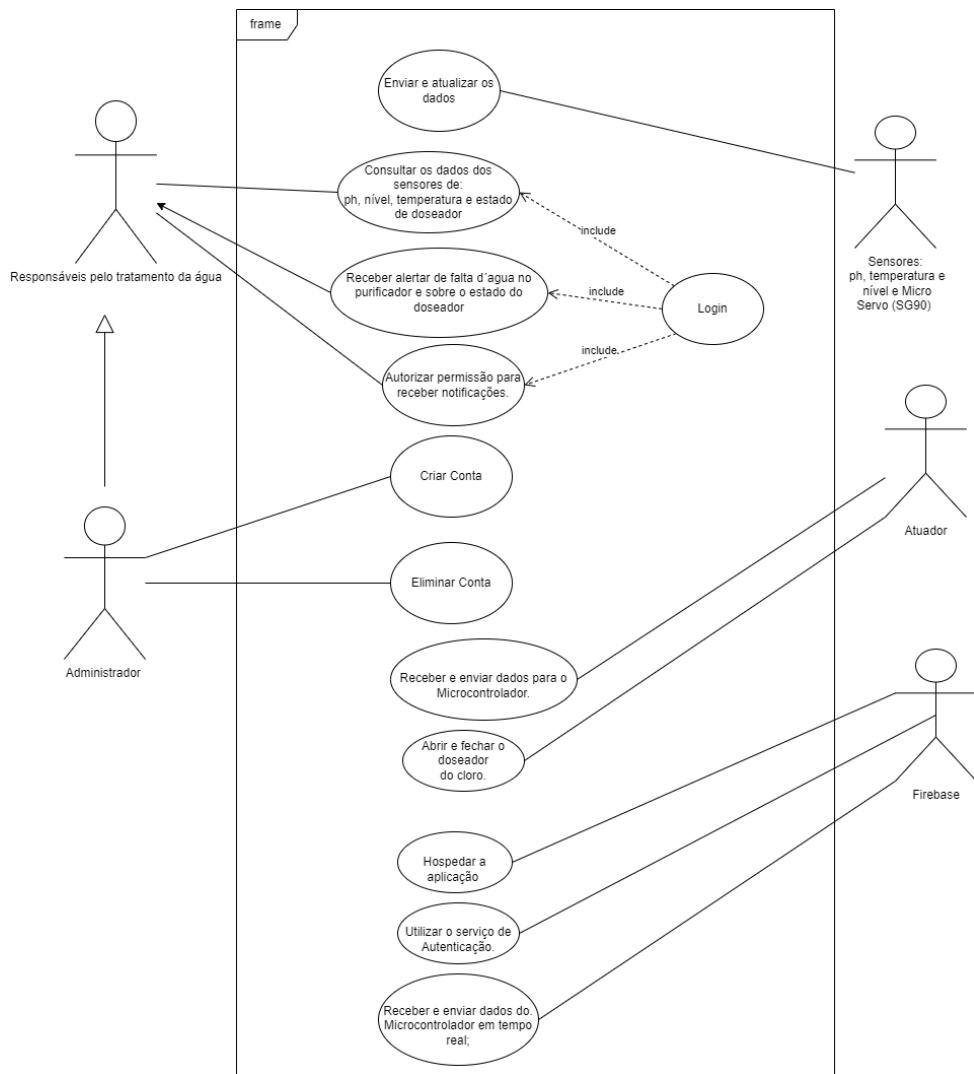


Figura 11-Diagrama de casos de uso do sistema APPT

Neste sistema, os requisitos não funcionais são listados da seguinte forma:

- Portabilidade - Sistema pode ser executado em diferentes sistemas operativos, dispositivos sem ter de modificar o código-fonte.
- Desempenho e Confiabilidade - Permitir que os dados sejam recolhidos em tempo real a partir de sensores conectados à *internet* e enviados para a aplicação web, permitindo ao utilizador visualizar e monitorizar os parâmetros da água em tempo real.
- Segurança - A integração com o IoT permite que os dados sejam armazenados em nuvem, o que garante segurança e acessibilidade dos dados a partir de qualquer lugar.
- Usabilidade - A aplicação permite que os utilizadores realizem a visualização e controlo dos dados de forma eficiente e satisfatória.

## 5 Implementação

Este capítulo descreve as tecnologias utilizadas, a arquitetura e o desenvolvimento do sistema APPT. O objetivo é desenvolver uma aplicação web que inclui uma *interface* para o utilizador final, onde é possível controlar o doseador de cloro, ler os dados de sensores de temperatura, ph e nível da água e enviar notificação, utilizando Firebase, ESP8266, HTML, CSS e JavaScript.

### 5.1 Tecnologias Utilizadas

Nesta secção será feita a apresentação das tecnologias utilizadas que contribuíram para o desenvolvimento da aplicação. As tecnologias usadas foram as seguintes:

- **Visual Studio Code** - É um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS;
- **Node.JS** - É um “software” de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador “web”;
- **HTML** - É uma linguagem de marcação utilizada na construção de páginas na Web;
- **CSS** - É um mecanismo para adicionar estilos a uma página web, aplicado diretamente nas tags HTML ou ficar contido dentro das tags <style>;
- **JavaScript** - É uma linguagem de programação de alto nível utilizada principalmente para criar páginas “Web” interativas e dinâmicas, mas também pode ser usado para programação do lado do servidor, aplicações de “desktop” e desenvolvimento de aplicações móveis;
- **Arduino IDE** - “Arduino Integrated Development Environment” é um “software” que fornece uma plataforma para programação e desenvolvimento de programas para placas Arduino, entre outras. O Arduino IDE oferece suporte às linguagens de programação C e C++ e projetado para ser fácil de usar tanto para iniciantes quanto para utilizadores experientes;
- **Firebase** - Uma plataforma de back-end como serviço (BaaS) desenvolvida pela Google. Fornece um conjunto de ferramentas e serviços para desenvolver e gerir aplicação móveis e da web. O Firebase fornece recursos como autenticação, Real-time Databases, Hosting e Cloud Functions, tornando-o uma solução completa para o para o desenvolvimento da app.

## 5.2 Sensores e Atuador

Em termos de sensores, microcontrolador e atuador para o desenvolvimento do sistema APPT, foram usados os seguintes.

### **D1 mini Pro ESP8266**

O D1 mini Pro ESP8266, demonstrado na Figura 12 é uma placa de desenvolvimento compacta e poderosa baseada no microcontrolador Wi-Fi ESP8266. O ESP8266 é um microcontrolador versátil e de baixo custo com recursos Wi-Fi integrados, tornando-o ideal para aplicações IoT.

A placa D1 mini Pro ESP8266 é equipada com uma variedade de recursos que facilitam o desenvolvimento e o protótipo de projetos de IoT, incluindo uma porta micro-USB para alimentação e programação, um botão de *reset* e uma variedade de entradas/saídas (I/O) pinos para conexão com outros dispositivos e sensores.

O D1 mini Pro ESP8266 é compatível com o Arduino IDE, para que os utilizadores possam escrever código em C++ e carregá-lo na placa para execução. Além disso, o D1 mini Pro ESP8266 é compatível com uma variedade de bibliotecas e estruturas, como a biblioteca ESP8266WiFi e NodeMCU, que facilitam a interação com a placa e o desenvolvimento de aplicações IoT.



*Figura 12-Microcontrolador Wemos D1 mini Pro ESP8266 [9]*

Possui uma série de pinos disponíveis para o utilizador, que podem ser utilizados para conectar sensores, atuadores, ecrãs, entre outros dispositivos. Na Figura 13 está ilustrado uma lista dos pinos disponíveis no D1 Mini e as suas funções:

- GPIO1: Pino utilizado para comunicação serial (TX).
- GPIO2: Pino utilizado para comunicação serial (RX).
- GPIO3: Pino que pode ser utilizado para controle de um LED.
- GPIO4: Pino que pode ser utilizado para controle de um LED.
- GPIO5: Pino que pode ser utilizado para controle de um LED.



- GPIO9: Pino que pode ser utilizado para comunicação serial (TX).
- GPIO10: Pino que pode ser utilizado para comunicação serial (RX).
- GPIO12, GPIO13, GPIO14: Pinos de controle.
- GPIO15: Pino que deve ser mantido na “Terra” (GND) durante a inicialização para que o módulo ESP8266 possa ser inicializado corretamente.
- A0: Pino utilizado para leitura de valores analógicos.
- SDA: Pino utilizado para comunicação I2C.
- SCL: Pino utilizado para comunicação I2C.

