



# **Desenvolvimento de Soluções de Monitorização e Controlo em Sistemas de Recursos Limitados Relatório de Estágio Curricular – CapTemp**

Mestrado em Engenharia Informática – Computação Móvel

Tomás Marto Mendes

Leiria, novembro de 2023



# **Desenvolvimento de Soluções de Monitorização e Controlo em Sistemas de Recursos Limitados**

Mestrado em Engenharia Informática – Computação Móvel

Tomás Marto Mendes

Estágio realizado sob a orientação do Professor Doutor José Carlos Bregieiro Ribeiro.

Leiria, novembro de 2023

## **Originalidade e Direitos de Autor**

O presente relatório de estágio é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Mestrado em Engenharia Informática – Computação Móvel, no ano letivo 2023/2024 da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos (se aplicável).

## **Dedicatória**

Dedico este trabalho a todas as pessoas que, de alguma forma, contribuíram para a minha caminhada ao longo desta jornada académica.

Aos meus pais, pelo apoio incondicional, pelos valores transmitidos e pela confiança que sempre depositaram em mim. Sem o seu suporte constante e inabalável, este percurso seria muito mais difícil.

À minha família, pela compreensão e pelo incentivo em momentos de dúvida, e aos amigos, pelo encorajamento e pelas palavras de força nos momentos mais desafiadores.

Por fim, dedico esta conquista a todos os professores e colegas que me inspiraram e ajudaram ao longo deste caminho, tornando esta etapa tão enriquecedora e gratificante.

## Agradecimentos

Gostaria de expressar a minha profunda gratidão a todos aqueles que, de várias formas, contribuíram para a realização deste relatório de estágio de mestrado.

Em primeiro lugar, agradeço ao meu orientador, Professor Doutor José Carlos Bregieiro Ribeiro, pelo apoio incansável, orientação e valiosos conselhos ao longo deste percurso. A sua experiência e dedicação foram fundamentais para o sucesso deste trabalho.

À equipa da empresa “CapTemp”, agradeço pela oportunidade de realizar o estágio, bem como pela acolhida calorosa e o ambiente de trabalho inspirador. Agradeço especialmente ao supervisor João Agostinho e aos restantes colegas de trabalho, que sempre se mostraram disponíveis para partilhar conhecimentos e experiências.

Aos meus colegas de mestrado e amigos, Pedro Fernandes e Diogo Rodrigues, pela camaradagem e pelo apoio mútuo ao longo desta jornada académica. As discussões e colaborações enriqueceram imensamente a minha experiência e contribuíram para o meu crescimento profissional e pessoal.

Aos meus restantes amigos e familiares, pelo constante encorajamento, paciência e compreensão durante os momentos desafiantes. O seu apoio emocional foi essencial para que eu pudesse concluir esta etapa com êxito.

Por fim, mas não menos importante, agradeço a todos os professores da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria por proporcionarem um ambiente académico estimulante e por todo o suporte prestado ao longo do curso.

A todos, o meu sincero obrigado.

## Resumo

Este relatório, elaborado no âmbito do estágio curricular do Mestrado em Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria, realizado na empresa “CapTemp”, tem como objetivo transmitir os conhecimentos adquiridos durante o estágio. Este proporcionou a aplicação prática de conhecimentos teóricos, desenvolvimento de competências técnicas e compreensão da importância da colaboração em equipa e comunicação eficaz.

A “CapTemp”, sediada em Pombal, Leiria, especializa-se no desenvolvimento de soluções personalizadas de monitorização, controlo e supervisão. A empresa integra sensores, atuadores, coletores de dados e software dedicado para criar sistemas de monitorização eficazes, abrangendo desde o desenvolvimento dos sensores até à criação do software para análise e armazenamento de dados, garantindo uma atuação abrangente com base nos valores obtidos.

Durante o estágio, foram desenvolvidos projetos nas áreas de programação de *web services*, comunicação cliente-servidor, desenvolvimento de sistemas IoT automatizados e monitorizados, comunicação UDP e serviços de rede. Os principais projetos foram: "Senslive", "Nidus" e "CT Discover Mobile".

A plataforma online “Senslive” é uma solução baseada na nuvem que centraliza a monitorização de sensores, permitindo aos utilizadores gerir dispositivos e sensores, monitorizar em tempo real, modificar dados, criar relatórios automáticos, entre outras funcionalidades. A plataforma suporta dispositivos em diferentes locais e implementa um sistema de permissões para gestão eficaz e personalizada dos recursos.

O dispositivo “Nidus” permite a conexão de diversos sensores e dispositivos de atuação, transmitindo os dados para uma plataforma na nuvem para análise. O projeto focou-se em melhorar a mobilidade do dispositivo, desenvolvendo software para um módulo GPS externo que, em conjunto com a “Nidus”, permite a obtenção em tempo real da localização geográfica de múltiplos dispositivos conectados.

O projeto "CT Discover Mobile" envolveu a implementação de uma extensão móvel do projeto “CT Discover” usando a linguagem “Flutter”. Esta aplicação permite gerir as “Nidus” de forma intuitiva em dispositivos móveis, facilita a procura e identificação de dispositivos, visualização e controlo dos sensores, análise detalhada dos dados, alteração das configurações da “Nidus” e criação de relatórios detalhados.

**Palavras-chave:** sensores, rede, dispositivos, serviços *web*, cliente-servidor, sistemas IoT, comunicação UDP, módulo GPS

## Abstract

This report, prepared within the scope of the curricular internship of the Master's in Computer Engineering – Mobile Computing at the School of Technology and Management of the Polytechnic Institute of Leiria, carried out at “CapTemp”, aims to convey the knowledge acquired during the internship. It provided the practical application of theoretical knowledge, the development of technical skills, and an understanding of the importance of teamwork and effective communication.

“CapTemp”, based in Pombal, Leiria, specializes in developing customized monitoring, control, and supervision solutions. The company integrates sensors, actuators, data collectors, and dedicated software to create effective monitoring systems, encompassing everything from sensor development to the creation of software for data analysis and storage, ensuring comprehensive action based on the obtained values.

During the internship, projects were developed in the areas of web services programming, client-server communication, development of automated and monitored IoT systems, UDP communication, and network services. The main projects were: "Senslive," "Nidus," and "CT Discover Mobile."

The online platform “Senslive” is a cloud-based solution that centralizes sensor monitoring, allowing users to manage devices and sensors, monitor in real time, modify data, create automatic reports, among other functionalities. The platform supports devices in different locations and implements a permissions system for effective and customized resource management.

The "Nidus" device allows the connection of various sensors and actuating devices, transmitting the data to a cloud platform for analysis. The project focused on improving the device's mobility by developing software for an external GPS module which, in conjunction with the "Nidus," allows real-time acquisition of the geographical location of multiple connected devices.

The "CT Discover Mobile" project involved implementing a mobile extension of the "CT Discover" project using the "Flutter" language. This application allows managing "Nidus" intuitively on mobile devices, facilitates the search and identification of devices, visualization and control of sensors, detailed data analysis, modification of "Nidus" settings, and creation of detailed reports.

**Keywords:** sensors, network, devices, web services, client-server, IoT systems, UDP communication, GPS module

# Índice

<b>Originalidade e Direitos de Autor</b> .....	<b>1</b>
<b>Dedicatória</b> .....	<b>2</b>
<b>Agradecimentos</b> .....	<b>3</b>
<b>Resumo</b> .....	<b>4</b>
<b>Abstract</b> .....	<b>6</b>
<b>Índice</b> .....	<b>7</b>
<b>Lista de Figuras</b> .....	<b>10</b>
<b>Lista de Tabelas</b> .....	<b>12</b>
<b>Lista de Blocos de Código</b> .....	<b>13</b>
<b>Lista de Siglas e Acrónimos</b> .....	<b>14</b>
<b>1. Introdução</b> .....	<b>16</b>
<b>1.1. Caracterização da Entidade de Acolhimento</b> .....	<b>17</b>
<b>1.2. Organização do Documento</b> .....	<b>18</b>
<b>2. Estado da Arte</b> .....	<b>19</b>
<b>2.1. Introdução ao Campo de Estudo</b> .....	<b>19</b>
<b>2.2. Soluções e tecnologias disponíveis</b> .....	<b>19</b>
2.2.1. Tecnologias disponíveis.....	20
2.2.2. Produtos similares.....	21
<b>3. Trabalho desenvolvido</b> .....	<b>26</b>
<b>3.1. Ferramentas Utilizadas</b> .....	<b>26</b>
3.1.1. Laragon .....	26
3.1.2. pgAdmin .....	26
3.1.3. Postman.....	27
3.1.4. Hercules .....	27
3.1.5. WinSCP.....	28

<b>3.2. Senslive – Plataforma Web .....</b>	<b>28</b>
3.2.1. Metodologia de Desenvolvimento .....	29
3.2.2. Requisitos e Funcionalidades.....	30
3.2.3. Distribuição Temporal do Desenvolvimento .....	32
3.2.4. Arquitetura e Design da Aplicação .....	33
3.2.5. Desenvolvimento Frontend com Vue 3 .....	36
3.2.6. Apresentação da Aplicação Resultante.....	37
3.2.7. Comparação com outras <i>frameworks frontend</i> .....	47
3.2.8. Desenvolvimento Backend com Laravel .....	48
3.2.9. Comparação com outras <i>frameworks backend</i> .....	49
3.2.10. Testes e validações.....	51
3.2.11. Integração com PostgreSQL .....	53
3.2.12. Aplicação dos Conhecimentos Adquiridos .....	54
<b>3.3. Relacionamento entre Teoria e Prática.....</b>	<b>54</b>
3.3.1. Frontend com Vue 3.....	54
3.3.2. Backend com Laravel .....	55
<b>3.4. “Nidus” – Módulo GPS .....</b>	<b>57</b>
3.4.1. Contexto – “Nidus” .....	57
3.4.2. Características principais da “Nidus” .....	58
3.4.3. Áreas de aplicação da “Nidus” .....	59
3.4.4. Contexto – GL-X300B-GPS .....	60
3.4.5. Características Principais do GL-X300B-GPS .....	60
3.4.6. Aplicações Comuns do Módulo.....	61
3.4.7. Metodologia de desenvolvimento .....	62
3.4.8. Requisitos e funcionalidades.....	62
3.4.9. Arquitetura e design.....	62
3.4.10. Envio de dados para a plataforma.....	63
3.4.1. Testes .....	65
<b>3.5. CT Discover – Plataforma mobile .....</b>	<b>66</b>
3.5.1. Metodologia de desenvolvimento .....	67
3.5.2. Desenvolvimento com Flutter.....	67
3.5.3. Requisitos e funcionalidades.....	67

3.5.4.	Distribuição temporal do desenvolvimento .....	69
3.5.5.	Arquitetura e design .....	70
3.5.6.	Descoberta de dispositivos por UDP .....	72
3.5.7.	Descoberta de sensores <i>wired / wireless</i> .....	75
3.5.8.	Localização dos sensores .....	77
3.5.9.	Definições da Nidus .....	78
3.5.10.	Módulo F.....	79
3.5.11.	Cálculo de <i>hashes</i> .....	81
3.5.12.	Cálculo de <i>checksum</i> .....	82
<b>4.</b>	<b>Análise crítica e proposta de melhorias .....</b>	<b>86</b>
<b>4.1.</b>	<b>Análise Crítica.....</b>	<b>86</b>
<b>4.2.</b>	<b>Propostas de Melhoria.....</b>	<b>86</b>
<b>5.</b>	<b>Conclusão.....</b>	<b>88</b>
<b>6.</b>	<b>Bibliografia .....</b>	<b>90</b>

# Lista de Figuras

Figura 1 - Distribuição Temporal Senslive .....	33
Figura 2 - Arquitetura Senslive .....	34
Figura 3 - Arquitetura Relação Dispositivos-Senslive .....	35
Figura 4 - Arquitetura Comunicação SMS/Email .....	36
Figura 5 - Dashboard Senslive .....	37
Figura 6 - <i>Status</i> Plataforma Senslive .....	38
Figura 7 - Página de <i>Roles</i> Senslive .....	41
Figura 8 - Formulário de edição de uma <i>role</i> .....	41
Figura 9 - Utilizadores Senslive .....	42
Figura 10 - Relatórios Automáticos Senslive .....	42
Figura 11 - Empresas Senslive .....	43
Figura 12 - Gateways Senslive .....	44
Figura 13 - Dados Dispositivo Senslive .....	44
Figura 14 - Gráfico de Sensor .....	45
Figura 15 - Dispositivos Virtuais Senslive .....	45
Figura 16 - Dados Dispositivo Virtual Senslive .....	46
Figura 17 - Certificados Senslive .....	47
Figura 18 - Módulo GPS GL-X300B-GPS (adaptado de [10]) .....	60
Figura 19 - Arquitetura GPS .....	63
Figura 20 - Distribuição temporal do desenvolvimento - CTDDiscover .....	70
Figura 21 - Arquitetura CTDDiscover .....	71
Figura 22 - Pedidos HTTP e UDP - CTDDiscover .....	71
Figura 23 - Lista de Dispositivos Encontrados via Comunicação UDP .....	75
Figura 24 - Lista de Dispositivos Encontrados e Respetivos Sensores Conectados .....	76
Figura 25 - Detalhes dos Sensores Conectados a um Dispositivo .....	77
Figura 26 - Página de localizações dos sensores .....	78
Figura 27 - Página de definições da "Nidus" .....	79
Figura 28 - Página do Módulo F .....	80

Figura 29 - Monitorização dos sensores em formato de lista.....	81
Figura 30 - Monitorização dos Sensores em Formato de Gráfico.....	81
Figura 31 - Verificação de Checksum dos Ficheiros da "Nidus".....	82
Figura 32 - Algoritmo SHA1 (adaptado de [11]).....	82

## Lista de Tabelas

Tabela 1 - Comparação <i>frameworks frontend</i> .....	48
Tabela 2 - Comparação <i>frameworks backend</i> .....	50
Tabela 3 - Principais Características das Nidus .....	59

## Lista de Blocos de Código

Bloco de Código 1 - Encriptação SHA1 de cada ficheiro .....	38
Bloco de Código 2 - Encriptação SHA1 de cada ficheiro em Linux.....	38
Bloco de Código 3 - Obtenção de Espaço Total e Disponível em Disco .....	39
Bloco de Código 4 - Utilização de CPU em Ambiente Windows.....	39
Bloco de Código 5 - Memória Física Total em Ambiente Windows .....	39
Bloco de Código 6 - Memória Física Disponível em Ambiente Windows .....	40
Bloco de Código 7 - Memória Física Livre em Ambiente Windows.....	40
Bloco de Código 8 - Memória Física Utilizada em Ambiente Windows.....	40
Bloco de Código 9 - Memória Física em Ambiente Linux .....	40
Bloco de Código 10 - Resultado do Comando "free -b" Após Formatação em Ambiente Linux .....	41
Bloco de Código 11 – Função de Envio de Dados de Localização GPS .....	65
Bloco de Código 12 - Inicialização de Socket Datagram para Comunicação UDP .....	72
Bloco de Código 13 - Envio de Mensagem por Broadcast via Socket UDP.....	73
Bloco de Código 14 - Receção e Processamento de Pacotes UDP .....	73
Bloco de Código 15 - Leitura e Limpeza de Buffer.....	74
Bloco de Código 16 - Decodificação e Processamento de Dados em String .....	74
Bloco de Código 17 - Cálculo de CRC para Requisição Nidus com Endereço MAC .....	85

## Lista de Siglas e Acrónimos

<b>API</b>	Application Programming Interface
<b>AWS</b>	Amazon Web Service
<b>CRUD</b>	Create, Read, Update, Delete
<b>CSV</b>	Comma-Separated Values
<b>ESTG</b>	Escola Superior de Tecnologia e Gestão
<b>GPS</b>	Global Positioning System
<b>HACCP</b>	Hazard Analysis and Critical Control Points
<b>HTML</b>	Hyper Text Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HVAC</b>	Heating, Ventilation and Air Conditioning
<b>IA</b>	Inteligência Artificial
<b>IIoT</b>	Industrial Internet of Things
<b>IP</b>	Internet Protocol
<b>IoT</b>	Internet of Things
<b>iOS</b>	iPhone Operating System
<b>LTE</b>	Long Term Evolution
<b>MAC</b>	Media Access Control
<b>MVC</b>	Model-View-Controller
<b>ORM</b>	Object Relational Mapper
<b>PDF</b>	Portable Document Format
<b>PHP</b>	Hypertext Preprocessor
<b>PLC</b>	Programmable Logic Controller
<b>RDP</b>	Remote Desktop Protocol
<b>SDK</b>	Software Development Kit
<b>SCADA</b>	Supervisory Control and Data Acquisition
<b>SHA1</b>	Secure Hash Algorithm 1
<b>SMS</b>	Short Message Service
<b>SQL</b>	Structured Query Language
<b>TCP</b>	Transmission Control Protocol
<b>TRL</b>	Technology Readiness Level
<b>UDP</b>	User Datagram Protocol

<b>UPS</b>	Uninterruptible Power Supply
<b>WMI</b>	Windows Management Instrumentation
<b>XLS</b>	Excel Spreadsheet

# 1. Introdução

Este relatório oferece uma visão abrangente do trabalho desenvolvido durante o estágio curricular do Mestrado em Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria, realizado na empresa “CapTemp”. O estágio foi projetado para permitir a aplicação prática dos conhecimentos teóricos adquiridos ao longo do curso, bem como para desenvolver competências técnicas essenciais e entender a importância da colaboração em equipa e da comunicação eficaz.

Durante o estágio, as atividades concentraram-se no desenvolvimento de software e na integração de tecnologias avançadas. As tarefas incluíram a programação de *web services* e a implementação de sistemas de IoT (*Internet of Things*), com atenção especial à comunicação cliente-servidor e ao uso de protocolos de comunicação como UDP (*User Datagram Protocol*) e serviços de rede. O estágio envolveu o trabalho em três projetos principais, cada um com suas particularidades e desafios.

O projeto “Senslive” envolveu a criação de uma plataforma online baseada na nuvem para a centralização da monitorização de sensores. Esta plataforma proporciona aos utilizadores a capacidade de gerir dispositivos e sensores, realizar monitorização em tempo real, modificar dados e criar relatórios automáticos. Além disso, a plataforma foi desenvolvida com uma flexibilidade que permite suportar dispositivos em múltiplas localizações, implementando um sistema de permissões que assegura que os utilizadores acedam apenas aos dispositivos associados às suas localizações específicas, o que garante uma gestão eficiente e personalizada dos recursos.

No projeto “Nidus”, o foco foi no desenvolvimento de um dispositivo que facilita a conexão de uma ampla gama de sensores e dispositivos de atuação, oferecendo uma solução centralizada para a monitorização. O dispositivo é capaz de transmitir dados para uma plataforma na nuvem, onde as informações são analisadas de forma eficaz. O projeto teve como objetivo principal aprimorar a mobilidade do dispositivo através do desenvolvimento de software para um módulo GPS externo. Este software, integrado com o dispositivo “Nidus”, possibilita a obtenção em tempo real da localização geográfica de múltiplos dispositivos conectados, proporcionando uma solução de monitorização móvel e eficiente.

O projeto "CT Discover Mobile" envolveu a implementação de uma extensão móvel do sistema "CT Discover", utilizando a linguagem de programação Flutter. Esta aplicação oferece funcionalidades essenciais para a gestão intuitiva dos dispositivos Nidus em plataformas móveis, permitindo a gestão

remota, a procura e identificação de dispositivos em diferentes redes, a visualização e controlo dos sensores associados, além de realizar análises detalhadas dos dados dos sensores e a criação de relatórios. O projeto visa proporcionar uma interface amigável e eficaz para a gestão e monitorização dos dispositivos.

Este relatório detalha as atividades realizadas durante o estágio, abordando as metodologias aplicadas e os resultados obtidos. A análise oferecida proporciona uma compreensão completa dos desafios enfrentados, das soluções implementadas e do impacto das experiências adquiridas ao longo do estágio.

### **1.1. Caracterização da Entidade de Acolhimento**

A empresa “CapTemp”<sup>1</sup>, com sede em Pombal, Leiria, foi fundada em 25 de julho de 2013 e destaca-se pela sua especialização no desenvolvimento de soluções personalizadas para monitorização, controlo e supervisão. A empresa assume a responsabilidade de criar sistemas de monitorização completos, desde a integração de sensores até à implementação de software especializado para análise e armazenamento de dados.

A “CapTemp” apresenta projetos como o “T-Trail”, que se trata de um sistema de IA (Inteligência Artificial) para rastrear a temperatura corporal humana. Possui funcionalidades como reconhecimento facial com um algoritmo de IA rápido e preciso, leitura da temperatura em várias pessoas ao mesmo tempo, deteção da utilização de máscara e rastreamento das pessoas detetadas com temperatura elevada. Trata-se de um sistema independente, modular e portátil. Este projeto tem aplicações em aeroportos e terminais de transporte, transportes públicos, hospitais, clínicas, farmácias e lares, hotéis, centros comerciais e espaços de lazer, onde houver necessidade de monitorar a temperatura de indivíduos ou um grupo de pessoas.

O foco principal da “CapTemp” reside na capacidade de desenvolver sistemas que não só recolhem dados, mas também os interpretam de forma inteligente. Esta capacidade é assegurada pela integração de software dedicado, permitindo uma análise eficiente e armazenamento dos dados recolhidos. Desta forma, a empresa oferece não só capacidades de monitorização, mas também a capacidade de resposta com base nos dados capturados, o que é crucial para diversos setores e aplicações.

---

<sup>1</sup> <https://www.capttemp.com>

## **1.2. Organização do Documento**

Este relatório está estruturado em 5 capítulos, começando com uma introdução ao tema, onde são abordados o contexto do estágio, os projetos realizados, a entidade de acolhimento e aspetos iniciais relevantes que moldaram a experiência.

O segundo capítulo dedica-se à apresentação das tecnologias e projetos estudados, com o propósito de fornecer o suporte técnico e teórico ao trabalho desenvolvido durante o estágio, elucidando as bases tecnológicas que fundamentaram as soluções propostas.

O terceiro capítulo descreve detalhadamente o trabalho realizado na empresa para a resolução dos problemas identificados. Neste capítulo, são expostos os aspetos técnicos das soluções implementadas, incluindo a metodologia seguida, os testes conduzidos e os resultados obtidos, proporcionando uma visão clara e aprofundada das atividades desenvolvidas durante o estágio.

O quarto capítulo oferece uma análise crítica das soluções propostas, avaliando a eficácia e eficiência das abordagens adotadas. Esta análise inclui uma revisão detalhada dos resultados alcançados em comparação com os objetivos estabelecidos, identificando pontos fortes e áreas de melhoria. São discutidos os desafios enfrentados durante o desenvolvimento e a implementação das soluções, bem como as limitações encontradas. Além disso, são apresentadas propostas de melhorias para otimizar as soluções existentes, incluindo recomendações para ajustes técnicos, refinamentos no processo e sugestões para a integração de novas tecnologias ou metodologias que possam contribuir para uma melhoria contínua.

Por fim, o quinto capítulo fornece uma conclusão geral sobre o trabalho realizado, sintetizando as principais conclusões e aprendizados adquiridos durante o estágio. São discutidas as dificuldades encontradas ao longo do percurso e as estratégias adotadas para superá-las, assim como são apresentadas recomendações para futuras implementações, com o objetivo de fomentar a evolução contínua e o aprimoramento das práticas e soluções desenvolvidas.

## 2. Estado da Arte

Nesta secção, será apresentada uma visão abrangente do estado atual das tecnologias e soluções relevantes para o desenvolvimento e implementação de sistemas de monitorização e gestão no contexto da IoT [1].

O objetivo é oferecer uma compreensão clara das soluções tecnológicas disponíveis e suas aplicações práticas, explorando as principais tendências, inovações e desafios do setor. Será realizada uma revisão crítica da literatura existente, destacando os principais avanços e a evolução das tecnologias envolvidas, bem como a análise das ferramentas e metodologias utilizadas.

Além disso, esta secção abordará as tendências e inovações mais recentes, identificando oportunidades emergentes e as implicações de tais desenvolvimentos para a eficiência operacional e a gestão de processos industriais. A compreensão aprofundada dessas tecnologias é essencial para avaliar a relevância e o impacto das soluções implementadas durante o estágio, proporcionando um contexto que facilita a análise crítica e a interpretação dos resultados obtidos.

### 2.1. Introdução ao Campo de Estudo

O campo de estudo abordado neste relatório é a aplicação prática de tecnologias avançadas no desenvolvimento de sistemas de monitorização e gestão, com foco específico na IoT e em tecnologias associadas. Este setor é crucial para a otimização dos processos industriais e para a melhoria da eficiência operacional, desempenhando um papel significativo em diversas indústrias modernas. A compreensão das tecnologias e soluções existentes é essencial para contextualizar e avaliar o trabalho realizado durante o estágio.

### 2.2. Soluções e tecnologias disponíveis

Nesta secção, será explorada uma variedade de soluções e tecnologias disponíveis no contexto da IIoT (Internet Industrial das Coisas) [1]. Desde sensores e atuadores até plataformas de gestão de dados e ciber-segurança, estas tecnologias são relevantes para o leitor, pois foram amplamente utilizadas durante o estágio, desempenhando um papel crucial na otimização dos processos industriais e na melhoria da eficiência operacional.

Conhecer estas tecnologias é relevante para o leitor pois são essenciais para compreender o contexto e as abordagens descritas no relatório. Durante o estágio, foram utilizadas tecnologias como

sensores, atuadores, plataformas de gestão de dados e medidas de cibersegurança em diversos projetos da empresa “CapTemp”. Estas tecnologias foram fundamentais para a otimização dos processos e a melhoria da eficiência operacional. Portanto, a familiaridade com essas tecnologias é crucial para uma compreensão completa das soluções e resultados apresentados no relatório.

