



Full Stack Developer

Mestrado em Engenharia Informática – Computação Móvel

Diogo Filipe Duarte Rodrigues

Leiria, setembro de 2024



Full Stack Developer

Mestrado em Engenharia Informática – Computação Móvel

Diogo Filipe Duarte Rodrigues

Estágio realizado sob a orientação do Professor Doutor Luís Alexandre Lopes Frazão e sob supervisão de João Agostinho

Leiria, setembro de 2024

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 Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos (se aplicável).

Agradecimentos

Finda esta etapa do meu percurso, importa agradecer aos que me acompanharam e apoiaram, ao longo desta longa jornada. Especialmente, aos meus pais e ao meu irmão, que sempre me apoiaram e incentivaram e que tornaram possível a concretização deste objetivo.

À Joana, por toda a paciência e apoio incondicional durante este percurso.

Ao professor Doutor Luís Frazão, meu orientador, pela disponibilidade e auxílio ao longo deste desafio.

Aos meus colegas de mestrado e parceiros de estágio, Pedro e Tomás, pela constante ajuda e boa vontade que traziam, quando os dias eram mais complicados.

Por fim, mas não menos importante, à CapTemp, nomeadamente ao Sr. Carlos Domingues, Luís Silva e João Agostinho, pelo acolhimento, oportunidade, compreensão e apoio dado durante os 9 meses de estágio.

Resumo

O presente relatório foi realizado no âmbito da unidade curricular de Estágio do Mestrado em Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão, a qual pertence ao Instituto Politécnico de Leiria. O Estágio teve duração de 9 meses e foi realizado de forma totalmente presencial na CapTemp, empresa que é especializada no desenvolvimento de soluções IIoT, de sistemas de monitorização, supervisão e de controlo remoto.

Esta unidade curricular tem como objetivo proporcionar uma experiência no contexto empresarial, permitindo a aquisição de novos conhecimentos e a aplicação dos diversos conceitos aprendidos ao longo do percurso académico, para alcançar os resultados esperados dentro do prazo estipulado. O estágio teve como propósito a migração da plataforma *web* existente, SensLive, na empresa, para uma versão mais eficiente e rentável, utilizando tecnologias modernas. Adicionalmente, foi desenvolvida uma aplicação *mobile* que desempenha a mesma função da plataforma *web*, mas com a particularidade de ser *cross-platform*.

O resultado do estágio foi uma plataforma *web* renovada e uma nova aplicação *mobile*, ambas desenvolvidas para aumentar a satisfação dos clientes e facilitar o acesso ao SensLive a partir de qualquer lugar. Além disso, tanto a plataforma *web* como a aplicação *mobile* estão agora prontas para avançar para a última fase antes de serem colocadas em produção, melhorando o desempenho geral do SensLive e atraindo novos clientes. Estas soluções, representam uma inovação no mercado de monitorização de dispositivos IIoT.

Palavras-chave: IIoT, monitorização, plataforma *web*, aplicação *mobile*, SensLive, *cross-pltaform*.

Abstract

This report was carried out as a part of the Internship course of the master's degree in computer engineering – Mobile Computing at the School of Technology and Management, which belongs to the Polytechnic Institute of Leiria. The Internship lasted 9 months and was carried out entirely in person at CapTemp, a company that specialises in developing IIoT solutions, monitoring, supervision and remote-control systems.

The aim of this curricular unit is to provide experience in a business context, enabling the acquisition of new knowledge and the application of the various concepts learnt throughout the academic course, in order to achieve the expected results within the stipulated timeframe. The purpose of the internship was to migrate the company's existing web platform, SensLive, to a more efficient and profitable version, using modern technologies. In addition, a mobile application was developed that performs the same function as the web platform, but with the particularity of being cross-platform.

The result of the internship was a revamped web platform and a new mobile application, both developed to increase customer satisfaction and facilitate access to SensLive from anywhere. In addition, both the web platform and the mobile application are now ready to move on to the final stage before being put into production, improving SensLive's overall performance and attracting new customers. These solutions, represent a breakthrough in the IIoT device monitoring market.

Keywords: IIoT, monitoring, web platform, mobile application, SensLive, cross-platform.

Índice

Originalidade e Direitos de Autor	I
Agradecimentos.....	II
Resumo.....	III
Abstract.....	IV
Lista de Figuras.....	VIII
Listagem de Código.....	IX
Lista de Tabelas.....	X
Lista de Siglas e Acrónimos	XI
1. Introdução	1
1.1. Enquadramento e Problema.....	2
1.2. Motivação e Objetivos	2
1.3. Estrutura do Relatório	3
2. Caracterização da Entidade de Acolhimento.....	4
3. Programa de Estágio	6
3.1. Plano de trabalho.....	6
3.2. Metodologia de desenvolvimento	7
3.3. Descrição de objetivos de cada requisito funcional	8
4. Estado da Arte	11
4.1. Tecnologias front-end.....	11
4.1.1. React.....	11
4.1.2. Vue.js.....	11
4.1.3. AngularJS	12
4.1.4. Ember.js.....	12
4.2. Tecnologias back-end	13

4.2.1.	Laravel.....	13
4.2.2.	Django	13
4.2.3.	CodeIgniter.....	14
4.2.4.	Symfony	14
4.3.	Sistemas de gestão de base de dados.....	15
4.3.1.	PostgreSQL	15
4.3.2.	SQLite	15
4.3.3.	MySQL.....	16
4.3.4.	MariaDB.....	16
4.4.	Tecnologias <i>cross-platform</i>.....	17
4.4.1.	Flutter	17
4.4.2.	React Native	17
4.4.3.	Xamarin.....	18
4.5.	Tecnologias de desenvolvimento local	19
4.5.1.	Laragon.....	19
4.5.2.	XAMPP	19
4.5.3.	WampServer.....	19
4.6.	Sistemas de controlo de versões	20
4.6.1.	Git.....	21
4.6.2.	Mercurial	21
4.6.3.	Subversion (SVN)	21
4.7.	Ferramentas de testes.....	22
4.7.1.	Postman	22
4.7.2.	Insomnia	23
4.7.3.	Swagger.....	23
4.8.	Ambientes integrados de desenvolvimento	24
4.9.	Indústria IIoT	24
5.	Trabalho relacionado.....	27
6.	Desenvolvimento.....	32

6.1.	Arquitetura	35
6.2.	Contexto geral da plataforma.....	39
6.3.	Processos desenvolvidos	40
6.3.1.	SensLive Web.....	41
6.3.2.	SensLive Mobile.....	65
7.	Conclusão	76
7.1.	Trabalho futuro	77
8.	Referências	78

Lista de Figuras

Figura 1 - Exemplo do funcionamento de <i>branches</i>	21
Figura 2 - Arquitetura generalizada de um sistema IIoT	25
Figura 3 - Dashboard da plataforma Akenza.....	28
Figura 4 - <i>Dashboard</i> da plataforma ThingsBoard	29
Figura 5 - <i>Dashboard</i> do serviço IoT Device Management	30
Figura 6 - Arquitetura geral do SensLive	38
Figura 7 - Hierarquia dos principais elementos do SensLive	39
Figura 8 - Diagrama de encadeamento de funcionalidades desenvolvidas	41
Figura 9 - Página do SensLive com distinção de ambas as partes.....	42
Figura 10 - Página de subscrições de produtos	44
Figura 11 - Formulário tradução de valores	46
Figura 12 - Lista de subscrições disponíveis para compra	48
Figura 13 - Formulário de confirmação das subscrições	50
Figura 14 - Diagrama de sequência referente à criação de pagamentos com cartão de crédito.....	52
Figura 15 - Tabela com o histórico dos pagamentos	53
Figura 16 - Pedido enviado ao <i>endpoint</i> do SensLive	59
Figura 17 - Página das notificações.....	61
Figura 18 - Página de estatísticas	62
Figura 19 - Página inicial do SenLive web.	67
Figura 20 - Página inicial do SensLive <i>mobile</i>	67
Figura 21 - Página de gestão de subscrições de produtos.....	70
Figura 22 - Página de definições de email.....	71
Figura 23 - Página de notificações	72
Figura 24 - Extensão para visualização de <i>dashboards</i>	75

Listagem de Código

Listagem 1 - Função com o pedido feito à API do easypay	52
Listagem 2 - Função que retorna o access token à API do PayPal	55
Listagem 3 - Função utilizada para criar uma <i>order</i>	57
Listagem 4 - Exemplo de uma resposta recebida pelo <i>webhook</i>	59
Listagem 5 - Exemplos de uso do <i>i18n</i>	64
Listagem 6 - Exemplo de um ficheiro <i>i18n.js</i>	64

Lista de Tabelas

Tabela 1 - Plano de estágio resumido.....	7
Tabela 2 - Objetivos dos requisitos funcionais desenvolvidos	9
Tabela 3 - Comparação de <i>frameworks</i> de <i>front-end</i>	12
Tabela 4 - Comparação de <i>frameworks</i> de <i>back-end</i>	14
Tabela 5 - Comparação de sistemas de gestão de base de dados.....	16
Tabela 6 - Comparação de tecnologias <i>cross-platform</i>	18
Tabela 7 - Comparação das tecnologias de desenvolvimento local	20
Tabela 8 - Comparação de sistemas de controlo de versões.....	22
Tabela 9 - Comparação das ferramentas de testes	23
Tabela 10 - SensLive comparado com as soluções concorrentes	31
Tabela 11 - Tecnologias utilizadas no desenvolvimento	34

Lista de Siglas e Acrónimos

ANA	Aeroportos de Portugal
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
CO2	Dióxido de Carbono
CoAP	<i>Constrained Application Protocol</i>
CSS	<i>Cascading Style Sheets</i>
CSV	<i>Comma-separated values</i>
DDoS	<i>Distributed denial-of-service</i>
EC2	<i>Elastic Compute Cloud</i>
EDP	Energias de Portugal
EFTA	<i>European Free Trade Association</i>
ESTG	Escola Superior de Tecnologia e Gestão
GSM	<i>Global System for Mobile communication</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IDE	<i>Integrated Development Environment</i>
IIoT	<i>Industrial Internet of Things</i>
iOS	<i>iPhone Operating System</i>
IoT	<i>Internet of Things</i>
Lda	Limitada
LoRaWAN	<i>Long Range Wide Area Network</i>
LwM2M	<i>Lightweight Machine to Machine</i>
MPA	<i>Multi-page Application</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
MVC	<i>Model-View-Controller</i>
NB-IoT	<i>NarrowBand-Internet of Things</i>
ORM	<i>Object-relational mapping</i>
PDF	<i>Portable document format</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
RDS	<i>Relational Database Service</i>

REST	<i>Representational State Transfer</i>
SHA-1	<i>Secure Hash Algorithm 1</i>
SMS	<i>Short Message Service</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SNMP	<i>Simple Network Management Protocol</i>
SPA	<i>Single-page Application</i>
SQL	<i>Structured Query Language</i>
TI	Tecnologias de Informação
UE	União Europeia
UI	<i>User Interface</i>
URL	<i>Uniform Resource Locator</i>
UWP	<i>Universal Windows Platform</i>

1. Introdução

Nos dias de hoje, vivemos numa era de inovação tecnológica sem precedentes, onde a digitalização e a automação desempenham papéis cruciais em diversos setores. A introdução de sensores inteligentes e dispositivos conectados transformou a forma como as indústrias trabalham, permitindo uma monitorização precisa e constante de leituras sensíveis. Esta revolução tecnológica é impulsionada pela Indústria 4.0 e pelo IIoT (*Industrial Internet of Things*), que têm promovido a análise avançada de dados e a comunicação em tempo real.

A Indústria 4.0 e a IIoT estão a redefinir os limites da eficiência e da produtividade industrial. Com o crescimento de sensores, as empresas podem agora recolher dados em tempo real de processos, máquinas e ambientes. Estes dados são essenciais para a tomada de decisões informadas e para a otimização dos processos operacionais. A capacidade de obter informações precisas e instantâneas permite às empresas identificar e resolver problemas de forma antecipada, reduzir tempos de inatividade e melhorar a qualidade dos produtos [1].

Uma plataforma que permita a leitura em tempo real dos dados fornecidos pelos sensores torna-se, assim, uma ferramenta indispensável. Estas plataformas não só agregam dados de diversas fontes, mas também facilitam a sua análise e visualização de forma compreensível e útil para os utilizadores. Esta capacidade de integrar e interpretar grandes volumes de dados de sensores em tempo real oferece às empresas uma visão completa das suas operações, permitindo-lhes reagir rapidamente a quaisquer anomalias ou oportunidades de melhoria.

Para os clientes, a existência de uma plataforma robusta e eficiente de monitorização em tempo real significa maior confiança nos produtos e serviços oferecidos. A transparência nos dados permite uma melhor gestão dos recursos, menor desperdício e uma operação sustentável. Além disso, a capacidade de obter informações precisas e atualizadas sobre o desempenho e o estado dos produtos em uso pode melhorar significativamente a experiência do cliente e aumentar a satisfação.

Resumindo, o desenvolvimento de soluções tecnológicas avançadas, como plataformas de monitorização em tempo real, é vital para a evolução contínua da Indústria 4.0 e da IIoT. Estas soluções permitem, não só a recolha e análise de dados críticos de sensores, mas também a tradução destes dados em ações práticas que beneficiam tanto as empresas quanto os clientes. No contexto deste estágio, o objetivo é precisamente contribuir para esta

inovação, desenvolvendo uma plataforma que potencialize as vantagens dos dados em tempo real, promovendo eficiência, fiabilidade e competitividade no mercado.

1.1. Enquadramento e Problema

O presente relatório foi elaborado no âmbito da unidade curricular de Estágio do Mestrado de Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria. Tendo como objetivo apresentar o trabalho desenvolvido ao longo do estágio na empresa CapTemp, Lda, o qual decorreu de forma totalmente presencial.

O trabalho a desenvolver, consiste principalmente na migração da solução atual do portal SensLive, MPA (*Multi-page Application*), para uma solução SPA (*Single-page Application*). Para além disto, o desenvolvimento de soluções com base nos requisitos dos clientes, integração de novos equipamentos IoT, alarmística e gestão de subscrições, foram alguns objetivos secundários a ter em atenção.

1.2. Motivação e Objetivos

Ao migrar de uma solução MPA para uma SPA, a diferença é notável. A experiência do utilizador torna-se mais fluída, graças à diminuição de recarregamentos de página e ao melhor desempenho associado à carga assíncrona de conteúdo. Esta transição destaca-se pelos benefícios em termos de agilidade e rapidez de resposta [2]. É devido a estas características, que o presente estágio foi elaborado e para o qual foram definidos objetivos.

Os objetivos foram divididos em objetivos técnicos e gerais. Os principais objetivos técnicos serão, o desenvolvimento de uma plataforma Web SPA, onde o intuito será uma melhoria significativa da experiência do utilizador, e da integração de novas funcionalidades, as quais desempenharão um papel crucial na nova versão da plataforma SensLive, nomeadamente a funcionalidade dos pagamentos que foi um objetivo que foi traçado como fundamental para esta nova versão.

Em relação aos objetivos gerais, estes incluem, a aplicação prática dos conceitos adquiridos ao longo do percurso académico em ambiente empresarial, demonstrando autonomia no desenvolvimento e na resolução de problemas e ainda, a oportunidade de aprender novas tecnologias e conceitos durante este processo.

Além dos objetivos técnicos e gerais delineados anteriormente, é importante destacar as contribuições significativas que esta migração de uma solução MPA para uma SPA trará tanto para os clientes quanto para a empresa. Para os clientes, a transição para uma plataforma Web SPA resultará numa experiência para o utilizador mais fluída e eficiente, com menos tempo de carregamento de páginas e maior agilidade na navegação. Isto irá traduzir-se numa maior satisfação do cliente e possivelmente num aumento na retenção e fidelização. Já para a empresa, a implementação desta nova solução representará uma atualização tecnológica importante, permitindo acompanhar as tendências do mercado e oferecer um produto mais competitivo. Além disso, a integração de novas funcionalidades na plataforma SensLive reforçará a posição da empresa no setor, ampliando a sua base de clientes e produzindo novas oportunidades de negócio. Adicionalmente, a migração para uma solução tecnologicamente mais recente, simplificará a escalabilidade do sistema, permitindo à empresa expandir e adaptar-se às necessidades do mercado de forma mais eficiente.

1.3. Estrutura do Relatório

O presente relatório está estruturado em seis capítulos:

- No segundo capítulo, será delineada a caracterização da instituição de acolhimento, e serão detalhados alguns aspetos específicos do estágio;
- No terceiro capítulo, é apresentado o programa de estágio, nomeadamente o plano, a metodologia e a descrição dos objetivos de cada requisito funcional;
- No quarto capítulo, serão abordados o estudo e a comparação das tecnologias referentes a cada área, assim como uma descrição geral sobre a Indústria IIoT, apresentando o estado da arte;
- No quinto capítulo, será apresentado o desenvolvimento das soluções, mais concretamente as tecnologias escolhidas para fazê-lo, um contexto geral da solução, os processos desenvolvidos, o trabalho relacionado e a arquitetura da solução;
- Por fim, no último capítulo, serão expostas as conclusões do trabalho realizado e sugestões para trabalho futuro.

2. Caracterização da Entidade de Acolhimento

O estágio curricular teve duração de 9 meses, tendo início no dia 25 de setembro de 2023 e fim no dia 17 de junho de 2024. Foi realizado na empresa CapTemp, Lda [3], situada em Pombal, numa modalidade totalmente presencial. A CapTemp, Lda, é uma empresa privada, fundada em 2013, que pertence ao setor de serviços de gestão estratégica. Esta empresa, é portuguesa, especializada no desenvolvimento de soluções IIoT, de sistemas de monitorização, supervisão e de controlo remoto [4]. Foram os primeiros a obter a certificação conforme a norma EN12830:2018, referente a registadores de temperatura para transporte, armazenamento e distribuição de mercadorias sensíveis à temperatura – teste, desempenho e competência, e o guia “WELMEC 7.2 – 2018”, que simboliza a cooperação entre as autoridades de metrologia legal dos estados membros da UE (União Europeia) e da EFTA (*European Free Trade Association*), garantindo assim, total conformidade com a legislação europeia para registadores de temperatura, atendendo integralmente aos requisitos da metrologia legal. A CapTemp tornou-se assim, a primeira empresa europeia com uma solução totalmente certificada, denominada de SensLive (*Software hardware*) e com aprovação do modelo, de acordo com a norma EN12830:2018 [4]. A CapTemp, conta com vários clientes nacionais, tais como Nobre ¹, Porminho ², M.Cunha ³, Panidor ⁴, ANA aeroportos ⁵, etc. e também clientes internacionais, como BETC FULLSIX ⁶, Thales ⁷, Planeta DeAgostini ⁸, entre outros.

A empresa aplica uma estratégia de expansão internacional, onde aposta no crescimento e desenvolvimento da sua rede de parceiros internacionais. Os principais objetivos destas parcerias é, promover a gestão, divulgação e promoção dos produtos CapTemp e o desenvolvimento e procura de novas soluções, que satisfaçam as exigências dos clientes [4].

Nos 11 anos de existência da CapTemp, a dedicação da equipa à melhoria contínua dos resultados, levou ao seu reconhecível prestígio. Desde a assistência pré e pós-venda para apoiar integralmente os clientes, à valorização de parcerias sólidas com fornecedores e

¹ <https://www.nobre.pt/>.

² <http://www.porminhoalimentacao.com/>.

³ <https://www.mcunha.net/>.

⁴ <https://www.panidor.pt/pt/home>.

⁵ <https://www.ana.pt/pt>.

⁶ <https://betcfullsix.com/en>.

⁷ <https://www.thalesgroup.com/pt-pt>.

⁸ <https://www.planetadeagostini.pt/>.

clientes, a empresa tem-se destacado. A disseminação da política da qualidade a todas as partes interessadas é uma prioridade, assim como o estabelecimento de um padrão de qualidade para os serviços, orientado para a inovação e desenvolvimento de aplicações informáticas adaptadas às necessidades específicas de cada cliente. A monitorização constante dos objetivos dos processos assegura a melhoria contínua dos serviços prestados pela CapTemp, enquanto a procura permanente pela melhor forma de otimizar os processos internos permanece no centro da estratégia da empresa. Este compromisso, ao longo de mais de uma década, solidifica a posição da CapTemp como referência no setor [4].

3. Programa de Estágio

Neste capítulo, será delineado o plano de trabalho realizado ao longo dos 9 meses de estágio, abordando a metodologia de desenvolvimento utilizada na implementação da plataforma SensLive, tanto para a versão web como para a *mobile*. Serão também apresentados os requisitos funcionais, como também uma breve descrição dos objetivos de cada requisito desenvolvido, fornecendo uma compreensão clara do âmbito e propósito de cada atividade realizada durante o estágio.

3.1. Plano de trabalho

O presente estágio teve início com a ambientação na CapTemp. Essa etapa abrangeu a familiarização com o espaço físico de trabalho e a apresentação dos principais elementos e pessoas que constituíram o apoio durante o período de estágio. Além disso, foi dedicado um tempo significativo para proporcionar uma compreensão aprofundada do contexto relacionado à solução já existente e da base de dados que estava a ser utilizada. Esta introdução, permitiu uma imersão integral no projeto e uma percepção mais clara dos requisitos fundamentais a ser desenvolvidos. Tendo em conta que eram três estagiários, houve naturalmente repartição dos requisitos funcionais a ser realizados.

Assim, após esta integração e a instalação das tecnologias necessárias, foi iniciado o desenvolvimento das novas funcionalidades da plataforma SensLive web. Foi desenvolvido tanto o *back-end* como o *front-end* simultaneamente. Em relação a algumas funcionalidades, nomeadamente aquelas que utilizavam APIs (*Application Programming Interface*) externas, foi necessário estabelecer alguns períodos para analisar e compreender a documentação associada a estas interfaces. Esse processo foi crucial para garantir uma integração correta e eficiente. Assim, quando a solução fosse mudada para o ambiente de produção, todos os procedimentos estariam devidamente implementados e testados, assegurando um funcionamento contínuo e sem falhas.

Depois do desenvolvimento da solução web do SensLive, procedeu-se ao desenvolvimento da solução *mobile*. Para esta, foi necessário dedicar algum tempo à compreensão e estudo da nova linguagem que ia ser utilizada para esta versão *cross-platform*. Ainda englobada nesta versão do SensLive, foi desenvolvida uma “extensão” que

funcionava como uma aplicação independente ao SensLive, que permitia aos utilizadores visualizarem os seus *dashboards* em qualquer dispositivo.

Por último, foram realizadas várias melhorias gerais na versão *mobile* do SensLive. Estas melhorias incluíram desde aprimoramentos visuais até à simplificação e eliminação de redundâncias no código. Com isto, o intuito foi otimizar a experiência do utilizador e tornar a aplicação mais eficiente e intuitiva.

É importante referir que, ao longo do desenvolvimento de ambas as soluções, foi sempre escrita a documentação referente às duas versões, de modo a possibilitar uma simples e fácil integração no relatório de estágio.

Com isto, na Tabela 1, estão representadas de forma resumida as tarefas elaboradas durante o estágio, distribuídas ao longo dos meses em que o estágio decorreu.

Tabela 1 - Plano de estágio resumido

Tarefas	Estado	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun
Ambientação à empresa	Concluído										
Instalação das tecnologias necessárias ao desenvolvimento	Concluído										
Levantamento de requisitos	Concluído										
Desenvolvimento da solução web - <i>backend</i>	Concluído										
Desenvolvimento da solução web - <i>frontend</i>	Concluído										
Estudo e compreensão da nova linguagem <i>cross-platform</i>	Concluído										
Desenvolvimento da solução <i>mobile</i>	Concluído										
Extensão dos <i>dashboards</i>	Concluído										
Melhorias gerais	Concluído										
Desenvolvimento da documentação (relatório de estágio)	Concluído										

3.2. Metodologia de desenvolvimento

Em relação à metodologia de desenvolvimento, não foi seguida uma metodologia formal. A abordagem praticada consistiu numa orientação direta e contínua por parte do supervisor, que desempenhou um papel crucial na distribuição e acompanhamento dos requisitos funcionais.

O desenvolvimento do projeto foi estruturado de forma interativa e incremental. Inicialmente, os requisitos funcionais eram atribuídos pelo supervisor conforme as necessidades da nova plataforma. Cada requisito recebia uma explicação detalhada para garantir a compreensão completa dos objetivos e funcionalidades necessárias. Esta abordagem permitiu uma adaptação flexível às mudanças e desafios que surgiam ao longo do desenvolvimento.

Uma vez atribuído o requisito, o desenvolvimento era iniciado, seguido de um processo de testes. Estes testes tinham como finalidade verificar a funcionalidade. Este ciclo de

desenvolvimento, testes e correções repetia-se até que o requisito fosse considerado concluído. Apenas após a finalização de um requisito é que se iniciava o próximo, seguindo assim um padrão de trabalho sequencial e bem definido.