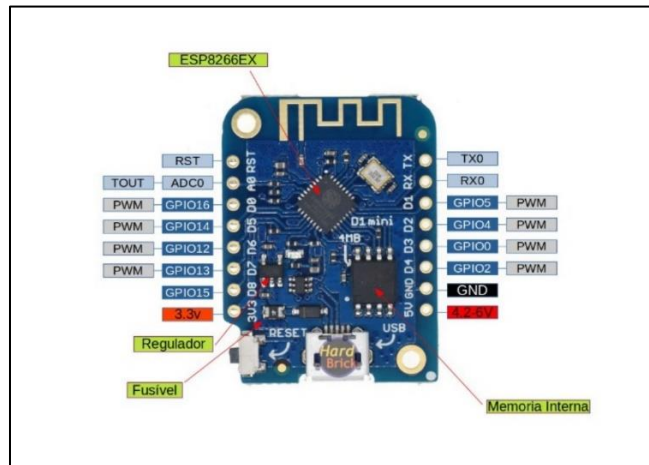


Figura 13-Imagem é do Wemos D1 mini Pro, a versão 3.0.0 [10]

## DHT11

É um sensor de temperatura e humidade de baixo custo, usado em projetos DIY IoT para medir a temperatura ambiente e a humidade numa sala ou ambiente. O sensor emite um sinal digital que pode ser lido por um microcontrolador. O sensor DHT11 é composto por um termistor para medir a temperatura e um sensor capacitivo para medir a humidade relativa do ar. Consegue medir temperatura na faixa de 0 a 50 graus Celsius com uma precisão de +/- 2 graus Celsius e humidade relativa do ar na faixa de 20 a 80% com uma precisão de +/- 5%.

O sensor, demonstrado na Figura 14, possui apenas três pinos de conexão: VCC, GND e DATA. O pino DATA é utilizado para a comunicação do sensor com o microcontrolador e deve ser conectado a uma porta digital do microcontrolador é preciso salientar que o barramento de comunicação é *single-wire*.

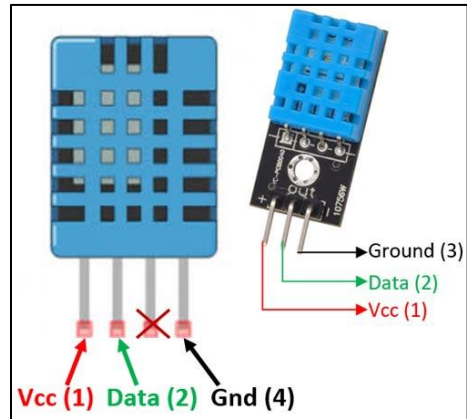


Figura 14-Sensores DHT11 e os pinos de ligação [11]

### HC-SR04

O HC-SR04, descrito na Figura 16, é um módulo sensor ultrassônico bastante utilizado em projetos de automação e robótica. Capaz de medir a distância entre o sensor e um objeto utilizando ondas sonoras.

O módulo HC-SR04 possui quatro pinos de conexão:

- VCC: Pino de alimentação do módulo. Deve ser conectado a uma fonte de 5V.
- GND: Pino de referência de terra do módulo. Deve ser conectado a um terminal de terra do circuito.
- Trig: Pino de envio do sinal ultrassônico. Este pino deve ser conectado a uma porta digital do microcontrolador.
- Echo: Pino de recepção do sinal ultrassônico. Este pino também deve ser conectado a uma porta digital do microcontrolador.

O funcionamento do módulo é relativamente simples. Para medir a distância entre o sensor e um objeto, o microcontrolador envia um pulso de sinal de 10 microssegundos pelo pino Trig. Este pulso dispara um sinal ultrassônico de frequência de 40 kHz pelo transmissor do módulo. O sinal ultrassônico é refletido pelo objeto mais próximo e retorna para o módulo, sendo recebido pelo pino Echo, como é referenciado na Figura 15. O microcontrolador então mede o tempo de ida e volta do sinal e calcula a distância utilizando a fórmula:

$$\text{Distância} = (\text{Tempo de ida e volta do sinal} \times \text{Velocidade do som}) / 2;$$

A velocidade do som é de aproximadamente 343 metros por segundo em temperatura ambiente. O valor calculado de distância é em centímetros.

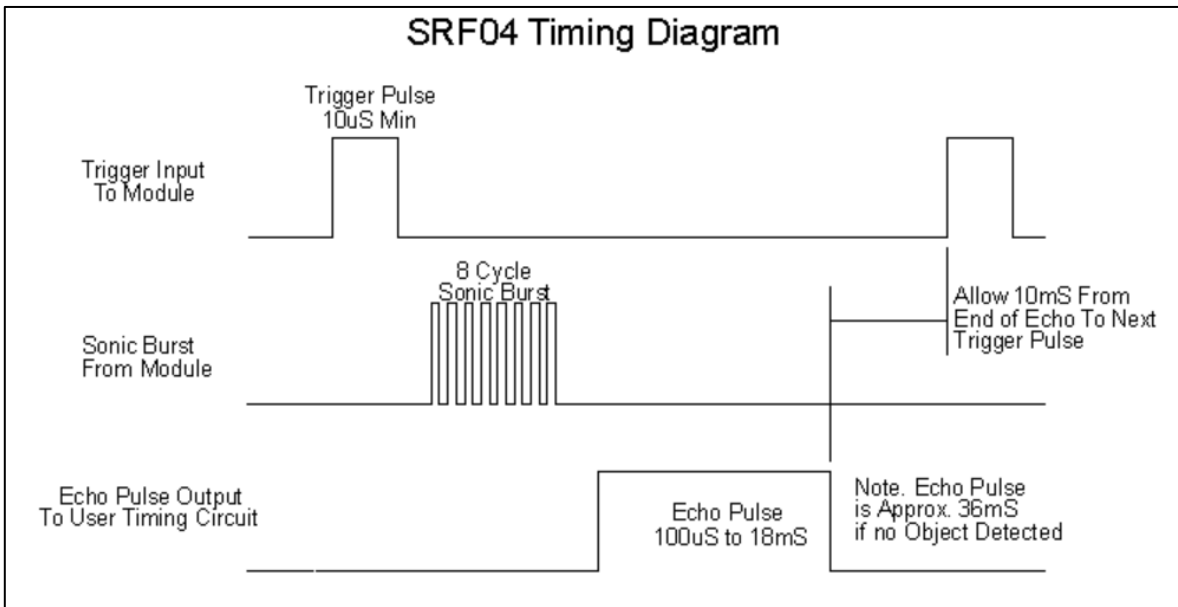


Figura 15-Diagrama de tempo do SRF04 [12]

O HC-SR04 é um módulo sensor bastante utilizado em projetos de robótica e automação, podendo ser utilizado para medir distâncias, evitar obstáculos, detectar a presença de objetos, entre outras aplicações. É importante lembrar que o módulo pode ser afetado por ruídos externos, como outras fontes de ultrassom, e deve ser utilizado com cuidado em ambientes com muitos obstáculos ou superfícies reflexivas.

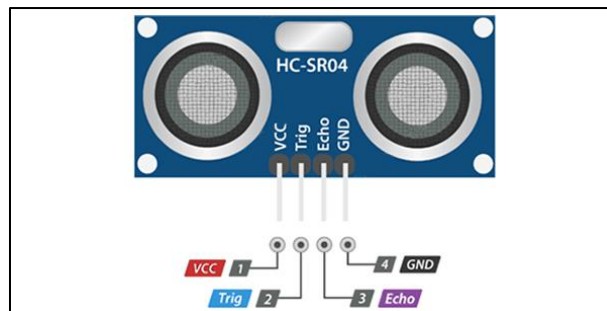


Figura 16- Sensor HC-SR04 [12]

### Sensor de PH (E-201C-Blue)

É um sensor de pH bastante utilizado em projetos que envolvem a medição do pH de soluções líquidas. Composto por um eletrodo de vidro sensível ao pH e um circuito amplificador para a leitura do sinal.

O sensor E-201C-Blue ilustrado na Figura 17, possui dois pinos de conexão:

- VCC: Pino de alimentação do sensor. Deve ser conectado a uma fonte de alimentação de 5 a 12V.

- GND: Pino de referência de terra do sensor. Deve ser conectado a um terminal de terra do circuito.
- O sensor também possui um cabo de conexão com um conector BNC para a conexão com um medidor de pH ou um microcontrolador com entrada analógica. É importante lembrar que o sensor deve ser calibrado antes da sua utilização e que o pH das soluções líquidas pode ser afetado por diversos fatores, como temperatura e presença de outros compostos.