Antes de se analisarem produtos específicos, é fundamental entender as diversas tecnologias que compõem o panorama da IIoT e como estas são aplicadas para atender às demandas industriais modernas. Serão, assim, discutidas algumas tecnologias disponíveis tal como as suas características, funcionalidades e benefícios.

### **2.2.1. Tecnologias disponíveis**

As tecnologias de IIoT representam a aplicação dos princípios da Internet das Coisas no setor industrial. Estas tecnologias têm como principal objetivo a otimização de processos produtivos, a melhoria da eficiência operacional, o aumento da segurança e a viabilização da manutenção preditiva, entre outras vantagens. Entre as principais tecnologias associadas ao IIoT, destacam-se os sensores e atuadores, a conectividade, as plataformas de gestão de dados, o *Big Data* e *Analytics*, o *edge computing* e a cibersegurança.

Os sensores e atuadores são dispositivos fundamentais que permitem a coleta de dados de máquinas e ambientes (sensores) e a atuação sobre máquinas e processos com base em comandos recebidos (atuadores). A conectividade refere-se aos protocolos e tecnologias de comunicação que viabilizam a transferência de dados entre dispositivos, como *Wi-Fi*, *Bluetooth*, “LoRaWAN”, “Zigbee”, 5G, etc. As plataformas de gestão de dados são sistemas que recolhem, armazenam, processam e analisam os dados provenientes de dispositivos IIoT, podendo ser locais (*on-premises*) ou baseadas na nuvem, como as oferecidas pela “AWS IoT”, “Microsoft Azure IoT” e “Google Cloud IoT”.

Os coletores de dados IIoT desempenham um papel vital na arquitetura IIoT, pois são responsáveis pela recolha, processamento e transmissão de dados provenientes de diversos sensores e dispositivos industriais para um sistema centralizado de gestão ou análise. Entre os principais tipos de coletores de dados IIoT encontram-se os PLC (*Programmable Logic Controller*) [2], que são dispositivos utilizados para automação de processos industriais, capazes de coletar dados de sensores e atuar sobre máquinas com base nesses dados. Os *data loggers*, que são dispositivos que recolhem e armazenam dados de sensores ao longo do tempo, frequentemente utilizados para monitorização de condições ambientais e operacionais. Por fim, os *gateways* IIoT, dispositivos que conectam sensores e equipamentos

industriais a redes de comunicação, facilitando a transferência de dados para plataformas de análise ou sistemas de gestão e realizando funções de pré-processamento e filtragem de dados.

É imperativo que os coletores de dados IIoT sejam robustos, confiáveis e capazes de operar em ambientes industriais adversos. Devem ainda ser compatíveis com diversos protocolos de comunicação e possuir capacidade de processamento suficiente para lidar com grandes volumes de dados em tempo real.

Em relação às plataformas *web* para monitorização e gestão de dispositivos dos clientes, estas constituem uma tecnologia essencial no contexto industrial e empresarial contemporâneo. Estas plataformas permitem a supervisão em tempo real e a administração centralizada de dispositivos conectados, proporcionando uma série de benefícios cruciais para a eficiência operacional.

A capacidade de monitorizar o status e o desempenho dos dispositivos em tempo real possibilita a deteção imediata de problemas, permitindo a implementação rápida de soluções e minimizando o tempo de inatividade. Além disso, estas plataformas permitem a gestão centralizada de dispositivos dispersos geograficamente, facilitando a supervisão e o controlo remoto, o que é particularmente vantajoso para empresas com operações distribuídas.

Outra funcionalidade fundamental é a análise de dados. Estas plataformas recolhem e analisam grandes volumes de dados gerados pelos dispositivos, oferecendo *insights* valiosos sobre o seu desempenho e utilização. Esta análise facilita a manutenção preditiva e a otimização dos recursos, melhorando a eficiência global.

As plataformas também permitem a configuração e atualização remota de dispositivos, reduzindo a necessidade de intervenções físicas e, conseqüentemente, os custos operacionais. As robustas medidas de segurança implementadas garantem a proteção dos dados e a integridade dos dispositivos, integrando encriptação de dados e autenticação de utilizadores para mitigar ameaças cibernéticas.

### **2.2.2. Produtos similares**

Nesta secção, serão apresentados e analisados produtos similares aos três principais projetos desenvolvidos durante o estágio: “Senslive”, “Nidus” e “CT Discover Mobile”. O objetivo é fornecer ao leitor uma visão abrangente de soluções comparáveis disponíveis no mercado, destacando as suas principais características, vantagens e limitações. Esta análise permitirá uma compreensão mais aprofundada das opções existentes e auxiliará na tomada de decisões informadas relativamente à escolha de tecnologias apropriadas para diversas aplicações.

“Senslive” é uma plataforma online baseada na nuvem que centraliza a monitorização de sensores, permitindo a gestão de dispositivos, a visualização de dados em tempo real, a geração automática de relatórios, entre outras funcionalidades.

O projeto “Nidus”, por outro lado, centra-se no desenvolvimento de um dispositivo que facilita a conexão de sensores e atuadores, proporcionando uma solução centralizada para a monitorização, com integração de um módulo GPS para rastreamento em tempo real..

Por fim, a aplicação “CT Discover Mobile” que visa expandir o sistema “CT Discover” para plataformas móveis, permitindo a gestão e monitorização remota de dispositivos Nidus.

Esta análise comparativa visa proporcionar uma visão clara das soluções alternativas existentes, permitindo uma escolha mais informada em função das necessidades específicas de cada aplicação.

### **1. Senslive: AWS IoT**

A “AWS IoT” [3] é uma plataforma de serviços oferecida pela “Amazon Web Services” (AWS) que facilita a conexão e gestão de dispositivos de IoT em larga escala. Esta plataforma permite que dispositivos conectados interajam de forma segura e eficiente com aplicações na nuvem e entre si.

Com a “AWS IoT”, os utilizadores podem monitorizar e gerir dispositivos remotamente, recolher e analisar dados em tempo real, e implementar soluções de manutenção preditiva. A plataforma oferece ferramentas avançadas para análise de dados, integração com outros serviços AWS e segurança robusta, incluindo encriptação e autenticação de dispositivos.

A “AWS IoT” é uma solução abrangente que suporta a inovação e eficiência operacional em diversas indústrias, facilitando a transformação digital através da integração de dispositivos conectados com a nuvem.

Comparativamente, a "Senslive" diferencia-se pela sua interface unificada e foco em simplificar a gestão integrada de múltiplos sistemas de monitorização em tempo real. Foi desenvolvida utilizando tecnologias modernas como “Vue” e “Laravel”, oferecendo uma solução escalável e eficiente para empresas que buscam centralizar e otimizar suas operações de monitorização na nuvem.

Enquanto a "AWS IoT" é amplamente reconhecida pela sua extensa gama de serviços e robustez, a "Senslive" destaca-se pela sua abordagem simplificada e adaptável, ideal para organizações que necessitam de uma solução ágil e de fácil implementação para gestão de dispositivos e dados de monitorização.

Ambas as plataformas oferecem vantagens significativas para os seus utilizadores, sendo a escolha entre elas influenciada pelas necessidades específicas de integração, escalabilidade e segurança de cada organização.

## 2. Dispositivo GPS: Teltonika RUT955

O “Teltonika RUT955” [4] é um *router* 4G *Long Term Evolution* (LTE) industrial concebido para fornecer conectividade fiável em ambientes robustos e remotos. Desenvolvido pela “Teltonika”, uma empresa reconhecida pelas suas soluções de conectividade, o “RUT955” é uma escolha popular para aplicações industriais, comerciais e residenciais que exigem conectividade estável e segura.

Este *router* oferece uma variedade de recursos avançados, incluindo suporte para redes 4G LTE, *Wi-Fi*, *Ethernet* e *Virtual Private Networks* (VPN). Foi projetado para satisfazer as necessidades de comunicação de dados em locais onde a conectividade é limitada ou não fiável.

Além disso, o “RUT955” possui uma interface de utilizador intuitiva e uma ampla gama de configurações que permitem a personalização para atender às necessidades específicas do utilizador. Seja para aplicações de telemetria, monitorização remota, vigilância por vídeo, acesso à Internet em locais remotos ou *backup* de rede, o “Teltonika RUT955” oferece uma solução robusta e fiável.

No âmbito do projeto realizado que envolveu o dispositivo “GL-X300B-GPS”, o objetivo principal era conectar dispositivos “Nidus” ao dispositivo GPS para obter as coordenadas geográficas, bem como os endereços IP e MAC de cada dispositivo “Nidus”. O dispositivo “GL-X300B-GPS” demonstrou ser altamente eficaz neste contexto, oferecendo todas as funcionalidades necessárias para cumprir estes requisitos.

Por outro lado, o “Teltonika RUT955”, apesar de ser um dispositivo robusto e versátil, apresenta algumas limitações específicas face à finalidade do projeto. Embora o “RUT955” também suporte GPS e ofereça uma ampla gama de interfaces, como portas RS232, RS485, Ethernet e USB, este é projetado principalmente para aplicações industriais complexas que requerem diversas opções de conectividade e funcionalidades adicionais, como suporte dual SIM, entradas e saídas digitais, e capacidades avançadas de VPN e firewall.

Essas funcionalidades avançadas, embora valiosas em muitos cenários, não são essenciais para a tarefa específica de rastreamento e obtenção de dados dos dispositivos “Nidus”. Além disso, o foco do “Teltonika RUT955” em ambientes industriais revelou adicionar uma camada de complexidade

desnecessária para este projeto, onde a simplicidade e a eficiência na obtenção de dados GPS, IP e MAC são cruciais.

O “GL-X300B-GPS”, em contrapartida, proporciona uma solução mais direcionada e otimizada para este tipo de aplicação. Este dispositivo oferece capacidades de rastreamento GPS precisas, bem como uma gestão eficaz dos dispositivos conectados através da plataforma “GoodCloud”. Esta plataforma facilita a monitorização em tempo real e a gestão remota dos dispositivos "Nidus", permitindo a obtenção de coordenadas geográficas e informações de rede de forma integrada e eficiente.

A escolha do “GL-X300B-GPS” para este projeto foi, portanto, motivada pela sua capacidade de fornecer exatamente as funcionalidades necessárias para rastrear e gerir os dispositivos "Nidus" de maneira simples e eficaz. A sua integração com plataformas de gestão como “GoodCloud” e a facilidade de obtenção de dados específicos tornaram-no a escolha ideal, superando as limitações do “Teltonika RUT955” para esta aplicação específica.

### **3. CTDDiscover: Advanced IP Scanner**

O “Advanced IP Scanner”<sup>2</sup> é uma ferramenta destinada à exploração e análise de redes locais. Projetado para ajudar administradores de redes e utilizadores avançados a gerir os seus ambientes de rede, o “Advanced IP Scanner” utiliza o protocolo UDP [5] para descobrir dispositivos conectados de forma rápida. Este software permite a identificação de computadores, *routers* e outros dispositivos com endereços IP e MAC, oferecendo funcionalidades adicionais como o acesso remoto através de RDP e outras interfaces. Com a capacidade de exportar resultados de análise para formatos como CSV e XML, o “Advanced IP Scanner” é uma solução versátil e eficiente para a monitorização e gestão de redes locais. Disponível para sistemas Windows, esta ferramenta destaca-se pela sua facilidade de uso e eficiência na deteção e análise de dispositivos de rede.

O “CTDiscover” e o “Advanced IP Scanner” são duas ferramentas distintas, embora partilhem algumas funcionalidades semelhantes. O “CTDiscover” é uma aplicação projetada especificamente para gerir e controlar dispositivos "Nidus" conectados à rede, permitindo operações detalhadas de configuração e controlo direto. Utiliza o protocolo UDP para descobrir dispositivos na rede e possibilita ações como ligar e desligar entradas e saídas do dispositivo, alterar definições como o endereço IP, adicionar ou remover sensores, e modificar a localização dos dispositivos.

---

<sup>2</sup> <https://www.advanced-ip-scanner.com>

Enquanto o “CTDiscover” é específico para dispositivos "Nidus" e oferece funcionalidades detalhadas de configuração e controlo, o “Advanced IP Scanner” é uma ferramenta mais genérica, destinada à descoberta e análise de dispositivos na rede. Ambas as ferramentas utilizam o protocolo UDP para a descoberta de dispositivos, mas diferem significativamente no seu alcance e nas funcionalidades específicas que oferecem. O “CTDiscover” é ideal para a gestão pormenorizada de dispositivos "Nidus", enquanto o “Advanced IP Scanner” é excelente para a rápida deteção e análise de uma variedade de dispositivos conectados à rede.

Embora o “Advanced IP Scanner” seja uma ferramenta poderosa e versátil para a exploração e análise de redes locais, apresenta algumas limitações. Uma das principais limitações é a sua disponibilidade apenas para sistemas operativos Windows, o que restringe a sua utilização em ambientes que utilizam outros sistemas operativos, como macOS ou Linux.

Além disso, enquanto o “Advanced IP Scanner” é excelente para a descoberta rápida de dispositivos e a identificação básica de endereços IP e MAC, não oferece funcionalidades avançadas de gestão e configuração detalhada dos dispositivos, como as oferecidas pelo “CTDiscover”. Por exemplo, não é possível utilizar o “Advanced IP Scanner” para ligar ou desligar entradas e saídas de dispositivos, alterar configurações específicas, adicionar ou remover sensores, ou modificar a localização dos dispositivos.

Outra limitação é a falta de integração direta com sistemas de gestão de dispositivos ou plataformas de monitorização em tempo real. Embora o “Advanced IP Scanner” possa exportar resultados de análise para formatos como CSV e XML, não oferece uma integração direta e contínua com sistemas mais complexos de gestão e monitorização.

Finalmente, o “Advanced IP Scanner”, apesar de permitir o acesso remoto através de RDP e outros protocolos, não oferece um nível profundo de controlo e configuração remota que algumas aplicações empresariais podem necessitar. Ele é mais adequado para a deteção inicial e análise básica de dispositivos na rede, mas não substitui ferramentas mais especializadas para a gestão e configuração detalhada de dispositivos de rede.

## 3. Trabalho desenvolvido

Durante o estágio na “CapTemp”, o papel de desenvolvedor de software foi desempenhado, responsável por conceber e implementar os diversos projetos mencionados anteriormente. Este relatório tem como propósito detalhar as atividades realizadas, salientando a aplicação dos conhecimentos adquiridos ao longo do percurso académico. Ao longo desse processo, uma variedade de tecnologias foi utilizada, tais como “Vue 3”, “Laravel”, “PostgreSQL”, “Flutter” (“Dart”) e *Bash scripting*. A seleção destas ferramentas visou garantir a eficiência e a escalabilidade dos projetos, bem como promover a integração entre elas, elemento essencial para o êxito das iniciativas desenvolvidas.

### 3.1. Ferramentas Utilizadas

Nesta secção, serão apresentadas e descritas as principais ferramentas que foram utilizadas ao longo dos projetos desenvolvidos durante o estágio. O objetivo é proporcionar uma visão detalhada das tecnologias e ambientes de desenvolvimento que facilitaram a implementação e gestão das tarefas, destacando as suas funcionalidades e benefícios no contexto dos projetos realizados.

#### 3.1.1. Laragon

Foi utilizada a ferramenta “Laragon”<sup>3</sup> para hospedar o *backend* e o *frontend* da aplicação. “Laragon” é um ambiente de desenvolvimento local que suporta diversas pilhas de desenvolvimento, incluindo PHP, “Node.js” e “Python”. Oferece um servidor *web* rápido e configurável, com suporte para “Apache” e “Nginx”, além de configurar automaticamente domínios locais para facilitar o acesso e a gestão de múltiplos projetos.

Com um painel de controle intuitivo, “Laragon” permitiu a fácil gestão de serviços como servidores *web* e bases de dados, integrando-se com “PostgreSQL”. A sua rápida instalação e portabilidade garantiram um ambiente de desenvolvimento consistente e eficiente, tornando-o uma escolha ideal para o desenvolvimento de aplicações *web* durante o estágio.

#### 3.1.2. pgAdmin

Foi utilizada a ferramenta “pgAdmin”<sup>4</sup> para a gestão e administração da base de dados “PostgreSQL” da aplicação. O “pgAdmin” é uma ferramenta de administração de código aberto,

---

<sup>3</sup> <https://laragon.org>

<sup>4</sup> <https://www.pgadmin.org>

amplamente reconhecida pela sua interface intuitiva e funcionalidades avançadas, que facilitam a gestão de bases de dados “PostgreSQL”.

O “pgAdmin” permitiu a criação, modificação e eliminação de objetos da base de dados, como tabelas, índices e funções, oferecendo uma visualização clara da estrutura da base de dados. As suas ferramentas de consulta avançadas possibilitaram a execução de *scripts* SQL, a análise e a optimização de consultas, bem como a visualização eficiente dos dados. Além disso, o “pgAdmin” disponibilizou recursos de monitorização e diagnósticos, ajudando a garantir o desempenho e a integridade da base de dados durante o desenvolvimento e a operação da aplicação.

### 3.1.3. Postman

Foi utilizada a ferramenta “Postman”<sup>5</sup> para testar e desenvolver APIs (*Application Programming Interface*). O “Postman” é uma ferramenta poderosa e intuitiva que facilita a criação, envio e análise de pedidos HTTP (*Hypertext Transfer Protocol*), sendo amplamente utilizada para testar e documentar APIs.

O “Postman” permitiu a construção de pedidos HTTP com diversos métodos (GET, POST, PUT, DELETE, entre outros) e a adição de parâmetros, cabeçalhos e corpos de mensagens. A ferramenta ofereceu uma interface gráfica que facilitava a visualização das respostas dos servidores, incluindo códigos de status, cabeçalhos de resposta e corpos de resposta em diversos formatos (JSON, XML, HTML). Além disso, o Postman permitiu a criação de coleções de pedidos, que podiam ser organizadas e reutilizadas, facilitando a automação e a documentação dos testes de API. Com funcionalidades avançadas, como testes automatizados, o “Postman” revelou ser uma ferramenta essencial para garantir a qualidade e a funcionalidade das APIs desenvolvidas durante o estágio.

### 3.1.4. Hercules

Foi utilizada a ferramenta “Hercules (HW-Group)”<sup>6</sup> para testes de comunicação em rede. O “Hercules” é uma ferramenta de software que oferece suporte abrangente para comunicação TCP/IP, UDP e serial, sendo particularmente útil para testes e diagnósticos em redes de comunicação e dispositivos de hardware.

O “Hercules” permitiu o envio e receção de dados através de diversas interfaces de comunicação, facilitando a análise e a resolução de problemas de conectividade. Com suporte para protocolos TCP

---

<sup>5</sup> <https://www.postman.com>

<sup>6</sup> <https://www.hw-group.com/software/hercules-setup-utility>

e UDP, foi possível simular conexões de rede e testar a resposta de servidores e dispositivos. Além disso, a ferramenta ofereceu funcionalidades para comunicação serial, permitindo a interação direta com dispositivos de *hardware* através de portas COM. A interface do Hercules é intuitiva e permitiu monitorizar e registar tráfego de dados, sendo uma ferramenta essencial para garantir a integridade e o funcionamento eficiente dos sistemas de comunicação desenvolvidos durante o estágio.

### 3.1.5. WinSCP

Foi utilizada a ferramenta “WinSCP”<sup>7</sup> para a transferência segura de arquivos entre máquinas locais e servidores remotos (ou dispositivos “Nidus”). WinSCP é um cliente SFTP, FTP, WebDAV e SCP de código aberto para Windows, conhecido pela sua interface intuitiva e robustez na transferência de arquivos.

O “WinSCP” permitiu a cópia segura de arquivos, facilitando a gestão de arquivos em diversos dispositivos (“Nidus” e módulo GPS). As suas funcionalidades incluem a sincronização de pastas, edição remota de arquivos e suporte para scripts e comandos, permitindo a automação de tarefas de transferência. A interface gráfica do “WinSCP” ofereceu uma visualização clara e organizada dos sistemas de arquivos locais e remotos, tornando a navegação e a transferência de arquivos simples e eficiente. Com suporte a diversos protocolos de transferência segura, o “WinSCP” assegurou a integridade e confidencialidade dos dados durante a transmissão, sendo uma ferramenta essencial para operações de manutenção durante o estágio.

## 3.2. Senslive – Plataforma Web

A plataforma “Senslive” é uma solução inovadora concebida para centralizar e unificar diversos sistemas de monitorização numa única plataforma *cloud*. Esta centralização facilita a gestão integrada e a obtenção de *insights* em tempo real, permitindo aos utilizadores gerir todos os seus sistemas a partir de um único ponto, melhorando a eficiência operacional e simplificando a supervisão de múltiplos processos.

No âmbito do estágio, foi realizada a modernização e atualização desta plataforma, com o objetivo de substituir uma versão anterior que utilizava tecnologias obsoletas e com suporte limitado. Devido a essas restrições tecnológicas, tornou-se necessária a migração para tecnologias mais modernas e amplamente suportadas. Assim, o desenvolvimento da nova versão da “Senslive” foi conduzido

---

<sup>7</sup> <https://winscp.net/>

utilizando as tecnologias “Vue” e “Laravel”, resultando numa solução mais robusta, eficiente e escalável.

Com esta nova arquitetura, a “Senslive” permite a integração de todos os sistemas de monitorização numa interface única baseada na *cloud*, facilitando a gestão centralizada e o acompanhamento em tempo real. Além de melhorar a eficiência operacional e a tomada de decisões, a plataforma oferece maior flexibilidade e segurança.

Portanto, a criação da “Senslive” foi impulsionada pela necessidade de atualizar a plataforma preexistente, adotando tecnologias contemporâneas que asseguram um melhor desempenho, escalabilidade e suporte pela comunidade de desenvolvedores. Esta modernização garante que a “Senslive” está preparada para atender às exigências atuais e futuras dos seus utilizadores, proporcionando um suporte contínuo e inovador.

### **3.2.1. Metodologia de Desenvolvimento**

A metodologia de desenvolvimento adotada para a plataforma “Senslive” foi caracterizada por uma abordagem direta e prática. A gestão do projeto foi realizada utilizando a plataforma “Git”, permitindo um controlo eficiente das versões e a colaboração contínua entre os membros da equipa de desenvolvimento.

Em termos de organização de tarefas, os programadores, tanto de *frontend* como de *backend*, receberam semanalmente diferentes atribuições. Estas tarefas foram cuidadosamente distribuídas e acompanhadas, sem a adoção de *frameworks* metodológicas rígidas como “Scrum” ou “Agile”. Para a abordagem escolhida era requerido simplicidade e eficiência, portanto optou-se por utilizar a *framework* “Kanban”, já que esta se baseia em princípios de gestão visual e fluxo contínuo, permitindo que um aumento da eficiência e a produtividade do trabalho, oferecendo também uma estrutura flexível que se revelou útil no projeto “Senslive”.

Semanalmente, os resultados e o trabalho desenvolvido eram apresentados e avaliados. Caso os requisitos das tarefas atribuídas fossem cumpridos satisfatoriamente, novas tarefas seriam então designadas. Este ciclo de atribuição, execução, avaliação e redistribuição de tarefas permitiu manter um fluxo constante de progresso e assegurar que os objetivos do projeto eram alcançados de forma eficaz e dentro dos prazos estabelecidos.

Esta metodologia prática e estruturada de revisão semanal garantiu que a equipa permanecesse focada e produtiva, permitindo um desenvolvimento contínuo e a entrega regular de funcionalidades, contribuindo para o sucesso do projeto “Senslive”.

### **3.2.2. Requisitos e Funcionalidades**

No que toca aos requisitos e funcionalidades do sistema, há diversos aspetos importantes a ter em conta. Começando pelas funções de utilizador, é crucial garantir que todas as operações básicas, como a criação, edição, visualização e eliminação de *roles*, sejam suportadas de forma robusta. Isso não só implica na implementação destas funcionalidades, mas também em assegurar que as permissões de acesso sejam aplicadas de forma precisa, de acordo com o papel de cada utilizador na plataforma.

Outro ponto fulcral diz respeito aos utilizadores em si. A capacidade de realizar todas as operações CRUD (*Create, Read, Update, Delete*) é essencial para garantir a eficácia do sistema. Para além disso, é importante distinguir entre utilizadores normais e “super” administradores, sendo estes últimos dotados de permissões e funcionalidades adicionais para gerir a plataforma de forma mais abrangente. Isso inclui o acesso a estatísticas exclusivas, que podem fornecer *insights* valiosos para a tomada de decisões.

No que toca aos relatórios automáticos, a capacidade de configurar e enviar relatórios por SMS ou Email é uma adição significativa ao sistema. Estes relatórios não só fornecem informações importantes aos utilizadores autorizados, como também podem ser personalizados de acordo com as necessidades específicas de cada empresa-cliente. A possibilidade de incluir estatísticas detalhadas sobre o desempenho dos sensores ao longo do tempo, juntamente com outras informações relevantes, aumenta ainda mais o valor desses relatórios.

Um aspeto crucial em relação às empresas é a implementação de um sistema de permissões de API personalizado para cada uma delas. Isso permite um controlo preciso sobre quais recursos e dados cada empresa pode aceder, proporcionando uma camada adicional de segurança e privacidade. Além disso, a introdução de funcionalidades para grupos e distribuidores oferece uma maneira eficaz de gerir relacionamentos entre empresas dentro da plataforma.

A gestão de dispositivos, sejam eles normais ou virtuais, é um ponto central do sistema. A capacidade de realizar operações CRUD sobre esses dispositivos, juntamente com a obtenção de estatísticas detalhadas e informações em tempo real sobre o seu estado, é fundamental para garantir o seu funcionamento eficaz. Além disso, o sistema de dispositivos virtuais oferece uma maneira flexível

de criar dispositivos personalizados com base numa combinação de sensores, ampliando ainda mais a funcionalidade do sistema.