A ferramenta Git ⁹ foi utilizada somente na fase final do projeto, especificamente para integrar o trabalho desenvolvido individualmente pelos estagiários. Esta etapa de integração envolveu a junção dos diferentes módulos e componentes, garantindo que todos se combinassem de maneira coerente e funcional no produto final. A utilização tardia do Git foi um ponto crítico, pois exigiu um esforço considerável para resolver alguns conflitos e assegurar a consistência da solução.

Esta abordagem, apesar de não seguir metodologias tradicionais como *Agile* ¹⁰ ou *Scrum* ¹¹, apresentou algumas vantagens significativas. A flexibilidade permitiu uma resposta rápida a mudanças e necessidades emergentes, enquanto o foco em requisitos específicos promoveu um compromisso total a cada funcionalidade desenvolvida. Adicionalmente, a interação com o supervisor e o feedback contínuo proporcionaram um ambiente de aprendizagem e melhoria contínua.

Contudo, a falta de um planejamento formal detalhado também trouxe desvantagens, como conflitos de desenvolvimento idêntico e algumas ineficiências. A integração do desenvolvimento apenas no final dificultou a resolução de conflitos.

Concluindo, a metodologia de desenvolvimento adotada, baseada na atribuição direta de requisitos funcionais e na utilização do Git para integração final, permitiu a realização bem-sucedida do projeto. Para futuros projetos, a consideração de metodologias mais estruturadas pode potencialmente melhorar o planejamento, organização e eficiência do processo de desenvolvimento.

3.3. Descrição de objetivos de cada requisito funcional

Neste subcapítulo, encontram-se todos os requisitos funcionais delineados que foram desenvolvidos durante o estágio, onde também são apresentados os respectivos objetivos de cada um. Os requisitos funcionais, foram planejados e atribuídos pelo supervisor conforme

⁹ <https://git-scm.com/>.

¹⁰ <https://www.agilealliance.org/agile101/>.

¹¹ <https://www.scrum.org/resources/what-scrum-module>.

as novas exigências da nova versão do SensLive. De seguida está a Tabela 2, que apresenta todos os requisitos funcionais desenvolvidos e os respetivos objetivos.

Tabela 2 - Objetivos dos requisitos funcionais desenvolvidos

Requisitos funcionais	Objetivos
Gestão de subscrições de Produtos, SMS e Email	Criar, editar e eliminar subscrições por um Administrador SensLive.
Ferramenta para tradução automática de ficheiros i18n.js	Criação de uma ferramenta, que traduza um ficheiro i18n.js para uma linguagem escolhida.
Editar perfil	Permitir ao utilizador escolher o <i>dashboard</i> da página inicial.
Relatórios personalizados	Permitir às empresas criar relatórios personalizados com base nos seus requisitos.
Gráficos	Visualizar e transferir gráficos com base nos <i>inputs</i> , sensores, intervalo de tempo e tipo de gráfico.
Tradução de valores	Criar, editar e eliminar tradução de valores.
Integração easypay API	Integrar a API da easypay, de forma a ser possível pagar subscrições dentro da plataforma SensLive, nomeadamente com Multibanco, Cartão de crédito e MB Way.
Subscrições de Produtos	Possibilidade de comprar subscrições de sensores através da plataforma SensLive.
Subscrições de SMS e Email	Possibilidade de comprar créditos de Email e saldo de SMS através da plataforma SensLive.
Pagamentos	Página com o histórico dos dados referentes a todos os pagamentos feitos pela empresa.
Tutorial	Facilitar a assistência ao cliente através da criação de um tutorial online que permita ao utilizador seguir com facilidade e adquirir

	competências essenciais de como utilizar as funcionalidades da plataforma SensLive.
Integração com vários ficheiros i18n.js de diferentes linguagens	Utilização de vários ficheiros i18n.js com diferentes linguagens, permitindo alterar a linguagem da plataforma facilmente.
Integração PayPal API	Integrar a API da easypay, de forma a ser possível pagar subscrições dentro da plataforma SensLive, nomeadamente com PayPal.
Notificações	Ter notificações para as subscrições.
<i>About</i>	Mostrar na página <i>About</i> , dados referentes à subscrição de produtos para o utilizador.
Adicionar eventos	Adicionar eventos a cada dispositivo externo.
Estatísticas	Permitir ao Administrador do SensLive visualizar diversas estatísticas dentro da plataforma SensLive.
Definições de SMS	Associar um dispositivo a cada local de uma empresa.
Manutenção	Agendar manutenções e alertar os utilizadores das mesmas.
Senslive <i>Mobile</i>	Desenvolver, numa tecnologia <i>cross-platform</i> , uma versão <i>mobile</i> do SensLive, com todas as funcionalidades anteriormente desenvolvidas para a versão web do SensLive.
Extensão para <i>dashboards</i>	Possibilitar aos utilizadores a exclusiva visualização de <i>dashboards</i> a partir de uma aplicação separada do SensLive.
Interfaces semelhantes	Manter sempre que possível a semelhança com a interface utilizada na versão anterior do SensLive.

4. Estado da Arte

Para alcançar a melhor performance possível ao longo do estágio, foi essencial dedicar tempo ao estudo e comparação de diversas tecnologias em várias áreas, nomeadamente, *front-end*, *back-end*, sistemas de base de dados, desenvolvimento *cross-platform*, plataforma local de desenvolvimento, controlo de versões e testes.

Assim, neste capítulo irão ser apresentadas algumas tecnologias para cada uma dessas áreas, sendo feitas comparações sempre que possível. Além disso, serão especificados alguns dos sensores utilizados pela empresa para monitorização e também uma breve abordagem sobre o conceito da indústria IIoT.

4.1. Tecnologias front-end

Para as tecnologias *front-end*, foi sempre tida em conta a sua flexibilidade e performance. Com isto, de seguida encontram-se algumas das tecnologias mais utilizadas atualmente para esta área.

4.1.1. React

React ¹², também conhecido como React.js ou ReactJS, é uma biblioteca de desenvolvimento de UI (*User Interface*) *open-source* baseada em JavaScript. Apesar de ser uma biblioteca em vez de uma *framework*, é amplamente utilizada no desenvolvimento web. Esta biblioteca surgiu pela primeira vez em maio de 2013, e é agora uma das bibliotecas mais comuns para o desenvolvimento web [5]. O React destaca-se pelo seu tamanho *lightweight*, pela sua performance e flexibilidade consistentes em comparação com algumas *frameworks*. No entanto, a sua curva de aprendizagem é frequentemente apontada como uma das poucas desvantagens desta tecnologia, pois em comparação com outras, pode não ser tão fácil de dominar [6].

4.1.2. Vue.js

Vue.js ¹³, é uma *framework front-end* JavaScript que se baseia em HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript, fornecendo um modelo de programação baseado em componentes, que facilita o desenvolvimento eficiente de UI e de

¹² <https://react.dev/>.

¹³ <https://vuejs.org/>.

SPA [7]. Lançada em fevereiro de 2024, esta é conhecida como sendo uma das *frameworks* mais simples, reutilizáveis e flexíveis. O seu tamanho *lightweight*, acrescenta aos seus pontos positivos quando comparada com outras *frameworks* [6].

4.1.3. AngularJS

AngularJS ¹⁴, lançada em outubro de 2010 é uma *framework* JavaScript MVC (*Model-View-Controller*), utilizada para criar aplicações web dinâmicas com uma arquitetura estruturada. Esta permite utilizar HTML como linguagem modelo e facilita a extensão da sintaxe do HTML para representar os componentes da aplicação de forma clara e concisa [8]. Apesar de apresentar uma performance equiparável com outras *frameworks*, esta tecnologia deixa a desejar quando comparada em termos de flexibilidade, integração, curva de aprendizagem e tamanho [6].

4.1.4. Ember.js

Ember.js ¹⁵, é uma *framework* JavaScript MVC, utilizada para criar aplicações flexíveis do lado do cliente. Esta possibilita o desenvolvimento de código organizado e estruturado, em comparação com outras *frameworks* JavaScript MVC. Esta abordagem facilita a criação de aplicações online mais robustas e flexíveis [9]. No entanto, comparando esta tecnologia com algumas das alternativas mais recentes, as diferenças são notórias. Desde a curva de aprendizagem até à performance, emerge uma distinção que evidencia as desvantagens das soluções mais antigas [6].

Após a breve descrição de cada *framework* de *front-end*, será apresentada a Tabela 3, que compara estas mesmas ferramentas entre si.

Tabela 3 - Comparação de *frameworks* de *front-end*

	React	Vue.js	AngularJS	Ember.js
Ano de Fundação	2013	2014	2010	2011
Curva de aprendizagem	Moderada	Baixa	Alta	Moderada
Flexibilidade	Alta	Alta	Baixa	Moderada
Performance	Alta	Alta	Alta	Moderada

¹⁴ <https://angularjs.org/>.

¹⁵ <https://emberjs.com/>.

Tamanho da framework	Leve	Leve	Grande	Médio
Documentação	Abundante	Abundante	Abundante	Abundante
Integração	Simples	Simples	Complexa	Complexa

Ao observar a tabela, conseguimos analisar que todas as *frameworks* têm alta performance e documentação abundante. A curva de aprendizagem, a flexibilidade e o tamanho, trazem as maiores diferenças entre estas tecnologias [6]. Apesar das diversas diferenças nas características apresentadas, a seleção de uma destas *frameworks* dependerá das necessidades específicas a desenvolver e das preferências adotadas pela empresa.

4.2. Tecnologias back-end

Para as tecnologias *back-end*, foi sempre tida em conta a utilização do padrão MVC e da sua curva de aprendizagem. Assim, de seguida encontram-se algumas das tecnologias mais utilizadas atualmente para esta área.

4.2.1. Laravel

Laravel ¹⁶, é uma *framework back-end* PHP (PHP: *Hypertext Preprocessor*), para desenvolvimento de sistemas web que utilizam o padrão MVC. Esta é descrita como uma “Framework PHP clara e sofisticada, com configuração mínima e sintaxe expressiva. Faz uso de roteamento RESTful e inclui ORM (*Object-relational mapping*)” [10]. Para além disso, destaca-se pela sua simplicidade, rapidez, flexibilidade e facilidade de manutenção. Comparando com outras tecnologias, as maiores diferenças seriam a sua curva de aprendizagem e a sua integração, sendo estas baixa e simples respetivamente [11].

4.2.2. Django

A web *framework* Django ¹⁷, é uma das ferramentas *open-source* mais populares atualmente, esta tem a capacidade de acelerar o desenvolvimento de aplicações web baseadas em Python. Com um foco de “desenvolvimento rápido e design limpo e pragmático”, esta ferramenta, quando implementada num servidor web, possibilita aos programadores a rápida criação de uma interface web completa, segura e escalável [12]. Django, apresenta uma

¹⁶ <https://laravel.com/>.

¹⁷ <https://www.djangoproject.com/>.

curva de aprendizagem um pouco mais elevada do que algumas *frameworks*, no entanto em relação às outras características, esta faz uso da arquitetura MVC, tem ORM integrado, tem suporte à API REST e apresenta uma integração simples [13].

4.2.3. CodeIgniter

O CodeIgniter ¹⁸, é um software *open-source* utilizado para criar rapidamente a estrutura web necessária para o desenvolvimento de páginas e sites dinâmicos em linguagem PHP. Essa estrutura adota um modelo *Loose Coupling* e segue o padrão de desenvolvimento MVC [14]. O modelo *Loose Coupling*, refere-se a um princípio de *software* em que os componentes ou sistemas interagem com dependências mínimas entre si, permitindo uma manutenção, flexibilidade e escalabilidade mais fáceis [15]. No entanto, esta tecnologia não apresenta ORM integrado, e a sua integração é bastante complexa quando comparada com outras *frameworks*.

4.2.4. Symfony

O Symfony ¹⁹, representa uma *framework* de alto desempenho para aplicações web baseada em PHP, que segue o padrão MVC. Como uma ferramenta *open-source*, esta oferece uma variedade de componentes e ferramentas destinadas a simplificar e acelerar o processo de desenvolvimento de aplicações web. O Symfony, visa a promover as melhores práticas no desenvolvimento web, procurando sempre fornecer um código estruturado que seja intuitivo e flexível [16]. Mesmo assim, esta *framework* é considerada difícil de aprender devido a sua alta curva de aprendizagem e complexidade [13].

Depois de referenciar algumas das *frameworks* de *back-end* existentes atualmente, será apresentada na Tabela 4, a comparação destas ferramentas entre si.

Tabela 4 - Comparação de *frameworks* de *back-end*

	Laravel	Django	CodeIgniter	Symfony
Linguagem Principal	PHP	Python	PHP	PHP
Arquitetura MVC	Sim	Sim	Sim	Sim

¹⁸ <https://www.codeigniter.com/>.

¹⁹ <https://symfony.com/>.

Curva de aprendizagem	Baixa	Moderada	Baixa	Alta
ORM integrado	Sim (Eloquent)	Sim (Django ORM)	Não	Sim (Twig)
Suporte à API REST	Sim	Sim	Possível com extensões	Sim
Integração	Simples	Simples	Complexa	Simples

Após observar a tabela, conseguimos concluir que todas utilizam a arquitetura MVC, sendo que a principal diferença entre elas é a curva de aprendizagem, sendo que ambas as tecnologias Django e Symfony apresentam curvas de aprendizagem superiores às suas contrapartes [13]. Também conseguimos verificar que a ferramenta CodeIgniter em comparação com as outras alternativas é a que se encontra mais limitada sobre as características comparadas. Mais uma vez, a seleção de uma destas *framework*, irá depender das necessidades do projeto e das preferências adotadas pela empresa.

4.3. Sistemas de gestão de base de dados

Em relação a sistemas de gestão de base de dados, foi tida em consideração, sistemas, onde o seu modelo de dados fosse relacional e ainda que apresentassem elevado desempenho e extensibilidade. Com isto, de seguida encontram-se alguns dos sistemas mais utilizados atualmente nesta área.

4.3.1. PostgreSQL

PostgreSQL²⁰, é um sistema relacional avançado de base de dados *open-source*, baseado em SQL (*Structured Query Language*) e reconhecido pela sua confiabilidade, flexibilidade e adesão a padrões técnicos. Com a capacidade de suportar tipos de dados relacionais e não relacionais, esta ferramenta destaca-se como uma das opções mais conformes, estáveis e equilibradas disponíveis atualmente [17].

4.3.2. SQLite

SQLite²¹, é um sistema de gestão de base de dados relacional que se destaca pela sua rapidez e pela exigência mínima de suporte do *host environment*. Este sistema não opera

²⁰ <https://www.postgresql.org/>.

²¹ <https://www.sqlite.org/index.html>.

como uma base de dados independente, mas sim como uma base de dados incorporada. Isto significa que os utilizadores podem incorporar funcionalidades de base de dados diretamente do código [18]. Esta ferramenta não apresenta replicação síncrona nem replicação assíncrona nativamente, e ainda, possui extensibilidade baixa em comparação com outras tecnologias [19].

4.3.3. MySQL

MySQL ²², desenvolvido pela Oracle ²³, é um sistema de gestão de base de dados relacionais *open-source* baseada em SQL. Com suporte de diversas plataformas, como Linux, Unix e Windows, este sistema é considerado bastante versátil. Embora este seja utilizado numa vasta gama de aplicações, o MySQL é mais frequentemente associado a aplicações web [20]. Em comparação de características como, o desempenho, uso de replicação síncrona e assíncrona e segurança integrada, o MySQL, destaca-se como uma das principais opções entre os sistemas de gestão de base de dados. Porém, no que diz respeito à extensibilidade, este sistema, apresenta algumas limitações [21].

4.3.4. MariaDB

MariaDB ²⁴, é uma plataforma de gestão de base de dados relacionais *open-source*, baseada no MySQL. Originada através da derivação do desenvolvimento do MySQL, o MariaDB, não apenas fornece uma alternativa ao MySQL, mas também atua como um substituto eficaz. Uma das características destacadas desta plataforma é a sua segurança da base de dados [22]. Devido a este sistema ser derivado do MySQL, as características entre si não apresentam grandes diferenças, sendo bastante idênticos entre si.

Após a breve análise de cada sistema de gestão de base de dados, será apresentada na Tabela 5, a comparação entre estes sistemas relacionais.

Tabela 5 - Comparação de sistemas de gestão de base de dados

	PostgreSQL	SQLite	MySQL	MariaDB
Modelo de dados	Relacional	Relacional	Relacional	Relacional
Desempenho	Excelente	Bom	Excelente	Excelente

²² <https://www.mysql.com/>.

²³ <https://www.oracle.com/pt/>.

²⁴ <https://mariadb.org/>.

Replicação Síncrona	Sim	Não	Sim	Sim
Replicação Assíncrona	Sim	Não	Sim	Sim
Extensibilidade	Alta	Baixa	Moderada	Moderada
Segurança Integrada	Mecanismos avançados de segurança	Segurança básica	Mecanismos avançados de segurança	Mecanismos avançados de segurança
Documentação	Abundante	Abundante	Abundante	Abundante

Depois de observar a tabela, conseguimos identificar que o único sistema que se encontra mais desalinhado com o resto dos sistemas de base de dados é o SQLite, visto que este, desde ao desempenho até à segurança, encontra-se inferior [23]. No entanto, o sistema tem de ser escolhido com base nas necessidades do projeto em causa e das preferências da empresa.

4.4. Tecnologias *cross-platform*

Para as tecnologias *cross-platform*, foi tida sempre em conta, no mínimo, a compatibilidade com os sistemas operativos Android, iOS e Windows. Tendo isto em consideração, de seguida, serão apresentadas algumas das tecnologias que foram ponderadas para o desenvolvimento multiplataforma.

4.4.1. Flutter

Desenvolvido pelo Google e lançado a maio de 2017, o Flutter ²⁵, destaca-se pela sua capacidade de criar aplicações visualmente apelativas e consistentes em diversas plataformas. Este oferece uma abordagem *cross-platform*, que permite o desenvolvimento de aplicações tanto para Android iOS e Windows. Com um ambiente de desenvolvimento reativo, o Flutter utiliza a linguagem Dart, que adota um modelo de *widgets*, permitindo a construção de interfaces flexíveis e dinâmicas.

4.4.2. React Native

React Native ²⁶, também conhecido como RN, é uma estrutura *cross-platform* amplamente utilizada para o desenvolvimento de aplicações móveis. Inicialmente lançado

²⁵ <https://flutter.dev/>.

²⁶ <https://reactnative.dev/>.

pelo Facebook como um projeto em março de 2015, o React Native rapidamente se tornou uma das principais escolhas para o desenvolvimento *mobile* em apenas alguns anos. Baseado em JavaScript, este possibilita a criação de aplicações renderizadas nativamente para iOS e Android [24] e também para UWP (*Universal Windows Platform*) e Web. Características como desempenho e flexibilidade encontram-se um pouco inferiores comparativamente a tecnologias como o Flutter [25].

4.4.3. Xamarin

Xamarin ²⁷, lançado a fevereiro de 2013, é uma estrutura *cross-platform* para desenvolver aplicações móveis utilizando a linguagem de programação, C#. Existem diversas estruturas disponíveis para o desenvolvimento de aplicações *cross-platform*, muitas das quais, utilizam HTML e JavaScript. O Xamarin destaca-se em relação a essas estruturas, pois, fornece uma abordagem única ao utilizar a linguagem, C# e *runtime*, que funciona em três plataformas móveis (Android, iOS, Windows) [26]. Tal como o React Native, o Xamarin, comparado com outra tecnologia, como o Flutter, apresenta ambos o desempenho e a flexibilidade limitada [27].

Após a breve descrição, de cada uma das ferramentas de desenvolvimento *cross-platform*, está apresenta na Tabela 6, a comparação destas ferramentas entre si.

Tabela 6 - Comparação de tecnologias *cross-platform*

	Flutter	React Native	Xamarin
Ano de Fundação	2017	2015	2013
Linguagem	Dart	JavaScript (React)	C#
Integração Nativa	Sim	Sim	Sim
Documentação	Abundante	Abundante	Abundante
Desempenho	Excelente	Bom	Bom
Flexibilidade	Alta	Moderada	Moderada

Ao observar a tabela, é possível perceber que todas as ferramentas são bastante semelhantes entre si, sendo que apenas há uma pequena diferença entre o desempenho e a flexibilidade entre as três tecnologias, dando a vantagem ao Flutter [28]. Ainda assim, a

²⁷ <https://dotnet.microsoft.com/en-us/apps/xamarin>.

tecnologia tem de ser escolhida com base nas necessidades do projeto e as preferências da empresa.

4.5. Tecnologias de desenvolvimento local

Em relação a tecnologias de desenvolvimento local, foi tida em consideração como requisito essencial a integração com PHP e Apache. Portanto, de acordo com este requisito, irão ser descritas algumas tecnologias ponderadas para o desenvolvimento local.

4.5.1. Laragon

Laragon²⁸, é uma *framework* de desenvolvimento local, que simplifica a criação e gestão de ambientes de servidor web. Esta, oferece uma solução segura, fácil e eficiente para os desenvolvedores que necessitam de um servidor local, para desenvolver e testar aplicações web [29]. Com a sua integração com PHP, Apache, Nginx, MySQL, NodeJS, etc, esta promove a facilidade do desenvolvimento de software de forma simples e simplificada.

4.5.2. XAMPP

XAMPP²⁹, derivado de *Cross-Platform* (X), Apache (A), MySQL (M), PHP (P) e Perl (P), é a ferramenta que junta todas estas tecnologias num só lugar. XAMPP, é um servidor web *cross-platform* utilizado para desenvolvimento de websites. Esta é considerada versátil e abrangente, pois proporciona a configuração rápida e conveniente de um servidor web completo. Esta tecnologia permite ser utilizada em Windows, Linux e macOS, tendo também uma flexibilidade alta, no entanto o nível de segurança é considerado baixo [30].

4.5.3. WampServer

WampServer³⁰, é um conjunto de software frequentemente utilizado por programadores para criar aplicações web dinâmicas. Este, possibilita a configuração de servidores web em máquinas locais, proporcionando um ambiente completo de desenvolvimento para testar websites. O termo “Wamp” é um acrónimo que representa as primeiras letras de Windows, Apache, MySQL e PHP. Estes componentes, desempenham papéis essenciais no desenvolvimento web, e a sua combinação forma uma ferramenta robusta e eficaz para a criação de aplicações web dinâmicas [31]. WampServer não tem suporte para outras

²⁸ <https://laragon.org/>.

²⁹ <https://www.apachefriends.org/>.

³⁰ <https://www.wampserver.com/en/>.

linguagens, sendo que a sua flexibilidade também está bastante limitada em comparação com outras tecnologias como o Laragon e o XAMPP [32].

Depois de entendermos um pouco de cada uma das tecnologias de desenvolvimento local, podemos observar na Tabela 7, estas tecnologias comparadas entre si.

Tabela 7 - Comparação das tecnologias de desenvolvimento local

	Laragon	XAMPP	WampServer
Plataformas compatíveis	Windows	Windows, Linux e macOS	Windows
Interface gráfica	Sim	Sim	Sim
Suporte para outras linguagens	Sim	Sim	Não
Flexibilidade	Alta	Alta	Baixa
Documentação	Abundante	Abundante	Média
Segurança	Alta	Baixa	Alta

Depois de uma análise aos resultados apresentados na tabela, podemos concluir que, o Laragon possui, em comparação com as outras tecnologias, os melhores resultados nas características comparadas. O XAMPP apresenta uma segurança baixa em comparação, sendo a única diferença em relação ao Laragon. Também, é importante notar que o WampServer, em comparação com as suas contrapartes, não oferece suporte para outras linguagens, tem uma flexibilidade e documentação reduzida [32]. Porém, a tecnologia a escolher, tem de ter em conta as necessidades do projeto e as preferências da empresa.

4.6. Sistemas de controlo de versões

Para os sistemas de controlo de versões, foi tido sempre em atenção a existência de histórico e *branches*. Em relação a recursos, procurou-se também sistemas onde a escalabilidade e a performance fossem de acordo com as necessidades do projeto. Com isto em mente, de seguida estão apresentados alguns dos sistemas de controlo de versões mais utilizados atualmente.

4.6.1. Git

O Git ³¹, é o sistema de controlo de versões mais utilizado atualmente. Este, é um sistema distribuído e permite rastrear as alterações que são feitas nos ficheiros, de forma a ter um histórico de alterações, possibilitando, se necessário a reversão para versões específicas. O Git também simplifica a colaboração entre colaboradores, com a utilização de Ramos (*Branches*) [33]. Estes permitem aos colaboradores trabalharem em funcionalidades isoladas sem interferir no código principal, tendo uma performance e escalabilidade lineares e facilitando a implementação de novas funcionalidades ou correções de bugs. No entanto é considerado um sistema onde a curva de aprendizagem por vezes não é baixa. Na Figura 1, está representado como a colaboração através da utilização de *branches* funciona.