O utilizador deve utilizar o código desenvolvido pelo fabricante [7] e após isso iniciar o processo de calibração do eletrodo. Para iniciar o processo de calibração é necessário preparar três soluções, uma neutra (cujo pH é 7), uma ácida (cujo pH é 4) e uma básica (cujo pH é 9,18). Ao inserir o eletrodo na solução neutra e executar o código do fabricante, será exibido no monitor serial o valor do pH com uma margem de erro embutida. Para corrigir essa margem, o utilizador deve definir (no campo 'offset') o valor exato da diferença entre o pH real da solução e o pH exibido pelo monitor serial do Arduino IDE.

Após esse teste o utilizador deverá limpar o eletrodo e inseri-lo na solução ácida, a partir do momento em que o pH exibido no monitor serial seja semelhante ao da solução, o eletrodo ficará aferido para a leitura de soluções ácidas. Para realizar a calibragem do eletrodo em soluções básicas, basta repetir o processo anterior de calibragem, porém deverá limpar o eletrodo e utilizar a solução básica. A frequência de calibragem, recomendada pelo fabricante, deve ser de 6 meses, porém caso a qualidade da água no local onde está o eletrodo seja má, é necessário que o mesmo seja aferido com uma frequência maior.

Com o circuito amplificador, é possível ler o sinal do sensor de pH através de uma entrada analógica do microcontrolador e processar a informação para exibir o valor de pH num ecrã, enviar informações para uma base de dados ou controlar um sistema de dosagem de reagentes para ajuste do pH da solução líquida.



*Figura 17-Sensor de PH [13]*

## Micro Servo SG90

O SG90, ilustrado na Figura 18 é um servo motor pequeno, leve e económico usado em vários projetos robóticos. É um micro servo de 9g que opera com fonte de alimentação de 4,8V a 6V DC e pode girar aproximadamente 180 graus. O Micro Servo SG90 tem três pinos:

- Pino que conecta ao microcontrolador para controlar a posição do servo. A posição do servo é determinada pelo *duty cycle*<sup>7</sup> do sinal de frequência enviado pelo microcontrolador.
- VCC: Conecta a uma fonte de energia para fornecer energia ao servo
- GND: Pino que deve ser conectado a um terminal de terra do circuito.



Figura 18-Servomotor FS90R POLOLU [14]

Por fim, foi preciso de um cabo Micro-USB para a alimentação e cabos Jumper, também conhecidos como cabos auxiliares usados para transferir energia elétrica. Para a seleção dos componentes, já mencionados, foram utilizados os seguintes parâmetros: especificações, a disponibilidade de documentação e o preço. No total foram utilizados 5 componentes, de forma que cada componente fosse compatível com o microcontrolador, e que o protótipo também pudesse obter, salvar e enviar os dados. Os sensores foram escolhidos por motivos económicos, de teste e de praticidade, no (Anexo A) foi feita uma comparação de alguns sensores utilizados e os mais adequados para o projeto.

---

<sup>7</sup> O duty cycle (ciclo de trabalho) é uma medida utilizada para descrever a relação entre o tempo em que um sinal está em nível alto (ou ligado) e o tempo em que está em nível baixo (ou desligado) durante um período. No caso do servo motor, o microcontrolador envia um sinal de frequência para controlar a posição do servo. O duty cycle desse sinal indica quanto tempo o sinal está em nível alto (ou ligado) relativamente ao tempo total do ciclo. Em outras palavras, o duty cycle é uma medida da largura do pulso do sinal enviado pelo microcontrolador. O servo motor utiliza essa informação para determinar a posição em que deve se posicionar.

A Figura 19, mostra como foi feita a ligação dos sensores e o Servo motor como o D1 mini Pro ESP8266.

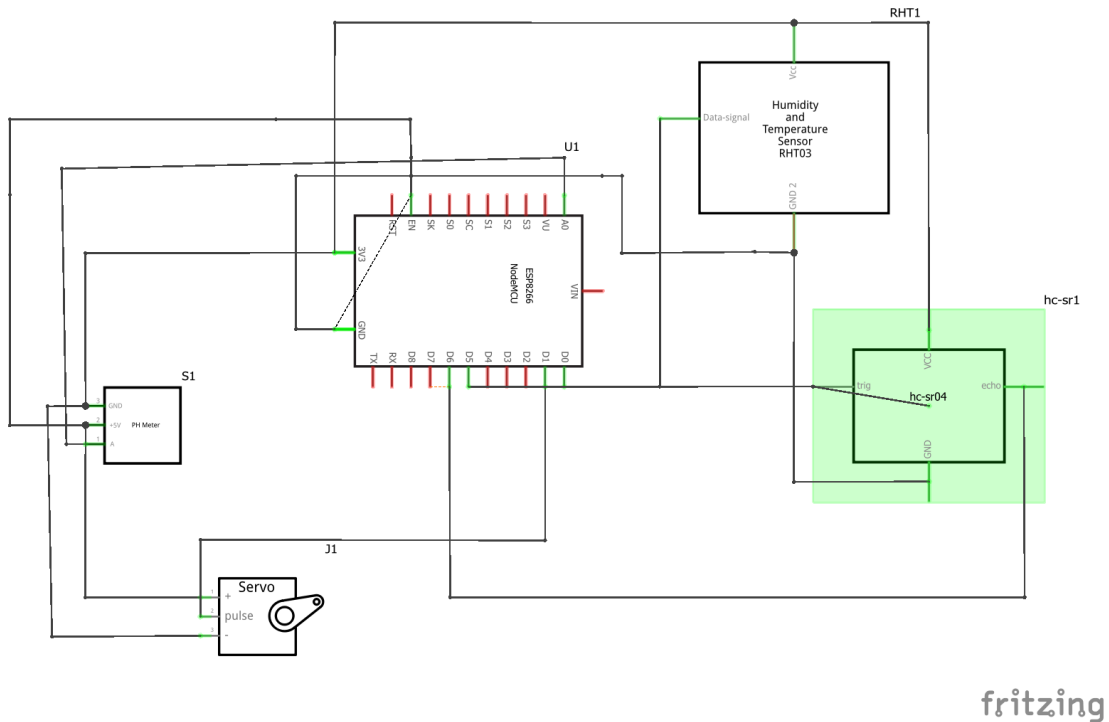


Figura 19-Ligações dos Sensores com o Esp8266

A Ligação do D1 mini Pro ESP8266 com os sensores decorreu da seguinte forma:

- Sensor de pH (Grove) conectou-se com o pino Analógica A0;
- Sensor HC-SR04 conectou-se com os pinos Digital D5 e D6, ou seja, o echoPin ao pino D5 e TriggerPin ao pino D6;
- Sensor DHT11 conectou-se o pino Digital D0;
- O Micro Servo SG90 conectou-se o pino Digital D1;

Com essa arquitetura, a solução permite a obtenção dos seguintes parâmetros físico-químicos: nível, temperatura e pH da água em tempo real. Também é possível salvar os dados obtidos na base de dados.

### 5.3 Arquitetura do Sistema

De início, é importante explicar o funcionamento geral do sistema. A solução consiste num protótipo usando:

- Um sensor de medição de pH (Grove);

- Uma placa de prototipagem D1 mini Pro Lolin<sup>8</sup> ESP8266;
- Um sensor HC-SR04 para aferir um medir do nível da água;
- Um sensor DHT11 para aferir um sensor de temperatura da água;
- Um Micro Servo SG90 responsável por abrir e fechar o doseador de cloro.

Serviços disponibilizados pelo Firebase:

- Real-time Databases: Para obter os valores dos sensores em tempo real;
- Hosting: Para hospedar a aplicação “web”;
- Authentication: Para utilizar o serviço de Autenticação.

E também HTML, CSS e JavaScript para modelar a “Interface web” do Sistema APPT.

Os sensores são conectados por meio dos pinos de entrada/saída (GPIO) do D1 mini Pro ESP8266, que recebe os dados dos sensores e os transmite para o *Firebase*. Os dados são armazenados na “Firebase Real-time Databases” e então exibidos na interface web criada usando HTML, CSS e JavaScript. A arquitetura do sistema APPT encontra-se ilustrada na Figura 20.