Finalmente, a integração com o módulo GPS e a implementação de um sistema de certificados são adições importantes ao sistema. A capacidade de transmitir informações de localização em tempo real através do módulo GPS, juntamente com a validação de dispositivos e sensores por meio de certificados, contribui para a segurança e confiabilidade do sistema como um todo.

Em termos de requisitos não funcionais, o desempenho do sistema foi uma prioridade, garantindo tempos de resposta rápidos para todas as operações de criação, edição, visualização e eliminação de roles. A plataforma demonstrou ser escalável, suportando um número crescente de utilizadores e dispositivos sem comprometer o desempenho, adaptando-se facilmente a futuras expansões.

Em termos de segurança, a plataforma implementou mecanismos robustos de autenticação e autorização (como “Laravel Sanctum”), garantindo que apenas utilizadores autorizados possam aceder ao sistema e que as permissões sejam aplicadas corretamente. A encriptação de dados sensíveis transmitidos entre o cliente e o servidor assegurou a proteção da privacidade e integridade das informações.

A confiabilidade do sistema foi assegurada através de uma alta disponibilidade, minimizando o tempo de inatividade. O sistema demonstrou tolerância a falhas, continuando a funcionar corretamente mesmo em caso de falhas parciais, graças à implementação de redundância e recuperação automática.

A manutenibilidade do sistema foi facilitada por um design modular, permitindo a manutenção e atualização de componentes específicos sem afetar o restante da plataforma. A documentação abrangente e clara forneceu suporte contínuo para desenvolvedores e administradores.

A usabilidade da plataforma foi garantida através de uma interface intuitiva e fácil de usar, permitindo que utilizadores de diferentes níveis de habilidade pudessem navegar e operar o sistema sem dificuldades. A acessibilidade também foi assegurada, aderindo às diretrizes de acessibilidade para permitir que todos os utilizadores, independentemente das suas necessidades, pudessem utilizar a plataforma de forma eficaz.

A compatibilidade do sistema foi comprovada pela sua capacidade de integração com outros sistemas e serviços, incluindo API de terceiros. A plataforma foi compatível com uma ampla gama de dispositivos e navegadores, garantindo acesso fácil a partir de diferentes plataformas.

Finalmente, a eficiência no uso de recursos foi assegurada. O sistema demonstrou ser eficiente na utilização de CPU, memória e largura de banda, garantindo um desempenho otimizado e custos operacionais reduzidos. A gestão de grandes volumes de dados foi realizada de maneira eficiente, assegurando que as operações de leitura e escrita fossem rápidas e confiáveis.

Em suma, a plataforma atingiu todos os requisitos funcionais e não funcionais, proporcionando uma experiência de utilização superior, mantendo a integridade e segurança dos dados, e demonstrando ser sustentável a longo prazo.

### **3.2.3. Distribuição Temporal do Desenvolvimento**

Nesta secção é apresentada uma visão geral da distribuição temporal do desenvolvimento das diversas funcionalidades da plataforma, detalhando as etapas e os períodos específicos dedicados a cada fase do projeto. A Figura 1 ilustra a distribuição temporal do desenvolvimento das diversas funcionalidades da plataforma, conforme detalhado abaixo:

- **26 de setembro – 29 de setembro:**

- Levantamento de Requisitos e Planeamento Inicial: Início da análise detalhada dos requisitos do sistema e definição do planeamento inicial do projeto.

- **2 de outubro – 10 de novembro:**

- Implementação da Gestão de Dispositivos: Este mês seria dedicado à implementação das operações CRUD para a gestão de dispositivos físicos, incluindo a obtenção de estatísticas detalhadas e informações em tempo real sobre seu estado.

- **10 de novembro - 30 de novembro:**

- Desenvolvimento dos Relatórios Automáticos: Durante este período, seria realizado o desenvolvimento do sistema de relatórios automáticos, incluindo a configuração e o envio automatizado de relatórios por SMS ou Email.

- **1 de dezembro - 15 de janeiro:**

- Desenvolvimento da Gestão de Dispositivos Virtuais: Este mês seria dedicado à implementação do sistema de dispositivos virtuais, permitindo a criação de dispositivos personalizados com base em uma combinação de sensores.

- **16 de janeiro – 22 de janeiro:**

- Implementação dos Certificados: Este mês seria dedicado à implementação do sistema de certificados, permitindo a validação de dispositivos e sensores por meio destes certificados.

• **22 de janeiro - 28 de fevereiro:**

- Implementação da Gestão de Empresas, Utilizadores e Roles: Durante este período, seria realizada a implementação da gestão de empresas, incluindo o sistema de permissões de API personalizado para cada empresa, assim como a implementação das funções de utilizador e roles.

• **1 de março - 7 de março:**

- Integração com Módulos GPS e Testes Finais: No último mês do estágio, seria realizada a integração final com módulos GPS, juntamente com testes finais e ajustes em todas as funcionalidades desenvolvidas, garantindo a qualidade e a robustez do sistema antes da entrega final.

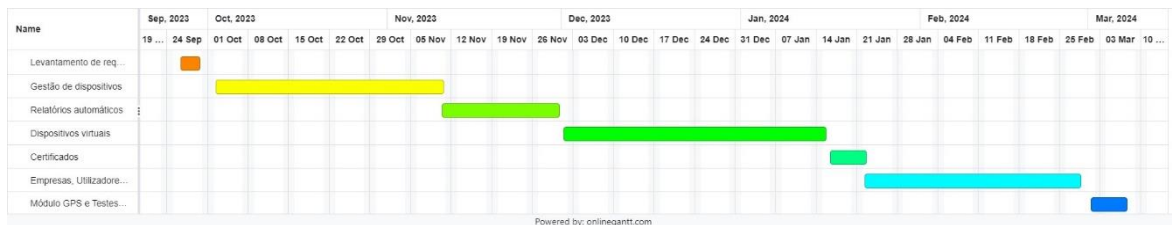


Figura 1 - Distribuição Temporal Senslive

### 3.2.4. Arquitetura e Design da Aplicação

A Figura 2 representa a arquitetura da plataforma “Senslive”, que foi desenvolvida num ambiente controlado local, e não na nuvem. No entanto, após a conclusão do projeto, está prevista a migração da plataforma para a nuvem e a sua colocação em produção.

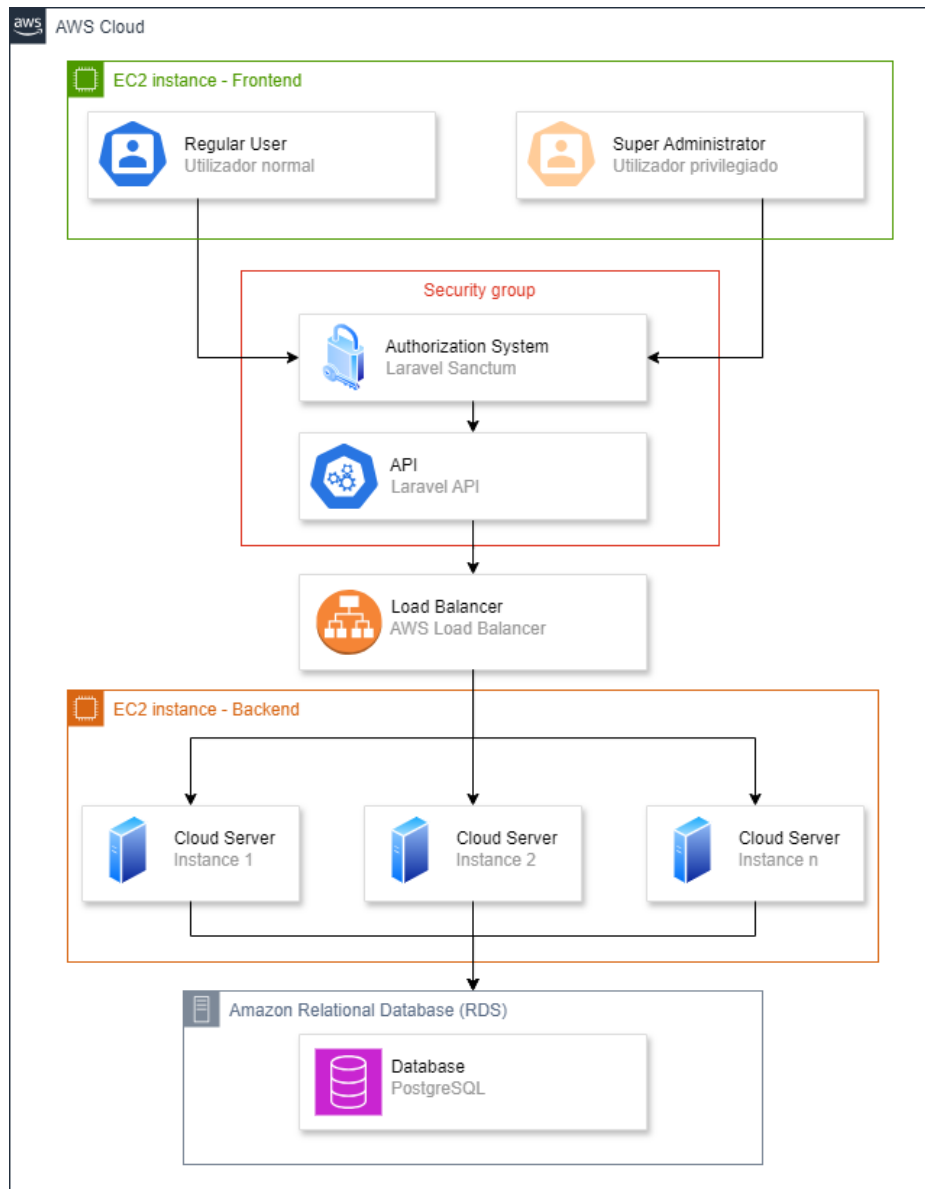


Figura 2 - Arquitetura Senslive

Relativamente ao envio de dados dos dispositivos “Nidus” para a plataforma, a Figura 3 representa a arquitetura desenvolvida para a transmissão de dados dos sensores para a plataforma “Senslive”.

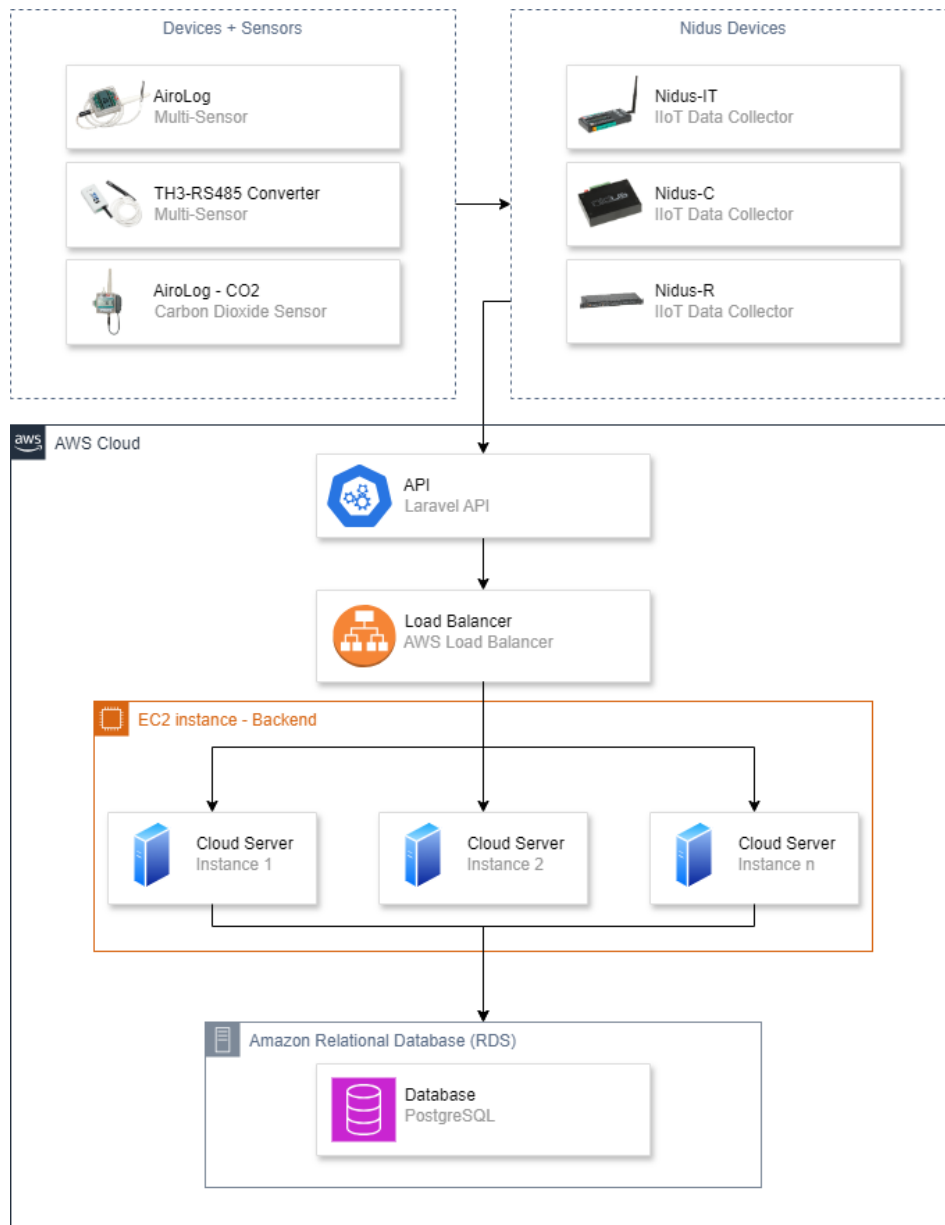


Figura 3 - Arquitetura Relação Dispositivos-Senslive

No entanto, existem casos especiais de dispositivos “Nidus” que não enviam apenas dados para a plataforma, mas também alertas via SMS, dando uso a APIs externas como “Clickatell”, “Vonage” e um servidor SMTP específico da empresa “CapTemp”. A Figura 4 representa a arquitetura desenvolvida para esta finalidade.

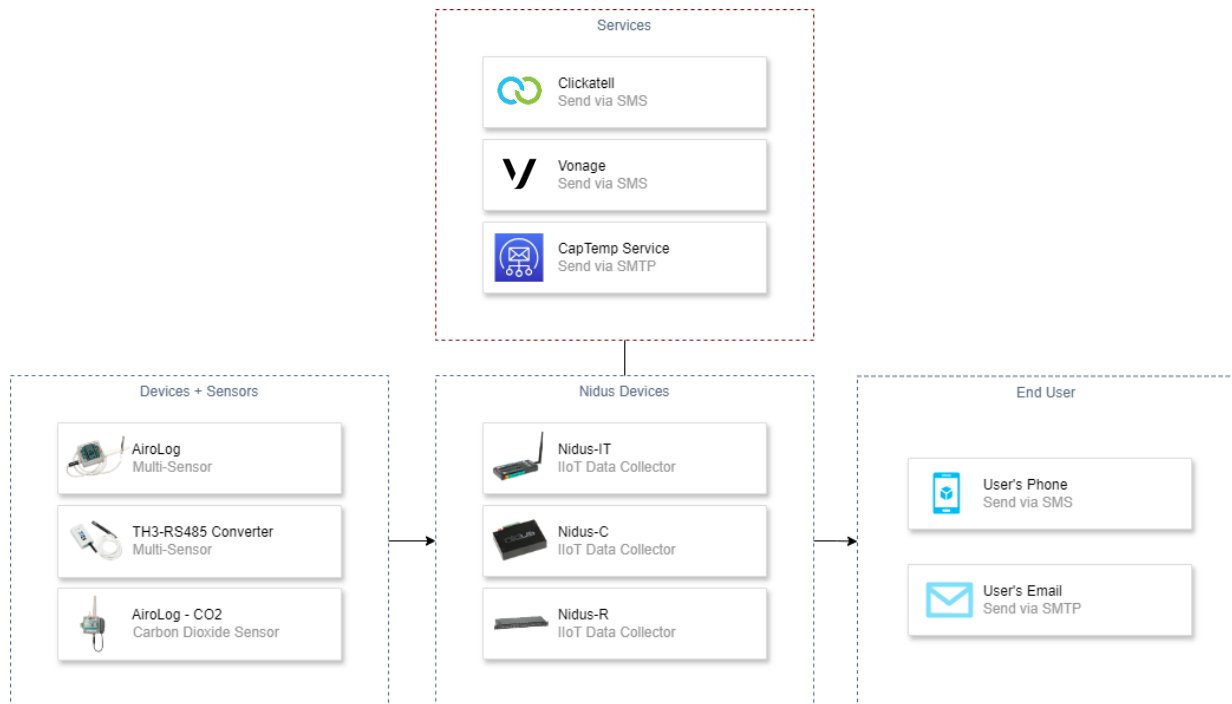


Figura 4 - Arquitetura Comunicação SMS/Email

### 3.2.5. Desenvolvimento Frontend com Vue 3

Todo o *frontend* da plataforma “Senslive” foi desenvolvido usando a *framework* de *Javascript* “Vue 3”. Esta é uma biblioteca progressiva usada na construção de interfaces modernas e interativas.

A escolha desta *framework* foi motivada pelas seguintes razões:

1. **Desempenho Aprimorado:** “Vue 3” introduziu melhorias significativas de desempenho em relação às versões anteriores, incluindo uma nova renderização reativa e uma árvore de componentes virtual mais eficiente. Isso pode resultar numa experiência de utilizador mais fluída e responsiva na plataforma.
2. **Composição e Reatividade Melhoradas:** “Vue 3” trouxe melhorias na forma como os componentes são compostos e como a reatividade é gerida. Isso pode facilitar o desenvolvimento e manutenção de interfaces complexas na plataforma, permitindo uma arquitetura modular e escalável.
3. **Ecosistema Robusto:** “Vue 3” possui um ecossistema robusto de bibliotecas e ferramentas que podem ser utilizadas para complementar o desenvolvimento da aplicação. Isso inclui bibliotecas como “Vue Router” para encaminhamento e “Vuex” para gestão de estado, entre outras.
4. **Curva de Aprendizagem Suave:** “Vue 3” é conhecida pela sua curva de aprendizagem suave, o que pode facilitar a integração de novos desenvolvedores na equipa de

desenvolvimento da plataforma. Isso pode resultar numa maior produtividade e eficiência no desenvolvimento da aplicação.

Também assim foram comparados alguns resultados com outras *frameworks* para assim poder ser garantida a melhor ferramenta a usar para o desenvolvimento da plataforma. Estes testes foram executados em diversos browsers e mostraram desempenhos semelhantes.

No geral, a escolha de “Vue 3” para o *frontend* da plataforma “Senslive” foi feita com base na sua combinação de desempenho, facilidade de uso e recursos avançados, que podem atender às necessidades de uma aplicação moderna e interativa como a “Senslive”.

### 3.2.6. Apresentação da Aplicação Resultante

A Figura 5 representa a página principal da plataforma. Após a autenticação de um utilizador, este será redirecionado para a página descrita na figura. É possível, nesta página, obter dados estatísticos sobre os últimos eventos que decorreram, o estado de dispositivos, assim como o seu local, e a localização geográfica dos mesmos.

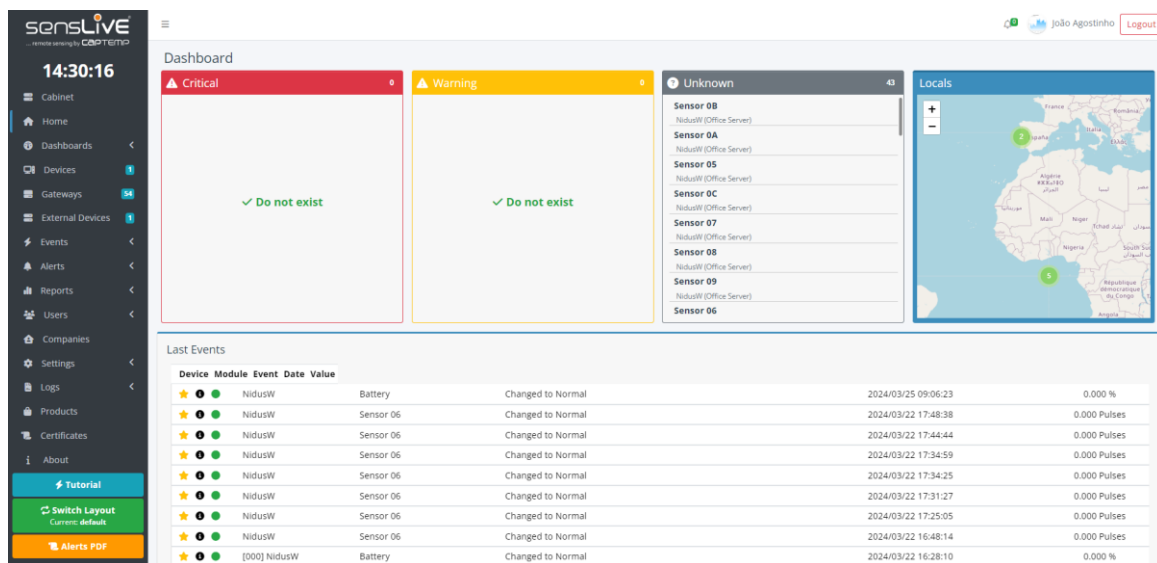


Figura 5 - Dashboard Senslive

A Figura 6 ilustra o estado atual do servidor onde está hospedada a plataforma “Senslive”. Nesta visualização, é possível observar informações como o estado atual da *cloud*, a versão da plataforma, a versão do certificado da plataforma, o certificado ISO/IEC 27001 (que atesta a segurança da informação na plataforma), os serviços atualmente ativos (“Apache”, “PostgreSQL”, “Senslive”, “Senslive NB-IoT” e “Kea Tracker”), o espaço disponível tanto no disco onde a plataforma está

armazenada quanto na base de dados. Além disso, é possível verificar o espaço alocado por cada tabela na base de dados, assim como a utilização de CPU e de RAM.

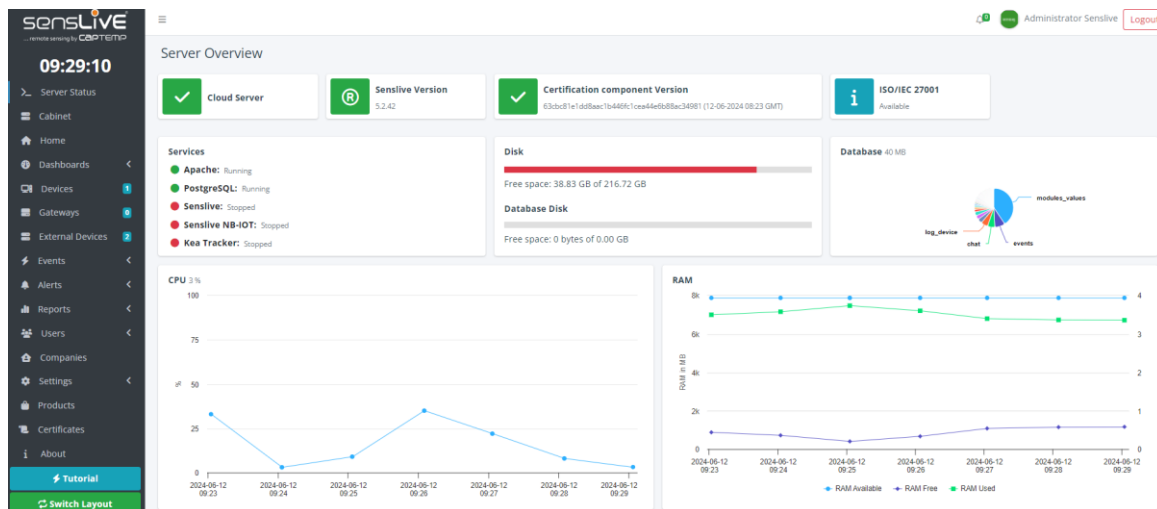


Figura 6 - Status Plataforma Sensive

Esta secção foi desenvolvida para ser compatível para sistemas operativos “Windows” e “Linux”, dando uso aos diferentes comandos para obter dados relevantes sobre o espaço disponível em disco e na base de dados, assim como utilização de CPU e de RAM.

Para realizar a verificação da certificação (integridade de todos os certificados), no sistema operativo “Windows”, foi utilizado o método de encriptação SHA1 [11], através da função nativa "hash\_file" do PHP, ilustrado no Bloco de Código 1. Esta função devolve um valor *hash* de 160 bits (20 bytes) correspondente ao conteúdo do ficheiro especificado como parâmetro de entrada. No caso em questão, foram processados todos os ficheiros da diretoria da plataforma.

```
foreach ($files as $file) {
    $hashOutput = hash_file('sha1', $file);
    $hexsum .= $hashOutput;
}
```

Bloco de Código 1 - Encriptação SHA1 de cada ficheiro

Já no sistema operativo “Linux” foi utilizado o comando nativo “sha1sum” ilustrado no Bloco de Código 2.

```
$hashes_output = rtrim(shell_exec("sha1sum [... ficheiros]"));
```

Bloco de Código 2 - Encriptação SHA1 de cada ficheiro em Linux

Para obter o espaço total e o espaço disponível em disco, foram utilizadas as funções nativas do PHP "disk\_total\_space" e "disk\_free\_space", ilustradas no Bloco de Código 3. A função "disk\_total\_space" devolve o valor total de bytes do sistema de ficheiros ou da partição de disco da diretoria especificada, que, neste caso, é a diretoria armazenada na variável "storage\_path". Por sua vez, a função "disk\_free\_space" oferece funcionalidades semelhantes à função "disk\_total\_space", mas devolve o valor total de bytes disponíveis no sistema de ficheiros ou partição de disco da diretoria especificada.

```
$serverStatus["disk"]["diskStorage"] = disk_total_space($storage_path);
$serverStatus["disk"]["diskFree"] = disk_free_space($storage_path);
```

**Bloco de Código 3 - Obtenção de Espaço Total e Disponível em Disco**

De forma a obter os valores da utilização de CPU e de RAM, foi também tido em conta o sistema operativo da máquina onde o servidor da plataforma "Senslive" se encontrava instalada. Assim, para os ambientes "Linux" e "Windows", revelou-se necessário tomar diversas abordagens.