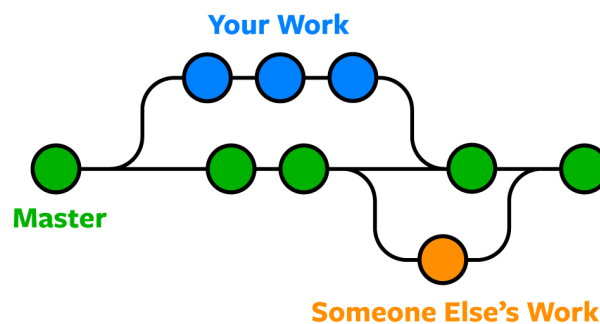


Figura 1 - Exemplo do funcionamento de *branches*

4.6.2. Mercurial

O Mercurial ³², destaca-se como uma excelente ferramenta de controlo de versões. Reconhecido pela sua rapidez, escalabilidade e eficiência no tratamento de projetos de diversas dimensões, esta ferramenta apresenta-se como uma alternativa ao Git. Para além disso, o Mercurial suporta uma ampla variedade de fluxos de trabalho e pode facilmente melhorar as suas funcionalidades por meio de extensões [34].

4.6.3. Subversion (SVN)

Subversion (SVN) ³³, é um sistema de controlo de versões que possibilita o rastreamento das alterações em arquivos e diretórios ao longo do tempo. O Apache Subversion, é um projeto *open-source*, responsável pelo desenvolvimento e manutenção do software Subversion. Este sistema, proporciona uma arquitetura servidor-cliente para a gestão de ficheiros e diretórios com versões [35]. Este sistema apresenta uma performance inferior

³¹ <https://git-scm.com/>.

³² <https://www.mercurial-scm.org/>.

³³ <https://subversion.apache.org/>.

comparado a outros sistemas de controlo de versões e utiliza uma abordagem diferente e limitada para funcionamento de *branches* [36].

De seguida, apresenta-se a Tabela 8, que mostra a comparação dos sistemas de controlo de versões anteriormente mencionados.

Tabela 8 - Comparação de sistemas de controlo de versões

	Git	Mercurial	Subversion (SVN)
Tipo de sistema	Distribuído	Distribuído	Centralizado
Histórico	Sim	Sim	Sim
Branches	Sim	Sim	Limitados
Curva de aprendizagem	Moderada	Baixa	Baixa
Escalabilidade	Alta	Alta	Alta
Performance	Alta	Alta	Moderada

Após a observação da tabela, é possível perceber que todos os sistemas são relativamente idênticos, apenas existindo diferença na curva de aprendizagem, onde o Git apresenta uma aprendizagem mais complexa em relação aos outros sistemas. Para além disso, o sistema Subversion (SVN), apresenta *Branches* limitadas e uma performance inferior, comparando com as suas contrapartes [36]. Apesar disto, o sistema de controlo de versões, depende totalmente das necessidades e preferências da empresa.

4.7. Ferramentas de testes

Em relação às ferramentas de testes, visto que estas servem especialmente para testar a API da plataforma, o único requisito a ter em atenção foi exatamente esse, a possibilidade de testar a API com facilidade. Assim, abaixo estão descritas algumas das ferramentas de testes de APIs mais utilizadas atualmente.

4.7.1. Postman

O Postman ³⁴, é uma plataforma API para contruir e utilizar APIs. O Postman simplifica cada etapa do ciclo de vida da API e agiliza a colaboração, para se poder criar melhores APIs

³⁴ <https://www.postman.com/>.

mais rapidamente [37]. Esta plataforma permite aos programadores enviar pedidos HTTP (*HyperText Transfer Protocol*), visualizar respostas e automatizar testes para garantir o funcionamento correto dos *endpoints*.

4.7.2. Insomnia

Insomnia³⁵, é uma ferramenta gratuita disponível para Windows, MacOS e Linux. Esta, oferece uma interface de usuário intuitiva para enviar pedidos à API e analisar respostas. Apesar de ser uma ferramenta leve, apresenta recursos avançados, como a criação de dados fictícios, encadeamento de pedidos e automação de testes. Estas funcionalidades facilitam o trabalho dos programadores, como também impulsionam significativamente a produtividade [38].

4.7.3. Swagger

O Swagger³⁶, é uma ferramenta *open-source* destinada ao design e documentação de APIs, que auxilia os programadores no desenvolvimento, criação, documentação e testagem de APIs RESTful mais eficientemente. O Swagger simplifica o desenvolvimento, testes e implementação de APIs, proporcionando uma abordagem mais rápida e fácil para os seus utilizadores [39].

Após a breve descrição de cada ferramenta de testes, está apresentada, na Tabela 9, a comparação destas.

Tabela 9 - Comparação das ferramentas de testes

	Postman	Insomnia	Swagger
Interface gráfica	Sim	Sim	Sim
Teste de API	Sim	Sim	Sim
Automatização de testes	Sim	Sim	Sim
Documentação	Abundante	Abundante	Abundante
Extensibilidade	Sim	Sim	Sim

³⁵ <https://insomnia.rest/>.

³⁶ <https://swagger.io/>.

Ao analisar a tabela, percebemos que as ferramentas de testes possuem características bastante idênticas entre si, sendo que de todas as características comparadas, todas elas têm os mesmos resultados [40]. Com isto, a escolha de uma ferramenta de testes, está dependente do das preferências da empresa e do utilizador.

4.8. Ambientes integrados de desenvolvimento

O Visual Studio Code [41], frequentemente abreviado de VS Code, é um ambiente de desenvolvimento integrado (IDE), desenvolvido pela Microsoft. Este editor *open-source* é utilizado para diversas linguagens de programação, graças à sua extensibilidade e suporte a uma variedade de tecnologias.

No que diz respeito a alternativas, existem várias opções disponíveis, tais como Sublime Text, Vim, Eclipse, entre outras. É importante notar que não há uma vantagem significativa em realizar comparações diretas entre ferramentas IDE, uma vez que a escolha de cada utilizador irá depender das suas preferências individuais.

4.9. Indústria IIoT

O conceito da Internet das Coisas (IoT) desempenha um papel crucial em contextos industriais, particularmente na forma da Internet das Coisas Industrial (IIoT). Este paradigma emergente, representa uma potencial revolução no fabrico e produção, aproveitando extensivamente uma variedade significativa de dispositivos de deteção interconectados e a integração de tecnologias informáticas modernas [42]. A Indústria 4.0, também conhecida como a 4ª Revolução Industrial, marca uma nova era na indústria, impulsionada pelo crescimento exponencial em disponibilidade, acessibilidade e capacidade de sensores, processadores e tecnologias de comunicação. A adoção em massa da tecnologia 4.0 é evidente no aumento significativo do uso de dispositivos incorporados em aplicações industriais. Este fenómeno desperta um crescente interesse na área da indústria IoT (*Internet of Things*), uma extensa rede de dispositivos, sistemas e aplicações que interagem, comunicam e compartilham informações entre si, com o ambiente externo e com os seres humanos [43].

Desta forma, é possível dividir a estrutura de uma arquitetura típica de um sistema IIoT em 4 partes, nomeadamente, Camada de perceção (*Perception Layer*), Camada de rede (*Network Layer*), Camada de processamento (*Processing Layer*) e Camada de aplicação

(*Application Layer*). De seguida encontra-se a Figura 2, onde está demonstrada uma arquitetura generalizada de um sistema IIoT [44].

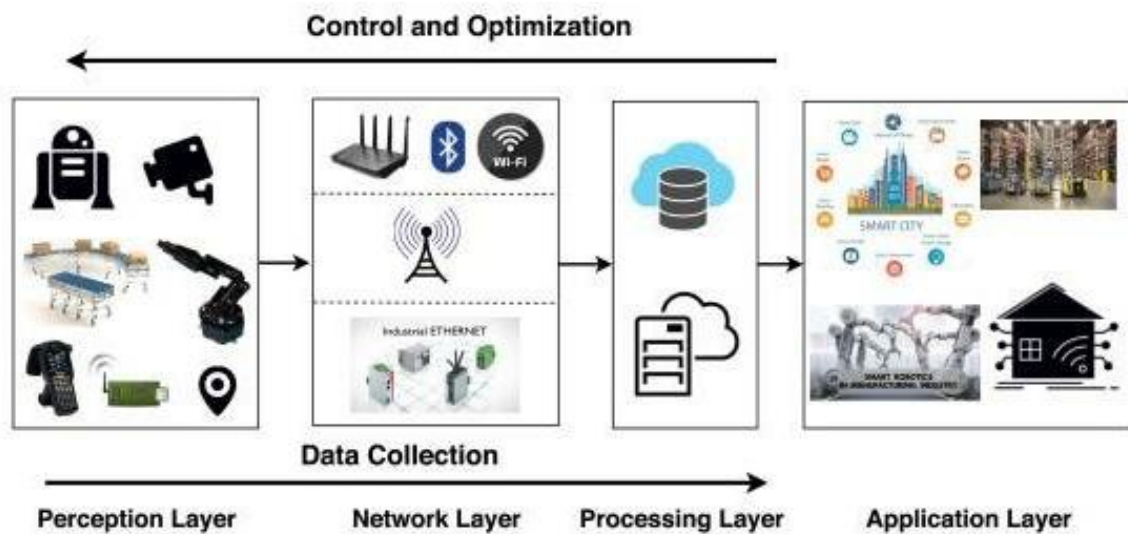


Figura 2 - Arquitetura generalizada de um sistema IIoT

A Camada de percepção, denominada por vezes de *Things*, (Coisas), constitui os elementos interligados em rede que se encontram na extremidade mais remota de uma rede IoT. Estes estão posicionados o mais próximo possível da fonte de dados, sendo normalmente, compostos por sensores e atuadores [45]. Estes dispositivos IoT podem abranger um grande espectro de funcionalidades, desde sensores de temperatura, humidade, inundação, gases até câmaras robóticas, contadores e medidores de energia.

Os sensores desempenham a função de coletar dados tanto do ambiente quanto dos objetos que estão a ser monitorizados. Estes dados são posteriormente convertidos em informações práticas que os utilizadores poderão observar na camada de aplicação. Já os atuadores, estes têm a responsabilidade de regular os processos que ocorrem no ambiente monitorizado, ajustando as condições em que os dados são gerados [45].

A Camada de rede, é onde os dados dos sensores e atuadores são agregados e convertidos em fluxos digitais para processamento futuro. Após receber os dados agregados e digitalizados, esta camada encaminha-os para que possam ser processados antes de chegarem à *cloud*. Os fluxos de dados provenientes dos sensores acumulam enormes quantidades de dados em constante mudança num curto espaço de tempo. Os dados são transmitidos de forma contínua, originando imensos fluxos de dados, que se revelam demasiado volumosos para serem transferidos diretamente para a *cloud*. A complexidade

umenta quando existe uma transmissão de dados de multiplicidade de sensores de um sistema IoT [45]. Devido a este facto, a camada de rede é considerada uma camada essencial quando se referencia uma arquitetura IIoT.

Camada de processamento, ou Camada de gerenciamento de dados, atua após a recolha dos dados pelos sensores e a sua conversão para formato digital. Estes são agregados e ficam prontos para serem processados por sistemas de TI (Tecnologias de Informação). Estes sistemas podem estar localizados nas proximidades dos sensores, embora também possam ser remotamente acedidos.

Uma vez obtidas informações relevantes a partir dos dados, torna-se desnecessário transmitir a totalidade das informações recolhidas até ao momento. Este processamento adicional tem como objetivo reduzir a carga de dados enviada para os centros de dados ou para a *cloud*. Esta abordagem contribui para mitigar preocupações relacionadas com armazenamento, segurança e tempo de inatividade [45].

Por último temos a Camada de aplicação, que engloba a interpretação de padrões promissores nos dados IoT e a sua tradução em sínteses de fácil compreensão para os utilizadores, como gráficos, tabelas, *dashboards* e relatórios. Nesta fase, destacam-se programas destinados ao controlo e monitorização de dispositivos, assim como software para a gestão de processos [46].

Com isto, pode-se concluir que, a integração da IIoT na Indústria 4.0 representa uma evolução significativa nos processos de fabrico e produção. A arquitetura delineada, composta pela camada de perceção, rede, processamento e aplicação, demonstra a complexidade e interconexão necessárias para viabilizar a comunicação eficiente entre dispositivos, sistemas e utilizadores. Esta abordagem proporciona não apenas eficiência operacional, mas também abre portas para inovações contínuas e melhorias substanciais na gestão industrial. À medida que avançamos nesta era digital, a convergência da IIoT e da Indústria 4.0 promete remodelar radicalmente o panorama industrial, elevando padrões de desempenho e impulsionando o crescimento em diversos setores.

5. Trabalho relacionado

Foi realizada uma análise detalhada do mercado de competição para plataformas de monitorização de IoT, especificamente aquelas que oferecem funcionalidades semelhantes ao SensLive. Esta análise revelou uma crescente variedade de soluções tecnológicas que têm como objetivo satisfazer as necessidades de monitorização, recolha e análise de dados em tempo real, provenientes de dispositivos IoT.

Entre vários competidores identificados, foram distinguidas três soluções que oferecem plataformas robustas com capacidades avançadas de integração, escalabilidade, suporte técnico de alta qualidade e capacidade de se adaptar a diversos setores industriais.

Em primeiro lugar, está a solução desenvolvida pela empresa Akenza AG ³⁷. Esta solução oferece uma plataforma intuitiva que facilita a conexão, gestão e monitorização de dispositivos IoT. As principais funcionalidades incluem a capacidade de integrar diversos dispositivos e sensores de forma simples, configurar fluxos de dados personalizados, e monitorizar o desempenho em tempo real. A plataforma da Akenza também suporta a gestão eficiente de dados através de *dashboards* personalizáveis, alertas configuráveis e ferramentas de análise avançada. Além disso, oferece conectividade segura e compatibilidade com múltiplos protocolos IoT, assegurando uma integração com infraestruturas existentes [47]. Na Figura 3, encontra-se representado um exemplo de um *dashboard* na plataforma Akenza.

³⁷ <https://akenza.io/>

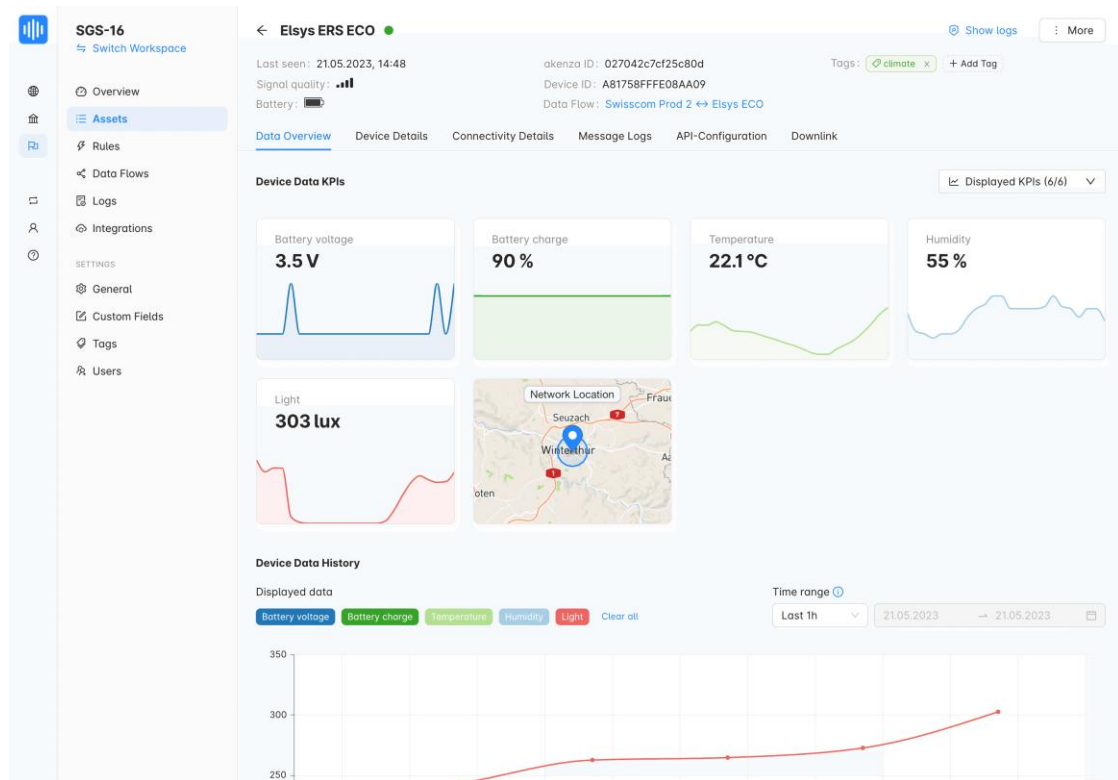


Figura 3 - Dashboard da plataforma Akenza

Outra solução bastante similar à anterior é a ThingsBoard ³⁸. Esta é uma plataforma *open-source* de gestão IoT para recolha, processamento, visualização de dados e gestão dispositivos IoT [48]. Entre as principais características destacam-se a gestão de dispositivos, a visualização de dados através de *dashboards* interativos e personalizáveis, e a criação de regras de processamento de dados em tempo real. A plataforma suporta múltiplos protocolos de comunicação, como MQTT (*Message Queuing Telemetry Transport*), CoAP (*Constrained Application Protocol*) e HTTP, garantindo flexibilidade na integração de diversos dispositivos [48]. Podemos observar na Figura 4, um exemplo de um *dashboard* na plataforma ThingsBoard.

³⁸ <https://thingsboard.io/>

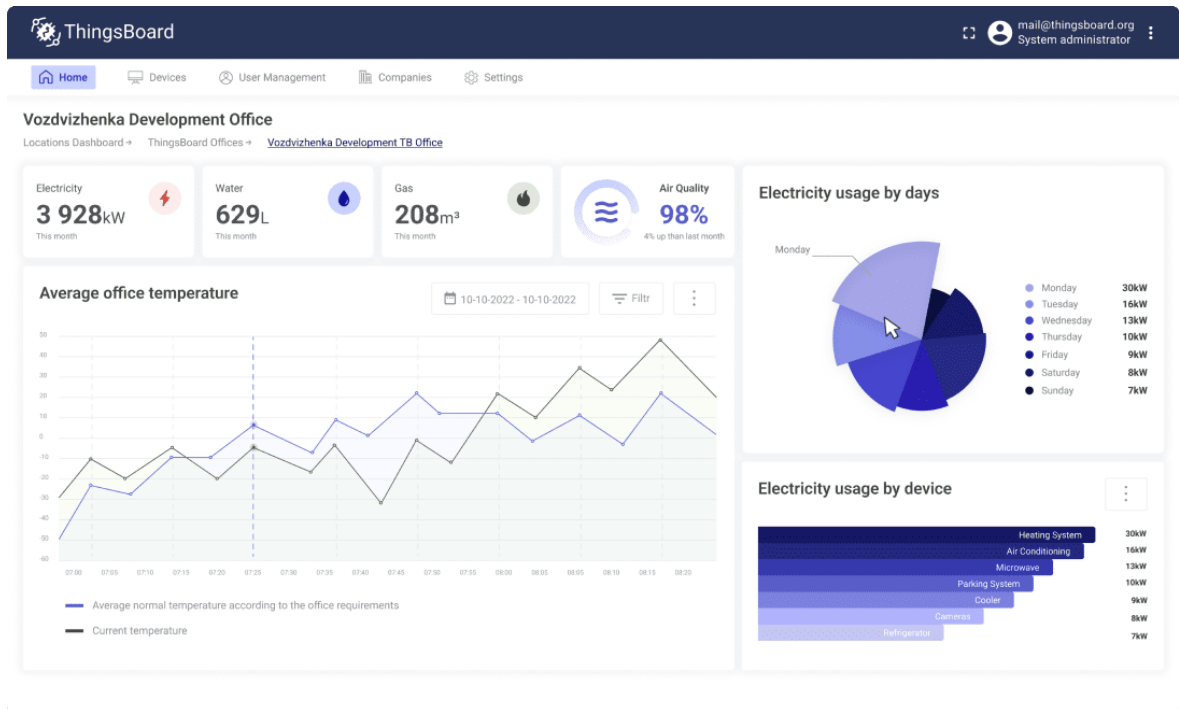


Figura 4 - Dashboard da plataforma ThingsBoard

Por último será abordado um serviço da *Amazon Web Services* (AWS), o AWS IoT Device Management ³⁹. Este oferece um conjunto abrangente de funcionalidades para gerir dispositivos IoT de forma eficiente e segura. Algumas destas incluem, a capacidade de registar e organizar dispositivos em tempo real, e atualizar *firmware* remotamente. Além disso, o AWS IoT Device Management permite a automatização de tarefas comuns de manutenção, como o acompanhamento dos dispositivos e a gestão de certificados. O serviço suporta múltiplos protocolos de comunicação, incluindo MQTT, HTTP e *WebSockets*, assegurando a flexibilidade e compatibilidade com uma ampla variedade de dispositivos [49]. Na Figura 5, está representado um *dashboard* exemplo do serviço AWS IoT Device Management.

³⁹ <https://aws.amazon.com/pt/iot-device-management/>

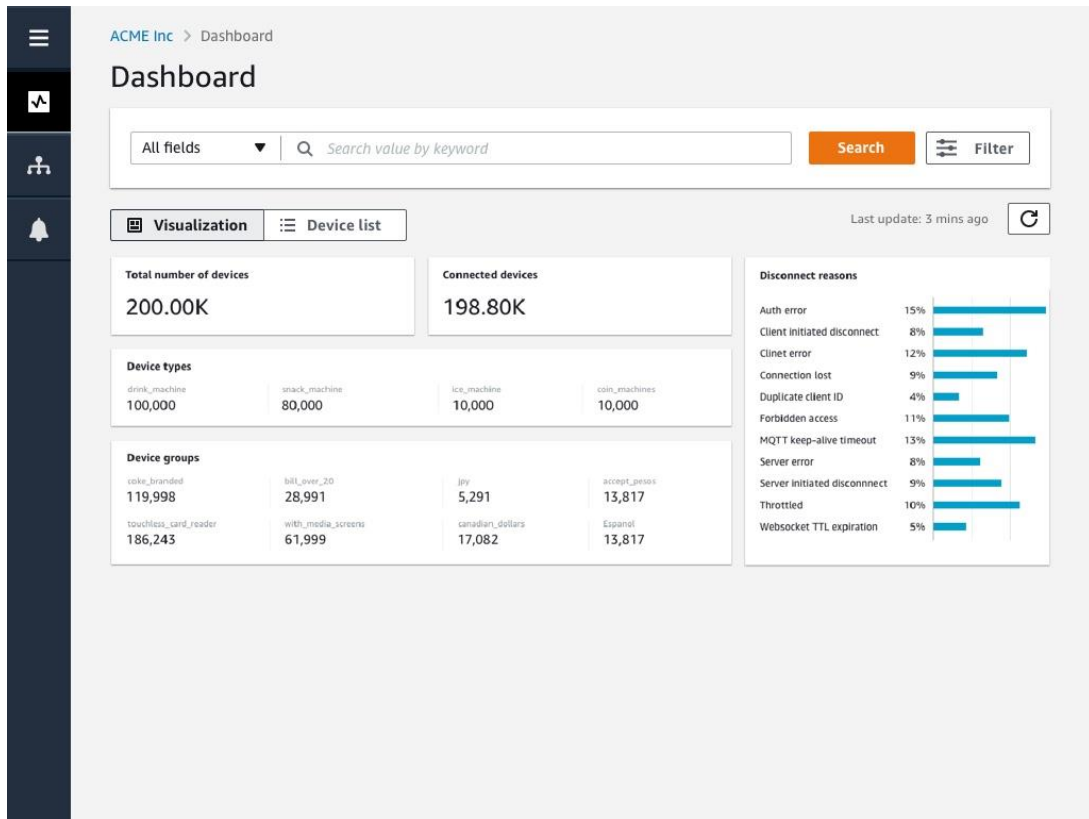


Figura 5 - Dashboard do serviço IoT Device Management

Na Tabela 10, encontra-se a comparação das soluções concorrentes com a plataforma do SensLive.

Tabela 10 - SensLive comparado com as soluções concorrentes

	SensLive	Akenza	ThingsBoard	AWS IoT DM
Ano de Criação	2011	2017	2016	2015
Dashboards personalizáveis	Sim	Sim	Sim	Não
Alertas/Alarmes	Sim	Sim	Sim	Sim
Regras	Sim	Sim	Sim	Sim
Análise em tempo real	Sim	Sim	Sim	Sim
Protocolos de comunicação	LoRaWAN, HTTP, MQTT, SNMP (Nidus), NB-IoT, Modbus (Nidus), PROFINET, BACnet, Ping	LoRaWAN, HTTP, MQTT, CoAP	MQTT, LwM2M, CoAP, HTTP	MQTT, HTTP, <i>WebSockets</i>
Gestão de dispositivos	Sim	Sim	Sim	Sim
Relatórios / Relatórios personalizáveis	Sim	Parcialmente	Parcialmente	Não

Observando a tabela, é possível identificar algumas diferenças significativas entre as soluções. A que apresenta maior déficit em termos de funcionalidades é a AWS IoT Device Management. No que diz respeito à funcionalidade de relatórios / relatórios personalizados, tanto a Akenza como a ThingsBoard apenas oferecem relatórios padrão, sem grande margem para personalização. Em contraste, o SensLive permite a criação de relatórios personalizados. Para esta comparação, foram consideradas apenas as funcionalidades mais importantes, embora pudessem ser incluídas outras características para uma comparação mais abrangente.