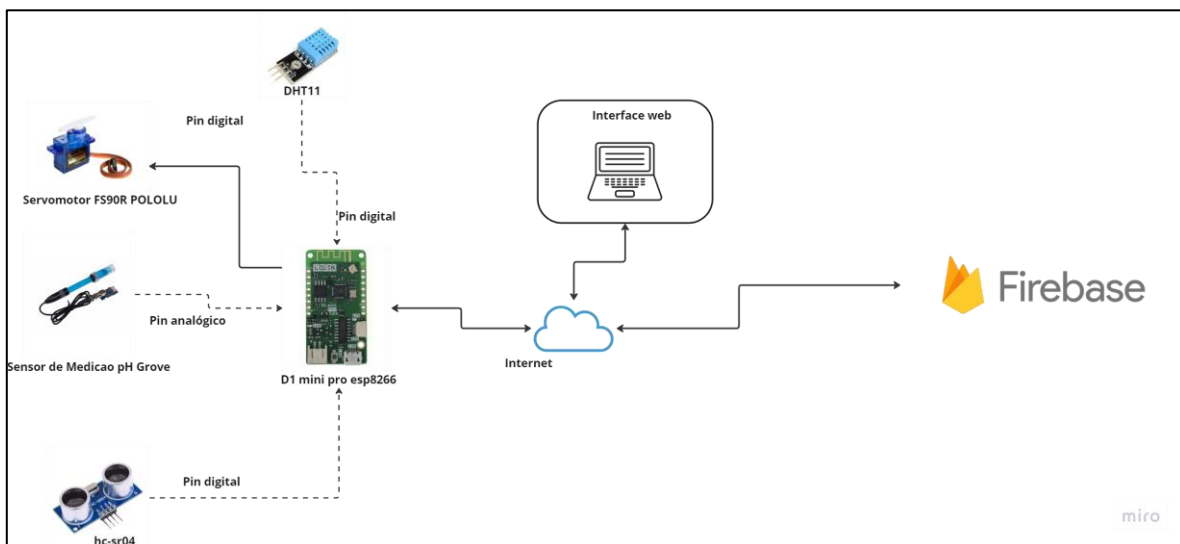


Figura 20- Arquitetura do Sistema APPT

O sistema APPT oferece uma maneira simples de monitorizar e manter os parâmetros de qualidade da água num purificador de água ou outro sistema de armazenamento de água. O diagrama representado na Figura 21 mostra uma visão geral de alto nível do projeto construído.

<sup>8</sup> Lolin é uma marca de moda portuguesa especializada na criação de vestuário feminino, incluindo vestidos, blusas, saias e calças. A marca é conhecida por seus designs exclusivos, tecidos de alta qualidade e atenção aos detalhes. As coleções da Lolin geralmente apresentam cores e padrões ousados, e as peças da marca são projetadas para serem confortáveis e elegantes. A Lolin tem várias lojas físicas em Portugal, bem como uma loja online onde os clientes podem navegar e adquirir as suas coleções.

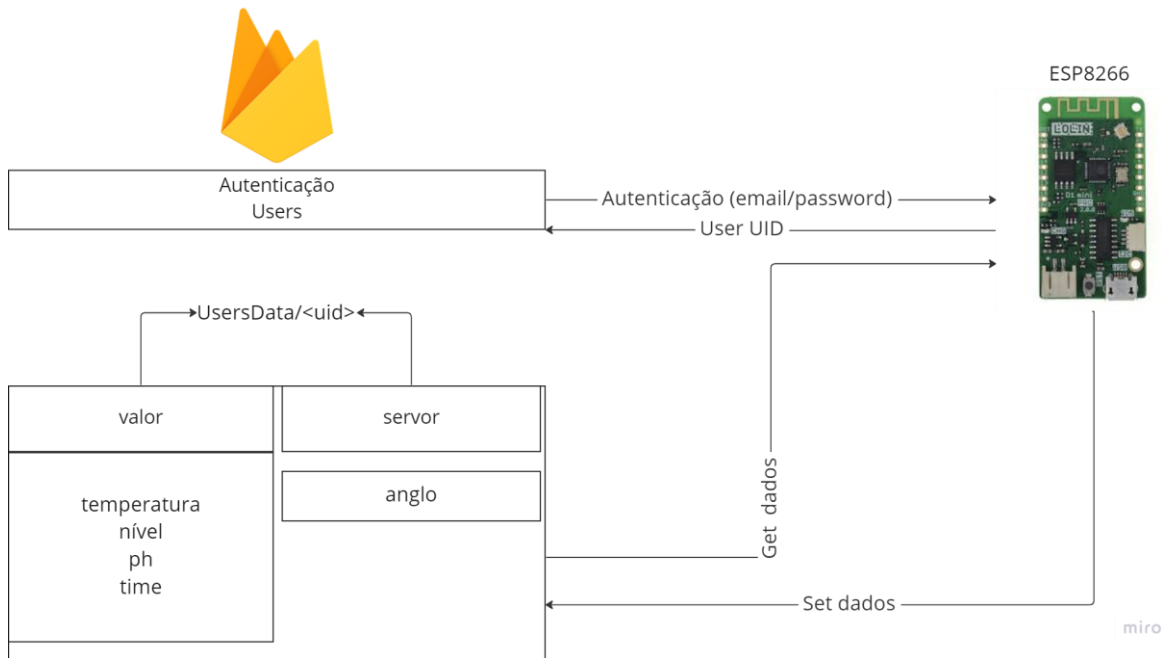


Figura 21-Diagrama Geral de Alto Nível

O ESP8266 autentica como utilizador com e-mail e *password* (este utilizador deve ser configurado nos métodos de autenticação de “Firebase”). Após a autenticação, o ESP8266 obtém o UID do utilizador. A base de dados é protegida com regras de segurança. O utilizador só pode conectar os nós da base de dados sob o nó com o seu UID de utilizador. Após obter o UID, o ESP8266 pode enviar os dados na base de dados. O ESP8266 obtém os valores dos sensores da temperatura, pH do nível da água e o estado do servo motor e obtém a hora certa após obter a leitura dos sensores.

O ESP8266 envia a temperatura, pH do nível da água para a base de dados. Novas leituras são adicionadas a base de dados periodicamente, definido por um tempo de atraso para 2 minutos e 5 segundos (15.000 milissegundos). Por fim, terá um registo de todas as leituras na base de dados em tempo real do “Firebase”.

#### 5.4 Conexão ao Firebase

Neste subcapítulo irei inserir os valores dos sensores utilizando D1 mini Pro ESP8266 no “Firebase Realtime Database” para ter um registo do histórico dos dados. Em seguida, aceder aos mesmos usando a aplicação da “Web” para exibir os resultados.

Para registar dados, demonstrado na Figura 26, é preciso seguir as etapas:

- a) **Configurar a placa D1 mini Pro ESP8266 e conectá-la à internet.**



É preciso ter o D1 mini Pro ESP8266 core instalado na IDE do “Arduino” e conectar a placa à rede WiFi. Para isso, inicializa-se o “Firebase Realtime Database” no código “Arduino” e inclui a biblioteca “Firebase Arduino”. Fornece as credenciais do projeto criado na plataforma do “Firebase”, mais especificamente o URL da base de *dados* (*DATABASE\_URL*), uma chave da conta de serviço (*API\_KEY*), o *email*(*USER\_EMAIL*) e *password* (*USER\_PASSWORD*) criada no serviço de autenticação do Firebase. Também necessita das credenciais do WiFi e instalar as bibliotecas do Servo Motor e o sensor DHT11.