Para obter dados relativos à utilização de CPU, foi implementada somente a função "getStatusCPU" para ambiente "Windows" ilustrada no Bloco de Código 4.

```
public function getStatusCPU(Request $request)
{
    $serverStatus = [];
    $cpuRaw = shell_exec("wmic cpu get loadpercentage");
    $cpuTrimmed = trim($cpuRaw);
    $cpuUsage = filter_var($cpuTrimmed, FILTER_SANITIZE_NUMBER_INT);
    $serverStatus['load'] = [intval($cpuUsage), 0, 0];
    return response()->json($serverStatus);
}
```

**Bloco de Código 4 - Utilização de CPU em Ambiente Windows**

De forma a obter o valor total de memória física do sistema (RAM) em "Windows", foi utilizado o comando "wmic computersystem get totalphysicalmemory", ilustrada no Bloco de Código 5, que consulta o sistema WMI (*Windows Management Instrumentation*), retornando o resultado final em bytes.

```
// Total RAM in system
exec('wmic computersystem get totalphysicalmemory', $ramTotal);
```

**Bloco de Código 5 - Memória Física Total em Ambiente Windows**

Para obter a memória física disponível e efetuar a conversão para *megabytes* foi utilizado o comando “wmic OS get TotalVisibleMemorySize /Value” ilustrado no Bloco de Código 6.

```
exec('wmic OS get TotalVisibleMemorySize /Value', $output);
$totalVisibleMemorySizeParts = explode('=', $output[2]);
$availablePhysicalMemory = $totalVisibleMemorySizeParts[1];
$serverStatus["ram"]["available"] = strval(intval($availablePhysicalMemory) *
1024);
```

#### Bloco de Código 6 - Memória Física Disponível em Ambiente Windows

Para obter a memória física livre e efetuar a conversão para *megabytes* foi utilizado o comando “wmic OS get FreePhysicalMemory /Value” ilustrado no Bloco de Código 7.

```
exec('wmic OS get FreePhysicalMemory /Value', $freePhysicalMemory);
$freePhysicalMemoryParts = explode('=', $freePhysicalMemory[2]);
$serverStatus["ram"]["free"] = $freePhysicalMemoryParts[1] * 1024;
```

#### Bloco de Código 7 - Memória Física Livre em Ambiente Windows

Finalmente, para obter a memória física utilizada, foi realizado o cálculo da diferença da memória disponível com a memória livre, ilustrado no Bloco de Código 8.

```
$serverStatus["ram"]["used"] = strval(intval($serverStatus["ram"]["available"]) -
intval($serverStatus["ram"]["free"]));
```

#### Bloco de Código 8 - Memória Física Utilizada em Ambiente Windows

Em ambiente “Linux”, é possível obter todos os mesmos dados obtidos em ambiente “Windows” utilizando o comando “free -b | grep Mem | sed -e's/ \*/ /g' | tr -d '\n'” ilustrado no Bloco de Código 9.

```
$ramRaw = shell_exec("free -b | grep Mem | sed -e's/ */ /g' | tr -d '\n' ");
$ramParts = explode(" ", $ramRaw);
$serverStatus["ram"]["total"] = $ramParts[1];
$serverStatus["ram"]["used"] = $ramParts[2];
$serverStatus["ram"]["free"] = $ramParts[3];
$serverStatus["ram"]["shared"] = $ramParts[4];
$serverStatus["ram"]["buff/cache"] = $ramParts[5];
$serverStatus["ram"]["available"] = $ramParts[6];
```

#### Bloco de Código 9 - Memória Física em Ambiente Linux

O comando completo filtra e formata a linha que contém as informações sobre a memória física a partir da saída do comando “free -b”, removendo espaços extras e novas linhas, resultando em uma única linha com a informação formatada de forma compacta.

Após passar por cada etapa do comando, a saída final pode ser algo o exemplo ilustrado no Bloco de Código 10.

```
Mem: 16384000 10485760 4096000 102400 1638400 8192000
```

#### Bloco de Código 10 - Resultado do Comando "free -b" Após Formatação em Ambiente Linux

A Figura 7 e Figura 8 representam as *roles*/papéis da plataforma da empresa do utilizador autenticado. As *roles* foram feitas de modo que cada empresa tenha as suas, não podendo ser alteradas por clientes de outras empresas.

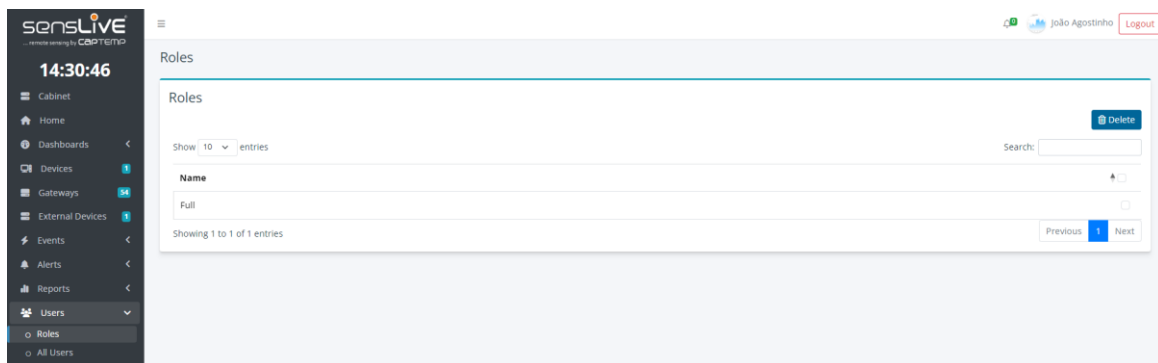


Figura 7 - Página de *Roles* Senslive

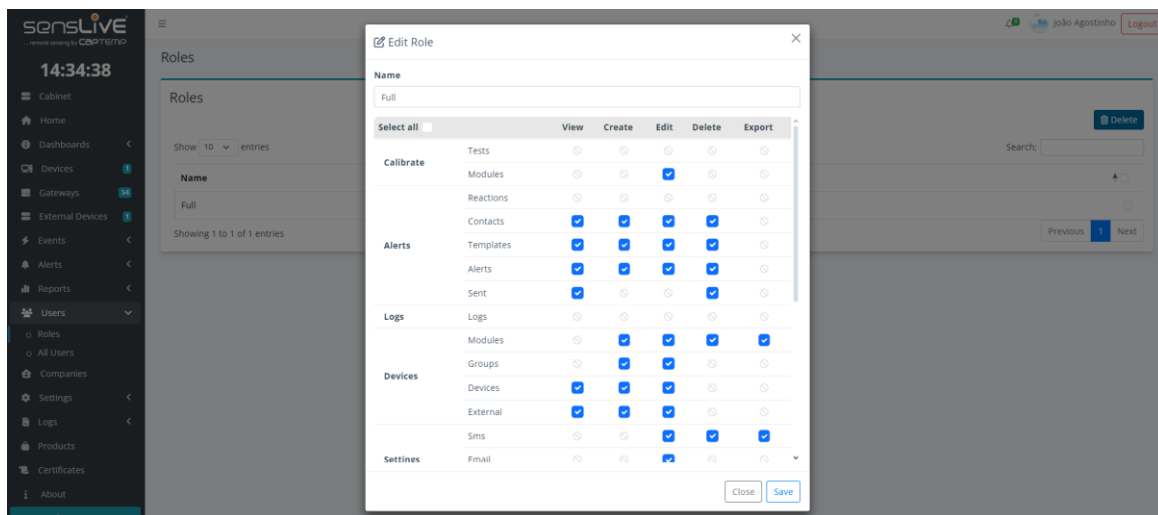


Figura 8 - Formulário de edição de uma *role*

A Figura 9 representa os utilizadores da plataforma da empresa do utilizador autenticado. Os utilizadores denotados a cor preta (Super Administradores) são representados como utilizadores superiores ao utilizador autenticado na plataforma. Desta forma, estes utilizadores superiores não podem ser alterados por utilizadores com permissões inferiores. De igual forma, o utilizador autenticado também é representado a cor preta, não só por razões estéticas, mas para encaminhar o

utilizador a utilizar a página correta para a atualização dos seus dados, que se encontra na sua página de perfil.

Nesta página também é possível ver dados dos utilizadores como o seu cargo e a última vez que acederam à plataforma.

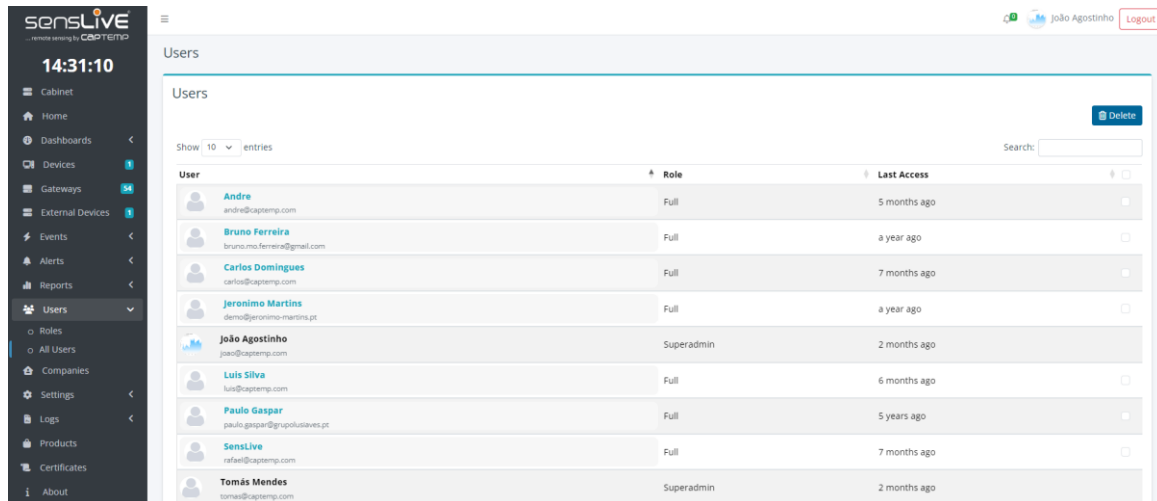


Figura 9 - Utilizadores Senslive

A Figura 10 representa os relatórios automáticos da empresa do utilizador autenticado. Um utilizador autorizado pode criar relatórios automáticos, sendo estes responsáveis por enviar dados sobre sensores selecionados aos utilizadores selecionados. Estes relatórios são automaticamente enviados em intervalos de tempo definidos, podendo ser estes visualizados nos formatos “Gráfico”, “Lista”, “Gráfico e Lista” e “Consumos”.

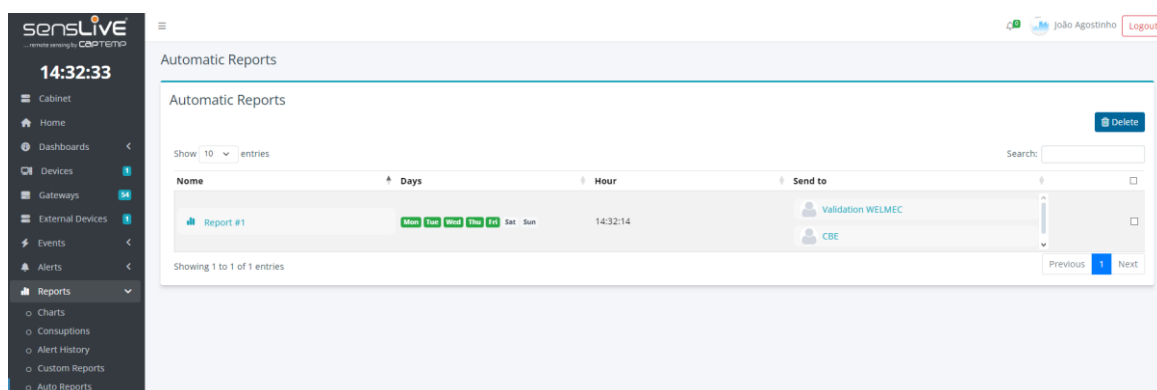


Figura 10 - Relatórios Automáticos Senslive

A Figura 11 representa as empresas visíveis ao utilizador autenticado. Estas empresas são apenas visíveis ao super administrador da empresa e podem ser categorizadas da seguinte forma: “Grupos”,

“Distribuidores” e “Clientes”, sendo assim, a visibilidade das empresas na lista alterada conforme o tipo de cada empresa.

Suponhamos que a Empresa A (empresa do utilizador autenticado) vende artigos à Empresa B. Nesta situação, o super administrador autenticado da Empresa A poderá visualizar informações e alterar definições da Empresa B. No entanto, o contrário não será possível, ou seja, a Empresa B não conseguirá aceder às informações da Empresa A.

Agora, consideremos um Grupo 1 (um grupo caracterizado por um conjunto de empresas) criado pela Empresa B, do qual faz parte a Empresa C. Apesar de a Empresa C não ser cliente da Empresa B, a sua pertença ao grupo permite que a Empresa B aceda aos dados da Empresa C.

Por fim, suponhamos a existência de uma Empresa D, que atua como distribuidora e na qual a Empresa C instalou recursos. Neste caso, a Empresa D poderá aceder aos dados da Empresa C, dado que foi a própria Empresa C que instalou os referidos recursos.

Name	Reseller	Credits	Actions
123tesjo22	SensLive Portal	0 emails	[Add] [Edit] [Delete] [Refresh]
123tesjoao2	SensLive Portal	0 emails	[Add] [Edit] [Delete] [Refresh]
123test	SensLive Portal	0 emails	[Add] [Edit] [Delete] [Refresh]
123test12joao	SensLive Portal	0 emails	[Add] [Edit] [Delete] [Refresh]
AFFAN Innovative Structures	Vacker Global	5098 emails	[Add] [Edit] [Delete] [Refresh]
Armando Portela Lda	SensLive Portal	100 emails	[Add] [Edit] [Delete] [Refresh]
ASM Industries	SensLive Portal	96 emails	[Add] [Edit] [Delete] [Refresh]
Avipronto	SensLive Portal	100 emails	[Add] [Edit] [Delete] [Refresh]
Avisabor	SensLive Portal	100 emails	[Add] [Edit] [Delete] [Refresh]
Bitcliq	SensLive Portal	0 emails	[Add] [Edit] [Delete] [Refresh]

Figura 11 - Empresas Senslive

As Figura 12 e Figura 13 representam os dispositivos físicos da empresa do utilizador autenticado. Nesta secção, caso o utilizador autenticado seja um super administrador da empresa, este poderá, então, aceder a todos os dispositivos da empresa, o que não é possível sendo um utilizador regular. Um utilizador regular poderá apenas aceder aos dispositivos a que pertencem ao seu local, ou seja, se o

utilizador pertencesse ao local “Leiria”, então, somente os dispositivos da empresa no local “Leiria” seriam apresentados na lista.

Name	Mac Address	Local	Last contact	Interval	Status	Status	Events
[000] NidusW	00:00:00:00:FE	Office Server	a year ago	30 seconds	Unknown	Normal	
NidusW	00:00:00:00:FF	Office Server	a year ago	30 seconds	Unknown	Normal	
-	192.168.1.1	Bruno Ferreira	-	30 minutes	Unknown	Normal	
0 Energia Nidus-C	00:C0:33:FC:6F:5A	Office Server	a year ago	15 minutes	Unknown	Normal	
0 Joao Nidus	90:38:0C:A9:33:9F	Office Server	a year ago	3 minutes	Unknown	Normal	
0- Nidus PC	70:B3:D5:A9:E7:74	Office Server	a year ago	1 minute	Unknown	Normal	
0-ENTGREEN	70:B3:D5:A9:E4:B9	Office Server	a year ago	15 minutes	Unknown	Normal	
00 EC Nidus-IT joao	70:B3:D5:A9:E0:8C	Office Server	a year ago	15 minutes	Unknown	Normal	
00- Nidus joao	70:B3:D5:A9:E7:75	Office Server	a year ago	2 minutes	Unknown	Normal	
00- Nidus Test	C4:DE:E2:CD:B5:AA	Office Server	8 months ago	15 minutes	Unknown	Normal	

Figura 12 - Gateways Senslive

Nome	Endereço Mac	Último contacto
[000] NidusW	00:00:00:00:FE	a year ago
Local	Endereço IP	Intervalo envio de dados
Office Server	192.168.1.201	30 seconds
Versão	Máscara rede	Alertas do dispositivo
230.509.041	255.255.255.0	
Versão Xml	Gateway	
230.509.001	192.168.1.254	

Figura 13 - Dados Dispositivo Senslive

Como a Figura 14 demonstra, é possível visualizar os dados de cada sensor em formato de gráfico. Esta funcionalidade permite uma análise visual e intuitiva dos dados coletados pelos sensores, facilitando a identificação de padrões, tendências e possíveis anomalias. Ao apresentar os dados em gráficos, os utilizadores podem monitorar de forma eficaz e em tempo real as variáveis medidas pelos sensores, melhorando a capacidade de tomada de decisões informadas e a resposta rápida a eventos críticos.

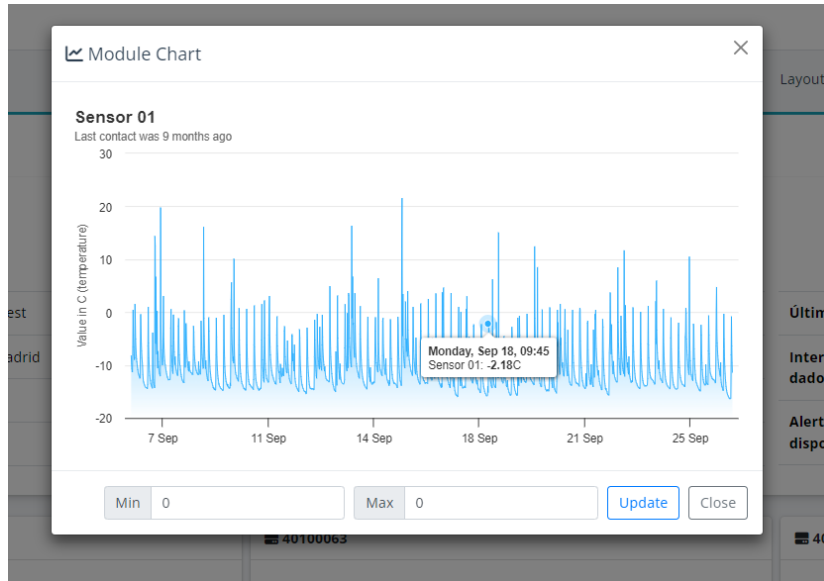


Figura 14 - Gráfico de Sensor

A Figura 15 e a Figura 16 ilustram os dispositivos virtuais da empresa associados ao utilizador autenticado. Um dispositivo virtual é uma representação agregada de vários dispositivos físicos reais. Por exemplo, no caso de haver três sensores de temperatura numa sala, pode ser criado um dispositivo virtual denominado "Termómetro Sala". Este dispositivo virtual coleta os valores dos três sensores e calcula um resultado conforme uma fórmula definida durante a sua criação. Por exemplo, o valor final do dispositivo virtual pode ser a média dos valores dos três sensores. Esta funcionalidade permite ao utilizador obter uma visão consolidada e simplificada dos dados, facilitando a análise e o monitoramento das condições ambientais de forma mais eficiente e intuitiva.

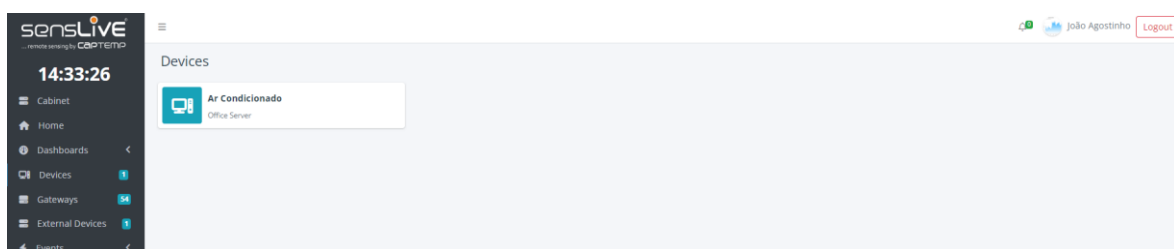


Figura 15 - Dispositivos Virtuais Senslive

The screenshot shows the Senslive web interface. On the left is a dark sidebar with navigation options: Cabinet, Home, Dashboards, Devices, Gateways, External Devices, Events, Alerts, Reports, Users, Companies, Settings, Logs, Products, Certificates, and About. The main content area is titled 'Devices' and shows a section for 'Ar Condicionado'. Below this, there's a 'Definições' section with a table:

Local	Office Server	Total Sensors	2
-------	---------------	---------------	---

Below the definitions is a 'Sensors' section with a table of sensor data:

Name	Serial number	Last value	Minimum	Maximum	Last contact	Status	Alerts	Graph	Minimum	Maximum
Sensor 0A	-	0.000	0.000	0.000	a year ago	Unkown	Normal		0.000	0.000
Sensor 0B	-	0.000	0.000	0.000	a year ago	Unkown	Normal		0.000	0.000

The interface also includes a search bar, a 'Delete' button, and pagination controls at the bottom of the sensors table.

Figura 16 - Dados Dispositivo Virtual Senslive

A Figura 17 representa os certificados da empresa do utilizador autenticado. Essencialmente, os certificados são criados pelos utilizadores e podem ser atribuídos a dispositivos e/ou sensores. O objetivo dos certificados é confirmar que os sensores e/ou dispositivos estão em conformidade com as normas estabelecidas, assegurando o seu funcionamento correto e dentro dos parâmetros esperados.

Os certificados podem ser de vários tipos:

1. **Calibração:** Este certificado confirma que o dispositivo ou sensor foi ajustado para fornecer medições precisas.
2. **Primeira verificação:** Emitido quando um dispositivo ou sensor é utilizado pela primeira vez, garantindo que cumpre os padrões necessários.
3. **Verificação periódica:** Realizada regularmente para garantir que o dispositivo ou sensor continua a funcionar corretamente ao longo do tempo.
4. **Verificação extraordinária:** Efetuada em situações especiais, como após uma reparação ou quando há suspeita de falha no equipamento.

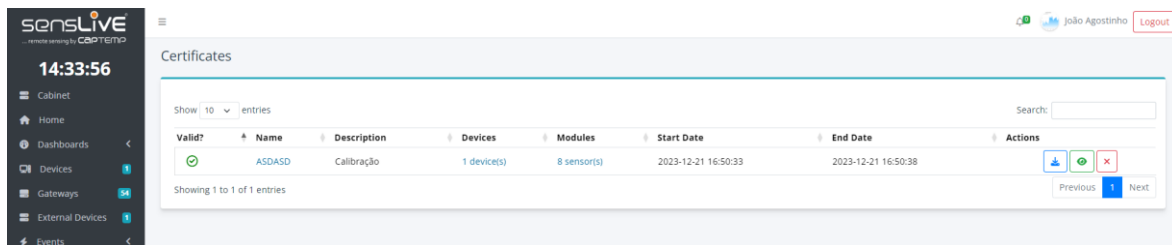
Os certificados desempenham um papel crucial na manutenção da qualidade e precisão dos dados coletados, assegurando que os dispositivos e sensores operam de acordo com os padrões exigidos. Além disso, facilitam a conformidade com normas legais e industriais, demonstrando que os equipamentos foram verificados e calibrados conforme os requisitos regulamentares.

A gestão de certificados na plataforma “Senslive” permite uma visão centralizada e organizada de todos os dispositivos e sensores certificados. Os utilizadores podem, assim, monitorizar facilmente o

estado de certificação dos seus equipamentos, planear manutenções preventivas e assegurar que todos os dispositivos críticos estão sempre operacionais e dentro dos padrões de qualidade.

A atribuição de certificados contribui também para a transparência e confiança entre a empresa e os seus clientes, garantindo que os dados fornecidos são precisos e fiáveis, baseando-se em equipamentos devidamente certificados. Esta prática é especialmente importante em setores onde a precisão dos dados é crucial, como na saúde, no ambiente e na manufatura.

Em resumo, os certificados são fundamentais para assegurar a conformidade, qualidade e integridade dos dispositivos e sensores utilizados pela empresa. A plataforma “Senslive” facilita a criação, gestão e atribuição destes certificados, garantindo que todos os equipamentos operam dentro dos parâmetros estabelecidos e proporcionando uma base sólida para a confiança e precisão dos dados monitorizados.



**Figura 17 - Certificados Senslive**

Por fim, a plataforma suporta múltiplas linguagens, tendo sido, de momento, inseridas a Portuguesa e Inglesa.

Os sistemas de internacionalização de páginas *web*, que fornecem ao utilizador final um sistema com tradução automática para a língua desejada, são cada vez mais utilizados. Contudo, isto implica um aumento da complexidade do sistema e, conseqüentemente, do espaço ocupado pelo código. No caso específico das páginas *web* com *JavaScript*, o sistema utilizado é o “I18N”, uma *framework* desenvolvida em *JavaScript*. Esta *framework* oferece várias funcionalidades além da tradução de páginas, tais como a pluralização das traduções e o carregamento dinâmico dos dicionários por “AJAX”, entre outras.