6. Desenvolvimento

O presente capítulo tem como objetivo apresentar todos os processos desenvolvidos ao nível da plataforma SensLive e da aplicação mobile, *Front-end* e *Back-end*. Para além disso, será feita uma comparação com algumas soluções existentes atualmente, que servem de mesmo propósito que o SensLive. Por último, é demonstrada a arquitetura do sistema de modo a dar uma melhor compreensão do contexto onde foram inseridas as soluções desenvolvidas.

Tecnologias utilizadas

Antes de dar início à descrição dos processos desenvolvidos é importante referir quais foram as tecnologias escolhidas para o desenvolvimento. Para isto, irão ser apresentadas as tecnologias que foram escolhidas, assim como as razões que fundamentam essas decisões.

- ***Front-end***

A tecnologia escolhida para o *front-end* foi o Vue.js. Esta foi um requisito da empresa, pelo que não houve qualquer autonomia para a escolha por parte dos estagiários. No entanto, esta tecnologia possuía grandes vantagens como, performance elevada, curva de aprendizagem baixa, entre outras características, como já referido no capítulo 4.1.

- ***Back-end***

Para o desenvolvimento *back-end*, a tecnologia adotada foi o Laravel, esta era um requisito imposto pela empresa o que limitou a liberdade na escolha de outra alternativa. Mesmo assim, o Laravel atualmente continua a ser uma das tecnologias mais reconhecidas. O facto de ter sido frequentemente utilizada durante a licenciatura também facilitou a sua implementação, dada a familiaridade prévia com a mesma. No capítulo 4.2 encontra-se com mais detalhe as características desta tecnologia.

- **Gestão de base de dados**

Em relação ao sistema de gestão de base de dados, foi utilizado PostgreSQL, o que foi igualmente um requisito imposto pela empresa. Este sistema, como já referido no capítulo 4.3, é um dos sistemas mais utilizados e distinguidos pela sua confiabilidade e flexibilidade.

- ***Cross-platform***

Para o desenvolvimento da aplicação mobile, optou-se pelo uso da tecnologia Flutter. Esta foi uma tecnologia que já tinha sido utilizada anteriormente para outros projetos dentro da empresa e então foi igualmente um requisito para o desenvolvimento desta solução. Mesmo assim, por esta ser uma tecnologia recente, as suas vantagens em relação a outras alternativas são evidentes, tal como mostrado no capítulo 4.4.

- **Desenvolvimento local**

Para o desenvolvimento local, foi utilizada a *framework* Laragon. A empresa aconselhou a utilização desta *framework* para gerir um servidor local para facilitar no desenvolvimento da plataforma SensLive. Dado que esta tecnologia foi vastamente usada ao longo da licenciatura, a sua escolha tornou-se ainda mais natural, beneficiando da já existente experiência com a mesma. As suas várias características, demonstradas no capítulo 4.5, também reforçam a escolha desta como uma preferência válida.

- **Controlo de versões**

Para o sistema de controlo de versões, foi adotada a ferramenta Git. A empresa já possuía um repositório centralizado no Git, utilizado ao longo do desenvolvimento para gerir as versões dos projetos em curso. Mesmo não tendo autonomia para escolher o sistema de controlo de versões, o Git, destaca-se como a ferramenta mais predominante atualmente, conforme discutido no capítulo 4.6.

- **Administração de base de dados**

Visto que para o sistema de gestão de base de dados foi escolhido o PostgreSQL, para a administração de base de dados, foi escolhido o pgAdmin ⁴⁰, que é uma ferramenta de administração *open-source* para base de dados PostgreSQL. Esta oferece uma interface gráfica intuitiva, através da web ou aplicação Windows, para criar, modificar e gerir objetos da base de dados, como tabelas e índices. Além disso, facilita a execução de consultas SQL e oferece uma visão hierárquica organizada para simplificar a navegação e administração da base de dados [50].

⁴⁰ <https://www.pgadmin.org/>

Como o pgAdmin, foi especialmente projetado para integração com sistemas PostgreSQL, tornou a escolha de outras alternativas irrelevante, devido à sua afinidade com essa plataforma.

- **Ferramenta de testes**

No que diz respeito aos testes, foi concebida total liberdade para escolher uma ferramenta, e a opção recaiu sobre o Postman. A escolha desta ferramenta foi bastante direta, uma vez que já tinha sido utilizada previamente durante a licenciatura, concedendo uma vantagem em relação a outras alternativas. Contudo, é apenas devido a este facto que a escolha da ferramenta de testes foi o Postman, para o propósito de testes, qualquer ferramenta teria sido válida, já que todas as mencionadas no capítulo 4.7 apresentam características semelhantes.

- **IDE**

Em relação ao ambiente de desenvolvimento integrado, ou IDE, optou-se pelo Visual Studio Code. Essa escolha baseou-se na familiaridade prévia com a plataforma, sendo um ambiente já conhecido. Assim como nas ferramentas de teste, é relevante notar que a escolha de outras alternativas seria válida, uma vez que as preferências de cada utilizador desempenhas um papel crucial nesse contexto.

Com isto, na Tabela 11, encontram-se as áreas e as respetivas tecnologias utilizadas durante o desenvolvimento.

Tabela 11 - Tecnologias utilizadas no desenvolvimento

Áreas	Tecnologias utilizadas
Front-end	Vue.js
Back-end	Laravel
Gestão de base de dados	PostgreSQL
Cross-platform	Flutter
Desenvolvimento local	Laragon
Controlo de versões	Git
Administração de base de dados	pgAdmin
Ferramenta de Testes	Postman

IDE (Ambiente integrado de desenvolvimento)	Visual Studio Code
--	---------------------------

6.1.Arquitetura

O planeamento da arquitetura foi delineado na totalidade pela entidade de estágio, não fazendo sentido incluir o plano detalhado neste documento, visto que não foi algo que foi pensado pelos estagiários.

No entanto, na Figura 6 encontra-se representada a arquitetura geral do SensLive. Alguns elementos da arquitetura já se encontravam desenvolvidos, nomeadamente os sensores, os coletores de dados IoT, a base de dados e parcialmente a API. A utilização de APIs externas, do uso da tecnologia Vue.js e Flutter foram os elementos incorporados na arquitetura para desenvolvimento da nova solução.

Comparando com a arquitetura anteriormente mostrada no capítulo 4.9, consegue-se traçar características idênticas. Dividindo a arquitetura do SensLive em 4 camadas, podemos observar uma arquitetura semelhante à de um sistema IIoT.

Para a camada de perceção, estão representados os sensores utilizados pela CapTemp. Entre eles encontram-se o TH3-RS485, que permite medir a temperatura, temperatura/humidade e gases, o Kea Tracker, que mede temperatura, humidade e pressão atmosférica e o Airo, que suporta medições de temperatura, temperatura/humidade e CO2 (dióxido de carbono). Quanto à conectividade dos sensores, o TH3 é um sensor com fios, enquanto o Kea Tracker e o Airo são sensores sem fios. Estes aparelhos serão instalados nos locais das empresas onde se pretende obter leituras.

Na camada de rede encontram-se os dispositivos como a NidusIT, NidusIT *plus*, Nidus-W, Nidus-C, Nidus-C *plus* e a Nidus-R. Estes dispositivos, desenhados e construídos pela CapTemp, funcionam como intermediários de comunicação com os sensores. Dependendo da conectividade dos sensores, será necessário ter um dispositivo compatível. Para sensores com fios, é possível utilizar quase todos dispositivos disponíveis, como a NidusIT, NidusIT *plus*, Nidus-C, Nidus-C *plus* e Nidus-R ⁴¹. No caso de sensores *wireless*, a comunicação é restrita aos dispositivos que suportam essa tecnologia, nomeadamente a NidusIT *plus*,

⁴¹ <https://www.capttemp.com/wired-data-collector>

Nidus-C *plus* e a Nidus-W ⁴². Entre os dispositivos, a principal diferença, além do suporte para sensores sem fios, reside no número de entradas e saídas que cada um possui.

Com exceção do KeaTracker, todos os outros sensores comunicam com as Nidus para enviar os dados recolhidos. O KeaTracker, por sua vez, comunica diretamente com o SensLive através de uma aplicação *mobile*.

No que diz respeito aos protocolos utilizados, é essencial salientar que os coletores de dados referidos, utilizam os protocolos de comunicação Modbus e SNMP (*Simple Network Management Protocol*). Por outro lado, os restantes protocolos mencionados, tais como o LoRaWAN (*Long Range Wide Area Network*), HTTP, MQTT, NB-IoT (*NarrowBand – Internet of Things*), PROFINET, BACnet e Ping, são exclusivamente utilizados pela API do SensLive. Cada um destes protocolos desempenha um papel fundamental na garantia de uma comunicação eficiente e sólida entre dispositivos e sistemas.

Em relação à camada de processamento, esta faz uso do serviço *cloud* da AWS. Para a base de dados, é utilizado o serviço Amazon RDS (*Amazon Relational Database Service*), que facilita a configuração, gestão e dimensionamento de uma base de dados relacional na AWS *cloud* [51]. Para a API e o SensLive web, é utilizado o serviço Amazon EC2 (*Amazon Elastic Compute Cloud*), que permite o aluguer de máquinas virtuais, para processar as aplicações [52]. Esta API será o ponto de receção de todos os dados recolhidos pelos sensores e dispositivos.

Além disso, a API integra outras APIs externas, como a do easypay, PayPal, Clickatell ⁴³, Vonage ⁴⁴, e SMTP ⁴⁵. Tanto a API do easypay como a do PayPal foram utilizadas para as subscrições de produtos, email e SMS. Para o envio de SMS, são usadas as APIs da Clickatell e da Vonage, e para o envio de emails, a API SMTP.

É importante salientar que, para todos os *endpoints* utilizados na API do SensLive, foram implementados múltiplos *middlewares* para garantir a segurança e acesso exclusivo a pessoas autorizadas. Os *middlewares* desempenham um papel crucial na filtragem de *requests*, autenticação de utilizadores e na proteção contra ataques, como injeções SQL e ataques de DDoS (*Distributed denial-of-service*). Além disso, ajudam a gerir o fluxo de

⁴² <https://www.capttemp.com/wireless-data-collector>

⁴³ <https://gatsby-staging.clickatell.com/>

⁴⁴ <https://www.vonage.com/communications-apis/>

⁴⁵ <https://www.smtp.com/resources/api-documentation/>

dados entre diferentes componentes do sistema, assegurando que apenas os *requests* válidos e seguros chegam à aplicação.

Falando ainda em segurança, é importante referir que existiu a utilização de um algoritmo criptográfico para as passwords, nomeadamente o SHA-1 (*Secure Hash Algorithm 1*). Este algoritmo recebe um *input* e produz um valor de *hash* de 160 *bits* (20 *bytes*) [53].

Por fim, a camada de aplicação. Esta camada engloba ambas as versões do SensLive, sendo os principais pontos de interação para os utilizadores, gerirem e observarem os dados dos seus sensores e dispositivos. Diferentemente da versão *web*, a versão *mobile* não estará incluída na *cloud*. Isto ocorre porque a aplicação *mobile* será instalada diretamente nos dispositivos dos utilizadores, permitindo uma experiência mais rápida e responsiva. A versão *mobile* irá conectar à API na *cloud* para sincronizar os dados, garantindo que os utilizadores tenham acesso às informações mais recentes, independentemente da plataforma que estejam a utilizar.

Em conclusão, pode-se afirmar que a arquitetura do SensLive se enquadra na totalidade numa arquitetura de um sistema IIoT. Ao existir integração da versão *web* e *mobile* com a API centralizada na *cloud*, o SensLive oferece uma solução robusta e eficiente para a monitorização e gestão de sensores e dispositivos industriais. Esta estrutura garante não só a disponibilidade e segurança dos dados através da *cloud*, mas também proporciona uma experiência de utilizador rápida e responsiva nas aplicações, confirmando-se assim, como uma solução completa e moderna no âmbito da IIoT.

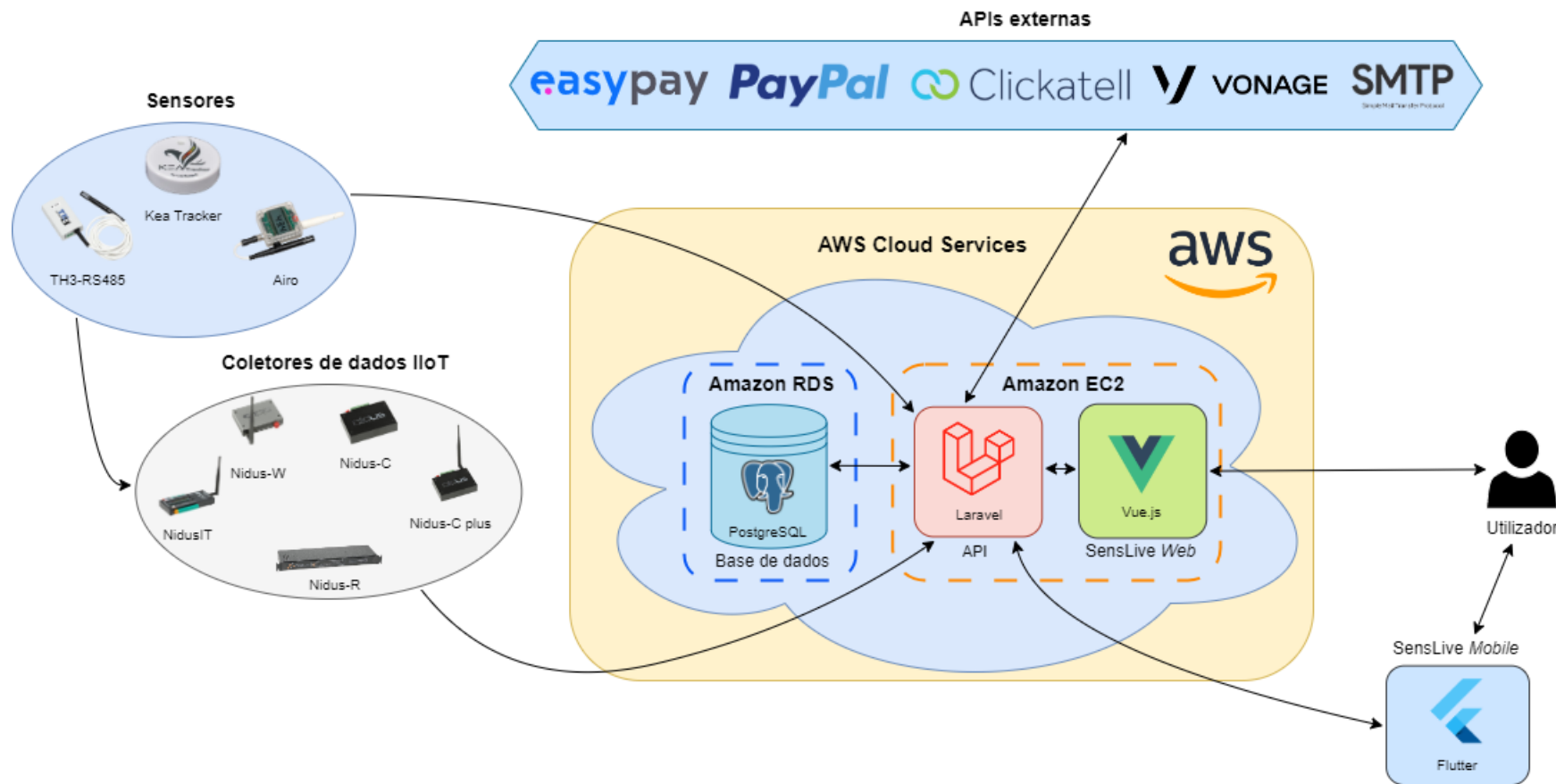


Figura 6 - Arquitetura geral do SensLive

6.2. Contexto geral da plataforma

De modo a simplificar o contexto geral do SensLive, está apresentada na Figura 7, a hierarquia dos principais elementos constituintes da solução. Esta estrutura hierárquica é fundamental para compreender como a plataforma organiza e gere os dados provenientes dos sensores. Ao clarificar as relações entre os diferentes níveis, desde o administrador do SensLive, às empresas, até aos seus dispositivos e sensores, é possível obter uma visão clara de como cada elemento contribui para o funcionamento integrado e eficiente da plataforma.

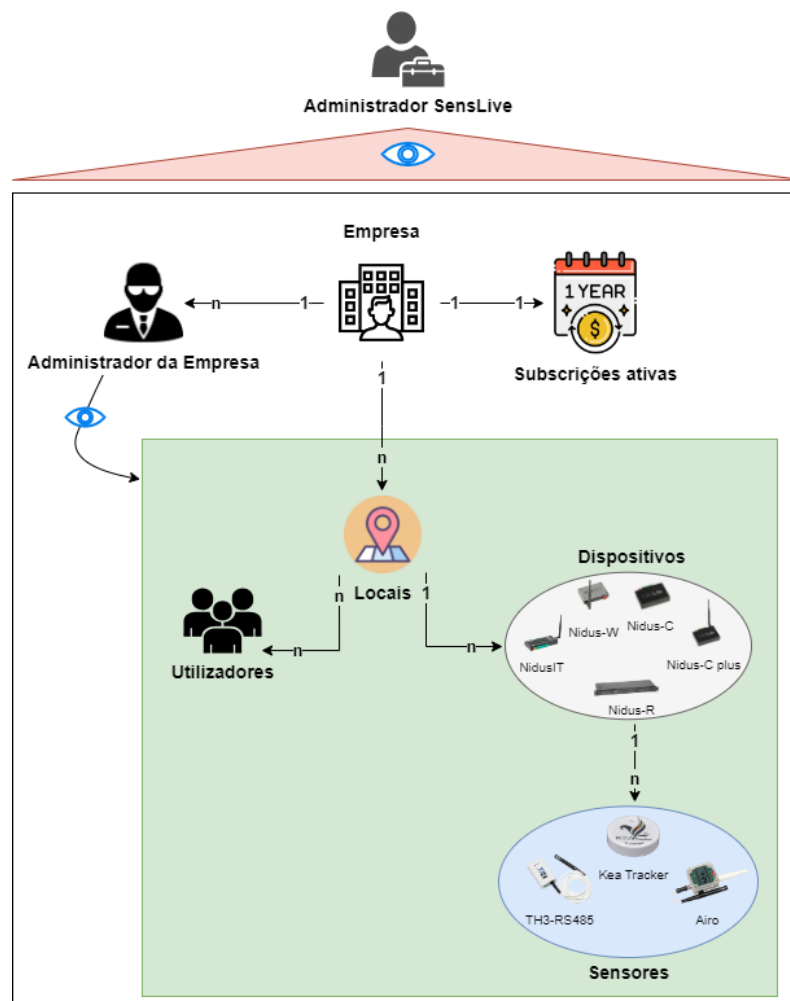


Figura 7 - Hierarquia dos principais elementos do SensLive

Começando no topo da hierarquia, temos o Administrador do SensLive. Este cargo pertence exclusivamente aos funcionários da CapTemp. Este cargo permite monitorizar todos os restantes elementos e gerir a plataforma através de várias funcionalidades exclusivas a este tipo de utilizador, como gestão de subscrições, agendamento de manutenções, entre outras.

De seguida, aparece a estrutura da Empresa. Podem existir várias empresas, sendo que cada uma delas pode ter múltiplos locais associados, vários administradores e uma subscrição ativa. Cada empresa possui apenas uma subscrição ativa associada, mas também é possível que tenha diversas subscrições no seu histórico, incluindo todas as subscrições que expiraram ao longo do tempo. Desta forma, é possível manter um registo detalhado de todas as subscrições que a empresa teve desde a sua criação.

Em relação aos administradores da empresa, este é o cargo mais elevado de uma empresa. Cada empresa pode ter múltiplos administradores. A principal função destes utilizadores é gerir a hierarquia dos vários utilizadores em cada local e administrar os diferentes locais da sua empresa. Além disso, são estes utilizadores que tratam de todas as subscrições de que a empresa usufrui e de todas as decisões referentes à empresa.

No próximo nível, temos os locais de uma empresa. Cada empresa pode possuir mais do que um local. Dentro de cada local, estão associados utilizadores e dispositivos, podendo ambos ser representados em n quantias. Normalmente, um local está associado a um local físico onde a empresa possui dispositivos e sensores que recolhem dados, estando, portanto, relacionado a coordenadas geográficas reais. Desse modo, os locais representam os diferentes pontos físicos onde a empresa opera e monitoriza as suas atividades.

Para os utilizadores de cada local, estes têm acesso aos dados gerados pelos sensores desse local. Este tipo de utilizador é especialmente designado para auxiliar na visualização das leituras dos valores recolhidos pelos vários sensores presentes no local, permitindo aos utilizadores acompanhar e analisar tanto os dados em tempo real quanto os dados previamente registados.

Por último, temos os dispositivos e os sensores. Estes desempenham um papel crucial na hierarquia, pois são responsáveis por fornecer e recolher todos os dados que são apresentados na plataforma do SensLive. Além disso, é importante referir que, um dispositivo pode conter múltiplos sensores.

6.3.Processos desenvolvidos

Este subcapítulo tem como objetivo detalhar o processo de desenvolvimento da plataforma SensLive para as versões Web e Mobile. Serão apresentadas as funcionalidades implementadas em ambas as plataformas, acompanhadas da explicação da lógica subjacente

sempre que possível. Além disso, serão incluídos excertos relevantes de código e representações visuais das interfaces quando apropriado. Para ser mais fácil entender o que vai aparecer dentro dos processos desenvolvidos, encontra-se na Figura 8, um diagrama que mostra o encadeamento de todos os passos e funcionalidades que irão ser explicados de seguida.

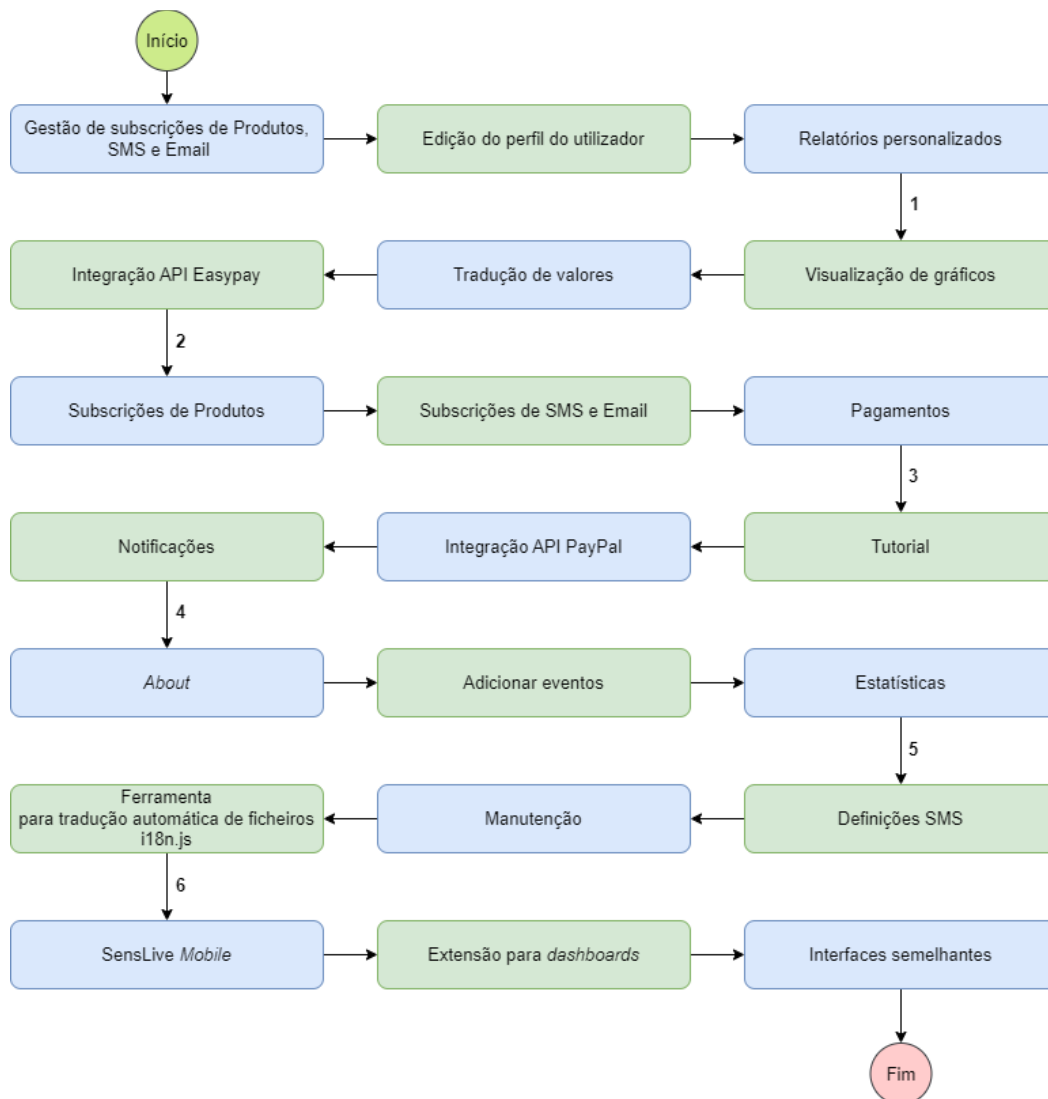


Figura 8 - Diagrama de encadeamento de funcionalidades desenvolvidas

6.3.1. SensLive Web

A plataforma SensLive Web, já existia numa versão anterior, desenvolvida unicamente em PHP. No entanto, diante de dificuldades de desempenho e a necessidade de uma solução mais moderna e eficiente, optou-se por uma migração com utilização de tecnologias como, Vue.js e Laravel. Esta migração visa melhorar significativamente a performance e escalabilidade do novo sistema, diante do aumento constante do volume de dados e clientes.