#### **b) Criar funções para ler os dados dos sensores.**

##### **Função `setdoseador()`**

A função “`setdoseador()`”, descrita na Figura 22, permite controlar o servo motor e assim conseguir abrir e fechar o doseador, que lê um valor na “Firebase Realtime Database” e define a posição do servo motor com base nesse valor. A função usa o método `Firebase.RTDB.getInt()` para ler o valor do “Firebase Realtime Database” no nó “/doseador/Anglo”. Se o valor for lido com sucesso, é armazenado na variável “estado” que representa a posição do servo motor. Então o objeto “myservo” define a posição do servo motor para o valor do estado. Se houver um erro ao ler o valor do “Firebase Realtime Database”, a função imprime uma mensagem de erro para o monitor serial.

```
void setdoseador() {  
  
    if (Firebase.RTDB.getInt(&fbdo, F("/doseador/Anglo")) == 1)  
    {  
        delay(300);  
        estado = fbdo.to<int>();  
        Serial.print("recebido: ");  
        Serial.println(estado);  
    }  
    else  
    {  
        Serial.println("erro na leitura do estado");  
    }  
    myservo.write(estado);  
}
```

Figura 22 Função para Controlar o Estado do Doseador

## Função getTem()

A função “getTem()”, descrita na Figura 23, lê os dados de temperatura do sensor DHT11 e calcula a temperatura em Celsius e retorna o valor como “float”. O código inclui primeiro a biblioteca “DHT”, que fornece métodos para leitura de dados de temperatura e humidade de sensor. A função introduz um atraso de 2 segundos para garantir que o sensor tenha tempo suficiente para fazer medições precisas. O código lê os dados de humidade do sensor usando o método “dht.readHumidity()” e armazena o resultado numa variável “h”. Em seguida, lê os dados de temperatura do sensor em Fahrenheit e Celsius usando o método “dht.readTemperature()” e armazena os resultados em “f” e “t” respetivamente. A função então calcula o índice de calor em Fahrenheit e Celsius usando o método “dht.computeHeatIndex()” e guarda os resultados em “hif” e “hic” respetivamente. Por fim, como só é preciso do valor da temperatura em Celsius, a função retorna a temperatura em Celsius, e o guarda na variável “t”.

```
float getTem() {  
  
    // Wait a few seconds between measurements.  
    delay(2000);  
  
    float h = dht.readHumidity();  
    // Read temperature as Celsius (the default)  
    float t = dht.readTemperature();  
    // Read temperature as Fahrenheit (isFahrenheit = true)  
    float f = dht.readTemperature(true);  
  
    // Compute heat index in Fahrenheit (the default)  
    float hif = dht.computeHeatIndex(f, h);  
    // Compute heat index in Celsius (isFahreheit = false)  
    float hic = dht.computeHeatIndex(t, h, false);  
  
    return t;  
}
```

Figura 23-Função para Calcular a Temperatura

## Função “getph()”

A função “getph()”, descrita na Figura 24, mede o nível de pH da água, usando o sensor de pH. A função usa a variável “samplingTime” para acompanhar a última vez que o pH foi mostrado na variável “printTime” e saber a última vez que o valor de pH foi impresso. Também usa a variável “pHValuepara” para armazenar o valor de pH calculado na variável

“voltage” e armazenar a leitura de tensão do sensor de pH. A função verifica se já passou tempo suficiente desde a última recolha do valor de pH, com base na variável “samplingInterval”. Se tiver passado tempo suficiente, pega uma nova amostra de pH usando o método “analogRead()” que a armazena na matriz “pHArray”. O método avergearray() é então usado para calcular os valores na “pHArraymatriz”, e esse valor é usado para calcular a tensão. Finalmente, o valor de pH é calculado com base numa fórmula linear, usando as variáveis “voltage” e “Offset” sendo armazenado na variável “pHValue”. Se passou tempo o suficiente desde que o último valor de pH foi impresso, com base na “printIntervalvariável”, a função retorna o valor atual na “pHValuevariável”.

```
float getph(void)
{
    static unsigned long samplingTime = millis();
    static unsigned long printTime = millis();
    static float pHValue, voltage;

    if (millis() - samplingTime > samplingInterval)
    {
        pHArray[pHArrayIndex++] = analogRead(SensorPin);
        if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;
        voltage = avergearray(pHArray, ArrayLenth) * 5.0 / 1024;
        pHValue = -19.18518519 * voltage + Offset;
        samplingTime = millis();
    }
    if ((millis() - printTime > printInterval))
    {
        return pHValue;
    }
}
```

Figura 24-Função para Calcular o ph

### Função getDistance()

A função “getDistance()”,descrita na Figura 25, usa sensor ultrassónico para medir a distâncias da água no purificador. A função primeiro define a saída “trigPin” LOW para garantir um sinal. Em seguida, define a saída “trigPin” HIGH em 10 microssegundos acionando o sensor ultrassónico para enviar uma onda sonora. Depois disso, define a saída “trigPin” de volta para LOW. A duração da viagem de ida e volta da onda sonora é medida usando a função “pulseIn()”, que mede o tempo que leva para um sinal ir de LOW para HIGH no “echoPin”. Esta duração é armazenada na variável “duration”. A distância até o objeto é calculada multiplicando a duração pela velocidade do som no ar (sendo aproximadamente

0,034 cm por microssegundo) e dividindo o resultado por 2 (já que a onda sonora percorre o dobro da distância até o objeto - primeiro até o objeto e após volta ao sensor). O valor resultante é armazenado na variável "distance". Finalmente, a função retorna a distância calculada para o código de chamada.

```
int getDistance() {  
  
    // Clears the trigPin condition  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    // Sets the trigPin HIGH (ACTIVE) for 10 microseconds  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
    duration = pulseIn(echoPin, HIGH);  
    // Calculating the distance  
    distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)  
  
    return distance;  
}
```

Figura 25-Função para Calcular a Distância

### c) Enviar os dados ao Firebase

Para isso, é preciso criar um *loop* que regista os dados do sensor no Firebase num intervalo regular de 15 segundos. É preciso usar o método "Firebase.set" ou "Firebase.push" para guardar os dados na base de dados. Todas as funções são chamadas num intervalo regular para atualizar a posição do servo motor com base no valor do "Firebase Realtime Database" e os valores da temperatura, nível e ph da água.

Como é possível observar na Figura 26, os dados são salvos num nó com o próprio UID do utilizador, para restringir o acesso a base de dados.



Figura 26-Imagem dos dados recebidos no Firebase

## 5.5 Aplicação Web

A aplicação consiste em uma interface web, onde é possível controlar o doseador e ler os dados dos sensores. Ao utilizar o HTML e o CSS, foi possível criar a estrutura e o estilo da “interface” do utilizador da aplicação web APPT e o JavaScript para adicionar interatividade além de comunicar com o Firebase para obter e enviar dados. O Firebase oferece um SDK (kit de desenvolvimento de software) para JavaScript que torna a integração com o Firebase fácil e rápida. Portanto, com essas tecnologias, é possível criar uma aplicação web completa e funcional, utilizando o Firebase como back-end para armazenar e gestão de dados.

Aqui está uma breve visão geral de cada parte:

### a) HTML Templates

O modelo HTML define a estrutura da Interface Web, incluindo o cabeçalho, e as seções de conteúdo principal. Também inclui espaços reservados para dados de medidor e tabela, que são atualizados dinamicamente usando JavaScript. O modelo da Interface inicial (index.html) inclui três medidores de temperatura, ph e distância, bem como um “slider” para alternar o doseador no ESP8266 e uma tabela para exibir todos os dados do sensor com um botão para eliminar os mesmos na base de dados.

### b) JavaScript Code

O código JavaScript lida com a atualização dinâmica dos dados em cada página. Usa-se o Firebase JavaScript SDK para recuperar os dados do sensor do Firebase Realtime Database e atualizar os medidores e tabela.

## Função setupUI()

Este pedaço de código descrito na Figura 27, configura a interface do utilizador para a Aplicação web e permite-o fazer “login” e aceder os seus dados armazenados no “Firebase Realtime Database”. A função “setupUI()”, recebe o objeto “user” como parâmetro, que representa o utilizador que faz o “login” no momento. Se um utilizador estiver conectado, a função define vários elementos da interface do utilizador a serem exibidos, incluindo o elemento de conteúdo, a barra de autenticação e o elemento de detalhes do utilizador, que mostra o endereço de e-mail.

```
const setupUI = (user) => {
  if (user) {
    //toggle UI elements
    loginElement.style.display = 'none';
    contentElement.style.display = 'block';
    authBarElement.style. = 'block';
    userDetailsElement.style.display = 'block';
    userDetailsElement.innerHTML = user.email;

    // get user UID to get data from database
    var uid = user.uid;

    // Database paths (with user UID)
    var dbPathDose = 'UsersData/' + uid.toString() + 'servo/angle';
    var dbPath = 'UsersData/' + uid.toString() + '/valor';

    var firebaseRef = firebase.database().ref(dbPath);
```