### 3.2.7. Comparação com outras *frameworks frontend*

Aspecto	Vue.js	React	Angular	Svelte
Curva de aprendizagem	Suave	Moderada	Íngreme	Moderada

<b>Popularidade</b>	Popular	Muito popular	Popular	Crescente
<b>Flexibilidade</b>	Alta	Alta	Baixa	Moderada
<b>Desempenho</b>	Bom	Bom	Bom	Excelente
<b>Ecosistema</b>	Coeso	Amplo	Amplo	Em desenvolvimento
<b>Comunidade</b>	Grande	Enorme	Grande	Em crescimento
<b>Tamanho da biblioteca</b>	Médio	Médio	Grande	Pequeno
<b>Documentação</b>	Boa	Boa	Boa	Boa
<b>Suporte a componentes</b>	Sim	Sim	Sim	Sim
<b>Gestão de estado</b>	Vuex	Redux	Angular services	Svelte store
<b>Suporte a SSR (Server-side rendering)</b>	Sim	Sim	Sim	Sim
<b>Popularidade em Portugal</b>	Alta	Moderada	Moderada	Moderada

Tabela 1 - Comparação *frameworks frontend*

Esta tabela proporciona uma visão abrangente das diferenças e semelhanças entre estas *frameworks* [6], [7], [8], contudo, é crucial ponderar as necessidades específicas do projeto, as competências da equipa de desenvolvimento e as preferências individuais ao selecionar a melhor opção. No caso da plataforma “Senslive”, a escolha recaiu sobre a *framework* “Vue.js”, pois esta se alinhava com as exigências do projeto, sendo estes o desenvolvimento de uma plataforma moderna, com bom desempenho, código-base e documentação intuitiva e código flexível utilizando componentes reutilizáveis, sendo assim eleita para o seu desenvolvimento.

### 3.2.8. Desenvolvimento Backend com Laravel

Todo o *backend* da plataforma “Senslive” foi desenvolvido usando a *framework* de PHP “Laravel” [9].

A escolha da linguagem PHP e da *framework* “Laravel” para o desenvolvimento do *backend* da plataforma “Senslive” foi motivada por várias razões:

1. **Ecosistema Maduro e Robusto:** PHP é uma das linguagens de programação mais populares para o desenvolvimento *web* e possui um ecossistema maduro, com uma ampla gama de bibliotecas, *frameworks* e ferramentas disponíveis. É uma *framework* moderna e poderosa, que oferece uma estrutura robusta e convenções bem definidas para o desenvolvimento de aplicações *web*.

2. **Facilidade de Aprendizagem e Uso:** PHP é conhecido por ser uma linguagem de programação de fácil aprendizagem e uso, especialmente para desenvolvedores iniciantes. Além disso, “Laravel” foi projetado com uma sintaxe expressiva e intuitiva, que facilita o desenvolvimento rápido e eficiente de aplicações *web*.

3. **Desempenho e Escalabilidade:** Embora PHP tenha sido criticado no passado por seu desempenho, as versões mais recentes da linguagem e otimizações de *framework* como as oferecidas pelo “Laravel” tornaram possível desenvolver aplicações *web* de alto desempenho e escaláveis. Fornece recursos como *cache*, sistema de *queue* e suporte a bases de dados relacionais e “NoSQL”, o que ajuda a otimizar o desempenho e a escalabilidade da aplicação.

4. **Comunidade Ativa e Suporte:** Tanto PHP quanto “Laravel” possuem comunidades ativas, com desenvolvedores que contribuem com correções de *bugs*, melhorias de desempenho e novos recursos regularmente. Isso significa que há uma ampla gama de recursos disponíveis para ajudar no desenvolvimento e solução de problemas ao trabalhar com PHP e “Laravel”.

5. **Compatibilidade e Integração:** PHP é suportado por uma grande variedade de servidores *web* e sistemas operacionais, o que facilita a implantação e hospedagem de aplicações *web*. Além disso, “Laravel” oferece integração com uma variedade de serviços e bibliotecas de terceiros, o que facilita a construção de aplicações *web* complexas e integradas.

No geral, a escolha da linguagem PHP e da *framework* “Laravel” para o desenvolvimento do *backend* da plataforma “Senslive” foi feita com base na sua combinação de facilidade de uso, desempenho, escalabilidade, suporte da comunidade e recursos avançados, que podem atender às necessidades de uma aplicação *web* moderna e robusta como a “Senslive”.

### 3.2.9. Comparação com outras *frameworks backend*

	Laravel	Express.js	Django	Ruby on Rails
<b>Curva de aprendizagem</b>	Suave	Moderada	Íngreme	Íngreme
<b>Popularidade</b>	Moderada	Muito popular	Popular	Popular

<b>Flexibilidade</b>	Alta	Alta	Moderada	Baixa
<b>Desempenho</b>	Bom	Bom	Bom	Bom
<b>Ecosistema</b>	Coeso	Amplo	Amplo	Amplo
<b>Comunidade</b>	Grande	Grande	Grande	Grande
<b>Tamanho da biblioteca</b>	Médio	Pequeno	Grande	Grande
<b>Documentação</b>	Boa	Boa	Boa	Boa
<b>ORM (Object-Relational Mapping)</b>	Eloquent ORM (integrado)	Não possui ORM integrado	Django ORM (integrado)	ActiveRecord (integrado)
<b>Linguagem de programação</b>	PHP	JavaScript (Node.js)	Python	Ruby
<b>Suporte a API Rest</b>	Sim	Sim	Sim	Sim
<b>Suporte a GraphQL</b>	Sim (via pacotes externos)	Sim (via pacotes externos)	Sim (via pacotes externos)	Sim (via pacotes externos)
<b>Suporte a middlewares</b>	Sim	Sim	Sim	Sim
<b>Suporte a autenticação</b>	Sim	Sim	Sim	Sim
<b>Popularidade em Portugal</b>	Moderada	Moderada	Moderada	Moderada

Tabela 2 - Comparação *frameworks* backend

Esta tabela proporciona uma visão abrangente das diferenças e semelhanças entre estas *frameworks*, contudo, é crucial ponderar as necessidades específicas do projeto, as competências da equipa de desenvolvimento e as preferências individuais ao selecionar a melhor opção. No caso da plataforma “Senslive”, a escolha recaiu sobre a *framework* “Laravel”, pois esta se alinhava com as exigências do projeto, sendo assim eleita para o seu desenvolvimento.

### **3.2.10. Testes e validações**

Com base nos requisitos e funcionalidades implementadas no "Senslive", foram realizados diversos testes rigorosos para garantir a robustez e a eficácia da plataforma. Primeiramente, no que diz respeito às operações CRUD de roles e utilizadores, testou-se a criação, edição, visualização e eliminação desses elementos, assegurando que as permissões de acesso fossem aplicadas de forma correta e precisa, de acordo com os diferentes níveis de utilizadores, incluindo administradores e "super" administradores. Esta distinção permitiu a cada tipo de utilizador executar as suas tarefas específicas, reforçando a gestão eficiente da plataforma.

No âmbito da gestão de relatórios automáticos, foram realizadas verificações cuidadosas na configuração e envio de relatórios por SMS e Email. Estes testes confirmaram que os relatórios continham estatísticas detalhadas e que poderiam ser personalizados conforme as necessidades específicas de cada empresa-cliente. A capacidade de incluir informações de desempenho dos sensores ao longo do tempo demonstrou-se especialmente útil para a análise contínua e a tomada de decisões informadas.

Além disso, foi testado o sistema de permissões de API personalizado para cada empresa, garantindo um controlo rigoroso sobre os recursos e dados acessíveis, oferecendo uma camada adicional de segurança e privacidade. A gestão de dispositivos, tanto físicos como virtuais, foi também um foco importante dos testes. As operações CRUD sobre esses dispositivos, juntamente com a obtenção de estatísticas detalhadas e informações em tempo real, foram verificadas para garantir a sua funcionalidade completa e eficiente. Particular atenção foi dada à criação de dispositivos virtuais a partir de múltiplos sensores, assegurando que os cálculos realizados, como a média dos valores dos sensores, fossem precisos e confiáveis.

No que concerne à integração com módulos GPS e ao sistema de certificados, foram confirmadas a capacidade de transmissão de informações de localização em tempo real e a validação de dispositivos e sensores, contribuindo para a segurança e confiabilidade geral do sistema.

Os testes de desempenho avaliaram os tempos de resposta para a operação de visualização de diversos dados, garantindo que permanecessem rápidos mesmo com o aumento do número de utilizadores e dispositivos. A escalabilidade da plataforma foi testada para assegurar que poderia suportar um número crescente de utilizadores e dispositivos sem comprometer o desempenho.

A segurança do sistema foi validada através de testes de autenticação e autorização, incluindo a implementação do "Laravel Sanctum", garantindo que apenas utilizadores autorizados pudessem

aceder ao sistema e que as permissões fossem aplicadas corretamente. A encriptação de dados sensíveis transmitidos entre o cliente e o servidor foi confirmada, assegurando a proteção da privacidade e integridade das informações.

A confiabilidade do sistema foi assegurada através de testes que demonstraram a alta disponibilidade e a capacidade de tolerância a falhas, com o sistema mantendo-se operacional mesmo em caso de falhas parciais, graças à implementação de redundância e recuperação automática.

Para garantir a manutenibilidade, o design modular da plataforma foi testado, permitindo a manutenção e atualização de componentes específicos sem afetar o restante do sistema. A documentação abrangente e clara forneceu suporte contínuo para desenvolvedores e administradores, facilitando a resolução de problemas e a implementação de novas funcionalidades.

A usabilidade da plataforma foi garantida através de uma interface intuitiva e fácil de usar, permitindo que utilizadores de diferentes níveis de habilidade navegassem e operassem o sistema sem dificuldades. A adesão às diretrizes de acessibilidade foi verificada, garantindo que todos os utilizadores, independentemente das suas necessidades, pudessem utilizar a plataforma de forma eficaz.

A compatibilidade do sistema foi testada e comprovada pela sua capacidade de integração com outros sistemas e serviços, incluindo APIs de terceiros, e pela compatibilidade com uma ampla gama de dispositivos e navegadores, assegurando acesso fácil a partir de diferentes plataformas.

Foram realizados diversos testes com vários pontos de entrada da API, utilizando uma variedade de cargas de trabalho e cenários de uso. Os testes foram executados repetidamente, e os resultados foram registados e analisados. Após um número significativo de tentativas, calculou-se a média dos resultados obtidos em cada métrica de desempenho.

Finalmente, a eficiência no uso de recursos foi assegurada. O sistema demonstrou ser eficiente na utilização de CPU, memória e largura de banda, garantindo um desempenho otimizado e custos operacionais reduzidos. A gestão de grandes volumes de dados foi realizada de maneira eficiente, assegurando que as operações de leitura e escrita fossem rápidas e confiáveis.

Com base nos testes realizados e nas funcionalidades implementadas, o nível de maturidade tecnológica da aplicação "Senslive" pode ser classificado como TRL (*Technology Readiness Level*) 7. Este nível indica que a plataforma foi testada com sucesso em um ambiente controlado que simula condições operacionais reais, demonstrando um desempenho robusto e confiável. A aplicação está

próxima de ser utilizada em produção, oferecendo uma experiência de utilização superior, mantendo a integridade e segurança dos dados, e mostrando-se sustentável a longo prazo.

### 3.2.11. Integração com PostgreSQL

Para a base de dados da plataforma Senslive, foi escolhido o “PostgreSQL” como sistema de gestão. O “PostgreSQL” é um sistema de gestão de bases de dados relacionais amplamente reconhecido pela sua robustez, desempenho e conformidade com padrões. Com uma vasta gama de funcionalidades avançadas, suporte à extensibilidade e uma comunidade ativa de desenvolvedores, o “PostgreSQL” oferece uma solução confiável e escalável para as necessidades de armazenamento de dados da plataforma Senslive.

Alguns motivos para o qual foi escolhido este sistema para a gestão da base de dados da plataforma foram:

1. **Recursos Avançados:** O “PostgreSQL” oferece uma vasta gama de funcionalidades avançadas, incluindo suporte para tipos de dados complexos como *arrays* e JSON, assim como a capacidade de extensibilidade através de funções definidas pelo utilizador e suporte para várias linguagens de programação, tais como PL/pgSQL, PL/Python e PL/Perl.

2. **Desempenho e Escalabilidade:** Reconhecido pela sua escalabilidade e capacidade de lidar com grandes volumes de dados e cargas de trabalho intensivas, o “PostgreSQL” possui otimizações de desempenho como índices eficientes, consultas paralelas e suporte a transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), garantindo confiabilidade e eficiência em ambientes de produção.

3. **Conformidade com Padrões:** O “PostgreSQL” adere aos padrões ANSI SQL, o que facilita a migração de aplicações de outros sistemas de base de dados relacionais. Além disso, suporta várias extensões SQL avançadas, oferecendo flexibilidade e compatibilidade com diversos requisitos de aplicação.

4. **Comunidade Ativa e Suporte:** Com uma comunidade ativa de desenvolvedores e utilizadores, o “PostgreSQL” beneficia de contribuições regulares, correções de erros e suporte em fóruns online e listas de discussão. Existem também recursos de documentação extensivos e tutoriais disponíveis para ajudar os desenvolvedores a aprender e usar o “PostgreSQL” de forma eficaz.

5. **Custo e Licenciamento:** O “PostgreSQL” é software de código aberto distribuído sob a licença “PostgreSQL”, permitindo o seu uso gratuito em projetos comerciais e não comerciais. Esta abordagem

de licenciamento pode reduzir os custos associados e fornecer uma solução económica para as necessidades de armazenamento de dados da plataforma “Senslive”.

No geral, a escolha do “PostgreSQL” como sistema de base de dados para a plataforma “Senslive” pode ter sido motivada pela sua robustez, desempenho, conformidade com padrões, apoio da comunidade e custo acessível.

### **3.2.12. Aplicação dos Conhecimentos Adquiridos**

No *frontend*, a utilização de “Vue 3” reflete a aplicação prática dos conceitos de componentes e reatividade. No *backend*, o uso de “Laravel” traduz a aplicação de princípios como MVC, *Eloquent* ORM e boas práticas de desenvolvimento *web*.

## **3.3. Relacionamento entre Teoria e Prática**

Nesta secção, será explorada a aplicação prática dos conhecimentos teóricos adquiridos durante o curso, focando-se na integração de tecnologias de *frontend* e *backend*. Especificamente, será discutida a implementação de interfaces com “Vue 3” e o desenvolvimento de funcionalidades do lado do servidor com “Laravel”. Este relacionamento entre teoria e prática será analisado, destacando como os princípios e técnicas aprendidos em ambiente académico foram adaptados e aplicados para resolver problemas reais e desenvolver soluções eficientes e robustas. A combinação destas tecnologias permitirá demonstrar a sinergia entre a construção de interfaces interativas e dinâmicas e a gestão eficaz de dados e lógica de negócio no servidor.

### **3.3.1. Frontend com Vue 3**

A aplicação de conceitos de “Vue 3”, como reatividade e diretivas, resultou em interfaces dinâmicas e responsivas. A utilização de “Vuex” para o gerenciamento de estado contribuiu para um fluxo de dados eficiente. A estrutura modular de “Vue 3” permitiu a criação de componentes reutilizáveis, facilitando a manutenção e a escalabilidade do código. Além disso, a implementação do sistema de encaminhamento com “Vue Router” assegurou uma navegação fluida e intuitiva entre diferentes vistas da aplicação.

A reatividade de “Vue 3” foi fundamental para a atualização automática da interface em resposta a alterações nos dados. Isso foi particularmente útil em *dashboards* de monitorização, onde a informação precisa ser atualizada em tempo real. As diretivas, por outro lado, simplificaram a manipulação do DOM, permitindo que o HTML fosse mais expressivo e menos propenso a erros.

A utilização de “Vuex” como um sistema de gerenciamento de estado centralizado assegurou que todos os componentes tivessem acesso a um estado consistente. Isso foi crucial em cenários onde múltiplos componentes precisavam compartilhar ou modificar os mesmos dados, como na gestão de dispositivos e sensores na aplicação Senslive. A abordagem de “Vuex” para gerenciar o estado de forma centralizada e previsível permitiu um melhor controle sobre o fluxo de dados e facilitou o processo de *debug* e desenvolvimento.

Além disso, a integração de plugins e bibliotecas complementares, como “Axios” para requisições HTTP, reforçou a capacidade de “Vue 3” em comunicar-se de forma eficiente com o *backend*, proporcionando uma experiência de utilizador mais robusta e interativa. A capacidade de “Vue 3” de lidar com requisições assíncronas e manipulação de dados em tempo real foi crucial para a funcionalidade da plataforma, permitindo atualizações contínuas sem a necessidade de recarregar a página.

A modularidade do “Vue 3” também possibilitou a criação de um sistema de componentes bem estruturado, onde cada componente representava uma parte específica da interface, como gráficos de monitorização, formulários de configuração e tabelas de dados. Esta abordagem modular não só melhorou a organização do código como também facilitou a colaboração entre diferentes membros da equipa de desenvolvimento, permitindo que componentes pudessem ser desenvolvidos, testados e mantidos de forma independente.

Em resumo, a implementação de “Vue 3” no desenvolvimento *frontend* da aplicação “Senslive” demonstrou como os conceitos teóricos podem ser aplicados para criar interfaces de utilizador eficazes, dinâmicas e responsivas. A combinação de reatividade, diretivas, gerenciamento de estado com “Vuex”, e a modularidade dos componentes resultou numa plataforma sólida e escalável, capaz de atender às necessidades dos utilizadores de maneira eficiente e intuitiva.

### **3.3.2. Backend com Laravel**

A implementação de rotas, controladores e modelos no “Laravel” demonstrou a aplicação direta dos conceitos do padrão arquitetural *Model-View-Controller* (MVC). Este padrão, amplamente reconhecido pela sua capacidade de organizar a lógica de negócios, apresentação e interação de uma aplicação, permitiu uma clara separação de responsabilidades, tornando o código mais organizado e fácil de manter.

A definição de rotas no “Laravel”, utilizando a sua sintaxe simples e expressiva, permitiu mapear requisições HTTP para os controladores correspondentes de maneira eficiente. Cada rota foi associada

a um controlador específico, que por sua vez continha a lógica necessária para processar a requisição e gerar a resposta apropriada. Esta abordagem garantiu que o fluxo de dados entre o cliente e o servidor fosse tratado de forma sistemática e previsível.

Os controladores em “Laravel” desempenharam um papel crucial na organização da lógica de aplicação. Cada controlador foi responsável por manipular as requisições, interagir com os modelos, e devolver as respostas adequadas ao cliente. A capacidade de utilizar *middleware* para pré-processar requisições antes que elas alcancem os controladores adicionou uma camada adicional de segurança e flexibilidade, permitindo a implementação de autenticação, autorização e outras verificações necessárias.

Os modelos em “Laravel”, representando a camada de dados da aplicação, foram fundamentais para a interação com a base de dados “PostgreSQL”. Utilizando o *Eloquent* ORM, a manipulação dos dados tornou-se intuitiva. O *Eloquent* permitiu definir relações entre tabelas, realizar consultas complexas e manipular registos da base de dados através de uma interface de alta abstração. A utilização de *Eloquent* não só simplificou a escrita de consultas SQL, mas também assegurou que as operações de banco de dados fossem realizadas de maneira eficiente e segura.

Além disso, a migração de banco de dados no “Laravel” facilitou a criação e manutenção das estruturas de tabelas no “PostgreSQL”. As migrações permitiram versionar as mudanças no esquema do banco de dados, tornando o processo de atualização e *rollback* mais controlado e menos propenso a erros. A utilização de *seeders* para a inserção de dados iniciais assegurou que o ambiente de desenvolvimento estivesse sempre preparado com os dados necessários para testes e desenvolvimento.

A utilização de *facades* e *helpers* do “Laravel” proporcionou uma forma conveniente de interagir com componentes comuns da aplicação, como cache, filas, serviços de email, entre outros. Estas ferramentas ajudaram a manter o código limpo e legível, promovendo boas práticas de desenvolvimento e facilitando a colaboração entre membros da equipa.

Em termos de segurança, o “Laravel” implementou robustos mecanismos de autenticação e autorização, utilizando “Laravel Sanctum” para a gestão de *tokens* de autenticação. Isso garantiu que apenas utilizadores autorizados pudessem aceder a recursos sensíveis, protegendo a integridade e confidencialidade dos dados.

A integração de “Laravel” com outras ferramentas e serviços externos foi facilitada pela sua arquitetura modular e suporte a pacotes. A utilização de bibliotecas de terceiros, através do “Composer”, permitiu a expansão das funcionalidades da aplicação de maneira eficiente e controlada.

Em resumo, a implementação do *backend* com “Laravel” no desenvolvimento da aplicação “Senslive” demonstrou a eficácia dos conceitos teóricos na prática. A aplicação do padrão MVC, a utilização do *Eloquent ORM* para interação com a base de dados “PostgreSQL”, e os robustos mecanismos de segurança e modularidade do “Laravel” resultaram numa solução *backend* sólida e escalável. Estes elementos combinados garantiram um desempenho confiável, uma manutenção facilitada e a capacidade de atender às necessidades dos utilizadores de maneira eficiente e segura.

### 3.4. “Nidus” – Módulo GPS

O texto que se segue fornece uma descrição detalhada do coletor de dados IIoT denominado “Nidus”, desenvolvido para captar, processar e transmitir dados de sensores e dispositivos industriais, assim como uma descrição detalhada do dispositivo “GL-X300B-GPS”, desenvolvido para captar, processar e transmitir dados geográficos relativos ao dispositivo. Esta secção explora o contexto, as funcionalidades e as principais características técnicas destes dispositivos, destacando a sua importância para a otimização e eficiência das operações industriais, assim como a sua integração entre si.

#### 3.4.1. Contexto – “Nidus”

A “Nidus” é um coletor de dados IIoT desenvolvido para capturar, processar e transmitir dados de diversos sensores e dispositivos industriais. Este dispositivo oferece conectividade com sensores que medem parâmetros como temperatura, pressão e humidade, utilizando protocolos industriais padrão como Modbus RTU. Processa os dados localmente, permitindo análise e filtragem em tempo real, e envia os dados processados de forma segura para plataformas de nuvem, sistemas SCADA e outros sistemas de gestão através de *Ethernet* ou Wi-Fi.

Além disso, a “Nidus” incorpora robustas medidas de segurança, incluindo criptografia, autenticação segura e *firewall* integrada, garantindo a proteção dos dados contra acessos não autorizados e ciberataques. A gestão remota é facilitada por uma interface amigável, que permite configuração, monitorização e atualização de *firmware* e *software* remotamente. A integração com plataformas de análise de dados e manutenção preditiva é simplificada através de APIs e SDKs, permitindo a personalização e o desenvolvimento de aplicações específicas.

Em resumo, a “Nidus” é uma ferramenta essencial para a indústria moderna, oferecendo uma solução robusta, segura e eficiente para a captura e análise de dados em tempo real, permitindo a otimização das operações industriais e melhorando a produtividade e eficiência das empresas que adotam a IIoT.

### 3.4.2. Características principais da “Nidus”

A Tabela 3 demonstra as características principais dos dispositivos “Nidus” utilizados na “CapTemp”, sendo estes as “Nidus-IT”, “Nidus-C” e “Nidus-R”.




	<b>Nidus-IT</b> 	<b>Nidus-C</b> 	<b>Nidus-R</b> 
<b>Comunicação Ethernet</b>	✓	✓	✓
<b>Suporte para sensores RS-485</b>	✓	✓	✓
<b>Suporte para sensores wireless</b>	✗	✗	✗
<b>Integração de equipamentos externos via Modbus RTU</b>	✓	✓	✓
<b>Inputs Digitais</b>	11	✗	18
<b>Outputs Digitais</b>	16	✗	6
<b>Suporte para Modem GSM com envio de alarmes SMS e controlo remoto</b>	✓	✓	✓
<b>Alarmes suportados</b>	Email, SMS, <i>web</i> , SNMP	Email, SMS, <i>web</i> , SNMP	Email, SMS, <i>web</i> , SNMP
<b>Comunicação M2M</b>	Protocolos SNMP, XML, Modbus-TCP	Protocolos SNMP, XML, Modbus-TCP	Protocolos SNMP, XML, Modbus-TCP
<b>Servidor WEB e ambiente gráfico integrado</b>	✓	✓	✓

Tabela 3 - Principais Características das Nidus

### 3.4.3. Áreas de aplicação da “Nidus”

A "Nidus" possui uma ampla gama de aplicações em diversas áreas, proporcionando monitorização e controlo de várias condições ambientais e sistemas. Algumas das áreas de aplicação incluem:

1. **Farmácias:** Monitorização e registo das condições de armazenamento dos medicamentos, garantindo o cumprimento dos padrões de segurança e qualidade.
2. **Padarias e instalações de congelação:** Monitorização das temperaturas durante a produção, facilitando a conformidade com os requisitos HACCP e garantindo a qualidade dos produtos.
3. **Armazenamento:** Monitorização de temperaturas, humidade e outras condições ambientais, com relatórios compatíveis com HACCP, assegurando condições ideais de armazenamento e conservação de produtos.
4. **Sistema SCADA:** Integração de inputs externos para monitorização e controlo de processos industriais em tempo real.
5. **Data Centers:** Monitorização de sistemas em *rack*, temperatura, humidade e outros parâmetros críticos, garantindo o funcionamento eficiente e seguro dos equipamentos de TI.
6. **Sistemas HVAC:** Monitorização dos parâmetros do sistema de aquecimento, ventilação e ar condicionado, assegurando o conforto e a eficiência energética em edifícios comerciais e industriais.
7. **Monitorização de Ar-Condicionado:** Detecção de fugas de água, controlo automático de temperatura e ambiente em espaços de TI e *data centers*.
8. **UPS / Gerador:** Monitorização remota, arranque remoto e controlo de temperatura e níveis de combustível ou bateria, garantindo a disponibilidade e fiabilidade de sistemas de energia de emergência.
9. **Sistemas de Segurança IP:** Vigilância por vídeo, controlo de acessos, e ativação automática de backups em caso de falha, garantindo a segurança e integridade de ambientes críticos.