O processo de desenvolvimento do novo SensLive já se encontrava em andamento, com a arquitetura e as tecnologias já definidas. O início do desenvolvimento foi marcado pela compreensão profunda do propósito desta nova iteração do SensLive, bem como pela análise minuciosa das tabelas de base de dados já existentes. É importante realçar, que nenhuma parte do código da versão anterior do SensLive foi fornecida durante o desenvolvimento desta nova iteração. Essa decisão foi tomada pelo supervisor de estágio, com o objetivo de assegurar que o desenvolvimento das funcionalidades da nova versão ocorresse de forma independente, sem influências externas.

A página do SensLive está estruturada em duas partes. A primeira é composta pelo *sideMenu* e *Toolbar*, que servem como elementos de navegação e controle. A segunda parte é dedicada ao conteúdo principal da página. Como esta nova solução fará uso da tecnologia SPA, grande parte das funcionalidades desenvolvidas serão integradas diretamente no conteúdo da página. Os respetivos *menus* e *submenus* das novas funcionalidades serão incluídos nos elementos de navegação. De modo a facilitar a compreensão do desenvolvimento descrito, encontram-se na Figura 9, ambas as partes representadas com cores distintas, verde para os elementos de navegação e azul para o conteúdo da página.

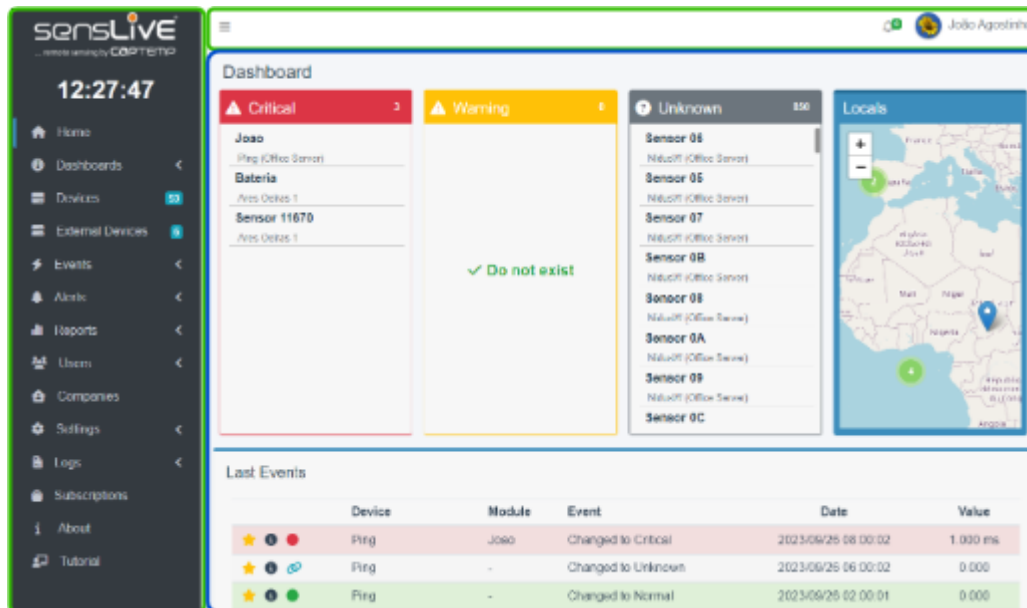


Figura 9 - Página do SensLive com distinção de ambas as partes

- **Gestão de subscrições de Produtos, SMS e Email**

Após se conhecer a estrutura da plataforma do SensLive, foi iniciado o desenvolvimento das funcionalidades, onde inicialmente tratou-se da funcionalidade de gestão de subscrições

de Produtos, SMS (*Short Message Service*) e Email. Cada uma dessas subscrições corresponde a uma área distinta de serviço: a subscrição de produtos abrange packs de sensores; a subscrição de SMS diz respeito ao saldo disponível para utilização; e a subscrição de Email envolve a quantidade de emails disponíveis para receber alertas.

Esta funcionalidade foi desenvolvida apenas para os administradores da plataforma, onde permitia a um administrador criar, editar e eliminar subscrições. Foi necessário criar uma tabela na base de dados para cada uma das subscrições, visto que cada uma tinha características diferentes.

Para criar uma subscrição havia alguns campos a preencher com base no tipo de subscrição, sendo que os campos que partilhavam entre si eram:

- Nome
- Referência
- Preços
- Ativa

Para além destes, existiam outros campos a preencher que faziam parte do formulário de criação. O campo “ativa” era responsável pela disponibilidade da subscrição no lado dos clientes, sendo que, caso alguma subscrição não estivesse ativa, esta não iria estar disponível para compra na página do cliente. Na Figura 10, encontra-se a página da gestão de subscrições, nomeadamente a de subscrições de produtos/sensores.



Active	Name	Reference	Price 1	Price 2	Price 3	Number of Sensors	Duration
✓	Free Subscription	SLS-SAFREE	0.00€	0.00€	0.00€	0	Free
✓	Network w/ up to 25 sensors	SLS-SA0025	██████████	██████████	██████████	25	1 Year
✓	Network w/ up to 50 sensors	SLS-SA0050	██████████	██████████	██████████	50	1 Year
✓	Network w/ up to 100 sensors	SLS-SA0100	██████████	██████████	██████████	100	1 Year
✓	Network w/ up to 1000 sensors	SLS-SA1000	██████████	██████████	██████████	1000	1 Year
✓	Network w/ up to 12 sensors	SLS-SA0012	██████████	██████████	██████████	12	1 Year
✓	Network w/ up to 6 sensors	SLS-SA0006	██████████	██████████	██████████	6	1 Year
✓	Network w/ up to 500 sensors	SLS-SA0500	██████████	██████████	██████████	500	1 Year

Figura 10 - Página de subscrições de produtos

- **Edição do perfil do utilizador**

Posteriormente, foi necessária uma simples adição à funcionalidade já existente de edição de perfil do utilizador. Essa adição permite que, o utilizador escolha qual a *dashboard*, dentro das disponíveis para o utilizador, que será exibida como página inicial após o login na plataforma. As opções de escolha incluem as *dashboards* individuais de cada utilizador, bem como a opção *default*, que se refere à página inicial comum do SensLive.

- **Relatórios personalizados**

Após isto, foi desenvolvida a funcionalidade dos relatórios personalizados. Para que haja este tipo de relatórios, é necessária uma empresa contactar a CapTemp, onde apresente os requisitos para a criação de um relatório único. Seguidamente, se possível, um componente à medida dessa empresa é criado, que irá conter as necessidades impostas pela mesma. Alguns exemplos de relatórios personalizados consistem em:

- Relatórios, onde apenas seja possível escolher sensores de temperatura e humidade.
- Relatórios, onde apenas seja possível escolher sensores de voltagem, frequência e amperagem.
- Relatórios, onde apenas seja possível escolher sensores de temperatura e pressão.

- Relatórios, onde apenas seja possível escolher sensores de um determinado local.

Com isto, a página onde o utilizador preenche os dados necessários para a criação do relatório foi desenvolvida. O utilizador, após preencher os dados do formulário, e escolher a data de início e fim pretendidas, é feita a criação, com os dados referentes aos *inputs*, no *back-end* e respetivo download do ficheiro em formato CSV (*Comma-separated values*) ou PDF. A criação do ficheiro, foi algo que já se encontrava desenvolvido anteriormente.

- **Visualização de gráficos**

Também foi desenvolvida uma funcionalidade semelhante à dos relatórios personalizados, que permite visualizar gráficos. Esta permite ver e, se necessário, transferir relatórios dos gráficos gerados em formato CSV ou PDF. Para gerar esses gráficos, é necessário selecionar os sensores cujo dados se pretende visualizar, especificar uma data de início e fim, e escolher o tipo de gráfico desejado: linhas, barras, consumo, média ou lista. Para criação destes gráficos foi utilizada uma biblioteca JavaScript, chamada de *Highcharts*⁴⁶. Nesta versão da funcionalidade, apenas os gráficos de linhas, média e lista foram implementados.

Após o utilizador preencher os *inputs* correspondente, estes são enviados para o servidor e processados. Os valores resultantes são então devolvidos ao *front-end*, onde serão utilizados para gerar os gráficos correspondentes. Para os gráficos de média ou linhas, os valores dos sensores são exibidos no eixo y, enquanto a escala temporal é apresentada no eixo x. No gráfico de médias, as médias são calculadas para grupos de 4 valores. Isso significa que apenas quando o número de valores é múltiplo de 4 é que a média é calculada e adicionada ao gráfico.

Quanto ao gráfico de lista, é mostrada uma tabela para cada sensor selecionado, mostrando os seguintes dados: máximo, mínimo, tolerância, valor e a data em que o valor foi lido.

Para as funcionalidades dos relatórios personalizados e dos gráficos, é de salientar que estas já estavam desenvolvidas na iteração anterior do SensLive, não fazendo parte das funcionalidades novas a desenvolver. Assim, para adaptar essas funcionalidades para a nova

⁴⁶ <https://www.highcharts.com/>

plataforma, mesmo não tendo acesso ao código base destas, foi necessário analisar o seu comportamento anterior e tentar reproduzir os mesmos resultados.

- **Tradução de valores**

De seguida, a funcionalidade da tradução de valores foi desenvolvida. Esta funcionalidade permite que os valores retornados pelos sensores instalados nas empresas, possam ser traduzidos em descrições personalizadas. Esta era também uma funcionalidade que já existia anteriormente.

É possível criar, eliminar e editar estas traduções de modo a cumprir com as necessidades dos clientes. Para criar uma tradução de valores é necessário preencher um formulário. Este formulário é composto por 2 passos, onde no primeiro passo é preciso inserir uma descrição para a tradução, selecionar quais os sensores que farão uso desta tradução e qual a regra padrão juntamente com o grau de severidade. Em relação ao segundo passo é necessário delinear quais as regras para a tradução de valores, onde o cliente escolhe quais são os valores onde será aplicada a regra e o seu grau de severidade respetivo. Na Figura 11, encontra-se o formulário desenvolvido para esta funcionalidade.

The image shows two side-by-side screenshots of a web form titled 'Edit Translation of Values'. The left screenshot shows the 'General' step (Step 1 of 2), and the right screenshot shows the 'Rules' step (Step 2 of 2).

General Step (Left):

- Description:** A text input field containing 'New Translation of Values'.
- Apply to:** A dropdown menu with 'Battery' and 'Signal' selected.
- Default Rule:** A text input field containing 'Teste' and a dropdown menu with 'Warning' selected.
- Next:** A teal button at the bottom right.

Rules Step (Right):

22,00	Quente	Critical	-
11,00	Médio	Normal	-
0,00	Zero	Warning	-
7,00	Baixo	Unknown	-
-7,5	Negativo	Critical	+

Back: A teal button at the bottom left. **Finish:** A teal button at the bottom right.

Figura 11 - Formulário tradução de valores

Os graus de severidade disponíveis para seleção são: *Normal*, *Warning*, *Critical* e *Unknown*. Para criação de uma tradução de valores é obrigatório ter a descrição e a regra padrão preenchida, pelo menos um sensor selecionado e uma regra definida.

Relativamente aos sensores que podem ser selecionados, apenas os sensores que não possuem qualquer tradução associada a eles é que podem ser selecionados, fazendo com que apenas uma tradução possa ser imposta a cada sensor.

Quando uma tradução é apagada, para além dos dados desta serem apagados, os sensores que estavam associados à tradução perdem a associação que tinham, fazendo com que, voltem a estar disponíveis para seleção.

- **Integração API Easypay e Subscrições de Produtos, SMS e Email**

A próxima funcionalidade desenvolvida, consistiu na integração do método de pagamento easypay ⁴⁷, na plataforma SensLive. Esta funcionalidade irá possibilitar aos clientes adquirir subscrições de Produtos, SMS e Email através do SensLive, as mesmas que são geridas pelos administradores da plataforma.

Para facilitar a integração e entender como a API do easypay funciona, foi necessário perder algum tempo a ler e analisar a documentação existente ⁴⁸. Depois disto, foi necessário criar uma conta no *backoffice*, para se obter credenciais para utilização da API e definir os URL's (*Uniform Resource Locator*) para as notificações. Estes URL's são utilizados para notificar quando existe alteração no estado dos pagamentos.

Depois deste processo, foi necessário criar uma tabela, *payments*, na base de dados que serviu para guardar as informações referentes a todos os pagamentos efetuados. Alguns dos campos nesta tabela constam de:

- *Status* – Estado do pagamento. (0 – pendente, 1 – pago e 5 - erro)
- *Payment ID* – Identificador do pagamento.
- *Method* – Método utilizado para o pagamento. (cc – cartão de crédito, mbw – MB Way, mb – Multibanco e pp - PayPal)
- *Value* – Valor da compra.
- *Requested by* – Identificador do utilizador que efetuou a compra.
- *Type* – Tipo de subscrição que foi adquirida. (0 – Email, 1 – SMS e 2 - Produtos)
- *Credits* – Quantidade adquirida de créditos dependendo do tipo de subscrição.
- *Start date* – Data de início da subscrição de Produtos guardada em segundos.

⁴⁷ <https://www.easypay.pt/>.

⁴⁸ <https://docs.easypay.pt/>.

- *End date* – Data de fim da subscrição de Produtos guardada em segundos.
- *Active* – Tem o valor de 1 ou 0, conforme se a subscrição de Produtos está ativa ou não.

Alguns destes campos são bastante autoexplicativos, no entanto há alguns que precisam de uma melhor explicação. Para que se entenda melhor, será feita uma breve explicação de como é que as subscrições funcionam.

As subscrições de produtos têm características diferentes das subscrições de SMS e Email. Em relação às subscrições de SMS e Email, quando compradas estas, irão acrescentar créditos aos créditos já existentes de uma empresa, ou seja, após o cliente efetuar o pagamento, os créditos serão logo adicionados à sua empresa. Estes tipos de subscrições encontram-se disponíveis no submenu SMS e Email respetivamente, dentro do menu *Settings*. No entanto, para as subscrições de Produtos não é assim tão simples. Para estas, devido a uma subscrição de produtos estar associada a uma duração, que neste caso é quase sempre 1 ano, é necessário verificar algumas condições antes e após a compra. Este tipo de subscrição encontra-se no menu com o nome *Subscriptions* e está disponível apenas para os administradores da empresa. Esta opção é exibida quando a empresa não tem uma subscrição associada ou quando a subscrição atual da empresa está prestes a expirar. Na Figura 12, encontra-se a lista de algumas subscrições disponíveis para compra.

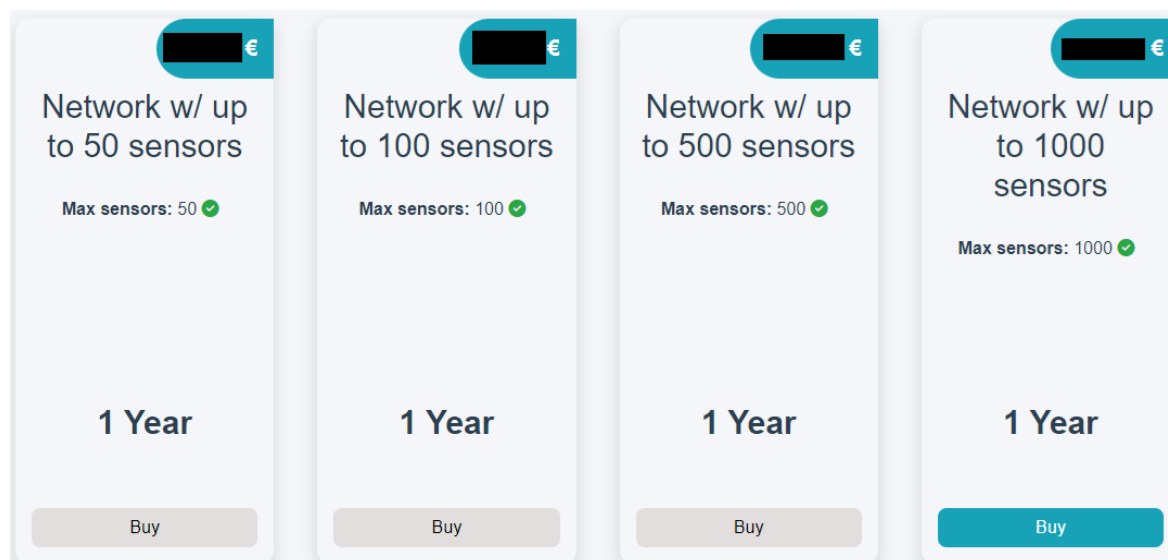


Figura 12 - Lista de subscrições disponíveis para compra

As condições antes da compra passam por verificar se a empresa já possui uma subscrição associada. Caso esta não tenha nenhuma subscrição, o menu para a compra das subscrições fica disponível, no entanto se existir uma subscrição associada, é necessário ver

se a *end_date*, está a menos de 30 dias de diferença, que no caso de estar, o menu também estará disponível. Em segundo lugar, após se verificar que a empresa possui uma subscrição que esteja para expirar, as subscrições ficam disponíveis para compra, porém nem todas poderão estar disponíveis, pois apenas será possível escolher subscrições que tenham packs onde o número de sensores é maior do que o número de sensores atual que a empresa tenha. Isto pode ser visto na Figura 12, onde podemos observar que para as subscrições de 50, 100 e 500 sensores, o botão *Buy*, encontra-se desativado, o que significa que neste exemplo, esta empresa teria mais de 500 sensores e a única subscrição disponível para compra seria a subscrição que possuía até 1000 sensores.

Em relação às condições após a compra, será necessário verificar se a subscrição associada já se encontra expirada ou se está prestes expirar. Caso a subscrição se encontre expirada, a subscrição será substituída pela nova, no entanto, caso esta ainda não se encontre expirada, a subscrição que acabou de ser comprada, será colocada em standby, o que fará com que a *start_date* desta seja o valor da *end_date* da subscrição atual e a *end_date* o valor da *start_date* adicionado à duração da nova subscrição. Para além disso, o campo *active* será 0 até a subscrição que se encontra em standby passe para ativa. Este processo de substituição é automatizado através da funcionalidade *schedule*⁴⁹ do Laravel.

É importante referir, que qualquer empresa que tenha deixado a sua subscrição de produtos expirar, não terá acesso à plataforma até voltar a fazer a compra de uma nova subscrição.

Depois de uma breve explicação de como as subscrições funcionam, está presente na Figura 13, o formulário referente aos pagamentos.

⁴⁹ <https://laravel.com/docs/11.x/scheduling>.

The figure displays three sequential steps of a subscription confirmation process:

- Step 1 (General):** Shows subscription details: Name: Network w/ up to 1000 sensors, Reference: SLS-SA1000, Price: 2174.00 €, Number of Sensors: 1000, Duration: 1 Year. A 'Next' button is visible.
- Step 2 (Details):** Contains input fields for Name, Email, Phone (with a country code dropdown), and Fiscal Number. Below these are radio buttons for payment methods: Mastercard, VISA, MB MULTIBANCO, MB WAY, and PayPal. 'Previous' and 'Next' buttons are at the bottom.
- Step 3 (Confirm Payment):** Displays a summary box for 'Payment by Multibanco or Homebanking' with the following details: Expiration Date: 04/19, Entity: 11683, Reference: 574 000 090, Amount/Value: 2174.00 €. A 'Done' button is at the bottom right.

Figura 13 - Formulário de confirmação das subscrições

No exemplo visto na **Erro! A origem da referência não foi encontrada.**, encontram-se os passos para compra de uma subscrição onde o método é multibanco. Todos os campos presentes no segundo passo são obrigatórios. Dependendo do método de pagamento selecionado, o terceiro passo pode variar. Como podemos ver na Figura 13, para o multibanco, é apresentado a data de expiração, juntamente com os dados do pagamento, sendo estes, entidade, referência e o valor da transação. No caso de escolher o pagamento com cartão de crédito, ao avançar para o terceiro passo, o cliente será automaticamente redirecionado para a página do easypay. Lá, ele poderá concluir a transação. No formulário, será exibido um aviso informando ao cliente sobre o redirecionamento para a nova página. Se, por algum motivo, o redirecionamento não ocorrer, será fornecido um link para que o cliente possa abrir manualmente a página do easypay. Se o método escolhido for MB Way, antes de ser possível passar para o terceiro passo, é necessário preencher mais um campo, que é referente ao número de telemóvel para onde será enviado o pedido de pagamento. Após isto, no terceiro passo, apenas será mostrada uma mensagem que alerta o cliente que o pedido de pagamento foi enviado para o telemóvel com o número que foi inserido anteriormente. Por último, caso o cliente pretenda pagar por PayPal, o processo é idêntico ao cartão de crédito, onde o cliente após preencher os dados é automaticamente redirecionado para a página do PayPal, para fazer login com a sua conta e terminar a transação, no formulário será exibido igualmente o aviso do redirecionamento.

Quando o cliente passa para o terceiro passo, caso o método selecionado tenha sido cartão de crédito, multibanco ou MB Way, é efetuado um pedido à API do easypay, para

este criar um *single payment*, que irá devolver os dados para efetuar o pagamento. A função que é utilizada para fazer este pedido encontra-se na Listagem 1.

```
//creates Single Payment using the EasyPay API
public function createSinglePayment (Request $request){
    $body = [
        "key" => $request->reference,
        "method" => $request->method,
        "type" => "sale",
        "value" => floatval($request->price),
        "currency" => "EUR",
        "expiration_time" => $request->expirationTime,
        "capture" => [
            "descriptive" => $request->description,
        ],
        "customer" => [
            "name" => $request->name,
            "email" => $request->email,
            "key" => $request->key,
            "phone_indicative" => $request->phoneCode,
            "phone" => $request->phone,
            "fiscal_number" => $request->fiscalNumber,
        ],
    ];

    $headers = [
        "AccountId: " . env("EASYPAY_ACCOUNTID"),
        "ApiKey: " . env("EASYPAY_APIKEY"),
        'Content-Type: application/json',
    ];

    $curlOpts = [
        CURLOPT_URL => "https://api.test.easypay.pt/2.0/single",
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_POST => 1,
        CURLOPT_TIMEOUT => 60,
        CURLOPT_POSTFIELDS => json_encode($body),
        CURLOPT_HTTPHEADER => $headers,
    ];

    $curl = curl_init();
    curl_setopt_array($curl, $curlOpts);
    $response_body = curl_exec($curl);
    curl_close($curl);
    $response = json_decode($response_body, true);

    return response()->json($response , 201);
}
```

}

Listagem 1 - Função com o pedido feito à API do easypay

Como podemos observar, existem vários campos opcionais que podem ser enviados no pedido, no entanto, apenas o *value* e o *method* são obrigatórios.

Depois de se receber a resposta deste pedido, são guardadas todas as informações na tabela *payments* e utilizadas as informações referentes aos pagamentos para completar o terceiro passo. Inicialmente, todos os pagamentos começam com o seu estado com o valor 0 (pendente).

Após a configuração do URL de notificações, é necessário aguardar a alteração de estado da transação, seja para “pago” ou “erro”. Conforme o estado muda, serão realizadas verificações e diversos passos para garantir que tudo ocorra conforme esperado. Devido à sensibilidade do código envolvido no processamento dos pagamentos, não é possível divulgar detalhes específicos. No entanto, na Figura 14, presente na documentação do easypay, é apresentado um diagrama de sequência dos passos relacionados à criação de um pagamento [54].