*Figura 27-Função para gerir LOGIN/LOGOUT UI*

A função obtém o “uid” do utilizador conectado e usa o para construir os nós para os dados no Firebase Realtime Database ao qual o utilizador tem acesso. E também, para aceder aos dados relacionados ao servo motor (dbPathDose) e um valor (dbPath) para os dados dos sensores. Por fim, a função cria um objeto de referência do Firebase Realtime Database para o nós “dbPath” chamando “firebase.database().ref(dbPath)”.

```

firebaseRef.on('value', function (snapshot, prevChildKey) {
  const newPost = snapshot.val();

  presElement.innerHTML = newPost.ph;
  tempElement.innerHTML = newPost.temperatura;
  nivelElement.innerHTML = newPost.nivel;

  /**gauge */

  var gaugeI = createNivelGauge();
  gaugeI.value = newPost.nivel;

  var gaugeP = createTemperatureGauge();
  gaugeP.value = newPost.temperatura;

  /** chart*/

  var chartI = createTemperatureChart();
  var timestamp = (new Date()).getTime();
  console.log("timestamp: " + timestamp);
  plotValues(chartI, timestamp, newPost.temperatura);

  var chartH = createPhChart();
  plotValues(chartH, timestamp, newPost.ph);

  var chartN = createNivelChart();
  plotValues(chartP, timestamp, newPost.nivel);

  setdoseador(dbPathDose);
}

```

Figura 28-Código para Gerir a Atualização dos Dados na Interface web

O método “firebaseRef.on()”, descrita na Figura 28, configura os valores quando há uma alteração nos dados dentro da função “setupUI()”. O método é chamado sempre que uma alteração é detetada, e recebe “snapshot” como primeiro argumento, que contém os dados atualizados. O método “snapshot.val()” é usado para extrair os valores dos objeto ph, temperatura e nivel. Esses valores são usados para atualizar os dados da Interface Web, no medidor e na tabela, de temperatura, ph e o nível da água. As funções “createNivelGauge()” e “createTemperatureGauge()” criam objetos de medição, sendo então atualizados com os valores de ph e temperatura e o nível usando as propriedades gaugeN.value , gaugeP.value e gaugeT.value.

Por fim, a função “setdoseador()” é chamada com um caminho da base de dados do Firebase como argumento, que presumivelmente atualiza a posição do servo motor com base num valor armazenado no Firebase Realtime Database.

### Função “createTable()”

A função “createTable()”, descrita na Figura 29, é chamada para criar a tabela e usa “dbRef” para detetar alterações no “Firebase Realtime Database” e atualiza a tabela com os dados mais recentes.

A função recupera os dados mais recentes da base de dados utilizando “snapshot.val()”, e extrai os valores de temperatura, pH, nível e *timestamp* então cria uma linha para a tabela HTML com o *content* de data/hora, temperatura, pH e do nível da água. usa o método “prepend()” para adicionar a nova linha ao topo da tabela, segurando que os dados mais recentes sejam sempre exibidos primeiro. O código também salva a data/hora da primeira

leitura nos dados em “lastReadingTimestamp, usado para garantir que apenas novos dados sejam adicionados à tabela.

```
function createTable() {
  // append all data to the table
  var firstRun = true;
  dbRef.on('value', function (snapshot, prevChildKey) {
    const newData = snapshot.val();
    console.log(newData);
    var temperature = newData.Temperatura;
    var ph = newData.ph;
    var nivel = newData.nivel;
    var timestamp = newData.time;
    //var timestamp = (new Date()).getTime();
    console.log(timestamp);

    var content = '';
    content += '<tr>';
    content += '<td>' + timestamp + '</td>';
    content += '<td>' + temperature + '</td>';
    content += '<td>' + valueph + '</td>';
    content += '<td>' + nivel + '</td>';
    content += '</tr>';
    $('#tbody').prepend(content);
    if (firstRun) {
      lastReadingTimestamp = timestamp;
      firstRun = false;
      //console.log(lastReadingTimestamp);
    }
  });
}
```

Figura 29-Função para criar a Tabela

A implementação da aplicação foi concluída. Usando o “Firebase hosting” foi possível hospedar, a aplicação em uma CDN global. Ao aceder a aplicação web pela primeira vez, verá um formulário para inserir a *password* e o e-mail do utilizador como demonstrado na Figura 30.



**Login**

✉ Enter your email

🔒 Enter Password

Login

*Figura 30-Tela de login da aplicação web*

A aplicação web é responsiva, possibilitando acedê-lo usando o telefone, computador ou tablet. Deve-se inserir o e-mail e a password do utilizador autorizado adicionado pelos métodos *Firebase Authentication*. Depois disso, pode-se aceder a Interface web com as leituras. A Figura 31 mostra as leituras dos dados dos sensores a serem exibidos em cartões, medidores e uma tabela.

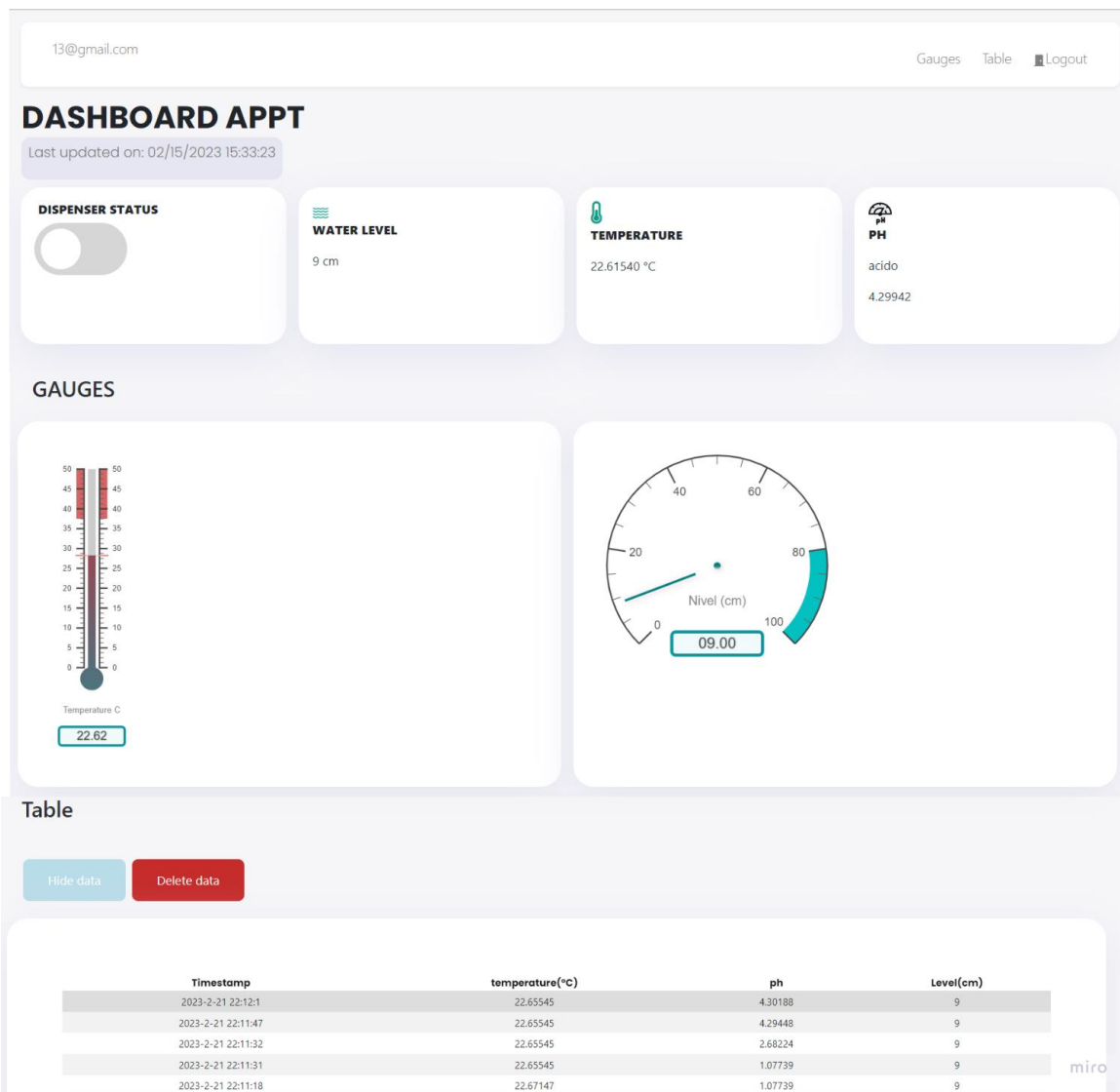


Figura 31-Imagem da Interface web

Como é visível na Figura 31, a Interface demonstra o horário e a data da última alteração ou utilizador, no processo de desenvolvimento é descrito no (Anexo B).

```
function setdoseador(dbPathDose) {
  if (dosaElement.checked) {
    console.log("ON");
    firebase.database().ref(dbPathDose).set(180);
  }
  else {
    console.log("OFF");
    firebase.database().ref(dbPathDose).set(0);
  }
}
```

Figura 32-Função para controlar o doseador na Interface Web

A função `setdoseador`, descrita na Figura 32, é acionada quando ocorre um evento de clique no elemento HTML com id "dosaElement". O objetivo da função é definir um valor para um nó específico na base de dados. A função começa a registrar e verifica se o elemento HTML "dosaElement" está marcado ou não, usando a propriedade `checked`. Se o elemento estiver marcado o valor 180 será definido para o nó especificado na base de dados Firebase. Caso contrário, o valor 0 será definido para o mesmo nó na base de dados Firebase, como se pode ver na Figura 33.