Estas aplicações abrangentes demonstram a versatilidade e utilidade da "Nidus" em várias indústrias e ambientes, proporcionando soluções eficazes para monitorização e controlo de sistemas e condições ambientais.

### 3.4.4. Contexto – GL-X300B-GPS

O GL-X300B-GPS [10] (Figura 18) é um dispositivo de rede desenvolvido pela “GL.iNet”, uma empresa conhecida por produzir *routers* e dispositivos de rede portáteis e de alta performance. Este dispositivo, em particular, integra funcionalidades avançadas, incluindo um módulo GPS, o que o torna útil para aplicações que necessitam de rastreio e monitorização de localização, além de conectividade de rede robusta.



Figura 18 - Módulo GPS GL-X300B-GPS (adaptado de [10])

### 3.4.5. Características Principais do GL-X300B-GPS

O GL-X300B-GPS destaca-se pelas suas diversas funcionalidades e especificações técnicas, que o tornam adequado para uma ampla gama de aplicações. As suas principais características incluem:

1. **Integração GPS:** O módulo GPS integrado permite a localização precisa do dispositivo, ideal para aplicações de rastreamento em tempo real e monitorização de veículos. É útil para frotas de transporte, serviços de logística, e outros setores que necessitam de rastreamento contínuo.

2. **Conectividade LTE:** Apresenta suporte para redes LTE (4G), garantindo conectividade de internet em locais remotos ou em movimento, onde as redes Wi-Fi podem não estar disponíveis. Permite também que o dispositivo atue como um hotspot móvel, fornecendo acesso à internet para outros dispositivos através de uma rede celular.

3. **Wi-Fi Dual Band:** Apresenta suporte para Wi-Fi de 2.4GHz e 5GHz, proporcionando flexibilidade e desempenho aprimorado em diferentes ambientes de rede. Fornece, também, a possibilidade de usar o dispositivo como um *router* Wi-Fi, fornecendo conectividade de rede para dispositivos próximos.

4. **Portabilidade e Design Robusto:** Design compacto e robusto, facilitando a instalação em veículos ou uso em ambientes externos, ideal para aplicações móveis e em condições adversas.

5. **Versatilidade de Energia:** Suporte para múltiplas opções de alimentação, incluindo bateria interna, energia de veículo (12V DC), e adaptadores de energia padrão. Apresenta flexibilidade para diversas situações de uso, seja em campo, em veículos ou em instalações fixas.

6. **Segurança Avançada:** Recursos de segurança integrados, como VPN (“OpenVPN”, “WireGuard”) para conexões seguras e criptografadas. Possui proteção de *firewall* e suporte para autenticação segura, garantindo a integridade dos dados transmitidos.

7. **Interfaces de Conexão:** Fornece múltiplas portas *Ethernet* para conexão de dispositivos com fio. Suporta antenas externas, aumentando a capacidade de receção de sinal em áreas com cobertura fraca.

8. **Software e Suporte:** Possui uma interface de utilizador amigável e suporte para personalização avançada através de “OpenWRT”, uma plataforma de *firmware* de código aberto. Fornece, também, atualizações de software e suporte técnico da “GL.iNet” para resolver problemas e adicionar novas funcionalidades.

### 3.4.6. Aplicações Comuns do Módulo

1. **Gestão de Frotas:** Rastreio de veículos em tempo real, monitorização de rotas, e gestão eficiente da frota. Melhora, assim, a segurança dos veículos e otimiza a logística.

2. **Aplicações de IoT:** Utilizado em projetos de Internet das Coisas (IoT), onde a conectividade contínua e a localização precisa são essenciais. Este dispositivo facilita a monitorização remota de ativos e a gestão eficiente de dispositivos IoT distribuídos, assegurando que todos os componentes da rede se encontram continuamente conectados e localizados com precisão.

3. **Hotspot Móvel:** Proporciona conectividade à internet em áreas remotas ou durante viagens, atuando como um ponto de acesso móvel. É especialmente útil para profissionais que necessitam de uma conexão constante à internet enquanto estão em movimento.

4. **Ambientes Industriais:** Implementação em ambientes industriais para monitorização de máquinas, equipamentos, e outras aplicações críticas que requerem conectividade confiável e localização precisa.

### **3.4.7. Metodologia de desenvolvimento**

Foi desenvolvido software para utilizar as funcionalidades do módulo GPS, sem a utilização de um repositório no “Github”, no entanto, a metodologia utilizada no projeto foi “Kanban”. O objetivo primordial era criar uma solução robusta e altamente funcional para atender às demandas específicas do contexto de uso. Este software foi concebido para facilitar a integração e a comunicação eficiente com o módulo GPS, visando fornecer informações geográficas precisas e atualizadas.

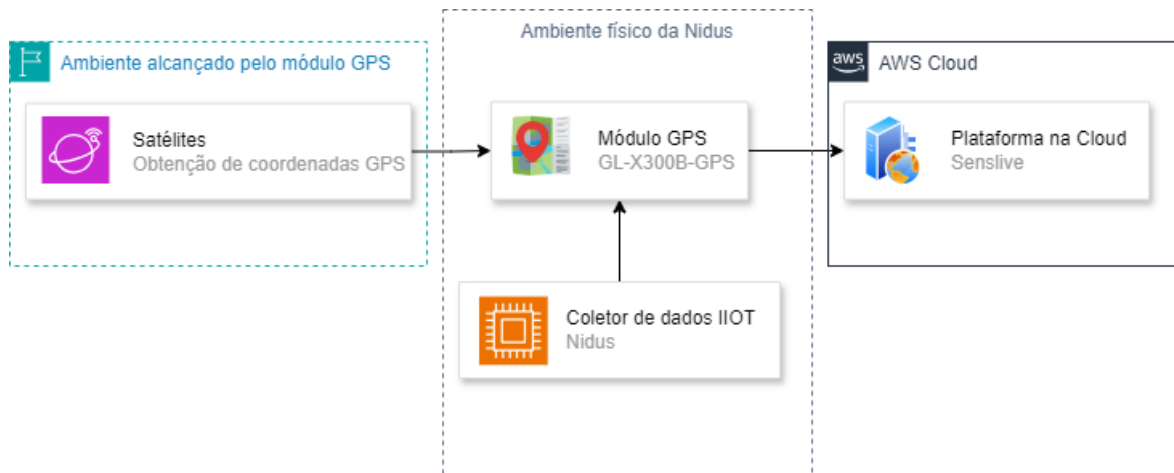
### **3.4.8. Requisitos e funcionalidades**

Para o desenvolvimento do projeto do módulo GPS, foram estabelecidos requisitos fundamentais visando assegurar a obtenção precisa das coordenadas geográficas do dispositivo, bem como a captura de dados relevantes provenientes das "Nidus" conectadas. Estes requisitos desempenharam um papel essencial na disponibilização de informações precisas e atualizadas à plataforma "Senslive". Adicionalmente, foi imperativo estabelecer uma conexão estável e segura com a referida plataforma, visando viabilizar o envio contínuo dos dados coletados.

As funcionalidades incorporadas no módulo GPS compreendem a capacidade de leitura e interpretação dos sinais GPS, essenciais para determinar a localização precisa do dispositivo. Além disso, foram desenvolvidas funcionalidades com a finalidade de identificar e adquirir dados das "Nidus" conectadas ao módulo, incluindo os seus endereços IP e MAC. Com o intuito de garantir o envio eficiente e seguro dos dados à plataforma "Senslive", foram adotados protocolos de comunicação seguros, juntamente com a implementação de um mecanismo de envio regular dos dados capturados. Também assim, foram introduzidas funcionalidades destinadas à gestão de erros e exceções, visando garantir a robustez e confiabilidade do sistema em diferentes cenários operacionais.

### **3.4.9. Arquitetura e design**

A Figura 19 representa a arquitetura definida para o projeto. A arquitetura demonstra o módulo GPS como dispositivo central, o foco do projeto, comunicando com satélites via ondas rádio para obter a localização em coordenadas geográficas. É representado um dispositivo “Nidus” conectado ao módulo GPS que fornece detalhes como o seu IP na rede do módulo GPS e o seu endereço MAC. Todos estes dados são assim enviados via pedidos HTTP para o portal “Senslive”.



**Figura 19 - Arquitetura GPS**

### 3.4.10. Envio de dados para a plataforma

A função “upload\_gps\_data()” é responsável por coordenar o envio dos dados GPS para a plataforma “Senslive”, seguindo uma série de passos estruturados para garantir a integridade e eficácia da transmissão.

O processo começa com a exibição da mensagem "sending gps data", indicando o início da operação de envio de dados.

#### 1. Obtenção do Caminho de Relatório:

- A função utiliza “uci get glgps.info.report\_path” para obter o caminho onde os dados GPS serão reportados.

#### 2. Verificação da Existência do Arquivo Temporário:

- Verifica-se a existência do arquivo “/tmp/gps\_json\_info” antes de prosseguir com o envio dos dados.

#### 3. Leitura e Formatação dos Dados:

- O conteúdo do arquivo “/tmp/gps\_json\_info”, contendo os dados GPS no formato JSON, é lido e armazenado na variável “gps\_info”.

- A função “get\_client\_list\_capttemp” é chamada para obter a lista de dispositivos conectados, armazenando o resultado na variável “devices”.

- Os dados de “gps\_info” e “devices” são combinados para formar o objeto JSON “content”, que será enviado ao servidor.

#### 4. Verificação de Condições Antes do Envio:

- São verificados três critérios antes de realizar o envio:
  - A existência do caminho “path” e do conteúdo “content”.
  - O *status* dos dados GPS, obtido através da função “getJsonValues”, deve ser igual a “true”.

#### 5. Envio da Requisição HTTP:

- Utiliza-se “curl” para realizar uma requisição POST ao servidor especificado em “path”, enviando o JSON “content” como *payload*.
- Os cabeçalhos adequados (“Content-Type: application/json” e “Accept: application/json”) são incluídos na requisição para garantir a correta interpretação dos dados pelo servidor.

#### 6. Registo de Resultados:

- Após o envio, o código captura a data e hora atuais utilizando ``date "+%Y-%m-%d %H:%M:%S"``.
- Dependendo do resultado do envio (“result”), regista-se no arquivo de log “/tmp/gps\_upload.log” detalhes como a data e hora do envio, intervalo de relatório configurado (“uci get glgps.info.report\_interval”), e o resultado específico da transmissão (“upload\_result”).

#### 7. Tratamento de Falhas:

- Caso ocorra uma falha no envio (quando “result” não é “ok”), uma mensagem indicando a falha é registada no arquivo de *logs*, mantendo a consistência na documentação dos eventos ocorridos.

Eu suma, a função “upload\_gps\_data()” foi desenvolvida para integrar diversas etapas críticas no processo de envio de dados GPS. Desde a configuração inicial e preparação dos dados até o envio seguro e o registo detalhado de resultados, cada passo é projetado para garantir que os requisitos funcionais e não funcionais, como integridade, segurança e robustez, sejam atendidos. O Bloco de Código 11 exemplifica uma abordagem organizada e eficiente para operações de envio de dados críticos em aplicações de monitoramento e telemetria.

```
upload_gps_data()
{
    echo "sending gps data"
    path=`uci get glgps.info.report_path`
```

```

if [ -f "/tmp/gps_json_info" ];then
    gps_info=`cat /tmp/gps_json_info`
    devices=$(get_client_list_capttemp)
    content="{\"devices\":'$devices','gps_info':'$gps_info'}"
    gps_staus=`getJsonValues "$content" "status" "default"`
    if [ "$path" != "" -a "$content" != "" -a "$gps_staus" = "true" ];then
        result=`curl -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d "$content" "$path`
        time=$(date "+%Y-%m-%d %H:%M:%S")
        [ "$result" == "ok" ] && echo time : "$time" --- interval : `uci get
glgps.info.report_interval` --- upload_result : "$result" --- "$content" >>
/tmp/gps_upload.log 2>/dev/null
        if [ "$result" != "ok" ];then
            time=$(date "+%Y-%m-%d %H:%M:%S")
            echo fail --- time : "$time" --- interval : `uci get
glgps.info.report_interval` --- upload_result : "$result" --- "$content" >>
/tmp/gps_upload.log 2>/dev/null
        fi
    fi
fi
}

```

Bloco de Código 11 – Função de Envio de Dados de Localização GPS

### 3.4.1. Testes

Realizaram-se testes exaustivos ao dispositivo “GL-X300B-GPS” num ambiente controlado para avaliar o seu desempenho e conformidade com os requisitos estipulados. Durante estes testes, o dispositivo excedeu todas as expectativas, demonstrando uma capacidade excecional de captar a sua localização geográfica com precisão. Além disso, revelou-se apto a detetar todas as unidades "Nidus" conectadas a si e a enviar todos estes dados de forma eficiente e em curtos períodos para a plataforma "Senslive".

A plataforma "CTDiscover", que será introduzida no capítulo, desempenhou um papel crucial na otimização do módulo GPS. Para associar uma unidade "Nidus" ao *router*, foi necessário ajustar a gama de IP da "Nidus" para a gama de IP do *router*, passando de 192.168.1.100 para 192.168.8.100. Esta alteração permitiu que a “Nidus” fosse conectada ao dispositivo de GPS e assim realizar os testes necessários.

Os testes confirmaram que o dispositivo “GL-X300B-GPS” não só cumpre como também supera os requisitos operacionais esperados. Este desempenho fiável e eficiente é crucial para a integração do

dispositivo em sistemas industriais que dependem da precisão e rapidez na transmissão de dados para a otimização das operações.

Com base nos resultados dos testes realizados, o dispositivo “GL-X300B-GPS” foi avaliado em termos do seu Nível de Maturidade Tecnológica (TRL). Este sistema de avaliação mede a maturidade de uma tecnologia ao longo de nove níveis, desde a conceção inicial até à implementação plena. O desempenho do “GL-X300B-GPS” nos testes indica que este módulo se encontra num TRL 7. Este nível corresponde a uma fase em que o protótipo foi demonstrado em ambiente operacional, mostrando que a tecnologia está pronta para ser utilizada em condições reais e tem um potencial elevado de integração em sistemas industriais.

Em suma, o “GL-X300B-GPS” demonstrou, através de testes controlados, uma capacidade excepcional de captar e transmitir dados de localização geográfica e de dispositivos conectados. O seu elevado desempenho e fiabilidade reforçam a sua adequação para aplicações industriais, sendo classificado como TRL 7, o que evidencia a sua prontidão para integração e utilização em ambiente real.

### **3.5. CT Discover – Plataforma mobile**

A “CTDiscover” é uma aplicação destinada à gestão de dispositivos conectados à rede, nomeadamente as “Nidus”. Esta ferramenta poderosa permite a comunicação direta com as “Nidus” selecionadas, ao contrário da plataforma “Senslive”, que comunica com um servidor externo às “Nidus”. Esta comunicação direta possibilita o controlo de cada dispositivo de forma mais detalhada e em um nível bastante baixo, permitindo operações como ligar/desligar entradas e saídas do dispositivo, alterar definições como o endereço IP, adicionar/remover sensores, modificar a sua localização, entre outras configurações.

A aplicação “CTDiscover” original foi desenvolvida exclusivamente para *desktop*. O desenvolvimento desta versão limitava-se a sistemas operativos *desktop*, restringindo a sua compatibilidade e escalabilidade. Face a esta limitação, surgiu a necessidade de criar uma aplicação que mantivesse as funcionalidades da “CTDiscover” antiga, mas que fosse compatível com diversos sistemas operativos.

Com o novo desenvolvimento, pretendeu-se garantir que todas estas funcionalidades estivessem disponíveis em múltiplas plataformas, aumentando a compatibilidade e a flexibilidade de utilização da “CTDiscover” em diferentes sistemas operativos, incluindo dispositivos móveis.

### 3.5.1. Metodologia de desenvolvimento

O desenvolvimento da aplicação foi realizado com base em atividades semanais ou diárias, permitindo um acompanhamento contínuo do progresso e facilitando ajustes conforme necessário. Para gerir o código de forma eficiente e garantir a sua integridade, foi utilizado o sistema de controlo de versões “Git”. Esta abordagem permitiu manter um histórico detalhado das alterações, evitando a perda de código e mantendo o projeto organizado.

O uso de “Git” proporcionou uma forma estruturada de rastrear o progresso do desenvolvimento, facilitando a colaboração e a revisão de código. Através de *commits* regulares, foi possível documentar cada etapa do processo, assegurando que todas as modificações fossem registadas de maneira clara e acessível.

É importante salientar que não foi adotada uma metodologia formal como “Scrum” ou “Agile”, mas sim “Kanban”, dado a sua simplicidade e eficiência. Assim, o foco foi na execução prática das tarefas, com base nas necessidades imediatas do projeto e nos objetivos estabelecidos. Esta abordagem mais flexível permitiu uma adaptação rápida às exigências do desenvolvimento, mantendo um fluxo de trabalho eficiente e produtivo.

### 3.5.2. Desenvolvimento com Flutter

A aplicação foi desenvolvida utilizando a *framework* “Flutter”. “Flutter” é uma tecnologia *open-source* criada pela “Google”, que permite o desenvolvimento de aplicações nativas para *Android*, iOS, *web* e *desktop* a partir de um único código base. Utiliza a linguagem de programação “Dart” e oferece ferramentas e widgets que facilitam a criação de interfaces de utilizador rápidas e flexíveis.

A escolha de “Flutter” deve-se à sua capacidade de gerar aplicações para múltiplas plataformas com um único código, reduzindo significativamente o tempo e os recursos necessários para o desenvolvimento e manutenção.

A vasta biblioteca de *widgets* personalizáveis e a robusta comunidade de desenvolvedores, juntamente com extensa documentação, tornaram “Flutter” a escolha ideal para este projeto, combinando eficiência, performance e facilidade de uso.

### 3.5.3. Requisitos e funcionalidades

Os requisitos para o funcionamento eficaz da aplicação incluem a identificação e manutenção das “Nidus” conectadas à rede, juntamente com a sua monitorização contínua. Para atender a esses requisitos, diversas funcionalidades foram implementadas:

A aplicação possibilita a descoberta das "Nidus" presentes na rede local por meio do protocolo UDP. Além disso, oferece a capacidade de editar as configurações das "Nidus" conforme necessário.

Outra funcionalidade essencial é a identificação dos sensores atribuídos a cada "Nidus", permitindo a edição dos mesmos, como adição ou remoção de sensores. Além disso, é possível identificar e editar a localização geográfica de cada "Nidus".

Para garantir a integridade dos dados, a aplicação realiza a verificação de *hashes* para todos os ficheiros da "Nidus", assegurando que estão validados e não corrompidos.

Os *logs* dos sensores de cada "Nidus" podem ser descarregados e gravados ao longo de um período pré-definido. A aplicação foi desenvolvida de forma a lidar com situações em que uma "Nidus" deixe de responder durante a gravação ou registo dos *logs*, garantindo a continuidade do processo.

Os *logs* podem ser extraídos em formatos convenientes, como PDF, CSV ou XLS, para análise e processamento posterior. Por outro lado, a gravação (extração contínua) dos *logs* é realizada em formato XLS, facilitando a sua manipulação e utilização para fins de monitorização e análise de dados.

Para além das funcionalidades acima mencionadas, a plataforma foi concebida para cumprir uma série de requisitos não funcionais cruciais para a sua operação eficaz e eficiente. Em primeiro lugar, a plataforma foi projetada para processar e transmitir dados em tempo real ou quase real, minimizando a latência e assegurando que a informação esteja disponível para análise sem atrasos significativos. Isso garante que os utilizadores possam tomar decisões com base em dados atualizados e precisos.

A fiabilidade da plataforma foi uma prioridade durante o desenvolvimento, assegurando alta disponibilidade e a capacidade de recuperação rápida em caso de falhas. Foram implementados mecanismos de redundância e recuperação de erros que garantem a continuidade dos serviços, mesmo em situações de falha técnica, reforçando a robustez da solução.

Em termos de segurança, a plataforma foi dotada de medidas rigorosas para proteger os dados transmitidos e armazenados, incluindo a criptografia dos dados tanto em trânsito quanto em repouso. Foram adotadas práticas de autenticação e autorização rigorosas para garantir que apenas utilizadores autorizados possam aceder às funções e dados da aplicação, proporcionando um ambiente seguro para a operação dos dados.

A usabilidade também foi um aspeto chave no desenvolvimento da plataforma. A interface da aplicação foi projetada para ser intuitiva e fácil de usar, permitindo aos utilizadores realizar todas as operações necessárias com um esforço mínimo e uma curva de aprendizagem reduzida. Isso garante

que a plataforma pode ser utilizada eficazmente por um amplo leque de utilizadores, independentemente do seu nível de experiência técnica.

Além disso, a plataforma foi desenvolvida para ser compatível com diferentes sistemas operativos e plataformas. Isso assegura que a aplicação pode ser utilizada em diversos ambientes tecnológicos sem necessidade de ajustes significativos, promovendo a sua adaptabilidade e flexibilidade em vários contextos operacionais.

A manutenibilidade foi um fator considerado durante o desenvolvimento. O código e a arquitetura da plataforma foram projetados de forma modular e bem documentada, facilitando a manutenção, atualização e expansão das funcionalidades da aplicação. Isso permite que a plataforma possa ser facilmente adaptada para novas exigências ou corrigida em caso de problemas, garantindo a longevidade e a evolução contínua da solução.

Finalmente, a eficiência energética da plataforma foi otimizada para minimizar o consumo de recursos computacionais, como processamento e memória, e para ser eficiente em termos de consumo energético, especialmente em dispositivos móveis ou sistemas com recursos limitados. Este aspeto é essencial para garantir uma operação sustentável e economicamente viável, especialmente em cenários onde os recursos são limitados.

#### **3.5.4. Distribuição temporal do desenvolvimento**

Nesta secção é apresentada uma visão geral da distribuição temporal do desenvolvimento das diversas funcionalidades da aplicação, detalhando as etapas e os períodos específicos dedicados a cada fase do projeto. A Figura 20 ilustra a distribuição temporal do desenvolvimento das diversas funcionalidades da plataforma, conforme detalhado abaixo:

- **9 de janeiro – 12 de janeiro**

- Levantamento de Requisitos e Planeamento Inicial: Início da análise detalhada dos requisitos do sistema e definição do planeamento inicial do projeto.

- **15 de janeiro – 2 de fevereiro | 11 de março – 14 de março**

- *Device Scanner*: Neste período foi feita a descoberta e edição de dispositivos ligados à rede utilizando tecnologia UDP.

- **5 de fevereiro – 16 de fevereiro**

- *Sensor Search*: Neste período foram implementados os métodos de procura de sensores sem-fio e sensores com-fio conectados à “Nidus” selecionada, assim como a sua possível adição/remoção do sensor à “Nidus”.

- **13 de fevereiro – 20 de fevereiro**

- *Sensor Location*: Neste período foi implementado o método de procura das localizações de todos os sensores conectados à “Nidus” assim como a sua possível edição.

- **21 de fevereiro – 28 de fevereiro**

- *Device Settings*: Neste período foi implementada a edição da “Nidus” selecionada, via pacotes HTTP. Isto inclui definições como relógio, *logs*, definições de portal, COM *ports*, definições GSM e manutenção.

- **29 de fevereiro – 14 de março**

- *Module F*: Neste período foi implementado o Módulo F que inclui funcionalidades como, verificação de *hashes* para validação de ficheiros da “Nidus”, verificação de integridade de software utilizando sistema de *hashing* “CRC-32” [12], descarregamento de *logs* nos formatos CSV, XML e XLS, e por fim, a gravação e monitorização em tempo real da “Nidus” e dos seus sensores, assim como visualização dos dados em formato tabela e formato de gráfico.

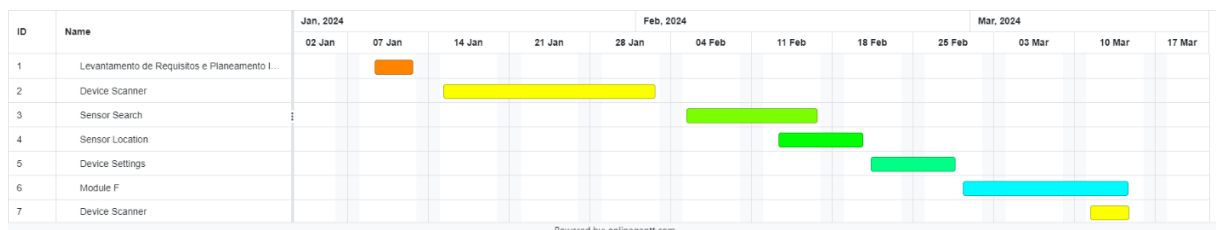


Figura 20 - Distribuição temporal do desenvolvimento - CTDDiscover

### 3.5.5. Arquitetura e design

Na fase de descoberta dos dispositivos (Figura 21), ocorre a emissão de um pacote *broadcast* para a rede, partindo da aplicação e direcionado ao *router*. O *router*, por sua vez, encarrega-se de difundir o pacote *broadcast* pela rede, alcançando todas as "Nidus" presentes na mesma. Posteriormente, as "Nidus" respondem ao pacote de acordo com a sua configuração e funcionalidades específicas.