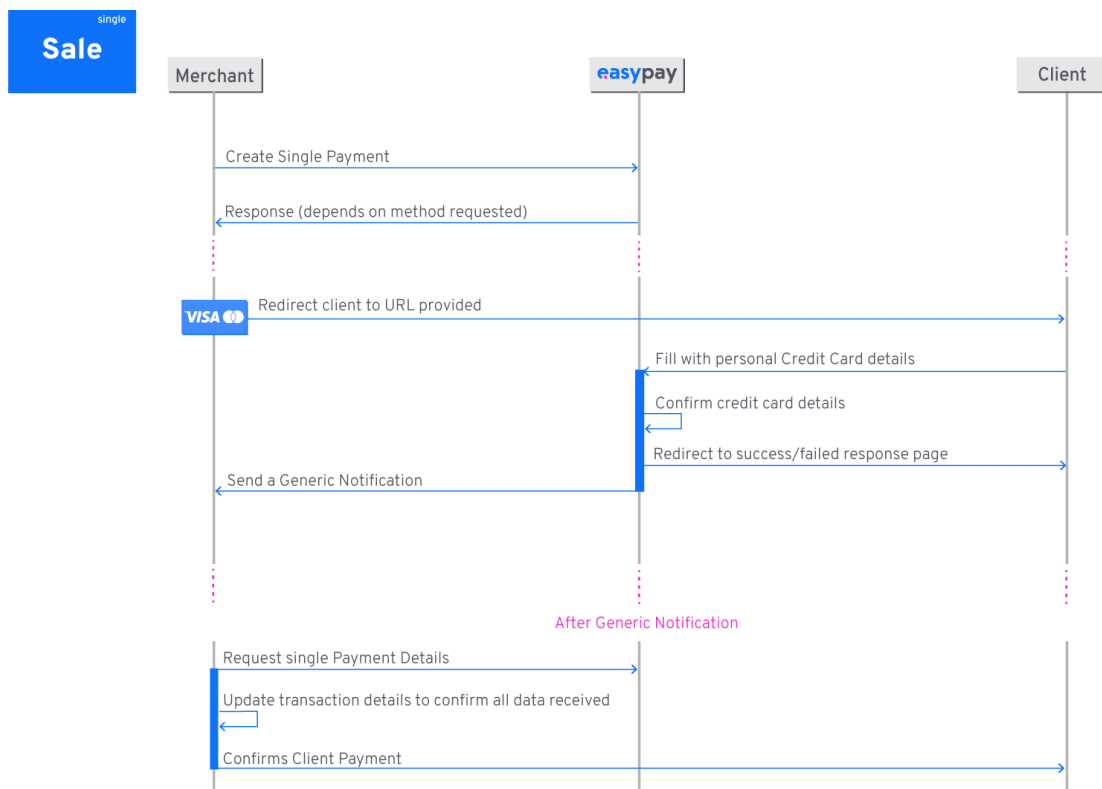
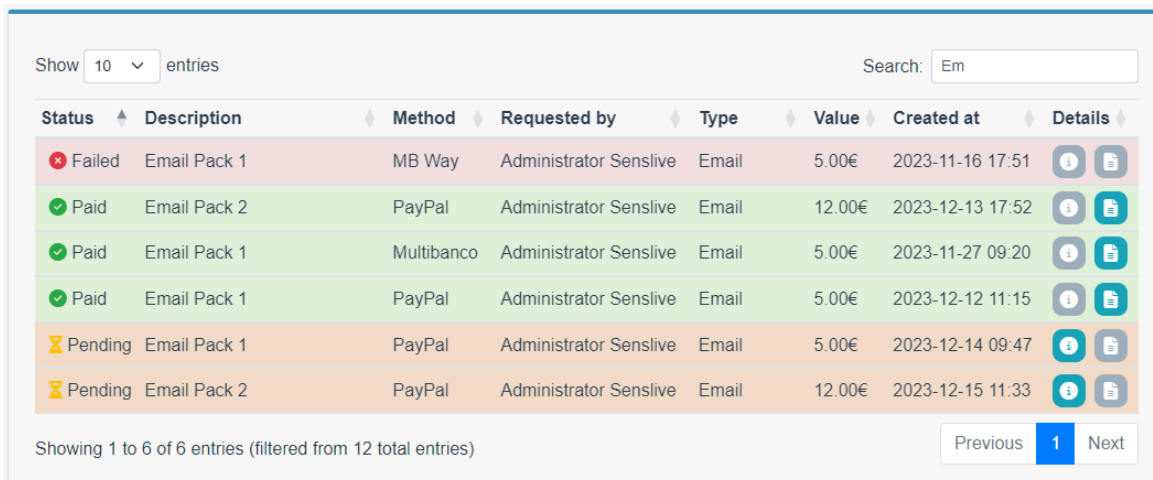














Figura 14 - Diagrama de sequência referente à criação de pagamentos com cartão de crédito

- **Pagamentos**

De seguida, foi necessário desenvolver uma página que mostrasse o histórico de pagamentos efetuados pela empresa. Esta página irá apresentar uma tabela que contenha os dados relevantes dos pagamentos, nomeadamente, o status do pagamento, a descrição, o método utilizado, quem é que fez o pedido de compra, o tipo de subscrição, o valor, a data de criação. Para além disso, numa coluna dedicada aparecem dois botões, onde o primeiro serve para mostrar as informações dos pagamentos que ainda não foram concluídos e o segundo para criar uma fatura do pagamento para os pagamentos que já foram concluídos. Na Figura 15 **Erro! A origem da referência não foi encontrada.**, encontra-se a tabela r



Status	Description	Method	Requested by	Type	Value	Created at	Details
Failed	Email Pack 1	MB Way	Administrator Senslive	Email	5.00€	2023-11-16 17:51	 
Paid	Email Pack 2	PayPal	Administrator Senslive	Email	12.00€	2023-12-13 17:52	 
Paid	Email Pack 1	Multibanco	Administrator Senslive	Email	5.00€	2023-11-27 09:20	 
Paid	Email Pack 1	PayPal	Administrator Senslive	Email	5.00€	2023-12-12 11:15	 
Pending	Email Pack 1	PayPal	Administrator Senslive	Email	5.00€	2023-12-14 09:47	 
Pending	Email Pack 2	PayPal	Administrator Senslive	Email	12.00€	2023-12-15 11:33	 

Showing 1 to 6 of 6 entries (filtered from 12 total entries) Previous 1 Next

Figura 15 - Tabela com o histórico dos pagamentos

eferente ao histórico dos pagamentos.

O primeiro botão permite visualizar as informações dos pagamentos que estão em estado “Pendente” (*Paid*). Essas informações incluem detalhes semelhantes aos explicados anteriormente para o terceiro passo do formulário de confirmação das subscrições. Dependendo do método de pagamento utilizado, diferentes informações podem ser exibidas.

Já o segundo botão, só está disponível quando um pagamento está marcado como “Pago” (*Paid*) e é utilizado para criar uma fatura referente à compra da subscrição. Vale referir que a funcionalidade de faturas era algo que já estava implementado e, portanto, não foi desenvolvida novamente.

- **Tutorial**

A próxima funcionalidade desenvolvida foi o tutorial da plataforma. Esta iniciativa surgiu como uma necessidade identificada pelo supervisor durante o estágio. Com o aumento significativo de telefonemas de clientes à procura de suporte sobre como utilizar o SensLive, e com o número crescente de formações agendadas, o tempo dedicado a resolver estes problemas tornou-se crucial. Para combater essa demanda, foi concebida uma página que visa abordar essas questões, permitindo aos utilizadores esclarecerem as suas dúvidas ao realizar formações autónomas dentro do SensLive.

A página do tutorial apresenta um cartão para cada menu disponível. Se um menu tiver submenus, um modal é aberto, permitindo ao utilizador escolher qual submenu é que deseja explorar no tutorial. Depois de selecionar o menu ou submenu desejado, o utilizador é redirecionado para a página correspondente e um tour pela página é iniciado. Durante o tour, o utilizador tem a oportunidade de explorar todas as operações possíveis naquela página, seguindo as instruções apresentadas. A funcionalidade do tour foi desenvolvida por outro estagiário.

Todos os utilizadores podem aproveitar esta funcionalidade sempre que precisarem. O sistema guarda individualmente todas as páginas nas quais um utilizador já completou o tutorial. Após concluir o tutorial de uma página específica, o utilizador tem a opção de realizar o tutorial novamente, caso necessário. Quando completar todos os tours de todos os menus e submenus disponíveis, o utilizador terá a possibilidade de gerar um certificado que comprove que completou com sucesso o tutorial completo do SensLive.

- **Integração API PayPal**

Depois disso, foi necessário voltar às subscrições, especificamente para integrar o PayPal nos métodos de pagamento já existentes. Isto foi fundamental, visto que a API do easypay não incluía integração com o PayPal, exigindo assim o uso da API do PayPal. Tal como o easypay, foi preciso dedicar algum tempo à análise da documentação disponível desta nova API ⁵⁰.

Após esta análise, foi necessário obter credenciais para poder fazer uso da API. Neste caso, o supervisor já constava de uma conta de programador, sendo que apenas foi preciso

⁵⁰ <https://developer.paypal.com/api/rest/>

ir buscá-las à página de programadores do PayPal. No entanto, para conseguir fazer uso da API, para além das credenciais, foi necessário obter o *access token*, que servia para autenticar as chamadas à API do PayPal. Na Listagem 2, encontra-se a função que retornava o *access token* quando esta era chamada.

```
public function getAccessToken(){
    //PayPal API credentials
    $clientId = env("PAYPAL_CLIENT_ID");
    $clientSecret = env("PAYPAL_CLIENT_SECRET");

    //Encode credentials in Base64
    $credentials = base64_encode($clientId . ':' . $clientSecret);

    //PayPal API endpoint
    $url = 'https://api-m.sandbox.paypal.com/v1/oauth2/token';

    //Guzzle HTTP client
    $client = new Client();

    //Make a POST request to get the access token
    $response = $client->post($url, [
        'headers' => [
            'Authorization' => 'Basic ' . $credentials,
            'Content-Type' => 'application/x-www-form-urlencoded',
        ],
        'form_params' => [
            'grant_type' => 'client_credentials',
        ],
    ]);

    //Decode the response JSON
    $data = json_decode($response->getBody(), true);

    //Access token
    $accessToken = $data['access_token'];

    return $accessToken;
}
```

Listagem 2 - Função que retorna o access token à API do PayPal

Podemos observar que, para se obter o *access token*, é necessário ter as credenciais da API do PayPal e chamar o *endpoint* que retorna o *token*, que neste caso foi *'https://api-m.sandbox.paypal.com/v1/oauth2/token'*.

Tendo acesso ao *token*, passou-se para o passo seguinte, que implica a criação de uma *order* após o utilizador passar para o terceiro passo da Figura 13. O código utilizado para se criar a *order* encontra-se representado na Listagem 3.

```
public function pay (Request $request)
{
    (...)
    $accessToken = $this->getAccessToken();

    try {
        //PayPal API endpoint
        $url = 'https://api-m.sandbox.paypal.com/v2/checkout/orders';

        //Request payload
        $payload = [
            'intent' => 'CAPTURE',
            'purchase_units' => [
                [
                    'description' => $request->description,
                    'amount' => [
                        'currency_code' => env("PAYPAL_CURRENCY"),
                        'value' => $request->price,
                    ]
                ]
            ],
        ],
        (...)
    ];

    //Guzzle HTTP client
    $client = new Client();

    //Make a POST request to create a PayPal order
    $response = $client->post($url, [
        'headers' => [
            'Content-Type' => 'application/json',
            'Authorization' => 'Bearer ' . $accessToken,
        ],
        'json' => $payload,
    ]);

    //Decode the response JSON
    $order = json_decode($response->getBody(), true);

    //Extract the approve URL
    $approveUrl = null;
    foreach ($order['links'] as $link) {
        if ($link['rel'] === 'payer-action') {
```

```
        $approveUrl = $link['href'];
        break;
    }
}
(...)
//Redirect to the approve URL
return response()->json($approveUrl, 201);
(...)
}
```

Listagem 3 - Função utilizada para criar uma *order*

Observando o excerto de código, é possível ver o *endpoint* utilizado para a criação da *order*, '<https://api-m.sandbox.paypal.com/v2/checkout/orders>'. Este *endpoint* necessita de dois campos obrigatórios no *body*, sendo eles o *intent* e o *purchase_units*, que servem para caracterizar o tipo de compra que se quer criar. O *intent* será *CAPTURE*, que significa que, “O comerciante pretende capturar o pagamento imediatamente após o cliente efetuar o pagamento”⁵¹, e o *purchase_units* indica o preço, a moeda e a descrição da compra.

Quando o pedido é enviado, a *order* é criada e o URL para redirecionar o utilizador para a página do PayPal é extraído da resposta. Este URL será então retornado ao *front-end*, permitindo que o utilizador seja redirecionado para efetuar o pagamento.

Após o utilizador efetuar o pagamento, o URL configurado será chamado para receber a notificação referente ao pagamento. Este URL é configurado através de um *webhook*. Para criar um *webhook*, é necessário ter acesso ao *access token* e enviar no *body* do pedido um URL e o tipo de evento para o qual se deseja receber notificações. Para isto, foi utilizada a ferramenta Postman.

Como as notificações de pagamento precisam de estar protegidas, foi imprescindível configurar o *webhook* com um URL que utilize o protocolo HTTPS (Hyper Text Transfer Protocol Secure). Visto que a aplicação ainda estava em desenvolvimento, não foi possível obter um URL com este protocolo, o que levou à necessidade de utilizar um *website* que gerava URLs com protocolo HTTPS e permitia a receção de respostas⁵².

Assim, com o URL seguro, foi apenas preciso definir qual o tipo de evento para o qual era preciso receber notificações. Para isso, foi preciso ver a documentação que enumerava

⁵¹ https://developer.paypal.com/docs/api/orders/v2/#orders_create

⁵² <https://webhook.site/>

os diversos tipos de eventos possíveis. No entanto, para configuração de uma notificação de confirmação de pagamento, foi apenas necessário escolher um tipo de eventos, “*CHECKOUT.ORDER.APPROVED*”.

Com ambos os campos preenchidos, seguiu-se para a criação do *webhook* através da ferramenta Postman. O PayPal continha um *workspace* no Postman ⁵³, pronto a ser utilizado para realizar este tipo de operações, por isso, o processo foi bastante simplificado, sendo apenas necessário preencher os vários *headers* obrigatórios.

Após isto, foram realizados pagamentos utilizando o método PayPal no SensLive para testar o *webhook* criado. Conforme previsto, as respostas dos vários pagamentos de teste começaram a chegar ao *website* que continha o URL usado no *webhook*. Tendo a resposta de confirmação do pagamento de uma *order*, procedeu-se à criação de um *endpoint* na API do SensLive para processar essas confirmações e atualizar o estado dos respetivos pagamentos de subscrições. Na Listagem 4, encontra-se um exemplo de uma resposta recebida pelo *webhook*.

```
{
  "id": "WH-9F0761074A487122S-4UN20311AB837192F",
  "event_version": "1.0",
  "create_time": "2024-05-27T09:42:22.339Z",
  "resource_type": "checkout-order",
  "resource_version": "2.0",
  "event_type": "CHECKOUT.ORDER.APPROVED",
  "summary": "An order has been approved by buyer",
  "resource": {
    "create_time": "2024-05-27T09:41:28Z",
    "purchase_units": [
      {
        "reference_id": "default",
        "amount": {
          "currency_code": "EUR",
          "value": "5.00"
        },
      },
    ],
    "payee": {
```

⁵³ <https://www.postman.com/paypal/workspace/paypal-public-api-workspace/overview>

```

    "email_address": "sb-9fi8128583582@business.example.com",
    "merchant_id": "UULZ3K7LMUD5L"
  },
  "description": "Email Pack 1",
}
],
(...)
"status": "APPROVED"
},
}

```

Listagem 4 - Exemplo de uma resposta recebida pelo *webhook*

Assim, depois da configuração deste novo *endpoint* e utilizando novamente a ferramenta Postman, foi copiada a resposta que tinha chegado ao *website* e inserida no corpo do pedido enviado para o novo *endpoint* que fazia o tratamento desta. Na

Figura 16, encontra-se o pedido na ferramenta Postman.

```

POST http://capttemp.dyndns.org:11988/api/v1/public/api/v1/notifyPayPal
Body
raw
{
  "id": "WH-3G198765YR3117821-3V512149N13945509",
  "event_version": "1.0",
  "create_time": "2023-12-15T11:33:45.749Z",
  "resource_type": "checkout-order",
  "resource_version": "2.0",
  "event_type": "CHECKOUT.ORDER.APPROVED",
  "summary": "An order has been approved by buyer",
  "resource": {
    "create_time": "2023-12-15T11:33:33Z",
    "purchase_units": [
      {
        "reference_id": "default",
        "amount": {
          "currency_code": "EUR",
          "value": "12.00"
        },
        "payee": {
          "email_address": "sb-9fi8128583582@business.example.com",
          "merchant_id": "UULZ3K7LMUD5L"
        },
        "description": "Email Pack 2",
        "shipping": {
          "name": {
            "full_name": "John Doe"
          }
        }
      }
    ]
  }
}

```

Figura 16 - Pedido enviado ao *endpoint* do SensLive

Em relação ao que acontece após o pedido ser recebido, em primeiro lugar verifica-se se o *payment_id* que consta na resposta existe na tabela *payments*. Só após esta confirmação é que o estado do pagamento é atualizado. Os restantes passos são idênticos aos anteriormente descritos para os outros métodos.

Com isto, é possível notar algumas diferenças entre as duas APIs utilizadas. No entanto, o processo interno é idêntico em ambas, pois ambas exigem credenciais para obter um *token*, necessitam de uma chamada a um *endpoint* específico para criar a *order/payment* e utilizam um URL para notificar que o pagamento foi efetuado. Além disso, é importante destacar que, independentemente da API utilizada, os dados dos pagamentos são armazenados na mesma tabela *payments* e o processo de tratamento do pagamento das subscrições é o mesmo.

- **Notificações**

Ainda no contexto das subscrições, foi necessário amplificar a funcionalidade existente de notificações para incluir mensagens relacionadas aos pedidos, compras de subscrições e quando as subscrições de produtos estão prestes a expirar. Para os dois primeiros tipos de mensagens, os utilizadores receberão notificações quando um pedido for feito e quando esse pedido for confirmado como pago. Para isso, adicionou-se estas mensagens às funções dos *endpoints* que lidam com as notificações das APIs do easypay e PayPal. Em relação ao outro tipo de mensagens, estas foram implementadas através de uma *schedule* diária, quando uma subscrição de produtos estiver com menos de 1 mês de duração, será enviada uma mensagem para o utilizador para o lembrar que a sua subscrição está prestes a expirar. Quando faltar uma semana, o utilizador é lembrado outra vez com uma mensagem e, caso este não a renove até à data de fim da subscrição, também receberá uma mensagem que irá referenciar isso mesmo. Estas mensagens de expiração, não serão enviadas caso o utilizador já possua uma subscrição comprada em *standby*.

Todas as mensagens são armazenadas numa tabela dedicada na base de dados, chamada chat. Nesta tabela, são registados o texto da mensagem, o ID do utilizador a quem a mensagem é enviada, a data de envio, entre outros campos. As notificações podem ser visualizadas conforme apresentado na Figura 17.

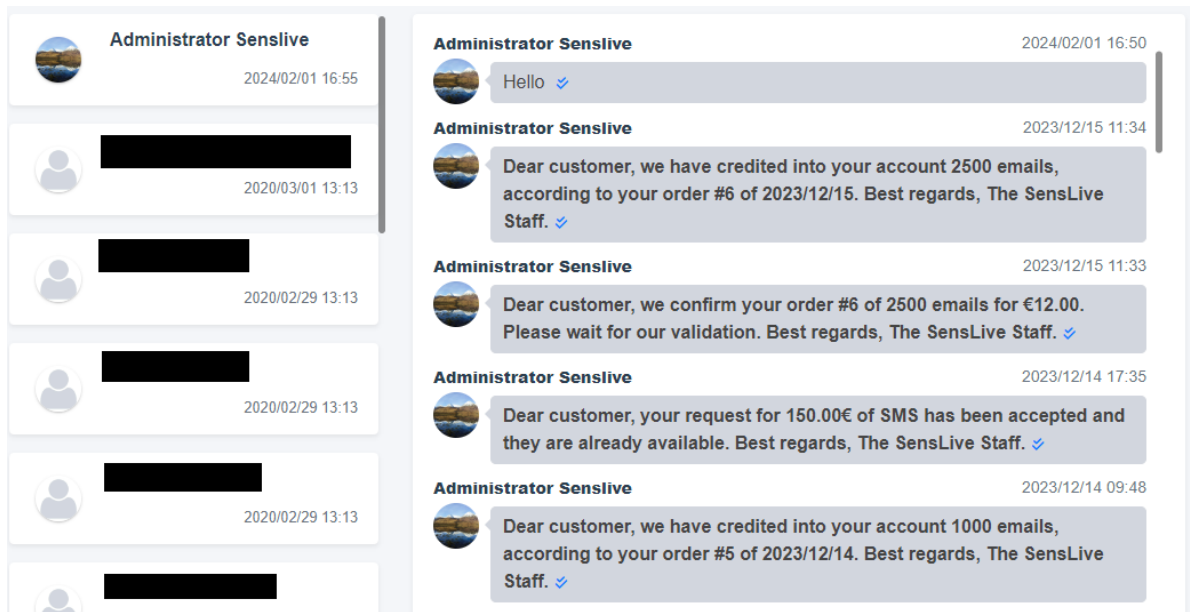


Figura 17 - Página das notificações

- **About**

Tal como as notificações, foi necessário aumentar o espetro da página do *About* do SensLive para incluir as informações da subscrição de Produtos. Com isto, será possível para o utilizador verificar se a sua subscrição se encontra válida, inválida ou prestes a expirar. Além disto, nesta página já era possível observar a versão do SensLive atual, a versão do componente de certificação e ainda verificar esta certificação através de um botão. Os passos para esta verificação já se encontravam implementados.

Após isto, todas as funcionalidades das subscrições encontravam-se terminadas, salvo alguns pontos extras que teriam de ser mais tarde desenvolvidos antes desta nova versão do SensLive passar para produção. No entanto, ainda havia algumas funcionalidades a acrescentar à restante plataforma.

- **Adicionar eventos**

Um dessas funcionalidades consistia em permitir aos utilizadores adicionar eventos aos dispositivos externos, caso assim o desejassem. Esta funcionalidade permitiria a um utilizador adicionar um evento a um dispositivo externo específico, o qual seria composto por uma data e uma descrição. A estrutura da tabela na base de dados para os eventos já se encontrava feita, pois os eventos já estavam a ser criados para os dispositivos “não externos”.

- **Estatísticas**

Outra funcionalidade solicitada, foi a criação de uma página de Estatísticas, destinada a administradores da plataforma, com o intuito de oferecer uma visão abrangente dos dados estatísticos. Esta página inclui um gráfico de barras onde está apresentado o tráfego mensal da plataforma por parte dos utilizadores, um gráfico circular que mostra a distribuição dos envios de dados pelos dispositivos e, por último, várias contagens de diversos parâmetros, tais como o número de empresas, dispositivos, dispositivos inativos, sensores, eventos e subscrições próximas de expirar. Esta funcionalidade foi desenvolvida para simplificar a visualização desses dados pelo administrador da plataforma e melhorar a sua gestão. Na Figura 18 encontra-se esta página representada.