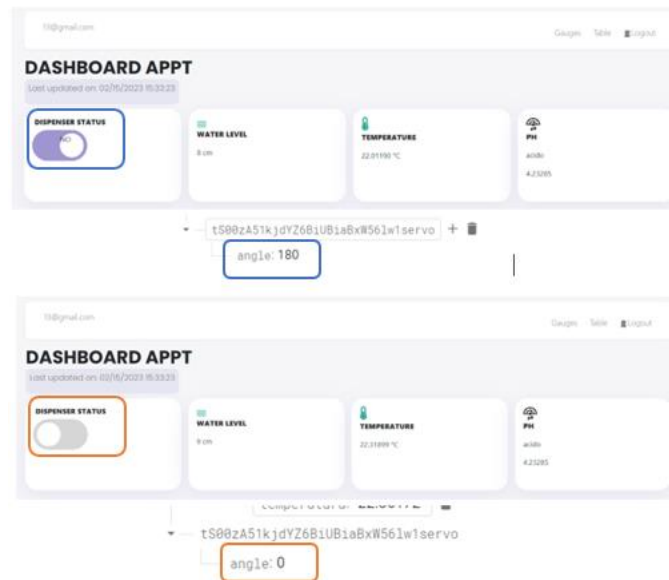


Figura 33-Imagem do doseador do cloro ligado e desligado

A Figura 34 mostra a tabela criada para recuperar todos os dados do sensor do "Firebase Realtime Database".

Table

Hide data Delete data

Timestamp	temperature(°C)	ph	Level(cm)
2023-2-21 22:12:1	22.65545	4.30188	9
2023-2-21 22:11:47	22.65545	4.29448	9
2023-2-21 22:11:32	22.65545	2.68224	9
2023-2-21 22:11:31	22.65545	1.07739	9
2023-2-21 22:11:18	22.67147	1.07739	9

Timestamp	temperature(°C)	ph	Level(cm)
2023-2-21 22:5:51	22.31899	4.23285	10
2023-2-21 22:5:38	22.23621	4.23285	10
2023-2-21 22:5:37	22.23621	4.23285	9
2023-2-21 22:5:23	22.19349	4.23285	9
2023-2-21 22:5:22	22.19349	4.23285	55
2023-2-21 22:5:11	22.12406	4.23285	55
2023-2-21 22:5:8	22.12406	4.23285	11
2023-2-21 22:4:53	22.04128	4.23285	11
2023-2-21 22:4:51	22.04128	4.23285	9
2023-2-21 22:4:39	22.02525	4.23285	9
2023-2-21 22:4:37	22.02525	4.23285	8
2023-2-21 22:4:25	22.01190	4.23285	8
2023-2-21 22:4:24	22.01190	4.23285	12
2023-2-21 22:4:11	21.98520	4.23285	12
2023-2-21 22:4:10	21.98520	4.23285	10
2023-2-21 22:3:57	21.99855	4.23285	10
2023-2-21 22:3:56	21.99855	4.23285	8
2023-2-21 22:3:44	22.01190	4.23285	8
2023-2-21 22:3:42	22.01190	4.23285	10
2023-2-21 22:3:29	22.02525	4.23285	10
2023-2-21 22:3:26	22.02525	4.23285	34

More results...

Figura 34-Imagem da tabela exibida na Interface web

Por fim, referir que no Anexo C, encontra-se o código para enviar notificações sobre o estado do doseador, nível e ph da água.

## 6 Verificação e Validação

A verificação e validação são processos independentes que são utilizados em conjunto para verificar se um produto, serviço ou sistema cumpre os requisitos e as especificações e que são adequados aos seus propósitos. A verificação confirma os testes e com provas objetivas que os requisitos especificados foram implementados.

Na Tabela 3 observa-se um sumário dos resultados obtidos após os testes do sistema APPT.

*Tabela 3-Sumário de resultados dos testes realizados*

Nome do Teste	Mês	Nºs de testes com sucesso	Nºs de testes sem sucesso	Total de testes
Verificar o e-mail e <i>password</i> do utilizador	Novembro	7	3	10
Enviar notificação sobre estado do doseador, nível e ph da água.	Fevereiro	3	1	4
Consultar os dados dos sensores (ex: PH, temperatura, nível da água);	Dezembro	10	5	15
Controlar o doseador	Dezembro	6	2	8
	Total %	70 %	30 %	37 = 100 %

Como se pode observar na Tabela 3, os testes realizados passaram com sucesso. É de salientar que ao longo do desenvolvimento do código da aplicação APPT foram executados vários testes e por questões de simplicidade não foram apresentados nem contabilizados.

## 7 Conclusão

Em conclusão, o grande objetivo do projeto foi a implementação de uma aplicação web para o sistema APPT. Que permite ao utilizador consultar os dados dos sensores (ph, temperatura e o nível da água), controlar o doseador e receber notificações. A aplicação responde à necessidade de uma solução de baixo custo para manter a qualidade da água num purificador. O sistema fornece monitorização e controle em tempo real dos parâmetros da qualidade da água, o que pode ajudar a prevenir riscos relativos a saúde.

A utilização de Firebase, ESP8266, HTML e CSS revelou-se ser eficaz e acessível. Acredita-se que o projeto tem um potencial enorme, pois alcançou os objetivos e expectativas propostos destacando a importância de ter um purificador de água caseiro de baixo custo que possa fazer a gestão dos recursos hídricos. Embora a versão atual do sistema tenha algumas limitações, ela demonstra o potencial dessa tecnologia para maior desenvolvimento e otimização em termos de trabalho futuro.

Em termos de trabalho futuro, poder-se-á melhorar a “interface” da aplicação “web” e adicionar mais recursos: como visualização de dados por gráficos de linha e receber notificação por SMS. Também poderia integrar sensores adicionais, como oxigénio dissolvido na água, para fornecer uma descrição mais completa dos parâmetros da água.

## Bibliografia

- [1] S. Srivastava, S. Vaddadi e S. Sadistap, "Smartphone-based System for water quality analysis," *Applied Water Science*, pp. 1-13, 2018 August 2018.
- [2] Demetillo, A. T, M. V. Japitana, Taboada e E. B., "A system for monitoring water quality in a large aquatic area using wireless sensor network technology," *Demetillo et al. Sustainable Environment Research*, p. 29:12, 23 April 2019.
- [3] J. L. HARDCASTLE, "In-Situ Launches Water Monitoring Smartphone App," 11 APRIL 2013. [Online]. Available: <https://www.environmentalleader.com/2013/04/in-situ-launches-water-monitoring-smartphone-app/>. [Acedido em 20 05 2020].
- [4] M. B. e. R. D. Robert Bosh GmbH, "Scrum Workshop," em *Robert Bosh GmbH, Scrum Workshop*, M. Bauerdick e R. Dias, Guarda, 2019.
- [5] M. Rodríguez, "Scrum Guide: The Past, Present and Future of Scrum," 12 01 2021. [Online]. Available: <https://netmind.net/en/the-past-present-and-future-of-scrum/>. [Acedido em 13 02 2023].
- [6] CRIARH Consultoria, "SPRINT BACKLOG: O QUE É E COMO FAZER FACILMENTE," 2022. [Online]. Available: <https://criarh.com.br/sprint-backlog/>. [Acedido em 13 02 2023].
- [7] M. C. d. S. P. Silveira, "A Reutilização de Requisitos no Desenvolvimento e Adaptação de Produtos de Software," *Dissertação apresentada para obtenção do grau de Doutor em Engenharia Electrotécnica e de Computadores*, p. 342, Janeiro 2006.
- [8] M. C. d. S. P. Silveira, "A Reutilização de Requisitos no Desenvolvimento e Adaptação de Produtos de Software," Janeiro 2006. [Online]. Available: [https://repositorio-aberto.up.pt/bitstream/10216/12020/2/Texto%20integral.pdf](https://repositorio-aberto.up.pt/bitstream/10216/12020/2/Texto%20integral.pdf;);. [Acedido em 20 11 2022].
- [9] arduino, "Wemos D1 mini pro 16MB clone," 2017 Outubro 07. [Online]. Available: <https://forum.arduino.cc/t/wemos-d1-mini-pro-16mb-clone/539411/6>. [Acedido em 19 02 2023].
- [10] hardbrick, "Conhecendo o Wemos D1 mini / Pro a fundo," 26 JANEIRO 2021. [Online]. Available: <https://hardbrick.com.br/2021/01/26/conhecendo-o-wemos-d1-mini-pro-a-fundo/>. [Acedido em 19 02 2023].
- [11] aprendiendoarduino, "Aprendiendo Arduino," 14 noviembre 2018.. [Online]. Available: <https://aprendiendoarduino.wordpress.com/tag/dht11/>. [Acedido em 19 2 2023].
- [12] Robot Electronics, "SRF04 - Ultra-Sonic Ranger Technical Specification," 2003 May . [Online]. Available: <https://www.robot-electronics.co.uk/htm/srf04tech.htm>. [Acedido em 08 03 2023].