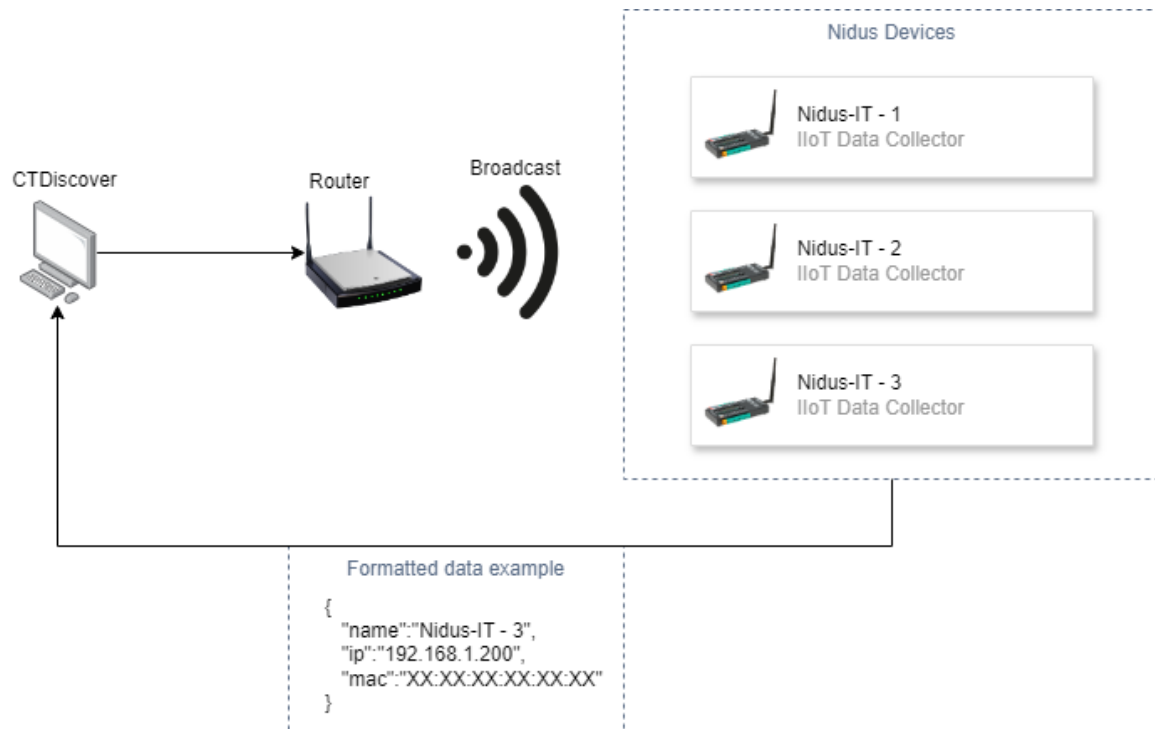


Figura 21 - Arquitetura CTDDiscover

Após a descoberta das "Nidus", os pedidos subsequentes a uma "Nidus" selecionada são realizados utilizando-se quer o protocolo UDP ou HTTP (Figura 22), dependendo dos dados a enviar e dos requisitos da operação. Essa escolha é determinada pelo tipo de comunicação mais adequado para as necessidades específicas da transmissão de dados em questão.

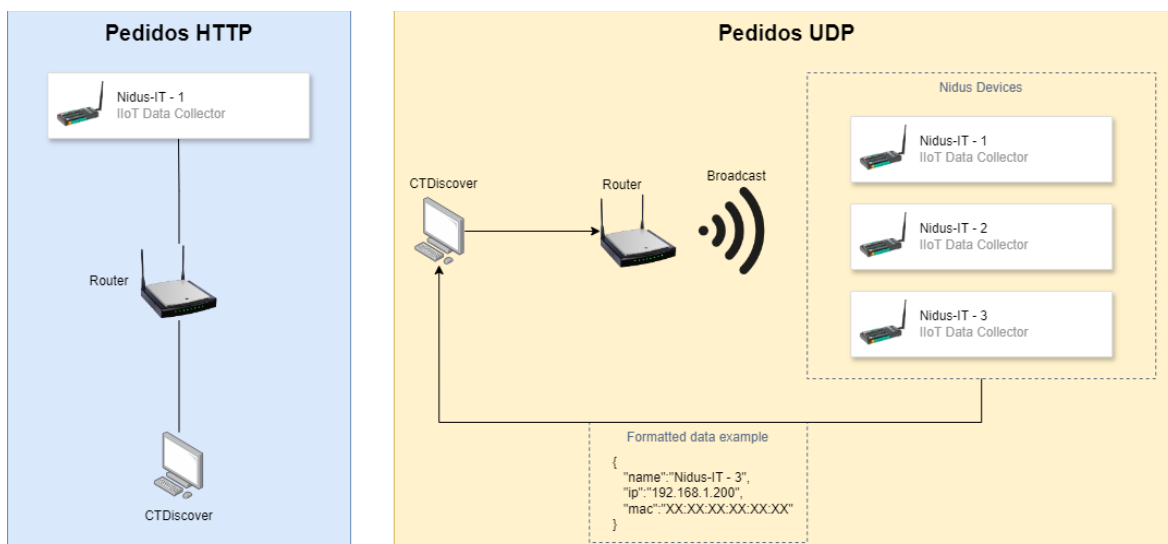


Figura 22 - Pedidos HTTP e UDP - CTDDiscover

O protocolo UDP é utilizado para transmissões rápidas e eficientes, ideal para comunicações onde a velocidade é crítica e a perda ocasional de pacotes é tolerável, como na monitorização contínua de

sensores. Por outro lado, o protocolo HTTP é utilizado para transmissões que requerem fiabilidade e confirmação de entrega, sendo ideal para operações onde a integridade dos dados é essencial, como na configuração de dispositivos ou no envio de comandos específicos.

A flexibilidade de escolher entre UDP e HTTP permite otimizar a comunicação com os dispositivos "Nidus" conforme necessário, melhorando o desempenho geral da rede. O UDP oferece menor latência e menor *overhead*, enquanto o HTTP proporciona maior fiabilidade e segurança na entrega dos dados. Esta arquitetura é eficiente na descoberta de dispositivos, pois utiliza pacotes *broadcast* que facilitam a identificação rápida de todos os dispositivos na rede, reduzindo o tempo e a complexidade de configuração manual. A utilização do *router* para difundir pacotes *broadcast* e gerir a comunicação entre os dispositivos e a aplicação centralizada resulta numa utilização mais eficiente da infraestrutura de rede existente. Além disso, a arquitetura é facilmente escalável, permitindo a adição de novos dispositivos "Nidus" à rede e a sua descoberta automática sem necessidade de ajustes significativos na sua configuração.

### 3.5.6. Descoberta de dispositivos por UDP

A descoberta de dispositivos utilizando comunicação UDP envolve diversos passos, iniciando-se pela criação e configuração do *socket* UDP.

No primeiro passo, é ligado um *socket* UDP a qualquer endereço IPv4, com a porta a ser atribuída automaticamente através da utilização do valor 0.

```
try {  
    RawDatagramSocket socket = await  
RawDatagramSocket.bind(InternetAddress.anyIPv4, 0);  
}
```

Bloco de Código 12 - Inicialização de Socket Datagram para Comunicação UDP

De seguida, é necessário ativar o mecanismo de *broadcast*, atualizando o parâmetro "*broadcastEnabled*", e assim, enviada uma mensagem de *broadcast* para o endereço e porta específicos.

```
// Enable broadcast  
socket.broadcastEnabled = true;  
  
// Send a broadcast message  
socket.send(Uint8List.fromList(numbers), InternetAddress(broadcastAddress  
, port));
```

**Bloco de Código 13 - Envio de Mensagem por Broadcast via Socket UDP**

O pacote enviado por *broadcast* é especificamente desenhado para ser reconhecido na rede pelos dispositivos "Nidus". Este pacote é composto por uma sequência pré-definida de bits, que inclui uma assinatura única e identificável por todos os dispositivos "Nidus". Esta sequência de bits é padronizada e idêntica para todos os dispositivos, garantindo que a comunicação UDP seja eficiente e que cada "Nidus" na rede possa identificar e responder ao pacote de *broadcast* corretamente. A assinatura inclui cabeçalhos específicos, códigos de identificação, e verificações de integridade que asseguram a autenticidade e a precisão do pacote recebido.

Após o envio do pacote, é necessário configurar o *socket* UDP para escutar as respostas dos dispositivos. Este processo envolve a espera passiva por mensagens de retorno, que contêm dados dos dispositivos "Nidus" disponíveis na rede. Estas respostas são então processadas para identificar e listar os dispositivos encontrados, permitindo uma gestão eficaz e em tempo real dos mesmos.

```
socket.listen((RawSocketEvent event) {
    if (event == RawSocketEvent.read) {
        try {
            Datagram datagram = socket.receive();
            var packet = UDPHelper.parsePacket(datagram.data);
            if (packet != null) {
                bool foundDevice = false;
                for (var device in devicesFound) {
                    if (device.mac == packet.mac) {
                        foundDevice = true;
                    }
                }
                if (!foundDevice) {
                    devicesFound.add(packet);
                }
            }
        } catch (e) {
            // handle error
        }
    }
});
```

**Bloco de Código 14 - Receção e Processamento de Pacotes UDP**

Ao receber uma resposta, o pacote é analisado e convertido de "Uint8List" para uma instância de um objeto da classe "Device". A função "parsePacket" é responsável por validar diversos campos do pacote para assegurar que os dados estão corretos antes de criar e retornar o objeto "Device". Este

processo de validação inclui a verificação de identificadores únicos, endereços MAC, tipos de dispositivos, e outros parâmetros críticos. Somente após a confirmação da integridade e da precisão dos dados, o objeto “Device” é instanciado e adicionado à lista de dispositivos encontrados, garantindo que apenas dispositivos válidos e corretamente identificados sejam considerados para gestão e monitorização na rede. Esta função é composta pelos seguintes passos:

- Remoção do último carácter ( “\n” ) da variável “*buffer*”.

```
int bytesRead = buffer.length;
// Remove trailing newline character
if (buffer[bytesRead - 1] == 10) {
    buffer.removeLast();
}
```

#### Bloco de Código 15 - Leitura e Limpeza de Buffer

- Decodificação e tratamento da *string*

É necessário remover caracteres especiais como “0x01” e “0x04”.

```
String sData = utf8.decode(buffer.subList(0, bytesRead));
sData = sData.replaceAll(String.fromCharCode(0x01), '');
sData = sData.replaceAll(String.fromCharCode(0x04), '');

List<String> sFields = sData.split(String.fromCharCode(0x02));

if (sFields.length != 13) {
    return null;
}
```

#### Bloco de Código 16 - Decodificação e Processamento de Dados em String

Por fim, é aplicada uma estratégia para evitar a inclusão de dispositivos duplicados na lista de dispositivos, visto que nem sempre estes respondem apenas uma vez ao mesmo pacote UDP. Para tal, é utilizado um mapa, denominado “*devicesFound*”, responsável por armazenar entradas únicas, assegurando que dispositivos repetidos não sejam guardados. Esta abordagem garante que cada dispositivo “Nidus” é representado uma única vez na lista final, evitando redundâncias e possíveis conflitos na gestão e monitorização dos dispositivos na rede.

A Figura 23 representa visualmente a lista de dispositivos encontrados, assim como as funcionalidades de procurar por outros dispositivos e/ou editar os atuais.

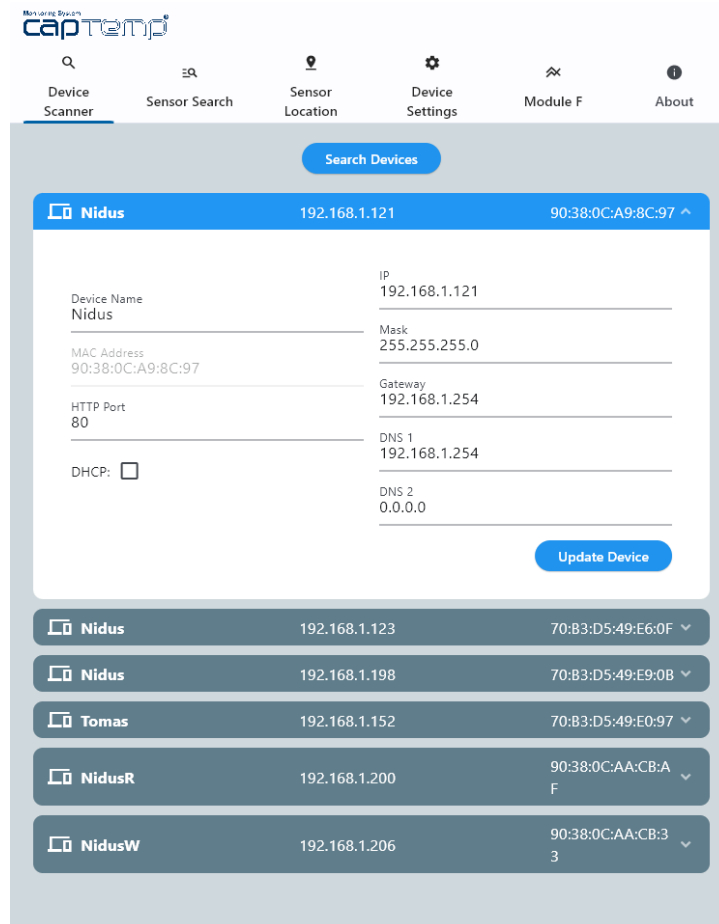


Figura 23 - Lista de Dispositivos Encontrados via Comunicação UDP

### 3.5.7. Descoberta de sensores *wired* / *wireless*

A solução "CT Discover" integra funcionalidades como a identificação de sensores, realizada mediante solicitações HTTP a um arquivo denominado "setup.xml" armazenado nas "Nidus". Após a obtenção deste arquivo, ocorre a análise da resposta, em formato XML, para extrair os valores essenciais, procedendo à atualização da lista de sensores identificados. Estes sensores fornecem detalhes como número de série, designação, valor atual e unidade de medida associada.

Adicionalmente, a aplicação possibilita a modificação das configurações de procura de sensores sem fio. Esta alteração é realizada através da configuração da chave, permitindo ajustar a potência de transmissão e o canal de rede utilizado pela "Nidus" na procura de sensores sem fio a ela conectados. Esta funcionalidade facultada pela aplicação permite uma personalização das configurações de busca, adaptando-as às exigências específicas do ambiente e dos dispositivos conectados.

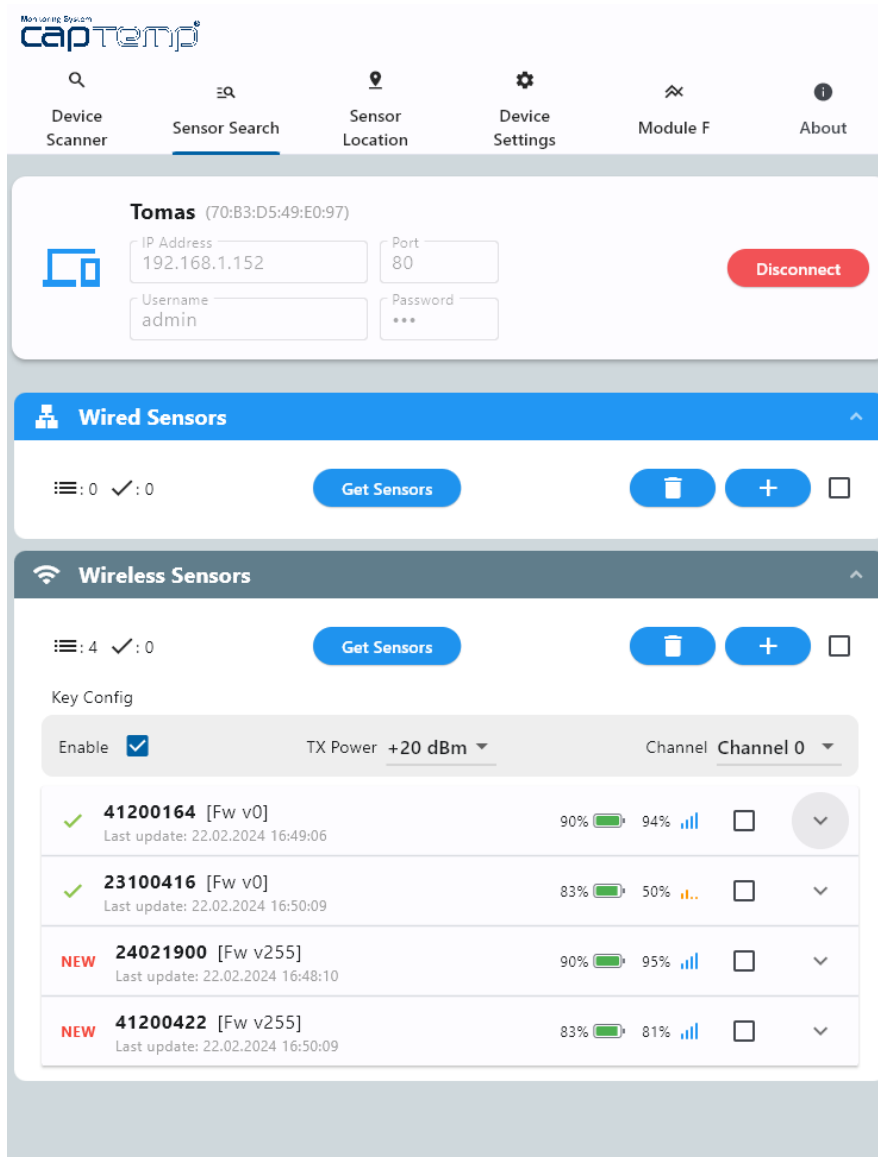


Figura 24 - Lista de Dispositivos Encontrados e Respetivos Sensores Conectados

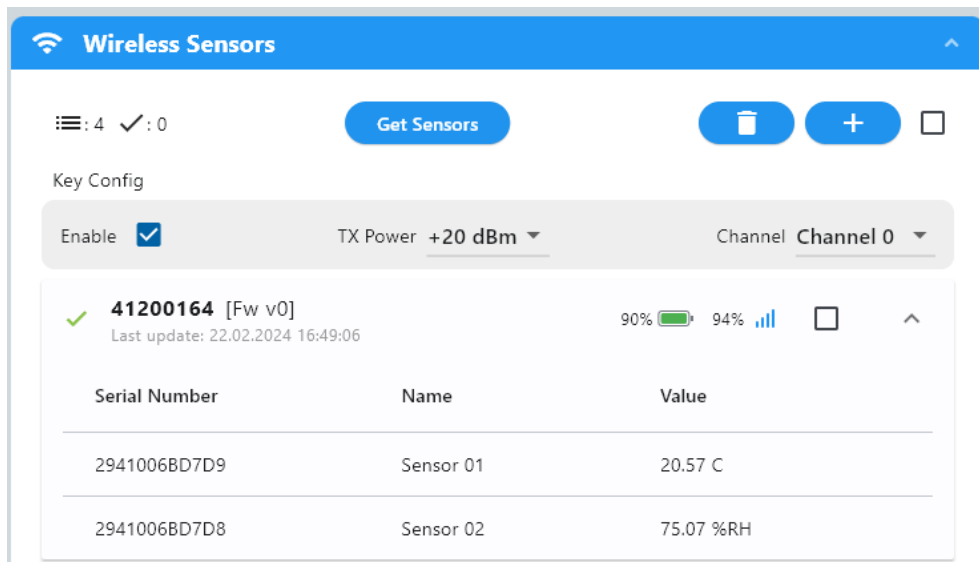


Figura 25 - Detalhes dos Sensores Conectados a um Dispositivo

### 3.5.8. Localização dos sensores

Nesta secção (Figura 26), é possível aceder aos sensores conectados à "Nidus" seleccionada e obter informações como o tipo (com-fio ou sem-fio), endereço, nome e localização. Adicionalmente, é possível atualizar a localização do sensor. Esta atualização é efetuada através de um pedido HTTP contendo código XML formatado especificamente para esse fim.

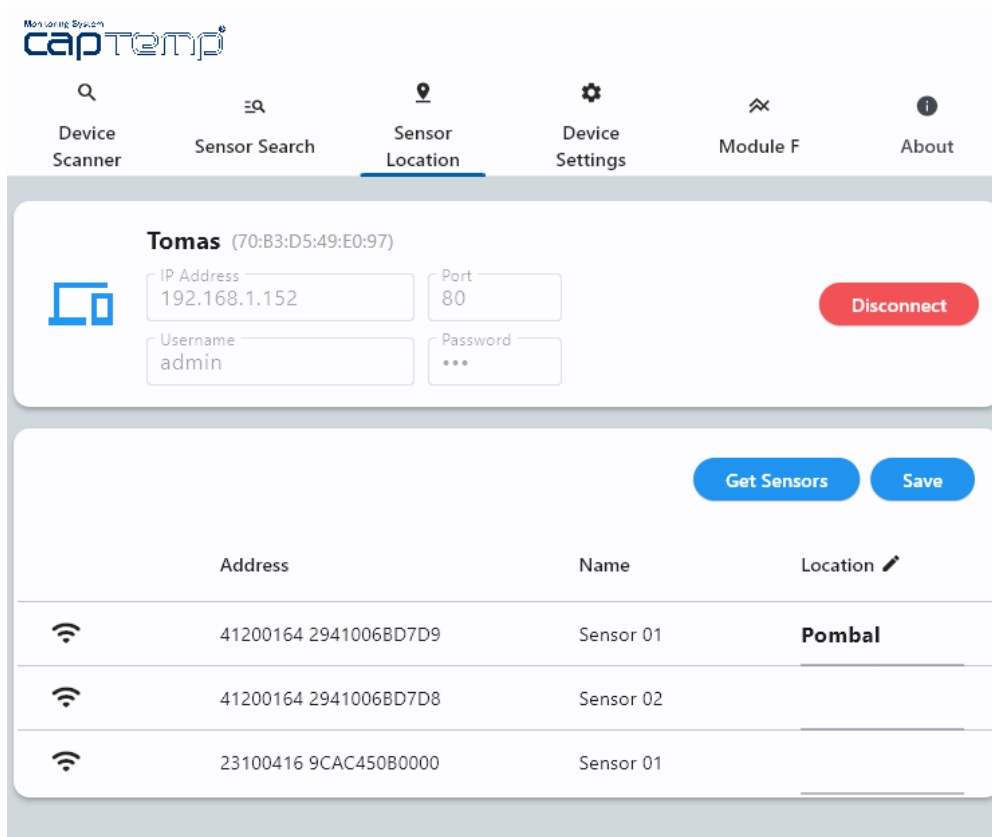


Figura 26 - Página de localizações dos sensores

### 3.5.9. Definições da Nidus

Nesta secção (Figura 27), é efetuada a configuração do dispositivo “Nidus”. É possível efetuar alterações como a alteração do seu relógio, servidor SNTP, temporização de registo de *logs*, *endpoint* para onde deseja enviar os seus dados, portas de série, definições GSM, e definições de manutenção como o restauro do estado do dispositivo para predefinições de fábrica.

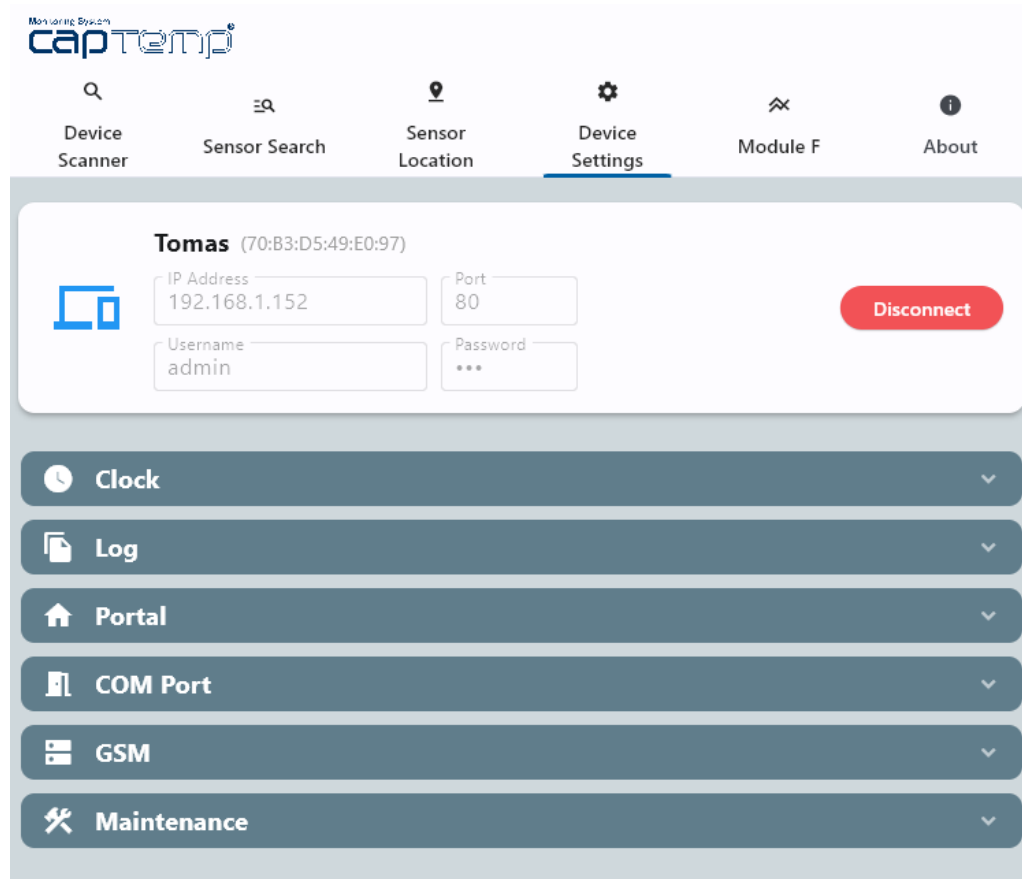


Figura 27 - Página de definições da "Nidus"

### 3.5.10. Módulo F

Nesta secção, é apresentado o Módulo F (Figura 28), que inclui funcionalidades como o cálculo de *hashes* para a verificação da integridade dos certificados no portal, cujo URL é inserido pelo utilizador, sendo este, por defeito, "https://senslive.com". As *hashes* válidas são previamente armazenadas na "Nidus" para que possam ser comparadas com as *hashes* do portal. Se todas forem idênticas, os certificados são considerados válidos, caso contrário, um ou mais certificados (no portal ou no "Nidus") são considerados inválidos, alertando assim o utilizador para esta ocorrência. É também possível para o utilizador descarregar estes ficheiros.