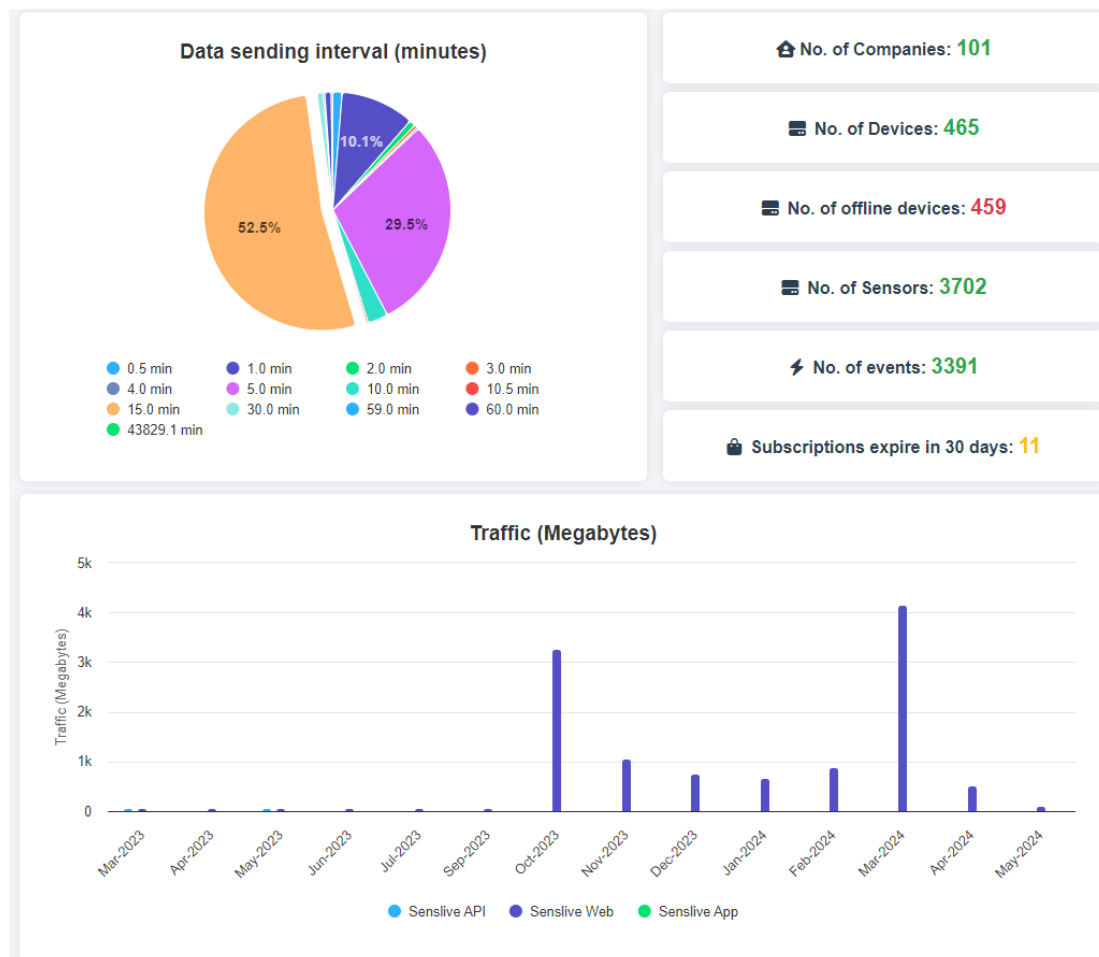


Figura 18 - Página de estatísticas

- **Definições de SMS**

Em relação às definições de SMS, estas também requereram a adição de uma funcionalidade nova. Dentro das definições de SMS, os utilizadores podem escolher utilizar

o serviço do SensLive para receber notificações dos alarmes ou utilizar o seu próprio servidor GSM (*Global System for Mobile communication*). Para usufruir da primeira opção, é necessário comprar subscrições de SMS, as quais irão fornecer saldo para que as mensagens de alarmes possam ser enviadas. No caso de o utilizador escolher a utilização do seu próprio servidor GSM, este irá ter de preencher um formulário para configuração da *gateway*. Nesta segunda opção é quando a nova funcionalidade entra em vigor. Esta, irá permitir aos utilizadores escolherem que dispositivo associar a cada local da sua empresa, tendo como principal objetivo determinar que dispositivo é que irá receber os SMS consoante o local. Isto é especialmente importante para empresas que tenham vários dispositivos em diferentes locais.

- **Manutenção**

Por último, foi desenvolvida uma nova funcionalidade para o SensLive, a funcionalidade de manutenção. Esta inclui dois tipos de manutenção: Manutenção regular e Manutenção para atualização de versões. A primeira permite agendar uma manutenção na plataforma, escolhendo um intervalo de datas, e enviar um alerta global para todos os utilizadores para que tenham conhecimento da ocorrência. Para o segundo tipo, o utilizador envia uma notificação a todos os utilizadores, especificando as novas funcionalidades trazidas pela nova versão. Foi também implementado um pop-up para todos os utilizadores quando ocorre uma nova atualização de versão. Este pop-up informa sobre a disponibilidade de uma nova versão e lista todas as novas funcionalidades. Para garantir que o utilizador recebeu esta informação, foi adicionado um novo campo à tabela dos utilizadores, que guarda o *id* da última versão da plataforma a que o utilizador teve acesso. Sempre que o utilizador se autenticar na plataforma, o sistema verifica se a versão associada ao utilizador é a mais recente disponível na tabela de versões. Se não for, o alerta aparece quando o utilizador se autenticar, caso contrário, nada acontece. A funcionalidade de manutenção, apenas pode ser utilizada pelos administradores da plataforma SensLive.

- **Ferramenta para tradução automática de ficheiros i18n.js**

Durante o desenvolvimento do SensLive, foi necessário ter sempre em conta todas as *labels* que se inseriam, desde títulos, descrições, mensagens, etc. Estas *labels* foram sempre escritas em inglês e seguiam as regras da internacionalização, ou i18n⁵⁴. A

⁵⁴ <https://www.i18next.com/>

internacionalização ajuda a facilitar a tradução automática, ao fazer uso de “códigos” que com base na linguagem selecionada, carrega a respetiva tradução. Na Listagem 5, está presente o excerto de código com alguns exemplos do uso do `i18n`.

```
this.optionsAlert.push({ id: 0, label:
this.$t("unitsTranslate.normal")}); // Alert is normal
    this.optionsAlert.push({ id: 1, label:
this.$t("unitsTranslate.warning") }); // Alert is warning
    this.optionsAlert.push({ id: 2, label:
this.$t("unitsTranslate.critical") }); // Alert is critical
    this.optionsAlert.push({ id: 3, label:
this.$t("unitsTranslate.unknown") }); // Alert is unknown
```

Listagem 5 - Exemplos de uso do `i18n`

Como podemos observar no excerto, estamos a usar o código `$t("unitsTranslate.normal")`, que irá carregar o respetivo valor que está associado a ele. Tal como já referido, conforme a linguagem do utilizador, o ficheiro respetivo com as traduções dessa linguagem é chamado e as traduções são carregadas. Na Listagem 6, encontra-se um exemplo de um ficheiro `i18n.js`, onde podemos ver como é que os códigos estão a ser definidos.

```
export default{
  (...)
  unitsTranslate:{
    normal: "Normal",
    warning: "Warning",
    critical: "Critical",
    unknown: "Unknown",
    (...)
  },
  (...)
}
```

Listagem 6 - Exemplo de um ficheiro `i18n.js`

Tendo isto em consideração, surgiu a ideia de criar uma aplicação que, com base no ficheiro `i18n.js`, realize a tradução de forma automática de acordo com a nova língua selecionada, criando um ficheiro traduzido.

Esta aplicação foi escrita em *python* e com a ajuda da livreria da *googletrans*⁵⁵ foi possível traduzir todos os valores. Foi delineado no código que os valores a ser traduzidos seriam apenas aqueles que se encontrassem dentro de aspas. Finalmente, após a tradução total dos valores, um novo ficheiro é criado com os novos valores traduzidos.

- **Interfaces semelhantes**

É importante referir que durante o desenvolvimento, foi sempre tentado seguir as regras principais de usabilidade e *design*, garantindo uma experiência de utilizador eficiente e intuitiva. Para além disso, foi tida em consideração a interface existente da versão anterior do SensLive, de forma a preservar a familiaridade dos utilizadores e facilitar a transição para a nova versão. Mantendo uma interface semelhante e integrando a nova funcionalidade do tutorial, espera-se que, no futuro haja uma poupança significativa de tempo, algo que atualmente não acontece. Sempre que os clientes possuírem alguma dúvida, esta, poderá ser esclarecida de imediato através do tutorial, evitando reuniões/chamadas desnecessárias com a empresa. Além disso, a continuidade do *design* da interface anterior irá evitar futuros problemas que os clientes teriam devido à mudança drástica de interface gráfica, poupando assim também tempo crucial, que pode ser utilizado para desenvolver novas funcionalidades.

6.3.2. SensLive Mobile

A aplicação SensLive Mobile foi um projeto que já tinha sido pensado anteriormente pela empresa, no entanto, nunca foi possível iniciar o seu desenvolvimento. Com o desenvolvimento do SensLive Web a terminar antes do tempo, foi possível dar início à variante do SensLive *cross-platform*.

Esta aplicação foi concebida especialmente para os clientes que já tinham questionado a empresa sobre a possibilidade de uma aplicação *mobile* para aceder ao SensLive, mas também para facilitar o acesso aos dados sem necessitar de ir ao *website* do SensLive. O único requisito imposto pelo supervisor foi, a utilização de Flutter para o desenvolvimento, pois um dos objetivos desta aplicação era ser *cross-platform*, ou seja, que pudesse ser instalada em qualquer dispositivo, independentemente do sistema operativo (Windows, MacOS, iOS, Android, Linux).

⁵⁵ <https://pypi.org/project/googletrans/>

Assim, deu-se início ao desenvolvimento da aplicação. Como a aplicação utiliza a API da plataforma web, o desenvolvimento foi bastante simplificado, sendo que o maior desafio consistiu em aprender e entender a nova linguagem de programação, Dart. Para isso, foi primeiro necessário seguir um tutorial de introdução à linguagem para adquirir competências sobre como utilizar esta *framework*.

Para além disso, tentou-se sempre seguir os padrões visuais que já tinham sido delineados na plataforma web e garantindo que todas as páginas desenvolvidas nesta versão fossem adaptáveis a dispositivos móveis, considerando a diferença de dimensões de ecrãs.

Nesta versão do SensLive houve algumas funcionalidades que não foram implementadas da versão web, tais como a compra de subscrições. A decisão do não desenvolvimento destas funcionalidades recaiu sobre o supervisor. Será mais à frente mencionado o porquê e quais foram as funcionalidades que não foram implementadas.

Com isto, a primeira funcionalidade desenvolvida foi a página de *login*, que permite ao utilizar aceder ao SensLive, com as mesmas credenciais que são utilizadas para a plataforma SensLive. Dado que esta é uma aplicação *mobile*, ao desenvolver esta funcionalidade, teve-se em consideração o facto de que a conta do utilizador deve permanecer autenticada mesmo quando este fecha a aplicação, facilitando assim a sua utilização. Caso o utilizador pretenda realmente sair da sua conta, poderá fazê-lo através do *logout*, e assim, quando voltar a iniciar a aplicação, será necessário preencher novamente o formulário de *login*.

Para esta funcionalidade se comportar assim, foi necessário guardar o *access_token* do utilizador no armazenamento do dispositivo em causa, quando este faz *login* na plataforma. Isto foi possível através da utilização do package *SharedPreferences*⁵⁶, que é um *package* que possibilita guardar “*simple data*”, no armazenamento específico do sistema operativo. Assim, mesmo o utilizador fechando a aplicação, sempre que este voltava a abri-la, o *access_token* guardado no armazenamento, seria utilizado para fazer as demais chamadas à API.

Com o *token* do utilizador a ser guardado, o acesso aos dados foi bastante simplificado. O próximo passo foi desenvolver a *Home Page*. Esta página no SensLive web já se encontrava desenvolvida quando o desenvolvimento foi iniciado, desta forma foi necessário observar o seu comportamento e o código que estava a ser utilizado e assim, foi possível

⁵⁶ https://pub.dev/packages/shared_preferences

replicar o seu desempenho. Esta página apresenta três listas de sensores, diferenciadas pelo seu estado, um mapa que mostra os locais da empresa do utilizador e ainda uma tabela que apresenta os últimos eventos dos dispositivos. Na Figura 19 e na Figura 20 encontram-se as páginas iniciais do SensLive web e *mobile* respetivamente.

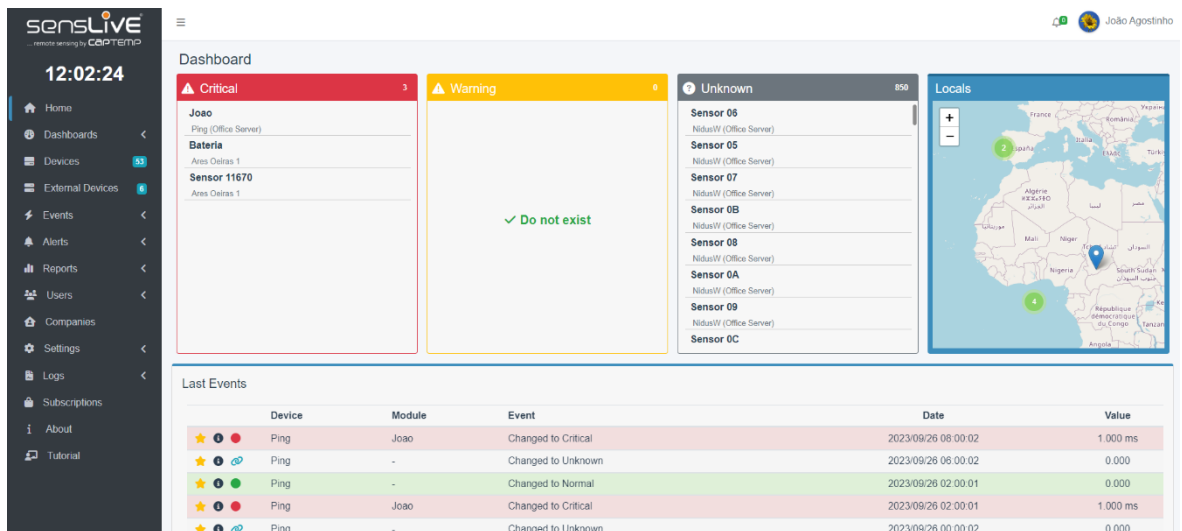


Figura 19 - Página inicial do SenLive web.

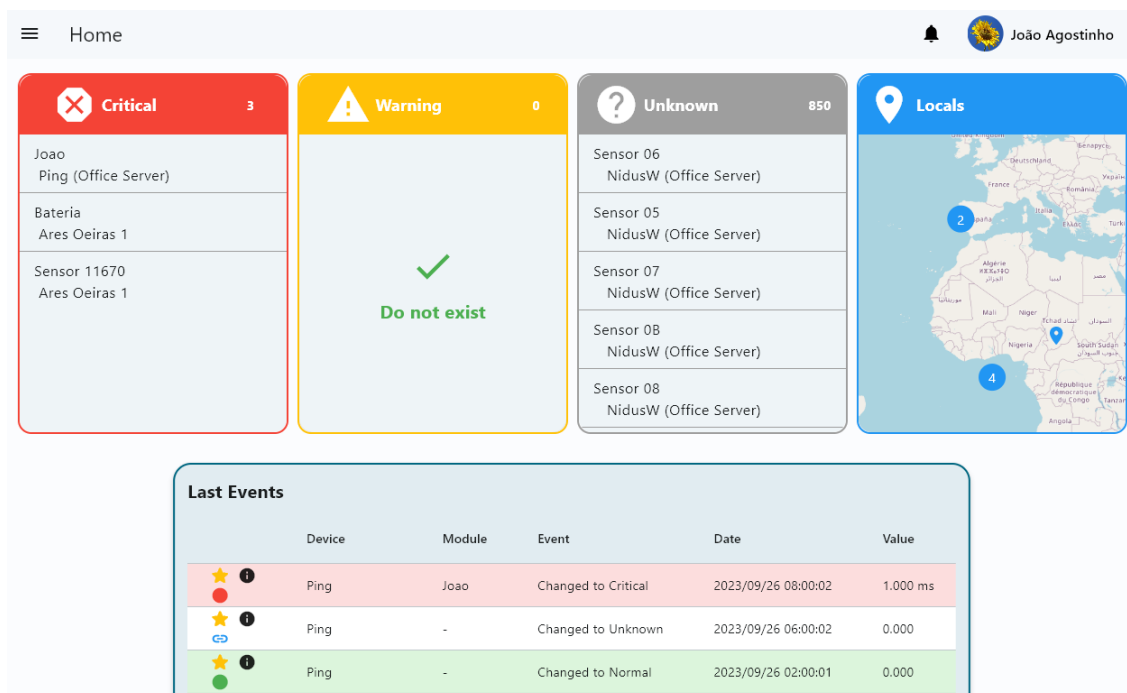


Figura 20 - Página inicial do SensLive *mobile*

Observando ambas as figuras, é possível verificar que visualmente ambas as páginas são bastante idênticas, sendo que a única diferença aparente é o esquema de cores diferente. Este cuidado visual, foi sempre mantido ao longo de todas as funcionalidades desenvolvidas para esta versão do SensLive.

Após se tratar da página inicial, foi necessário implementar funcionalidades essenciais para o bom funcionamento da aplicação. Estas funcionalidades incluem uma *navigation drawer* (navegação lateral) para facilitar a navegação entre as várias páginas da aplicação, uma *toolbar* que permite adicionar e facilitar ações relevantes, como botões de navegação mais utilizados. Ainda a utilização de um *package* de internacionalização, que funcionará de forma semelhante ao SensLive web, ajudando na tradução automática de textos. Para esta versão, também foi adotada a abordagem de dividir as páginas em duas partes, referenciada no subcapítulo 6.3.1. Esta abordagem consiste em separar as páginas em elementos de navegação e conteúdo da página.

Em relação ao *navigation drawer*, teve-se especial cuidado na definição das opções exibidas com base na *role* do utilizador autenticado. Dependendo do seu *role*, certas opções podem estar omitidas ou visíveis para esse utilizador. Para a internacionalização, foi utilizada a *package flutter_i18n*⁵⁷, que recria a mesma experiência proporcionada pelo i18n, mas aplicada à tecnologia Flutter.

Com estas funcionalidades essenciais criadas, passou-se ao desenvolvimento da funcionalidade de perfil. Esta funcionalidade permite atualizar qualquer dado referente ao utilizador, como o nome, o email, a password, a foto, etc. Como esta funcionalidade já estava desenvolvida e os *endpoints* já tinham sido criados, a sua implementação foi relativamente simples. No entanto, surgiram alguns problemas ao fazer o *upload* da foto do utilizador, pois não estava a ser possível guardá-la corretamente. Após se tentar várias soluções, o problema foi resolvido com a utilização de um pedido HTTP *Multipart*. Este tipo de pedido serve para resolver o problema de *upload* eficiente de ficheiros [55]. Com o uso de um pedido HTTP *Multipart*, a foto passou a ser guardada com sucesso no servidor.

De seguida, foi feito o desenvolvimento da funcionalidade dos dispositivos externos. Esta funcionalidade já tinha sido implementada no SensLive, mas como já descrito anteriormente no subcapítulo 6.3.1, foi apenas necessário desenvolver os eventos destes dispositivos. No entanto, para esta nova versão, foi necessário desenvolver na totalidade esta funcionalidade. Visto que na parte do servidor já estava tudo desenvolvido, foi apenas necessário implementar a parte visual desta página. Esta página lista todos os dispositivos externos da empresa, mostrando o seu nome e o local associado. Para cada dispositivo,

⁵⁷ https://pub.dev/packages/flutter_i18n


também é possível abrir o URL associado, editar, adicionar um evento e eliminar o dispositivo.

Foi também necessário desenvolver a funcionalidade dos gráficos. Para esta página foi tida como base a página já desenvolvida no SensLive web. Assim como na versão anterior, foram implementados apenas os gráficos de linhas, médias e lista. Tal como as outras funcionalidades já descritas, esta também já tinha todas as funções na parte do servidor desenvolvidas, sendo necessário apenas implementar a parte visual da página. No entanto, para esta versão, foi necessário adaptar a criação dos gráficos, pois não era possível utilizar a mesma biblioteca usada na versão web da plataforma. Para isso, foi utilizada a *package syncfusion_flutter_charts*⁵⁸, criada nativamente em Dart para permitir a visualização de gráficos de alta qualidade em aplicações móveis. Ultrapassando este obstáculo, o resto da implementação tornou-se bastante trivial.

É de salientar que a opção de transferir os gráficos em formato de CSV e PDF e a funcionalidade dos relatórios personalizados, na versão *mobile* não foi implementada, visto que essa funcionalidade é exclusiva da versão web do SensLive.

Foi decidido que, para esta versão da plataforma, não seria implementada a funcionalidade da compra de subscrições independentemente do seu tipo. No entanto, os administradores da plataforma SenLive poderão fazer gestão das subscrições. Assim, esta funcionalidade foi implementada com as mesmas regras seguidas para a versão web, sendo que o mesmo pode ser observado na Figura 21.

⁵⁸ https://pub.dev/packages/syncfusion_flutter_charts



Active ↑	Name	Reference	Price 1	Price 2	Price 3	Number of Sensors	Duration	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Free Subscription	SLS-SAFREE	0.00 €	0.00 €	0.00 €	0	Free	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 25 sensors	SLS-SA0025	[REDACTED]	[REDACTED]	[REDACTED]	25	1 Year	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 50 sensors	SLS-SA0050	[REDACTED]	[REDACTED]	[REDACTED]	50	1 Year	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 100 sensors	SLS-SA0100	[REDACTED]	[REDACTED]	[REDACTED]	100	1 Year	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 1000 sensors	SLS-SA1000	[REDACTED]	[REDACTED]	[REDACTED]	1000	1 Year	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 12 sensors	SLS-SA0012	[REDACTED]	[REDACTED]	[REDACTED]	12	1 Year	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 6 sensors	SLS-SA0006	[REDACTED]	[REDACTED]	[REDACTED]	6	1 Year	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Network w/ up to 500 sensors	SLS-SA0500	[REDACTED]	[REDACTED]	[REDACTED]	500	1 Year	<input type="checkbox"/>

Figura 21 - Página de gestão de subscrições de produtos

Posteriormente, foram desenvolvidas todas as funcionalidades presentes no menu de Definições (*Settings*), que incluem: locais, email, SMS, tradução de valores e pagamentos. Cada uma destas funcionalidades foi representada como um submenu dentro do menu dos *Settings*.

Referentemente às definições dos locais, esta já se encontrava desenvolvida na versão web, no entanto, para se conseguir replicar o seu comportamento nesta nova versão foi necessário rever o código já implementado. Esta página tem como objetivo, permitir ao utilizador: adicionar, editar e eliminar locais da sua empresa. Caso este queira adicionar um local, terá de dar um nome ao local, uma descrição, em que cidade e país este está situado e ainda selecionar o local num mapa ou inserir as suas coordenadas. Após preencher estes dados, um local será adicionado à empresa e aparecerá na tabela com os outros locais já existentes.

Para as definições de email e SMS, foram tidos os mesmos pontos de referência que constavam da solução web do SensLive. Tal como referido anteriormente no subcapítulo 6.3.1, as definições de SMS estão divididas em duas opções que os utilizadores podem escolher. A utilização do serviço do SensLive para receber notificações dos alarmes ou utilizar o seu próprio servidor GSM. Para as definições de email, o princípio é o mesmo, irá haver uma opção para utilização do serviço do SensLive para envio de notificações dos alarmes e relatórios automáticos ou a opção de utilizar o próprio servidor SMTP (*Simple Mail Transfer Protocol*).

Com isto, foram desenvolvidas ambas as páginas onde cada opção estava representada por um separador. Se o utilizador optar por utilizar o serviço do SensLive, terá de utilizar a funcionalidade de pagamentos de subscrições para conseguir adquirir créditos. Como esta funcionalidade não está implementada na aplicação, existe um aviso informando o utilizador de que, para utilizar este serviço, é necessário aceder à plataforma web do SensLive.

Caso o utilizador prefira utilizar o seu próprio serviço, nas definições de email, será necessário preencher alguns dados, como o endereço de envio, o servidor SMTP utilizado, a porta do servidor, se pretende utilizar autenticação e que tipo de encriptação deseja para o envio.

Para as definições de SMS, independentemente da opção escolhida, o utilizador terá de fornecer alguns dados, incluindo a linguagem das mensagens de alerta, o número máximo de mensagens diárias e o número associado para receber alertas. Também será possível usufruir da funcionalidade mencionada no subcapítulo 6.3.1, que possibilita ao utilizador associar um dispositivo a cada local da empresa.

Para facilitar a compreensão, está representado visualmente na Figura 22, a página de definições de email.

SensLive Service	Your own SMTP Server
From Address: senslive@captemp.com	From Name: Senslive Portal teste
SMTP Server: mail.captemp.com	SMTP Port: 994
SMTP authentication <input type="checkbox"/>	SMTP Encryption: ssl
Username: senslive@captemp.com	Password: *****

Figura 22 - Página de definições de email

Em relação às páginas da tradução de valores e histórico de pagamentos, estas contêm as mesmas características da versão web já mencionadas no subcapítulo 6.3.1. Porém, a funcionalidade que é responsável por gerar as faturas dos pagamentos que já foram pagos, na tabela do histórico de pagamentos, não foi implementada. Para visualização e transferência das faturas, os utilizadores precisarão de aceder à plataforma SensLive web. Nesta versão, os utilizadores apenas poderão utilizar a funcionalidade que permite visualizar as informações de pagamentos que estejam em estado “Pendente”.

Após a implementação de todas as funcionalidades das definições, iniciou-se o desenvolvimento da página de notificações. Esta foi elaborada com base nos mesmos princípios da versão web. Como a funcionalidade já existia, foi essencial analisar o seu comportamento para replicar todos os pontos cruciais. Pontos estes que incluíam:

- Lista com os nomes dos utilizadores que enviaram mensagens;
- Identificação de mensagens por ler;
- Lista de mensagens referentes ao utilizador que a enviou;
- Hora e data de envio na listagem das mensagens;
- Marcar uma mensagem como lida ao clicar nela.