- [13] howtomechatronics, "Ultrasonic Sensor HC-SR04 and Arduino – Complete Guide," [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>. [Acedido em 19 2 2023].
- [14] seeedstudio, "Grove - PH Sensor Kit (E-201C-Blue)," [Online]. Available: <https://wiki.seeedstudio.com/Grove-PH-Sensor-kit/>. [Acedido em 24 05 2022].
- [15] mauser, "Motor micro servo 4.8V..6V DC SG90 - 180º," [Online]. Available: [https://mauser.pt/catalog/product\\_info.php?products\\_id=096-6477&gclid=Cj0KCQiAutyfBhCMARIsAMgcRjQd2wCOR0\\_vXFSEOivesWczKxiLB6YePS3hA1SfrKHeGkN76\\_1rj1QaAnHIEALw\\_wcB](https://mauser.pt/catalog/product_info.php?products_id=096-6477&gclid=Cj0KCQiAutyfBhCMARIsAMgcRjQd2wCOR0_vXFSEOivesWczKxiLB6YePS3hA1SfrKHeGkN76_1rj1QaAnHIEALw_wcB). [Acedido em 22 02 2023].
- [16] Lastminute Engineers, "Interfacing DS18B20 1-Wire Digital Temperature Sensor with Arduino," 2022. [Online]. Available: <https://lastminuteengineers.com/ds18b20-arduino-tutorial/>. [Acedido em 22 02 2023].
- [17] Farnell, "SEN0204," 2023. [Online]. Available: <https://pt.farnell.com/dfrobot/sen0204/non-contact-water-liquid-level/dp/2946127>. [Acedido em 19 02 2023].
- [18] "An IoT Based Monitoring and Controlling System for Water Chlorination Treatment," Junho 2018. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/3231053.3231084>.



## Anexo A

### **Processo comparativo dos sensores no mercado mais adequados**

## Sensor de Temperatura

### DHT11

Foi usado como forma de simular os valores da temperatura da água. A temperatura da água pode ser influenciada pela temperatura ambiente, humidade relativa e outras variáveis ambientais. Então usou-se essas medições para estimar a temperatura da água. No entanto, a temperatura ambiente pode não refletir com precisão a temperatura da água, pois a água tem uma capacidade térmica maior do que o ar, o que significa que a água leva mais tempo para aquecer ou esfriar do que o ar. Portanto, é importante considerar as propriedades de outros sensores e o uso de fórmula ou equação que leva em consideração essas variáveis.

### DS18B20

O sensor mais indicado seria o Sensor DS18B20, demonstrado na Figura 35, para medir o valor da temperatura da água. Segundo a sua documentação, esse sensor é à prova d'água, alcança a precisão de até 0.5° C e também proporciona leituras de até 12-bits através da sua comunicação com o microcontrolador. É controlado com uma tensão de entrada entre 3 e 5,5V e possui uma faixa de medição entre -55° C a +125°C.



Figura 35- Sensor DS18B20 [15]

## Sensor de Nível da Água

Foi preciso saber o nível da água antes do tratamento com cloro, pois, é importante para garantir a eficácia do processo de desinfecção, bem como para a proteção da saúde humana e do meio ambiente.

### HC-SR04

O sensor de distância ultrassônico HC-SR04 pode ser utilizado para medir a distância entre o sensor e a superfície da água. Dessa forma, é possível estimar o nível da água num tanque

ou recipiente. No entanto, é importante ter em mente que a medição da altura da água com o sensor HC-SR04 é afetada por diversos fatores, como a temperatura da água, a humidade do ar e a presença de obstruções no caminho do sinal ultrassónico. Além disso, o sensor pode apresentar leituras inconsistentes ou imprecisas em algumas situações. Portanto a mais adequado seria utilizar um Sensor de Nível de Líquidos.

#### SEN0204

O DFRobot SEN0204, exibido na Figura 36 é um sensor de nível de líquidos à base de capacitância projetado para medir o nível de líquidos em tanques ou reservatórios. Esse sensor pode ser utilizado para medir o nível de água, óleo, gasolina, entre outros líquidos. ´

Ele possui os seguintes pinos:

1. VCC: Pino de alimentação positiva (normalmente 5V).
2. GND: Pino de alimentação negativa ou terra.
3. S: Pino de saída de sinal analógico que varia relativamente ao nível do líquido.
4. D: Pino de saída de sinal digital acionado quando o nível do líquido atinge um limite definido pelo utilizador.
5. E1: Pino do primeiro elétrodo entrando em contacto com o líquido.
6. E2: Pino do segundo elétrodo entrando em contacto com o líquido.

É importante lembrar que a conexão dos elétrodos do sensor com o líquido a ser medido é fundamental para a correta operação do dispositivo. Além disso, o circuito condicionador de sinal e a “interface” com um microcontrolador ou outro dispositivo eletrónico podem variar dependendo da aplicação.



Figura 36-Sensor DFRobot SEN0204 [15]

## Anexo B

### **Código para exibir a data e hora da última atualização**

### Código para exibir a data e hora da última atualização

Para exibir a hora da última atualização usando JavaScript acessando a data da última modificação da página atual e atualizar o elemento específico no documento HTML. Aqui está um trecho do código:

```
<p>Last updated on: <span id="lastUpdated"></span></p>
```

```
var lastUpdated = document.lastModified;  
document.getElementById("lastUpdated").textContent = lastUpdated;
```

Este código adiciona um elemento de parágrafo ao documento HTML com um elemento vazio 'span' que possui 'id' 'lastUpdated'. O código JavaScript acessa a 'lastModified' propriedade do objeto 'document' e define o conteúdo de texto do 'span' elemento para esse valor. Este exibirá a data e a hora em que a Interface foi atualizada pela última vez no formato "MM/DD/AAAA HH:MM:SS".

## Anexo C

### **Código para enviar notificações**

Para enviar uma notificação *push* com um link para um site usa-se também JavaScript e HTML, com o método “showNotification()”. Aqui está o código:

```
if ('Notification' in window && navigator.serviceWorker) {
  Notification.requestPermission().then(perm => {
    if (perm === "granted") {
      const notification = new Notification("Aleta Nivel da água
APPT!", {
        body: "Nivel da água " + valornivel,
        data: {
          url: ' http://127.0.0.1:5501/',
          tag: 'Nivel baixo' + valornivel
        },
        icon: "Logo teste.png",
        //tag: "Welcome Message",
      })

      notification.addEventListener("click", e => {

    })
  })
}
```

Nesse código, o objeto *options* passado para o método “showNotification()” inclui uma data a propriedade com a URL da aplicação web APPT, como é demonstrado na Figura 37 . Quando a utilização clicar na notificação, o *service worker* pode recuperar esses dados e usá-los para abrir o site em uma nova janela ou guia. Aqui está um exemplo de como lidar com o evento de clique de notificação para abrir o site:

```
self.addEventListener('notificationclick', function(event) {
  const url = event.notification.data.url;
  event.waitUntil(clients.openWindow(url));
  event.notification.close();
});
```

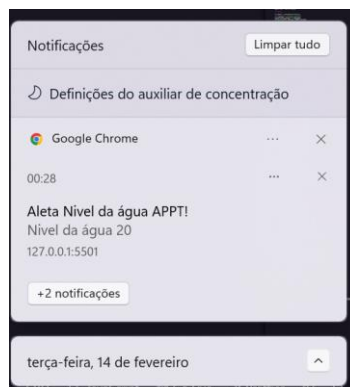


Figura 37-Notificação do nível da água