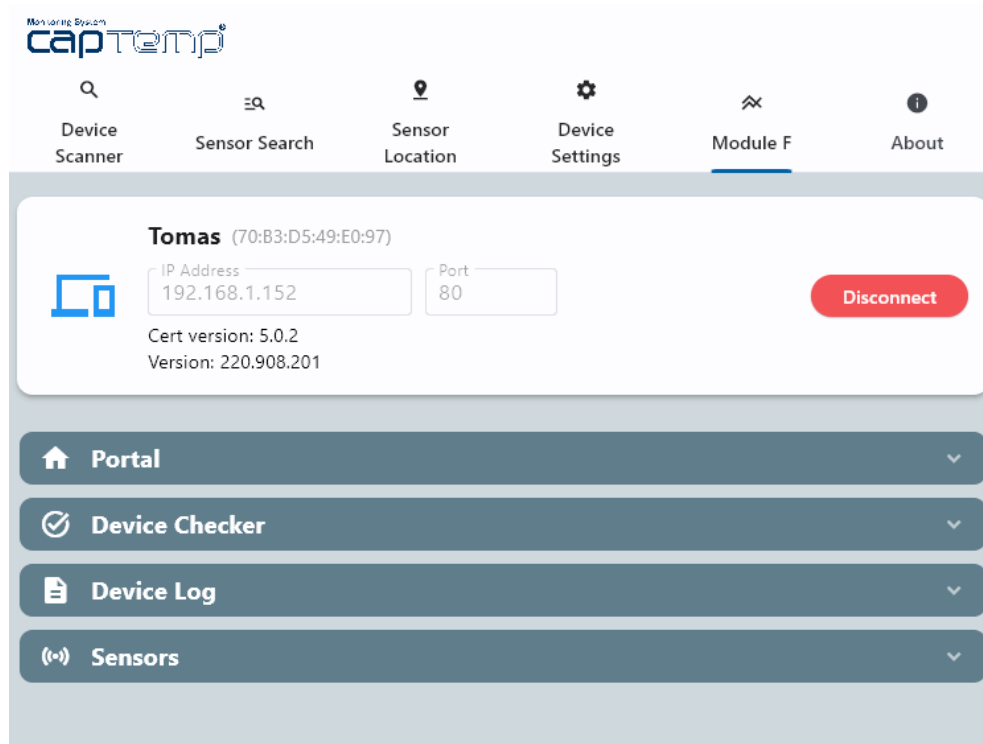


Figura 28 - Página do Módulo F

Além disso, o Módulo F disponibiliza uma funcionalidade de monitorização em tempo real das “Nidus” e dos respetivos sensores (Figura 29). Esta monitorização contínua permite ao utilizador acompanhar, em tempo real, o estado e a integridade dos sensores, bem como visualizar as alterações e tendências nos dados através de gráficos interativos (Figura 30). Este sistema de monitorização fornece uma camada adicional de segurança e transparência, permitindo identificar de forma imediata quaisquer discrepâncias ou irregularidades que possam surgir.

Adicionalmente, o Módulo F suporta quebras de conexão com a “Nidus”. Caso deixe de haver conexão entre a ferramenta “CTDiscover” e a “Nidus”, a aplicação foi programada para aguardar um período de tempo definido até ocorrer o *timeout*. Durante este período, a aplicação tenta restabelecer a comunicação. Se a conexão for retomada dentro deste prazo, os logs correspondentes serão registados e a monitorização prosseguirá normalmente. Esta funcionalidade assegura a continuidade do processo de monitorização, mesmo em situações de falha temporária de comunicação.

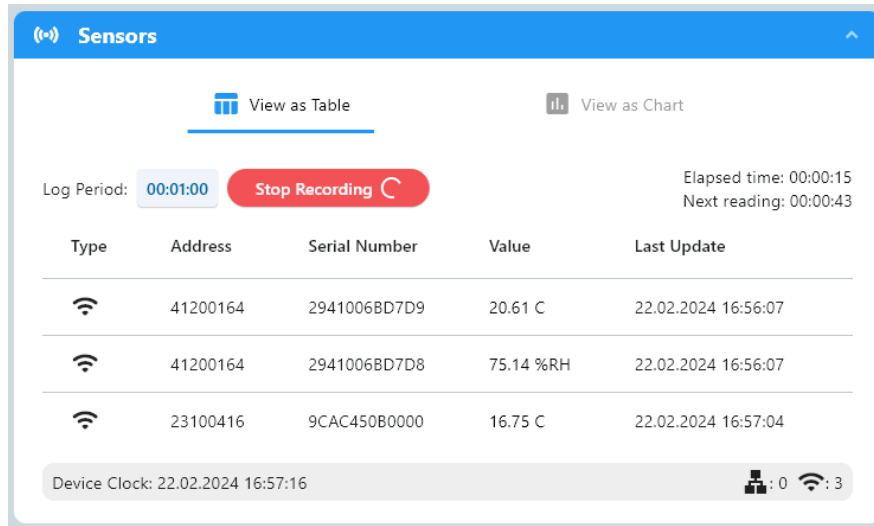


Figura 29 - Monitorização dos sensores em formato de lista

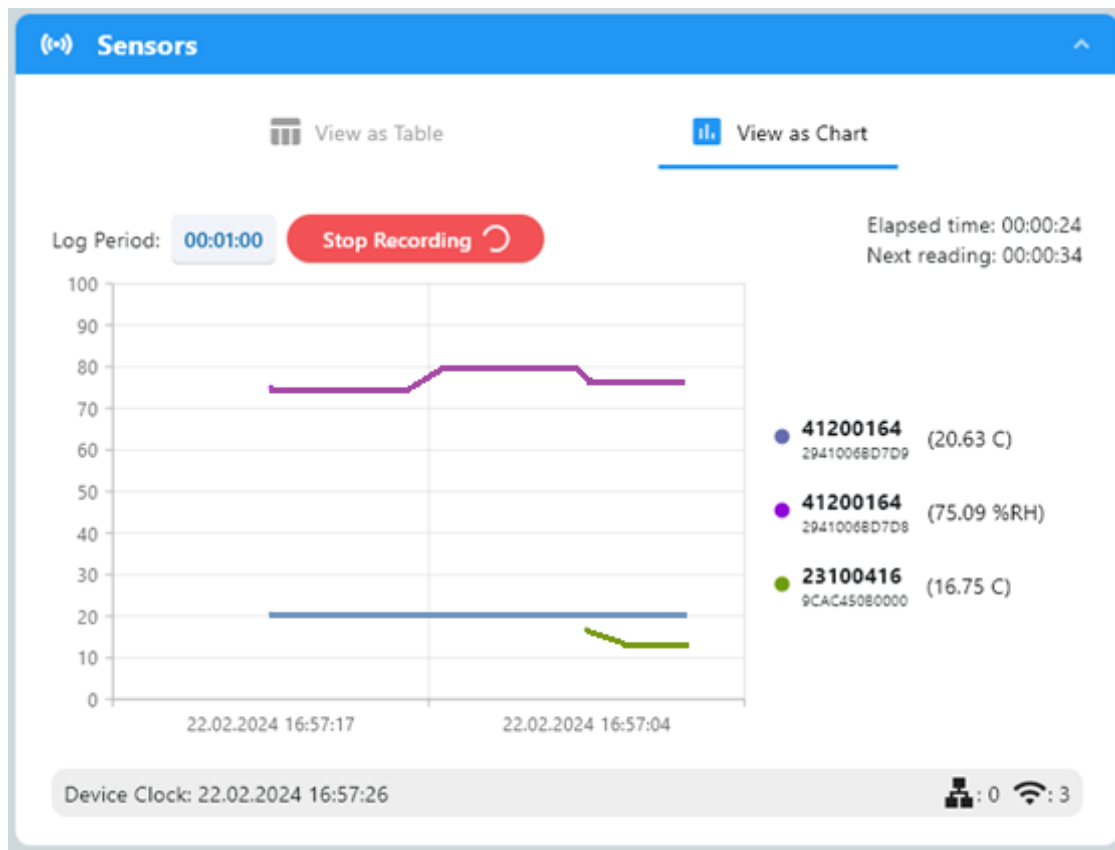
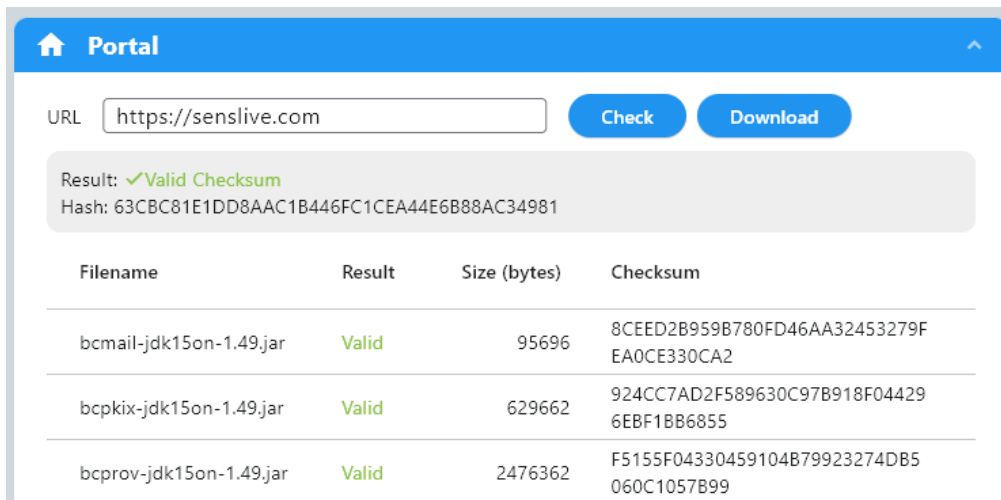


Figura 30 - Monitorização dos Sensores em Formato de Gráfico

### 3.5.11. Cálculo de *hashes*

O cálculo e verificação das *hashes* dos arquivos (Figura 31) são procedimentos cruciais para garantir a integridade e autenticidade dos dados. O processo de cálculo e verificação das *hashes* dos arquivos é realizado utilizando o método de encriptação SHA1 (*Secure Hash Algorithm 1*) [11].



Filename	Result	Size (bytes)	Checksum
bcmail-jdk15on-1.49.jar	Valid	95696	8CEED2B959B780FD46AA32453279F EA0CE330CA2
bcpkix-jdk15on-1.49.jar	Valid	629662	924CC7AD2F589630C97B918F04429 6EBF1BB6855
bcprov-jdk15on-1.49.jar	Valid	2476362	F5155F04330459104B79923274DB5 060C1057B99

Figura 31 - Verificação de Checksum dos Ficheiros da "Nidus"

O algoritmo SHA1 (Figura 32) [14] é uma função de *hash* criptográfica que produz uma representação única e irreversível de dados de entrada. Funciona dividindo os dados de entrada em blocos de 512 *bits*, aplicando operações *bitwise* e rotações aos blocos de dados e comprimindo-os iterativamente em um valor *hash* final de 160 bits.

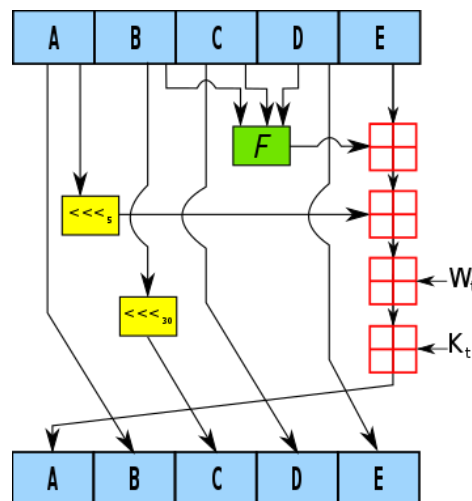


Figura 32 - Algoritmo SHA1 (adaptado de [11])

### 3.5.12. Cálculo de *checksum*

Por vezes, é necessário enviar dados para um dispositivo "Nidus". No entanto, ao fazer isso, é essencial verificar a autenticidade e integridade do dispositivo. Isso requer o envio de uma chave de confirmação para garantir que o dispositivo que está sendo editado seja o correto e que apresente um

*firmware* certificado. Para resolver este desafio, foi implementado um sistema de cálculo de *checksums*, utilizando o método CRC-32.

O método CRC-32 (*Cyclic Redundancy Check*) [12] é um algoritmo de soma de verificação que calcula um valor de verificação de 32 *bits* para dados, geralmente utilizado para detetar erros de transmissão ou corrupção de dados em redes de comunicação e armazenamento.

Neste método, os dados são processados *bit a bit*, gerando um valor de verificação de 32 *bits*. Esse valor é anexado aos dados e transmitido juntamente com eles. Na receção, o CRC-32 é recalculado e comparado com o valor recebido. Se houver discrepância entre os valores, isso indica uma possível falha de transmissão ou corrupção de dados.

O CRC-32 é rápido e eficiente em termos de processamento, sendo amplamente utilizado em comunicações digitais, armazenamento de dados e sistemas de arquivos para garantir a integridade dos dados.

A função utilizada na “CTDiscover” para calcular o *checksum* da chave tem como parâmetros de entrada uma *string*, sendo esta uma sequência XML dos dados a enviar para o dispositivo “Nidus” e o endereço MAC do dispositivo. A função envolve os seguintes passos:

### 1. Parâmetros de Entrada:

- “sString”: A *string* para a qual o CRC será calculado.
- “mac”: O endereço MAC usado no cálculo do CRC.
- “isPost”: Um parâmetro opcional que indica se é um pedido *POST*.

### 2. Preparação dos Dados:

- A *string* “sString” é convertida em uma lista de *bytes* “sCRCBytes”.
- Um novo *array* chamado “bytestoCalc” é criado com espaço suficiente para armazenar “sCRCBytes” e o endereço MAC.
- Os *bytes* do endereço MAC são convertidos de hexadecimal para decimal e colocados nos primeiros 6 elementos de “bytestoCalc”.

### 3. Cálculo do CRC:

- Os *bytes* de “sCRCBytes” são colocados após os *bytes* do endereço MAC em “bytestoCalc”.

- O método “hashCore” da instância de “crc32” é chamado com “bytestoCalc” como entrada para calcular o CRC.

- O método “hashFinal” é chamado para obter o valor final do CRC.

#### 4. Conversão e Formatação do CRC:

- Os *bytes* do CRC são convertidos numa *string* hexadecimal e formatados em duas posições (ex: '0A', 'FF').

- Se “isPost” for verdadeiro e o comprimento do CRC for menor que 8, zeros são adicionados à esquerda para garantir um comprimento de 8 caracteres.

- Se “isPost” for verdadeiro, a *string* “<Key>0x00000000</Key>” em “sString” é substituída pela *string* “<Key>” seguida do CRC calculado.

#### 5. Retorno:

- O CRC calculado é retornado como uma *string*.

Essencialmente, esta função adiciona um CRC calculado a uma *string*, substituindo uma marcação específica, usando o endereço MAC como parte do cálculo para garantir a integridade dos dados ao enviar requisições para um dispositivo "Nidus".

Para validar esta chave, a "Nidus" realiza o mesmo processo e, em seguida, as duas chaves são comparadas. Se as chaves forem idênticas, o processo pode continuar e a ação é concluída.

O **Bloco de Código** 17 demonstra como foi implementado o sistema de cálculo de *checksum* usando CRC-32.

```
String calculateNidusRequestCRCWithMac(String sString, String mac,
    [bool isPost = false]) {
    var crc32 = Crc32v_2();
    String hash = '';
    List<int> sCRCBytes = utf8.encode(sString);
    List<int> bytestoCalc = List<int>.filled(sCRCBytes.length + 6, 0);
    bytestoCalc[0] = int.parse(mac.substring(3, 5), radix: 16);
    bytestoCalc[1] = int.parse(mac.substring(0, 2), radix: 16);
    bytestoCalc[2] = int.parse(mac.substring(12, 14), radix: 16);
    bytestoCalc[3] = int.parse(mac.substring(9, 11), radix: 16);
    bytestoCalc[4] = int.parse(mac.substring(15, 17), radix: 16);
    bytestoCalc[5] = int.parse(mac.substring(6, 8), radix: 16);
    for (int i = 0; i < sCRCBytes.length; i++) {
```

```
        bytestoCalc[6 + i] = sCRCBytes[i];
    }

    // CRC Calc
    crc32.hashCore(UInt8List.fromList(bytestoCalc), 0,
        UInt8List.fromList(bytestoCalc).length);
    var hashBytes = crc32.hashFinal();
    for (var byte in hashBytes) {
        hash += byte.toRadixString(16).padLeft(2, '0').toUpperCase();
    }

    if (isPost) {
        if (hash.length < 8) {
            hash = '0' * (8 - hash.length) + hash;
        }
        return sString.replaceFirstMapped(
            RegExp(r'<Key>0x00000000</Key>'), (match) => '<Key>0x$hash</Key>');
    }

    return hash;
}
```

**Bloco de Código 17 - Cálculo de CRC para Requisição Nidus com Endereço MAC**

## 4. Análise crítica e proposta de melhorias

Este capítulo visa apresentar uma análise crítica das atividades desenvolvidas durante o estágio, refletindo sobre os pontos fortes e fracos observados, bem como propondo melhorias para futuras implementações e práticas.

### 4.1. Análise Crítica

Durante o estágio realizado na “CapTemp”, foram desenvolvidas várias atividades que contribuíram significativamente para o crescimento profissional do autor. Através da programação de *web services*, a implementação de sistemas de IoT, e o desenvolvimento de comunicações cliente-servidor, foi possível aplicar os conhecimentos adquiridos durante o curso de Mestrado em Engenharia Informática – Computação Móvel.

Contudo, durante a execução destas tarefas, alguns desafios foram identificados:

- **Integração de Sistemas:** A integração de novos dispositivos com a plataforma existente apresentou desafios técnicos significativos. A comunicação entre diferentes módulos, especialmente no que diz respeito à sincronização de dados em tempo real, revelou-se mais complexa do que o inicialmente previsto.

- **Documentação e Manutenção de Código:** Foi observada a necessidade de melhorar a documentação do código, essencial para garantir a sua manutenção futura e facilitar o trabalho de outros desenvolvedores. A falta de documentação clara pode resultar em dificuldades na compreensão e atualização dos sistemas implementados.

- **Gestão de Tempo:** A gestão de tempo durante o estágio foi um desafio constante, principalmente devido à complexidade dos projetos. A pressão para cumprir os prazos, sem comprometer a qualidade do trabalho, exigiu uma boa organização e priorização das tarefas.

### 4.2. Propostas de Melhoria

Com base na análise crítica acima, são propostas as seguintes melhorias para otimizar futuras práticas e projetos:

- **Melhoria na Integração de Sistemas:** Recomenda-se a implementação de um protocolo de comunicação mais robusto e a criação de um framework modular que facilite a integração de novos dispositivos. A utilização de técnicas de machine learning para prever falhas de comunicação e a otimização dos processos de sincronização de dados também deve ser considerada.

- **Mudar o Algoritmo de Encriptação das Nidus:** Atualmente, as Nidus utilizam o algoritmo de encriptação SHA1, que é considerado inseguro face aos padrões atuais de segurança. Sugere-se a migração para um algoritmo mais seguro, como o SHA-512, que oferece uma proteção muito mais robusta contra ataques de força bruta e outros métodos de quebra de encriptação.

- **Senslive – Encriptação de Dados Sensíveis:** Embora a encriptação de dados sensíveis esteja implementada no frontend da plataforma Senslive, é crucial que esta proteção seja estendida também ao lado do servidor. A falta de encriptação no servidor expõe dados críticos a potenciais vulnerabilidades. Portanto, recomenda-se a implementação de um sistema de encriptação de ponta-a-ponta, garantindo que os dados sejam protegidos em todas as etapas de processamento.

- **Gestão de Tempo:** Sugere-se a adoção de metodologias ágeis, como o “Scrum”, para melhorar a gestão do tempo e dos recursos. O uso de sprints curtos e revisões regulares do progresso permitiria um ajuste contínuo das prioridades e uma melhor adaptação às mudanças de requisitos.

## 5. Conclusão

O estágio curricular realizado na “CapTemp”, integrado no Mestrado em Engenharia Informática – Computação Móvel, constituiu uma experiência enriquecedora, onde se consolidaram conhecimentos teóricos e se desenvolveram novas competências técnicas. Ao longo do estágio, foram conduzidos vários projetos, destacando-se a implementação e otimização da plataforma “Senslive”, o desenvolvimento do dispositivo “Nidus” com capacidades de localização GPS, e a criação de aplicações móveis como o “CT Discover” e a versão móvel da plataforma “Senslive”. Estes projetos permitiram um contacto direto com tecnologias avançadas e emergentes, como a IoT, a programação de *web services*, e a comunicação cliente-servidor, fundamentais para o sucesso dos sistemas desenvolvidos.

O projeto “Senslive” exigiu uma abordagem metódica na centralização dos sistemas de monitorização de sensores, com foco na segurança e na eficiência de dados. A introdução de encriptação de dados no *frontend*, e a subsequente proposta de extensão dessa encriptação para o lado do servidor, sublinha a preocupação com a segurança de informações sensíveis, uma área de crescente importância no desenvolvimento de soluções tecnológicas. Da mesma forma, o desenvolvimento do “Nidus” com capacidades GPS e do “CT Discover”, envolveu desafios técnicos, nomeadamente na integração de diferentes sensores e na sincronização dos dados em tempo real. A adoção de estratégias como a comunicação via protocolo UDP e a utilização de algoritmos de encriptação, como o SHA1, demonstrou a capacidade de superar dificuldades e garantir a eficácia das soluções implementadas.

Durante o estágio, foram enfrentadas diversas dificuldades que, em última análise, contribuíram para o crescimento profissional. Entre estas, destacam-se os desafios na gestão de tempo, a complexidade da integração de novos dispositivos e a necessidade de assegurar a manutenção e a escalabilidade dos sistemas desenvolvidos. A aplicação de metodologias ágeis, como *Kanban*, e a implementação de práticas de documentação rigorosas, foram estratégias fundamentais para mitigar estas dificuldades e garantir a qualidade dos resultados.

A experiência adquirida ao longo do estágio não só permitiu a aplicação prática dos conhecimentos teóricos, mas também fomentou o desenvolvimento de competências essenciais, como a resolução de problemas, a gestão de projetos e a adaptação a novas tecnologias. O estágio revelou-se uma etapa crucial na formação, proporcionando uma compreensão aprofundada das práticas e desafios do setor de engenharia informática, preparando o autor deste relatório para uma integração eficaz no mercado de trabalho.

Em conclusão, o estágio curricular na “CapTemp” alcançou plenamente os seus objetivos, contribuindo de forma significativa para o desenvolvimento pessoal e profissional, e oferecendo uma base sólida para enfrentar futuros desafios na carreira de engenharia informática. A combinação de teoria e prática, aliada à capacidade de inovar e de resolver problemas complexos, destacou-se como um dos principais ganhos deste percurso académico, evidenciando o valor do estágio como parte integrante da formação de um engenheiro informático.

## 6. Bibliografia

- [1] R. Buyya e A. V. Dastjerdi, *Internet of Things: Principles and Paradigms*. Elsevier, 2016.
- [2] J. W. Webb e R. A. Reis, *Programmable Logic Controllers: Principles and Applications*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [3] A. Kurniawan, *Learning AWS IoT: Effectively manage connected devices on the AWS cloud using services such as AWS Greengrass, AWS button, predictive analytics and machine learning*. Packt Publishing Ltd, 2018.
- [4] “RUT955 RS232 / RS485 Cellular Router.” <https://teltonika-networks.com/products/routers/rut955> (accessed Jun. 17, 2024).
- [5] C. M. Kozierok, *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*, No Starch Press, 2005.
- [6] H. B. "React vs. Vue vs. Angular vs. Svelte," dev.to, 2021. [Online]. Available: <https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>. (accessed Nov 12, 2023).
- [7] T. Krotoff, "A quick comparison of React, Vue, Angular, and Svelte," GitHub Gist, 2018. [Online]. Available: <https://gist.github.com/TKrotoff/b1caa4c3a185629299ec234d2314e190>. (accessed Nov 12, 2023).
- [8] M. R. Mehrdad, "A Comparison of Vue.js, Angular, React.js, and Svelte," Medium, 2020. [Online]. Available: <https://medium.com/@mehrddad.rz/a-comparison-of-vue-js-angular-react-js-and-svelte-7042624fe32d>. (accessed Nov 12, 2023).
- [9] A. Cruz, *Getting started with Laravel 11, master the most popular PHP framework: Here continues your roadmap in the development of web applications in PHP with Laravel + Rest Api and Vue 3 Full Sack*. Andres Cruz.[] R. V. Yampolskiy, "Construction of an NP Problem with an Exponential Lower Bound," Computer Engineering and Computer Science, University of Louisville, 2015.
- [10] GL.iNet, "GL-X300B: Industrial Grade IoT Gateway." [Online]. Disponível: <https://www.glinet.com/products/gl-x300b/>. (accessed Jun. 17, 2024).
- [11] U.S. Department of Commerce, National Institute of Standards and Technology, "FIPS PUB 180-4: Secure Hash Standard," Apr. 2015. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>
- [12] Google, "CRC32 Implementation," Fuchsia, [Online]. Available: [https://fuchsia.googlesource.com/third\\_party/wuffs+/HEAD/std/crc32/README.md](https://fuchsia.googlesource.com/third_party/wuffs+/HEAD/std/crc32/README.md) (accessed May 10, 2024).