As mensagens são armazenadas numa tabela da base de dados, que regista a mensagem enviada, o remetente, o destinatário, a data e a hora de envio, entre outros campos. Na Figura 23, encontra-se a página de notificações desenvolvida para esta versão da plataforma.

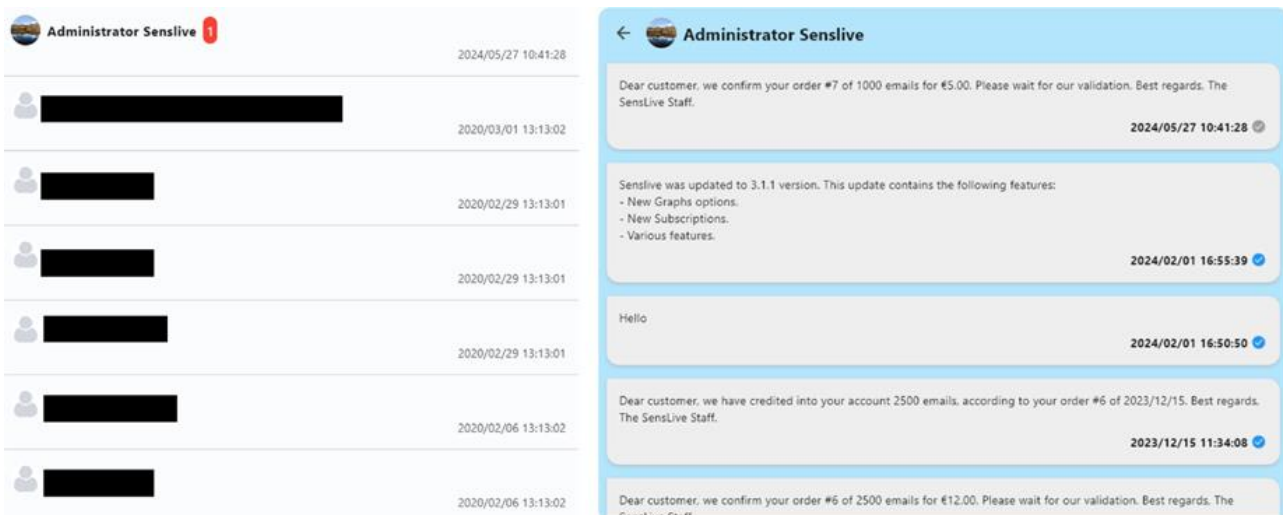


Figura 23 - Página de notificações

A figura representa as duas variantes da página de notificações. É importante realçar que a página ou apresenta o nome dos vários utilizadores, com a data da última mensagem, como

visualizado na parte esquerda da figura, ou as mensagens de um utilizador específico que tenha sido selecionado anteriormente, como visualizado na parte direita da figura.

De seguida, procedeu-se à implementação das páginas do *About* e das Estatísticas. As funcionalidades destas já se encontram descritas no capítulo anterior, por isso não serão novamente descritas.

Por fim, a última funcionalidade desenvolvida foi a capacidade dos utilizadores desta versão do SensLive receberem o alerta sobre atualizações da plataforma. Tal como na versão *web*, verifica-se se a última versão a que o utilizador teve acesso corresponde à mais recente disponível na tabela de versões. Se a versão for a mais recente, nenhum alerta será apresentado, caso contrário, será exibido um alerta sobre a nova atualização. É importante destacar que apenas esta funcionalidade foi implementada e não a funcionalidade completa da manutenção, que engloba a manutenção regular e manutenção para atualização de versões mencionadas anteriormente.

Tal como a funcionalidade de manutenção, o tutorial não foi implementado nesta versão da plataforma. Durante o desenvolvimento, teve-se o cuidado de garantir que um utilizador do SensLive web pudesse interagir com a versão *mobile* de maneira idêntica à qual já está habituado. Portanto, considerou-se que a implementação do tutorial seria redundante nesta versão.

Com o desenvolvimento concluído, tornou-se necessário dedicar algum tempo adicional para melhorar a aparência geral da aplicação. Esse esforço visou aprimorar o design e a interface visual, garantindo uma experiência de utilizador mais agradável e intuitiva. Desta forma, foi sempre procurado refinar todos os elementos visuais e estilísticos, assegurando que a aplicação não só funcionasse bem, mas também que seguisse os elementos característicos da plataforma SensLive web.

Considerando todas as funcionalidades que não foram implementadas nesta versão da plataforma, a principal razão para isso foi o fato de esta ser uma aplicação *cross-platform*. Foi tido sempre em consideração que esta deveria ser “leve” e que certas funcionalidades seriam demasiado exigentes computacionalmente para uma aplicação deste tipo. Da mesma forma, foi determinado pelo supervisor que algumas dessas funcionalidades seriam exclusivas da versão web do SensLive, justificando assim também a sua não implementação.

- **Extensão para *dashboards***

Por último, é relevante mencionar o desenvolvimento de uma extensão, específica para a visualização de *dashboards*. Esta aplicação foi concebida para permitir que os utilizadores apresentem os *dashboards* em qualquer ecrã ou local onde seja importante visualizar as leituras ou mapa. A funcionalidade dos *dashboards* original foi desenvolvida por outro estagiário, sendo que a principal diferença a esta extensão, reside no facto de, na versão original, ser possível editar, redimensionar, criar e mover *widgets*, ao contrário desta nova aplicação. A integração na “extensão” foi realizada, ao incorporar o código já desenvolvido da solução existente do SensLive *mobile*. Os *dashboards* são constituídos por diversos tipos de *widgets*, tais como leituras de capacidade, último valor registado, gráficos, mapa geográfico e *blueprints*.

A extensão oferece duas opções de *login*: através de QR Code, disponível na página dos *dashboards* em ambas as versões do SenLive, ou inserindo o código do *dashboard* manualmente, sem necessidade de usar o QR Code *scanner*. Assim como o SensLive *mobile*, quando o utilizador fecha a aplicação sem fazer *logout*, a sua sessão é guardada, mantendo a autenticação ativa para a próxima utilização. Para esquecer as credenciais, é necessário fazer *logout* na aplicação. Além disso, os utilizadores têm a liberdade de escolher entre o tema *light* ou *dark*, conforme a sua preferência.

Desenvolvida também em Flutter, esta extensão serve para que os utilizadores acedam aos *dashboards* de forma independente do SensLive *mobile* ou *web*. Isso permite, por exemplo, manter a aplicação aberta num dispositivo ou ecrã dedicado, mostrando os principais *widgets* que se quer monitorizar, facilitando a visualização contínua e simplificada dos dados mais importantes.

Na Figura 24, encontra-se um exemplo de um *dashboard* inserido nesta extensão.



Figura 24 - Extensão para visualização de *dashboards*

Observando a figura, é possível ver vários *widjets*, incluindo aqueles que mostram o último valor registado, destacados a verde, bem como as leituras de capacidade representadas pelos nomes “*Battery*” e “*Signal*”. Além disso, existem gráficos de linhas, barras e circulares com dados de sensores, um mapa geográfico que apresenta os locais e dispositivos da empresa, e também uma *blueprint* que apresenta um *heatmap* dentro de uma habitação. Os retângulos presentes no *heatmap* representam os sensores que o utilizador adicionou a este *blueprint*. É importante referir que qualquer modificação que se pretenda fazer, terá de ser realizada na versão web ou *mobile* do SensLive.

7. Conclusão

O estágio realizado teve como objetivo estudo e a elaboração de uma nova versão da plataforma SensLive, de modo a proporcionar uma experiência de utilizador mais fluída e simplificada.

Com o crescente número de pessoas a utilizar dispositivos tecnológicos, como computadores, telemóveis, tablets, etc., uma empresa que pretenda migrar de uma plataforma antiga para uma mais recente pode utilizar este relatório como referência. O documento oferece diversas estratégias e análises detalhadas sobre a melhor forma de realizar essa transição, bem como as tecnologias mais relevantes atualmente e as suas características.

Referenciando os objetivos técnicos traçados no início do estágio, estes foram alcançados dentro do prazo e conforme planeado. A funcionalidade dos pagamentos, que tinha sido traçada como crucial para os objetivos do estágio, foi desenvolvida com sucesso e encontra-se a ser utilizada na nova versão do SensLive conforme planeado inicialmente. Devido ao desenvolvimento do SensLive ter sido mais rápido do que o esperado, foi possível iniciar uma nova vertente desta plataforma, a versão *mobile*, que já era algo que a empresa internamente, já tinha planeado, mas devido à falta de tempo nunca tinha sido iniciada.

Os objetivos gerais foram também atingidos com sucesso, refletindo-se numa integração muito positiva na CapTemp. A disponibilidade para qualquer tipo de ajuda foi um fator crucial para este sucesso. Além disso, foi possível experienciar a forma de trabalhar numa empresa, o que contribuiu não só para o aumento das capacidades técnicas, mas também para o desenvolvimento da capacidade de comunicação.

Ao longo deste período, foram aprendidos novos conceitos e tecnologias que serão bastante benéficos para o futuro. A constante ajuda do nosso supervisor foi essencial para o cumprimento destes objetivos. A sua disponibilidade para esclarecer qualquer dúvida, mantendo uma postura crítica, garantiu que o desenvolvimento seguisse na direção correta.

Por fim, é importante referir que todo o estágio foi realizado de forma presencial. Esta experiência proporcionou um verdadeiro ambiente de empresa, oferecendo a primeira experiência no mundo do trabalho.

7.1. Trabalho futuro

Num desenvolvimento de *software*, é inevitável encontrar imperfeições. Mesmo quando uma solução parece estar completa, há sempre a possibilidade de surgirem erros ou situações não previstas, exigindo correções ou ajustes. Da mesma forma, embora um sistema possa estar operacional em todas as suas vertentes, é comum identificar áreas onde se pode melhorar, como por exemplo, otimização de código, redução do tempo de resposta do lado do servidor, ou até mesmo melhorar algum elemento visual da plataforma.

No *back-end*, é provável encontrar algumas funções que carecem de otimização e redundâncias entre as suas funcionalidades. Resolver essas questões no futuro poderia não só melhorar o desempenho, mas também reduzir o tamanho de certos ficheiros.

Uma funcionalidade que requer melhorias para o futuro, seria a funcionalidade do tutorial. Visto que esta funcionalidade, foi desenvolvida separadamente, não houve oportunidade de juntar ambas as partes e ver a versão final. Assim, terminar e colocar em produção para os clientes utilizarem, seria algo a ter em conta no futuro.

No contexto da versão *mobile* desenvolvida em Flutter, seria importante incorporar as funcionalidades que não foram inicialmente implementadas na primeira versão. Um tutorial na versão *mobile*, seria interessante desenvolver, visto que apesar de em teoria ser, a mesma plataforma, a mudança de suporte pode fazer com que os clientes tenham algumas dificuldades na utilização desta versão. Para além disso, é crucial que a aplicação mantenha a sua agilidade e rapidez, características próprias de uma aplicação *mobile*. Adicionalmente, é importante realçar a necessidade de otimização do código como meio de aprimorar o desempenho geral da aplicação.

Finalmente, é crucial reconhecer que o desenvolvimento de *software* é um processo contínuo de melhoria. Ao abordar as imperfeições e as áreas a aperfeiçoar em todas as fases do projeto, desde o *back-end* até à versão *mobile*, podemos assegurar que a aplicação atende continuamente às necessidades dos utilizadores, mantendo elevados padrões de desempenho e eficiência.

8. Referências

- [1] T. Ainsworth, J. Brake, P. Gonzalez, D. Toma e A. F. Browne, “A Comprehensive Survey of Industry 4.0, IIoT and Areas of Implementation,” *SoutheastCon 2021*, pp. 1-6, 2021.
- [2] T. Davidson, “Single Page Application (SPA) vs Multi Page Application (MPA): Which Is The Best?,” CleanCommit, 28 Fevereiro 2023. [Online]. Available: <https://cleancommit.io/blog/spa-vs-mpa-which-is-the-king/>. [Acedido em 25 Junho 2024].
- [3] CapTemp, “CapTemp - Monitoring Sytems with Wireless Sensor & sync Technology,” CapTemp, Lda, 2023. [Online]. Available: <https://www.captemp.com/>. [Acedido em 14 Dezembro 2023].
- [4] CapTemp, “CapTemp - About us,” CapTemp, Lda, 2023. [Online]. Available: <https://www.captemp.com/about>. [Acedido em 14 Dezembro 2023].
- [5] C. Deshpande, “What Is React?,” Simplilearn Solutions, 5 Outubro 2023. [Online]. Available: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>. [Acedido em 2 Fevereiro 2024].
- [6] J. Desai, “5 Best Frontend Frameworks for Web Development in 2024,” Positiwise Software Private Limited, 22 Dezembro 2023. [Online]. Available: <https://positiwise.com/blog/best-front-end-frameworks>. [Acedido em 2 Fevereiro 2024].
- [7] AlternativeTo, “Vue.js,” AlternativeTo, [Online]. Available: <https://alternativeto.net/software/vue-js/about/>. [Acedido em 4 Janeiro 2024].
- [8] H. Akhtar, “Introduction to AngularJS Framework (with Key Concepts),” BrowserStack, 12 Maio 2023. [Online]. Available: <https://www.browserstack.com/guide/what-is-angularjs-framework>. [Acedido em 2 Fevereiro 2024].

- [9] R. Harężlak, “What Is Ember.js?,” Selleo, 7 Agosto 2023. [Online]. Available: <https://selleo.com/blog/what-is-ember-js>. [Acedido em 2 Fevereiro 2024].
- [10] AlternativeTo, “Laravel,” AlternativeTo, 20 Fevereiro 2023. [Online]. Available: <https://alternativeto.net/software/laravel/>. [Acedido em 4 Janeiro 2024].
- [11] ICStudio, “17 Benefits of Laravel Framework,” ICStudio, 6 Setembro 2022. [Online]. Available: <https://icstudio.online/en/post/17-benefits-laravel-framework>. [Acedido em 23 Fevereiro 2024].
- [12] IBM, “What is Django?,” IBM, [Online]. Available: <https://www.ibm.com/topics/django>. [Acedido em 2 Fevereiro 2024].
- [13] K. Karakhainko, “What are the best backend development frameworks? The ultimate comparison,” TechWings, 1 Setembro 2023. [Online]. Available: <https://techwings.com/blog/backend-frameworks-ultimate-comparison>. [Acedido em 2 Fevereiro 2024].
- [14] P. Pedamkar, “What is CodeIgniter?,” EDUCBA, 5 Julho 2023. [Online]. Available: <https://www.educba.com/what-is-codeigniter/>. [Acedido em 2 Fevereiro 2024].
- [15] P. Kirvan, “What is loose coupling?,” TechTarget, Junho 2023. [Online]. Available: <https://www.techtarget.com/searchnetworking/definition/loose-coupling>. [Acedido em 12 Agosto 2024].
- [16] R. Singh, “What is Symfony and use cases of Symfony?,” DevOpsSchool, 25 Dezembro 2023. [Online]. Available: <https://www.devopsschool.com/blog/what-is-symfony-and-use-cases-of-symfony/>. [Acedido em 2 Fevereiro 2024].
- [17] IBM, “What is PostgreSQL?,” IBM, [Online]. Available: <https://www.ibm.com/topics/postgresql>. [Acedido em 2 Fevereiro 2024].
- [18] A. Hassan, “What Is SQLite?,” Built In, 21 Dezembro 2022. [Online]. Available: <https://builtin.com/data-science/sqlite>. [Acedido em 2 Fevereiro 2024].

-
- [19] R. A. S, “What is SQLite? Everything You Need to Know,” Simplilearn, 16 Fevereiro 2023. [Online]. Available: <https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite>. [Acedido em 23 Fevereiro 2024].
- [20] L. Moore, “What is MySQL?,” TechTarget, Julho 2018. [Online]. Available: <https://www.techtarget.com/searchoracle/definition/MySQL>. [Acedido em 2 Fevereiro 2024].
- [21] S. Ravooof, “PostgreSQL vs MySQL: Explore Their 12 Critical Differences,” Kinsta, 29 Dezembro 2023. [Online]. Available: <https://kinsta.com/blog/postgresql-vs-mysql/>. [Acedido em 23 Fevereiro 2024].
- [22] A. Custodio, “What is MariaDB?,” InMotion Hosting, Inc, 22 Dezembro 2023. [Online]. Available: <https://www.inmotionhosting.com/support/server/databases/what-is-mariadb/>. [Acedido em 2 Fevereiro 2024].
- [23] M. Drake, “SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems,” DigitalOcean, 9 Março 2022. [Online]. Available: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>. [Acedido em 2 Fevereiro 2024].
- [24] M. Budziński, “What Is React Native? Complex Guide for 2023,” Netguru S.A, [Online]. Available: <https://www.netguru.com/glossary/react-native>. [Acedido em 9 Fevereiro 2024].
- [25] V. Prudnikov, “Flutter vs React Native: Which Framework Wins in 2024?,” What the Flutter Blog, 31 Dezembro 2023. [Online]. Available: <https://blog.flutter.wtf/flutter-vs-react-native/>. [Acedido em 23 Fevereiro 2024].
- [26] Javatpoint, “What is Xamarin?,” Javatpoint, [Online]. Available: <https://www.javatpoint.com/what-is-xamarin>. [Acedido em 9 Fevereiro 2024].

- [27] V. Prudnikov, “Flutter vs Xamarin: What to Choose for Mobile App?,” What The Flutter Blog, 4 Dezembro 2023. [Online]. Available: <https://blog.flutter.wtf/flutter-vs-xamarin/>. [Acedido em 23 Fevereiro 2024].
- [28] A. Cooper, “Flutter Vs React Native Vs Xamarin – Top Cross Platform Mobile App Development Frameworks,” ValueCoders, [Online]. Available: <https://www.valuecoders.com/blog/technology-and-apps/flutter-vs-xamarin-vs-react-native-which-cross-platform-mobile-app-development-framework-to-choose/>. [Acedido em 9 Fevereiro 2024].
- [29] V. Thomas, “Laravel Framework: Secure, Flexible, Scalable.,” WebSight Design Inc., 28 Setembro 2023. [Online]. Available: <https://www.websightdesign.com/blog/post/2023/09/28/laravel-framework-security-flexibility-and-scalability>. [Acedido em 23 Fevereiro 2024].
- [30] S. Gyaan, “What is XAMPP and how does it work? Everything You Need to Know About the XAMPP Server.,” 27 Setembro 2023. [Online]. Available: <https://www.linkedin.com/pulse/what-xampp-how-does-work-everything-you-need-know-server-site-gyaan>. [Acedido em 9 Fevereiro 2024].
- [31] P. Iwuh, “What is WAMP Server?,” Wazobia , 9 Maio 2023. [Online]. Available: <https://wazobia.tech/blog/development/what-is-wamp-server>. [Acedido em 9 Fevereiro 2024].
- [32] Eina Virtual, “XAMPP VS WAMP VS LARAGON: WHICH IS THE BEST OPTION FOR LOCAL PHP DEVELOPMENT?,” Eina Virtual, [Online]. Available: <https://einavirtual.com/php/xamp-wamp-laragon>. [Acedido em 23 Fevereiro 2024].
- [33] Noble Desktop, “What is Git and Why Should You Use it?,” Noble Desktop, 22 Setembro 2023. [Online]. Available: <https://www.nobledesktop.com/learn/git/what-is-git>. [Acedido em 26 Janeiro 2024].

-
- [34] Containerize, “Mercurial | Free Distributed Version Control System,” Containerize, [Online]. Available: <https://products.containerize.com/version-control/mercurial/>. [Acedido em 9 Fevereiro 2024].
- [35] Miladev95, “Apache Subversion (SVN),” 29 Agosto 2023. [Online]. Available: <https://medium.com/@miladev95/apache-subversion-svn-ebd1a8baf72f>. [Acedido em 9 Fevereiro 2024].
- [36] D. Cortuk, “Version Control Software Comparison: Git, Mercurial,CVS, SVN,” 28 Outubro 2019. [Online]. Available: <https://medium.com/@derya.cortuk/version-control-software-comparison-git-mercurial-cvs-svn-21b2a71226e4>. [Acedido em 9 Fevereiro 2024].
- [37] Postman, Inc, “What is Postman?,” Postman, Inc, [Online]. Available: <https://www.postman.com/product/what-is-postman/>. [Acedido em 26 Janeiro 2024].
- [38] G. Fong, “A useful guide to API testing using Insomnia,” 10 Setembro 2023. [Online]. Available: <https://medium.com/@gavinklfong/a-useful-guide-to-api-testing-using-insomnia-35ccdaf35ae>. [Acedido em 9 Fevereiro 2024].
- [39] Apidog, “Swagger: Everything You Need to Know,” Apidog, [Online]. Available: <https://apidog.com/articles/what-is-swagger/>. [Acedido em 9 Fevereiro 2024].
- [40] SourceForge, “Insomnia vs. Postman vs. Swagger Comparison Chart,” SourceForge, [Online]. Available: <https://sourceforge.net/software/compare/Insomnia-vs-Postman-vs-Swagger/>. [Acedido em 9 Fevereiro 2024].
- [41] Microsoft, “Visual Studio Code - Code editing. Redefined.,” Microsoft, [Online]. Available: <https://code.visualstudio.com/>. [Acedido em 26 Janeiro 2024].
- [42] O. Peter, A. Pradhan e C. Mbohwa, “Industrial internet of things (IIoT): opportunities, challenges, and requirements in manufacturing businesses in

- emerging economies,” *Procedia Computer Science, Volume 217*, pp. 856-865, 2023.
- [43] M. K. Tange, X. F. De Donno e N. Dragoni, “A Systematic Survey of Industrial Internet,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489-2520, 2020.
- [44] T. D. D. K. S. C. Tsiknas K, “Cyber Threats to Industrial IoT: A Survey on Attacks and Countermeasures.,” *IoT 2*, no. 1, pp. 163-186, 2021.
- [45] Record Evolution, “The IIoT Architecture: How To Tap Into Its Full Potential?,” Record Evolution, 9 Dezembro 2020. [Online]. Available: <https://www.record-evolution.de/en/blog/the-iiot-architecture-design-an-overview/>. [Acedido em 1 Março 2024].
- [46] A. Simmons, “Internet of Things (IoT) Architecture: Layers Explained,” Dgtl Infra LLC, 13 Novembro 2022. [Online]. Available: <https://dgtlinfra.com/internet-of-things-iot-architecture/>. [Acedido em 1 Março 2024].
- [47] Akenza AG, “akenza | Self-Service IoT Platform to Build Smart Solutions,” Akenza AG, [Online]. Available: <https://akenza.io/features>. [Acedido em 05 Junho 2024].
- [48] ThingsBoard, “ThingsBoard - Open-source IoT Platform,” ThingsBoard, [Online]. Available: <https://thingsboard.io/>. [Acedido em 05 Junho 2024].
- [49] Amazon Web Services, “AWS IoT Device Management,” Amazon Web Services, [Online]. Available: <https://aws.amazon.com/pt/iot-device-management/>. [Acedido em 05 Junho 2024].
- [50] PostgreSQL Community Association of Canada, “pgAdmin - PostgreSQL Tools,” PostgreSQL Community Association of Canada, [Online]. Available: <https://www.pgadmin.org/>. [Acedido em 26 Janeiro 2024].
- [51] Amazon Web Services, “What is Amazon Relational Database Service (Amazon RDS)?,” Amazon Web Services, [Online]. Available:

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>.

[Acedido em 6 Junho 2024].

[52] Wikipedia, “Amazon EC2,” 28 Março 2024. [Online]. Available: https://pt.wikipedia.org/wiki/Amazon_EC2. [Acedido em 06 Junho 2024].

[53] R. Prabhu, “SHA-1 Hash,” GeeksForGeeks, 18 Julho 2024. [Online]. Available: <https://www.geeksforgeeks.org/sha-1-hash-in-java/>. [Acedido em 26 Agosto 2024].

[54] Easypay, “Easypay docs,” [Online]. Available: <https://docs.easypay.pt/#tag/Single-Payment>. [Acedido em 19 Abril 2024].

[55] A. Chalmers, “Introduction to HTTP Multipart,” 25 Abril 2023. [Online]. Available: <https://blog.adamchalmers.com/multipart/>. [Acedido em 27 Maio 2024].

[56] Python, “Python,” Python Software Foundation, [Online]. Available: <https://www.python.org/>. [Acedido em 19 Janeiro 2024].

[57] M. Yurko, “Comparing Windows XAMPP, WampServer, EasyPHP, and Other Alternatives,” CopyProgramming, 14 Abril 2023. [Online]. Available: <https://copyprogramming.com/howto/windows-xampp-vs-wampserver-vs-easyphp-vs-alternative>. [Acedido em 9 Fevereiro 2024].

[58] T. D. D. K. S. C. Tsiknas K, “Cyber Threats to Industrial IoT: A Survey on Attacks and Countermeasures,” pp. 163-186, 2021.